Modelowanie matematyczne

programowanie liniowe i całkowitoliczbowe

Wykład 1

Konstanty Junosza-Szaniawski
Armin Fügenschuh
Paweł Rzążewski
Joanna Sokół Krzysztof Węsek

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Zaliczenie - ostatnie laboratorium punktowane (25 pkt) + dokumentacja powykonawcza (25pkt)

Oprogramowanie - ZIBopt Solver Suite: ZIMPL + SCIP
http://scip.zib.de/

Literatura - H. Paul Williams, Model building in mathematical programming

Pojęcie modelu

Pojęcie modelu

- W wielu sytuacjach używamy **modeli**.

Pojęcie modelu

- W wielu sytuacjach używamy **modeli**.
- Jako model rozumiemy obiekt który ma imitować jakiś inny obiekt np.:

Ziemia
(obiekt)

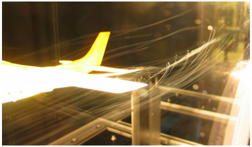
Proces abstrakcji

Globus
(model obiektu)
- idealna sfera
- równoleżniki i południki
- lądy i oceany
- państwa i miasta
- oś Ziemi

Pojęcie modelu

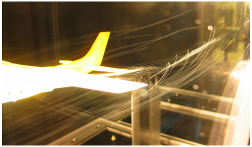
Pojęcie modelu

- Niektóre modele są fizyczne...

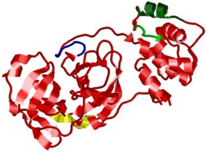
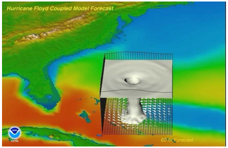
  

Pojęcie modelu

- Niektóre modele są fizyczne...


- A inne abstrakcyjne...

Pojęcie modelu

Pojęcie modelu

- Model abstrakcyjny nazywamy matematycznym, jeśli własności modelowanego obiektu są określone przez zależności algebraiczne, funkcje, relacje.

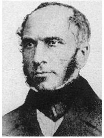
Pojęcie modelu

- Model abstrakcyjny nazywamy matematycznym, jeśli własności modelowanego obiektu są określone przez zależności algebraiczne, funkcje, relacje.
- Przykład: Gaz płynący rurociągiem.

Proces abstrakcji matematycznej

$$h_f = f \cdot \frac{L}{D} \cdot \frac{V^2}{2g}$$

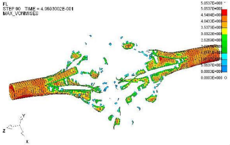
Równanie Darcy'ego - Weisbacha



Henry Darcy (1803-1853)

Julius Weisbach (1806-1871)

Rozwiązanie modelu



Interpretacja

Pojęcie modelu

Pojęcie modelu

- Istotą matematycznego modelu jest użycie:
  - równości
  - nierówności
  - logicznych zależności

Pojęcie modelu

- Istotą matematycznego modelu jest użycie:
  - równości
  - nierówności
  - logicznych zależności

- Matematyczne zależności odzwierciedlają zależności pochodzące z świata rzeczywistego.
  - prawa fizyki
  - wymagania techniczne
  - ograniczenia rynkowe

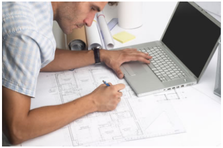
Po co modelować?

Po co modelować?

- Proces budowania modelu, tworzonego przez grupę ludzi, pozwala lepiej zrozumieć modelowane zjawisko.

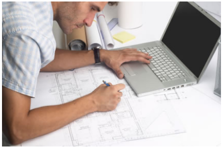
Po co modelować?

- Proces budowania modelu, tworzonego przez grupę ludzi, pozwala lepiej zrozumieć modelowane zjawisko.
- Rozwiązanie modelu może przynieść zaskakujące, których nie otrzymalibyśmy w inny sposób.



Po co modelować?

- Proces budowania modelu, tworzonego przez grupę ludzi, pozwala lepiej zrozumieć modelowane zjawisko.



- Rozwiązanie modelu może przynieść zaskakujące, których nie otrzymalibyśmy w inny sposób.

- Model pozwala na przeprowadzenie eksperymentów, które w świecie rzeczywistym są bardziej kosztowne lub w ogóle niemożliwe.

Model a dane

Model a dane

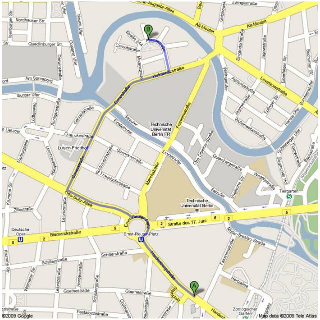
- Model powinien w miarę możliwości być jak najbardziej niezależny od danych.

Model a dane

- Model powinien w miarę możliwości być jak najbardziej niezależny od danych.
- Dzięki temu może być wykorzystany dla różnych danych.
  - zmiana kosztów
  - zmiana współczynników technologicznych
  - zmiana ilości dostępnych surowców

Model a dane

- Model powinien w miarę możliwości być jak najbardziej niezależny od danych.
- Dzięki temu może być wykorzystany dla różnych danych.
    - zmiana kosztów
    - zmiana współczynników technologicznych
    - zmiana ilości dostępnych surowców
- Przykład: Model najkrótszej ścieżki – Warszawa i Berlin



Standardowe modele

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego
- Mogą one być używane w wielu rzeczywistych sytuacjach.

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego
- Mogą one być używane w wielu rzeczywistych sytuacjach.
- Dla typowych modeli znanych jest wiele specjalistycznych algorytmów, które rozwiązują je w sposób efektywny.

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego
- Mogą one być używane w wielu rzeczywistych sytuacjach.
- Dla typowych modeli znanych jest wiele specjalistycznych algorytmów, które rozwiązują je w sposób efektywny.
- Jednak ich zastosowanie jest ograniczone i wystarczy niewielka zmiana założeń, a przestają pasować do naszego modelu.

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego
- Mogą one być używane w wielu rzeczywistych sytuacjach.
- Dla typowych modeli znanych jest wiele specjalistycznych algorytmów, które rozwiązują je w sposób efektywny.
- Jednak ich zastosowanie jest ograniczone i wystarczy niewielka zmiana założeń, a przestają pasować do naszego modelu.
- Wtedy należy używać ogólnych metod modelowania.

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego
- Mogą one być używane w wielu rzeczywistych sytuacjach.
- Dla typowych modeli znanych jest wiele specjalistycznych algorytmów, które rozwiązują je w sposób efektywny.
- Jednak ich zastosowanie jest ograniczone i wystarczy niewielka zmiana założeń, a przestają pasować do naszego modelu.
- Wtedy należy używać ogólnych metod modelowania.
- Wiele sytuacji można zamodelować na więcej niż jeden sposób.

Standardowe modele

- Jest wiele standardowych modeli takich jak
  - przepływ w sieci
  - najkrótszej ścieżki
  - kolorowania grafu
  - drzewa rozpinającego
- Mogą one być używane w wielu rzeczywistych sytuacjach.
- Dla typowych modeli znanych jest wiele specjalistycznych algorytmów, które rozwiązują je w sposób efektywny.
- Jednak ich zastosowanie jest ograniczone i wystarczy niewielka zmiana założeń, a przestają pasować do naszego modelu.
- Wtedy należy używać ogólnych metod modelowania.
- Wiele sytuacji można zamodelować na więcej niż jeden sposób.
- Korzystanie z więcej niż jednego modelu może być cenne:
  - jeśli dają różne rozwiązania, różnice mogą wiele wnieść w zrozumienie problemu.
  - jeśli dają podobne rozwiązania, są bardziej wiarygodne, w kolejnych zastosowaniach można wykorzystywać już tylko jeden model – ten, którego rozwiązania znajduje się szybciej.

Wartość modelu

Wartość modelu

- Modelowanie często jest koniecznością, np. gdy rozmiar problemu nie pozwala na rozwiązanie go wprost.

Wartość modelu

- Modelowanie często jest koniecznością, np. gdy rozmiar problemu nie pozwala na rozwiązanie go wprost.
- Modelowanie często spotyka sie z krytyką.

Wartość modelu

- Modelowanie często jest koniecznością, np. gdy rozmiar problemu nie pozwala na rozwiązanie go wprost.
- Modelowanie często spotyka sie z krytyką.
  - Trudno jest dostarczyć wszystkich danych niezbędnych dla modelu.

Wartość modelu

- Modelowanie często jest koniecznością, np. gdy rozmiar problemu nie pozwala na rozwiązanie go wprost.
- Modelowanie często spotyka sie z krytyką.
  - Trudno jest dostarczyć wszystkich danych niezbędnych dla modelu.
  - Jeśli w modelu występuje 100 000 danych liczbowych, a niektóre z nich są niepewne, to czy rozwiązanie pochodzące z modelu jest wiarygodne?

Wartość modelu

- Modelowanie często jest koniecznością, np. gdy rozmiar problemu nie pozwala na rozwiązanie go wprost.
- Modelowanie często spotyka sie z krytyką.
  - Trudno jest dostarczyć wszystkich danych niezbędnych dla modelu.
  - Jeśli w modelu występuje 100 000 danych liczbowych, a niektóre z nich są niepewne, to czy rozwiązanie pochodzące z modelu jest wiarygodne?
  - Niektóre zjawiska trudno wyrazić przez liczby np. użyteczność lub wartość społeczna.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.
  - Oczywiście model i rozwiązanie muszą być starannie sprawdzone i ew. poprawione.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.
  - Oczywiście model i rozwiązanie muszą być starannie sprawdzone i ew. poprawione.
  - W procesie decyzyjnym często dopuszczamy się wielu uproszczeń i uogólnień.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.
  - Oczywiście model i rozwiązanie muszą być starannie sprawdzone i ew. poprawione.
  - W procesie decyzyjnym często dopuszczamy się wielu uproszczeń i uogólnień.
  - Być może próba opisania pewnych zjawisk przez liczby jest najlepszym rozwiązaniem.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.
  - Oczywiście model i rozwiązanie muszą być starannie sprawdzone i ew. poprawione.
  - W procesie decyzyjnym często dopuszczamy się wielu uproszczeń i uogólnień.
  - Być może próba opisania pewnych zjawisk przez liczby jest najlepszym rozwiązaniem.
- Z drugiej strony nie można ślepo wierzyć rozwiązaniom pochodzącym z modelu.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.
  - Oczywiście model i rozwiązanie muszą być starannie sprawdzone i ew. poprawione.
  - W procesie decyzyjnym często dopuszczamy się wielu uproszczeń i uogólnień.
  - Być może próba opisania pewnych zjawisk przez liczby jest najlepszym rozwiązaniem.
- Z drugiej strony nie można ślepo wierzyć rozwiązaniom pochodzącym z modelu.
- Trzeba pamiętać, że model tylko w **przybliżeniu** opisuje własności rzeczywistego obiektu.

Wartość modelu

- Z drugiej strony, istnieje szereg argumentów wspierających modelowanie.
  - Nawet jeśli niektóre dane są błędne, rozwiązanie może być użyteczne.
  - Oczywiście model i rozwiązanie muszą być starannie sprawdzone i ew. poprawione.
  - W procesie decyzyjnym często dopuszczamy się wielu uproszczeń i uogólnień.
  - Być może próba opisania pewnych zjawisk przez liczby jest najlepszym rozwiązaniem.
- Z drugiej strony nie można ślepo wierzyć rozwiązaniom pochodzącym z modelu.
- Trzeba pamiętać, że model tylko w **przybliżeniu** opisuje własności rzeczywistego obiektu.
- Może okazać się, że czynniki nieujęte w modelu są kluczowe.

Przykład 1: Problem mieszania

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.
- Czyste oleje dzielą się na dwie podstawowe grupy:

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.
- Czyste oleje dzielą się na dwie podstawowe grupy:
    - Oleje roślinne: `VEG1`, `VEG2`

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.
- Czyste oleje dzielą się na dwie podstawowe grupy:
  - Oleje roślinne: `VEG1`, `VEG2`
  - Pozostałe oleje: `OIL1`, `OIL2`, `OIL3`

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.
- Czyste oleje dzielą się na dwie podstawowe grupy:
  - Oleje roślinne: `VEG1`, `VEG2`
  - Pozostałe oleje: `OIL1`, `OIL2`, `OIL3`

- Rafinacja olejów roślinnych i nieroślinnych jestm innym procesem technologicznym.

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.
- Czyste oleje dzielą się na dwie podstawowe grupy:
  - Oleje roślinne: `VEG1`, `VEG2`
  - Pozostałe oleje: `OIL1`, `OIL2`, `OIL3`
- Rafinacja olejów roślinnych i nieroślinnych jestm innym procesem technologicznym.
- Wytwórnia jest w stanie przetworzyć co najwyżej 200 ton olejów roślinnych i 250 ton nieroślinnych.

Przykład 1: Problem mieszania

- Olej kuchenny powstaje przez rafinację czystych olejów i mieszanie ich razem.
- Czyste oleje dzielą się na dwie podstawowe grupy:
  - Oleje roślinne: `VEG1`, `VEG2`
  - Pozostałe oleje: `OIL1`, `OIL2`, `OIL3`

- Rafinacja olejów roślinnych i nieroślinnych jestm innym procesem technologicznym.
- Wytwórnia jest w stanie przetworzyć co najwyżej 200 ton olejów roślinnych i 250 ton nieroślinnych.
- Zakładamy, że w procesie rafinacji nie ma strat masy, a jego koszt może być pominięty.

Przykład 1: Problem mieszania

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.
- Zakładamy, twardość składników liniowo wpływa na twardość mieszaniny.

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.
- Zakładamy, twardość składników liniowo wpływa na twardość mieszaniny.
- Koszt jednej tony i twardość czystych olejów przedstawia tabela.

	$VEG1$	$VEG2$	$OIL1$	$OIL2$	$OIL3$
Koszt	110	120	130	110	115
Twardość	8.8	6.1	2.0	4.2	5.0

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.
- Zakładamy, twardość składników liniowo wpływa na twardość mieszaniny.
- Koszt jednej tony i twardość czystych olejów przedstawia tabela.

	$VEG1$	$VEG2$	$OIL1$	$OIL2$	$OIL3$
Koszt	110	120	130	110	115
Twardość	8.8	6.1	2.0	4.2	5.0

- Cena gotowego oleju to 150 zł za tonę.

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.
- Zakładamy, twardość składników liniowo wpływa na twardość mieszaniny.
- Koszt jednej tony i twardość czystych olejów przedstawia tabela.

	$VEG1$	$VEG2$	$OIL1$	$OIL2$	$OIL3$
Koszt	110	120	130	110	115
Twardość	8.8	6.1	2.0	4.2	5.0

- Cena gotowego oleju to 150 zł za tonę.
- Jak wytwórca powinien zaplanować produkcję, by zmaksymalizować swój zysk?

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.
- Zakładamy, twardość składników liniowo wpływa na twardość mieszaniny.
- Koszt jednej tony i twardość czystych olejów przedstawia tabela.

	$VEG1$	$VEG2$	$OIL1$	$OIL2$	$OIL3$
Koszt	110	120	130	110	115
Twardość	8.8	6.1	2.0	4.2	5.0

- Cena gotowego oleju to 150 zł za tonę.
- Jak wytwórca powinien zaplanować produkcję, by zmaksymalizować swój zysk?
- Przykład ten jest typowym zastosowaniem modeli PL.

Przykład 1: Problem mieszania

- Dodatkowo istnieje technologiczne ograniczenie na *twardość* oleju.
- Twardość gotowego produktu powinna być pomiędzy 3 a 6 jednostkami.
- Zakładamy, twardość składników liniowo wpływa na twardość mieszaniny.
- Koszt jednej tony i twardość czystych olejów przedstawia tabela.

	$VEG1$	$VEG2$	$OIL1$	$OIL2$	$OIL3$
Koszt	110	120	130	110	115
Twardość	8.8	6.1	2.0	4.2	5.0

- Cena gotowego oleju to 150 zł za tonę.
- Jak wytwórca powinien zaplanować produkcję, by zmaksymalizować swój zysk?
- Przykład ten jest typowym zastosowaniem modeli PL.
- Problemy występujące w praktyce sa oczywiście znacznie większe.

Problem mieszania jako zagadnienie PL

Problem mieszania jako zagadnienie PL

- Zmienne $x_1, x_2, \ldots, x_5$ reprezentują ilości (w tonach) czystych olejów `VEG1`, `VEG2`, `OIL1`, `OIL2`, `OIL3`, które powinny być kupione i przetworzone w ciągu miesiąca.

Problem mieszania jako zagadnienie PL

- Zmienne $x_1, x_2, \ldots, x_5$ reprezentują ilości (w tonach) czystych olejów `VEG1`, `VEG2`, `OIL1`, `OIL2`, `OIL3`, które powinny być kupione i przetworzone w ciągu miesiąca.
- Zmienna y reprezentuje ilość gotowego oleju, który powinien zostać wyprodukowany.

Problem mieszania jako zagadnienie PL

- Zmienne $x_1, x_2, \ldots, x_5$ reprezentują ilości (w tonach) czystych olejów `VEG1, VEG2, OIL1, OIL2, OIL3`, które powinny być kupione i przetworzone w ciągu miesiąca.
- Zmienna y reprezentuje ilość gotowego oleju, który powinien zostać wyprodukowany.
- Naszym celem jest maksymalizacja zysku (z uwzględnieniem kosztów zakupu składników):

$$-100x_1 - 120x_2 - 130x_3 - 110x_4 - 115x_5 + 150y$$

Problem mieszania jako zagadnienie PL

- Zmienne $x_1, x_2, \ldots, x_5$ reprezentują ilości (w tonach) czystych olejów `VEG1, VEG2, OIL1, OIL2, OIL3`, które powinny być kupione i przetworzone w ciągu miesiąca.
- Zmienna y reprezentuje ilość gotowego oleju, który powinien zostać wyprodukowany.
- Naszym celem jest maksymalizacja zysku (z uwzględnieniem kosztów zakupu składników):
$$-100x_1 - 120x_2 - 130x_3 - 110x_4 - 115x_5 + 150y$$
- Limity ilości przetworzonych olejów roślinnych i nieroślinnych nakładają ograniczenia na zmienne.
  - $x_1 + x_2 \leq 200$
  - $x_3 + x_4 + x_5 \leq 250$

Problem mieszania jako zagadnienie PL

- Zmienne $x_1, x_2, \ldots, x_5$ reprezentują ilości (w tonach) czystych olejów `VEG1`, `VEG2`, `OIL1`, `OIL2`, `OIL3`, które powinny być kupione i przetworzone w ciągu miesiąca.
- Zmienna y reprezentuje ilość gotowego oleju, który powinien zostać wyprodukowany.
- Naszym celem jest maksymalizacja zysku (z uwzględnieniem kosztów zakupu składników):
$$-100x_1 - 120x_2 - 130x_3 - 110x_4 - 115x_5 + 150y$$
- Limity ilości przetworzonych olejów roślinnych i nieroślinnych nakładają ograniczenia na zmienne.
  - $x_1 + x_2 \leq 200$
  - $x_3 + x_4 + x_5 \leq 250$
- Ograniczenia na twardość finalnego produktu dają dwa dodatkowe ograniczenia.
  - $8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 - 6y \leq 0$
  - $8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 - 3y \geq 0$

- Zmienne $x_1, x_2, \ldots, x_5$ reprezentują ilości (w tonach) czystych olejów `VEG1`, `VEG2`, `OIL1`, `OIL2`, `OIL3`, które powinny być kupione i przetworzone w ciągu miesiąca.
- Zmienna y reprezentuje ilość gotowego oleju, który powinien zostać wyprodukowany.
- Naszym celem jest maksymalizacja zysku (z uwzględnieniem kosztów zakupu składników):
$$-100x_1 - 120x_2 - 130x_3 - 110x_4 - 115x_5 + 150y$$
- Limity ilości przetworzonych olejów roślinnych i nieroślinnych nakładają ograniczenia na zmienne.
  - $x_1 + x_2 \leq 200$
  - $x_3 + x_4 + x_5 \leq 250$
- Ograniczenia na twardość finalnego produktu dają dwa dodatkowe ograniczenia.
  - $8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 - 6y \leq 0$
  - $8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 - 3y \geq 0$
- Suma wag produktów jest równa wadze gotowego oleju.
$$x_1 + x_2 + x_3 + x_4 + x_5 - y = 0$$

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:
  - Zimpl – wysokopoziomowy język do modelowania.

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:
  - Zimpl – wysokopoziomowy język do modelowania.
  - SoPlex – solver liniowy (dualna metoda sympleks).

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:
  - Zimpl – wysokopoziomowy język do modelowania.
  - SoPlex – solver liniowy (dualna metoda sympleks).
  - SCIP – ogólny solver dla problemów programowania całkowitoliczbowego i mieszanego.

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:
  - Zimpl – wysokopoziomowy język do modelowania.
  - SoPlex – solver liniowy (dualna metoda sympleks).
  - SCIP – ogólny solver dla problemów programowania całkowitoliczbowego i mieszanego.
- Pakiet ZIBopt jest bezpłatny dla celów niekomercyjnych (i stosunkowo tani do celów komercyjnych).

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:
  - Zimpl – wysokopoziomowy język do modelowania.
  - SoPlex – solver liniowy (dualna metoda sympleks).
  - SCIP – ogólny solver dla problemów programowania całkowitoliczbowego i mieszanego.
- Pakiet ZIBopt jest bezpłatny dla celów niekomercyjnych (i stosunkowo tani do celów komercyjnych).
- brak ograniczeń na rozmiar problemu wejściowego (jak to ma miejsce w wielu innych solverach)

Formułowanie zagadnień programowania matematycznego

- Formułowanie dużych zagadnień programowania liniowego (lub całkowitoliczbowego) za pomocą arkuszy kalkulacyjnych (takich jak Excel) może być niewygodne i pracochłonne.
- Dużo wygodniejszym sposobem jest użycia oprogramowania zwanego **generatorami macierzy** (matrix generators – MG).
- MG umożliwiają formułowanie ograniczeń i celów za pomocą wysokopoziomowej składni - my będziemy używać Zimpl.
- **Zimpl**. został stworzony przez Thorstena Kocha (Zuse Institut Berlin) jako część jego rozprawy doktorskiej (2004).
- Jest częścią pakietu **ZIBopt Solver Suite**:
  - Zimpl – wysokopoziomowy język do modelowania.
  - SoPlex – solver liniowy (dualna metoda sympleks).
  - SCIP – ogólny solver dla problemów programowania całkowitoliczbowego i mieszanego.
- Pakiet ZIBopt jest bezpłatny dla celów niekomercyjnych (i stosunkowo tani do celów komercyjnych).
- brak ograniczeń na rozmiar problemu wejściowego (jak to ma miejsce w wielu innych solverach)
- Jako oprogramowanie na licencji open source może być łatwo modyfikowany.

Jak zdobyć ZIBopt Solver Suite?

http://scip.zib.de/ - > Download ->
Binaries:
Windows/PC, 64bit, vc10: linked to SoPlex 1.7.0, Zimpl 3.3.0
Zaznaczyć I certify that I will use the software only as a member of a noncommercial and academic institute and that I have read and accepted the ZIB ACADEMIC LICENSE. -> start download
Rozpokować -> scip-3.0.0.win.x86_64.vc10.opt.spx.mt

Następnie -> ZIMPL (http://zimpl.zib.de/)
Precompiled binaries are also available.
zimpl-3.3.0.win.x86_64.vc10.normal.opt.exe

Inne generatory macierzy i solvery

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com
- Istnieją także inne solvery

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com
- Istnieją także inne solvery
  - Gurobi (płatny), http://www.gurobi.com

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com
- Istnieją także inne solvery
  - Gurobi (płatny), http://www.gurobi.com
  - IBM ILOG CPlex (płatny), http://www.ilog.com/products/cplex

Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com
- Istnieją także inne solvery
  - Gurobi (płatny), http://www.gurobi.com
  - IBM ILOG CPlex (płatny), http://www.ilog.com/products/cplex
  - XPress (płatny), http://www.dashoptimization.com

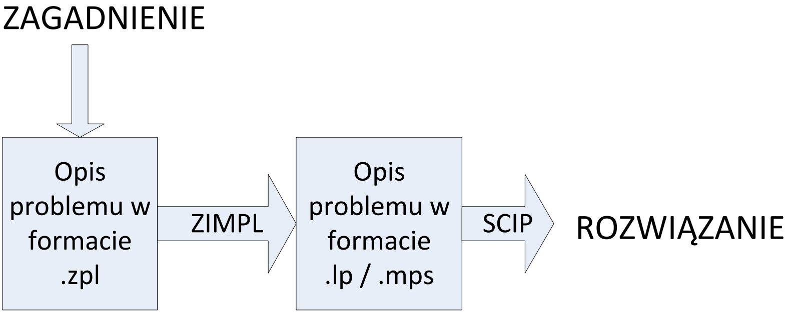
Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com
- Istnieją także inne solvery
  - Gurobi (płatny), http://www.gurobi.com
  - IBM ILOG CPlex (płatny), http://www.ilog.com/products/cplex
  - XPress (płatny), http://www.dashoptimization.com
  - Lindo What's Best (płatny), http://www.lindo.com

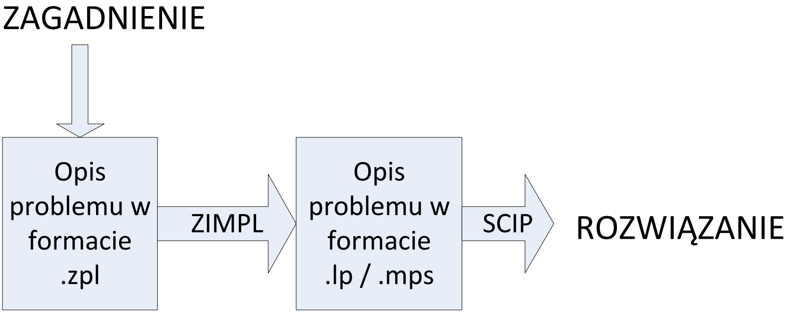
Inne generatory macierzy i solvery

- Istnieje też wiele innych (najczęściej płatnych) generatorów macierzy
  - GAMS, http://www.gams.com
  - AMPL, http://www.ampl.com
  - AIMMS, http://www.aimms.com
- Istnieją także inne solvery
  - Gurobi (płatny), http://www.gurobi.com
  - IBM ILOG CPlex (płatny), http://www.ilog.com/products/cplex
  - XPress (płatny), http://www.dashoptimization.com
  - Lindo What's Best (płatny), http://www.lindo.com
  - COIN-OR Cbc (bezpłatny), http://projects.coin-or.org/Cbc

Rozwiązywanie zagadnienia programowania matematycznego w ZIBOpt



Rozwiązywanie zagadnienia programowania matematycznego w ZIBOpt



Zobaczmy przykład wyk1p1.zpl

Formaty plików

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.
- Drugi plik w większości przypadków może zostać zignorowany (o ile nie używamy długich nazw).

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.
- Drugi plik w większości przypadków może zostać zignorowany (o ile nie używamy długich nazw).
- Format `.lp` jest rozpoznawany przez wiele solverów.

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.
- Drugi plik w większości przypadków może zostać zignorowany (o ile nie używamy długich nazw).
- Format `.lp` jest rozpoznawany przez wiele solverów.
- Niektóre solvery wymagają formatu `.mps`.

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.
- Drugi plik w większości przypadków może zostać zignorowany (o ile nie używamy długich nazw).
- Format `.lp` jest rozpoznawany przez wiele solverów.
- Niektóre solvery wymagają formatu `.mps`.
- `.mps` był kiedyś podstawowym formatem używanym do zapisu problemów programowania matematycznego. Jest on mało czytelny dla człowieka, za to łatwy to zapisu na kartach dziurkowanych.

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.
- Drugi plik w większości przypadków może zostać zignorowany (o ile nie używamy długich nazw).
- Format `.lp` jest rozpoznawany przez wiele solverów.
- Niektóre solvery wymagają formatu `.mps`.
- `.mps` był kiedyś podstawowym formatem używanym do zapisu problemów programowania matematycznego. Jest on mało czytelny dla człowieka, za to łatwy to zapisu na kartach dziurkowanych.
- Jeśli potrzebujemy pliku w formacie `.mps`, należy wywołać `zimpl -t mps plik.zpl`

Formaty plików

- Wywołanie `zimpl plik.zpl` utworzy dwa pliki:
  - Problem zapisany w pliku `plik.lp`.
  - Plik pomocniczy z kodowaniem długich nazw zmiennych i ograniczeń `plik.tbl`.
- Drugi plik w większości przypadków może zostać zignorowany (o ile nie używamy długich nazw).
- Format `.lp` jest rozpoznawany przez wiele solverów.
- Niektóre solvery wymagają formatu `.mps`.
- `.mps` był kiedyś podstawowym formatem używanym do zapisu problemów programowania matematycznego. Jest on mało czytelny dla człowieka, za to łatwy to zapisu na kartach dziurkowanych.
- Jeśli potrzebujemy pliku w formacie `.mps`, należy wywołać `zimpl -t mps plik.zpl`
- Na szczęście SCIP obsługuje format `.lp`.

Jak rozwiązać zagadnienie ?

Jak rozwiązać zagadnienie ?

1. Tak przygotowany plik należy uruchomić za pomocą Zimpla. `zimpl-3.3.0.win.x86_64.vc10.normal.opt [nazwa pliku].zpl`

Jak rozwiązać zagadnienie ?

1. Tak przygotowany plik należy uruchomić za pomocą Zimpla. `zimpl-3.3.0.win.x86_64.vc10.normal.opt [nazwa pliku].zpl`
2. Użyteczne będzie stworzenie pliku `.bat` i uruchamianie Zimpla za jego pomocą

```
zimpl-3.3.0.win.x86_64.vc10.normal.opt wyk1p1.zpl
pause
```

Jak rozwiązać zagadnienie ?

1. Tak przygotowany plik należy uruchomić za pomocą Zimpla. `zimpl-3.3.0.win.x86_64.vc10.normal.opt [nazwa pliku].zpl`
2. Użyteczne będzie stworzenie pliku `.bat` i uruchamianie Zimpla za jego pomocą
```
zimpl-3.3.0.win.x86_64.vc10.normal.opt wyk1p1.zpl
pause
```
3. Tak czy inaczej, Zimpl stworzy nam plik `.lp`, który tylko czeka na rozwiązanie.

Jak rozwiązać zagadnienie ?

1. Tak przygotowany plik należy uruchomić za pomocą Zimpla. `zimpl-3.3.0.win.x86_64.vc10.normal.opt [nazwa pliku].zpl`
2. Użyteczne będzie stworzenie pliku `.bat` i uruchamianie Zimpla za jego pomocą
   ```
   zimpl-3.3.0.win.x86_64.vc10.normal.opt wyk1p1.zpl
   pause
   ```
3. Tak czy inaczej, Zimpl stworzy nam plik `.lp`, który tylko czeka na rozwiązanie.
4. Uruchamiamy SCIP i w konsolce wpisujemy `r wyk1p1.lp` (*read*).

Jak rozwiązać zagadnienie ?

1. Tak przygotowany plik należy uruchomić za pomocą Zimpla. `zimpl-3.3.0.win.x86_64.vc10.normal.opt [nazwa pliku].zpl`
2. Użyteczne będzie stworzenie pliku `.bat` i uruchamianie Zimpla za jego pomocą
```
 zimpl-3.3.0.win.x86_64.vc10.normal.opt wyk1p1.zpl
pause
```
3. Tak czy inaczej, Zimpl stworzy nam plik `.lp`, który tylko czeka na rozwiązanie.
4. Uruchamiamy SCIP i w konsolce wpisujemy `r wyk1p1.lp` (*read*).
5. Następnie wpisujemy op (*optimize*).

Jak rozwiązać zagadnienie ?

1. Tak przygotowany plik należy uruchomić za pomocą Zimpla. `zimpl-3.3.0.win.x86_64.vc10.normal.opt [nazwa pliku].zpl`
2. Użyteczne będzie stworzenie pliku `.bat` i uruchamianie Zimpla za jego pomocą
```
zimpl-3.3.0.win.x86_64.vc10.normal.opt wyk1p1.zpl
pause
```
3. Tak czy inaczej, Zimpl stworzy nam plik `.lp`, który tylko czeka na rozwiązanie.
4. Uruchamiamy SCIP i w konsolce wpisujemy `r wyk1p1.lp` (*read*).
5. Następnie wpisujemy op (*optimize*).
6. Aby wyświetlić rozwiązanie wpisujemy `dis sol` (*display solution*).

Format .zpl

```
 set X :=1,2,3,4;
param c[<x> in X] := if x < 3 then 10 else 20 end;
var v[X] real >= 2 <= 15;
minimize cost: sum<x> in X do x*v[x];
subto c1: forall <x> in X do v[x]<= c[x];
```

Format .lp

```
 Minimize
cost: + v#1 +2 v#2 +3 v#3 +4 v#4
Subject to
c1_1:
+ v#1 <= 10
c1_2:
+ v#2 <= 10
c1_3:
+ v#3 <= 20
c1_4:
+ v#4 <= 20
Bounds
2 <= v#1 <= 15
2 <= v#2 <= 15
2 <= v#3 <= 15
2 <= v#4 <= 15
End
```

Format .mps

```
 NAME problem.
ROWS
N OBJECTIV
L c1_1
L c1_2
L c1_3
L c1_4
COLUMNS MARK0000 'MARKER'
'INTORG'
MARK0001 'MARKER' 'INTEND'
v#1 OBJECTIV 1
v#1 c1_1 1
v#2 OBJECTIV 2
v#2 c1_2 1
v#3 OBJECTIV 3
v#3 c1_3 1
v#4 OBJECTIV 4
v#4 c1_4 1
RHS
RHS c1_1 10
RHS c1_2 10
RHS c1_3 20
RHS c1_4 20
BOUNDS
LO BOUND v#1 2
UP BOUND v#1 15
LO BOUND v#2 2
UP BOUND v#2 15
LO BOUND v#3 2
UP BOUND v#3 15
LO BOUND v#4 2
UP BOUND v#4 15
ENDATA
```

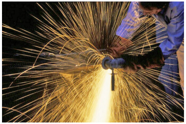
Przykład 2: optymalizacja produkcji

Przykład 2: optymalizacja produkcji

- Fabryka produkuje 5 typów produktów `PROD1`, `PROD2`, `PROD3`, `PROD4`, `PROD5`.

Przykład 2: optymalizacja produkcji

- Fabryka produkuje 5 typów produktów `PROD1`, `PROD2`, `PROD3`, `PROD4`, `PROD5`.
- W fabryce przeprowadza się dwa rodzaje procesów technologicznych – szlifowanie i wiercenie.



Przykład 2: optymalizacja produkcji

Przykład 2: optymalizacja produkcji

- Każda jednostka każdego produktu przynosi pewien zysk

PROD1	PROD2	PROD3	PROD4	PROD5
550	600	350	400	200

Przykład 2: optymalizacja produkcji

- Każda jednostka każdego produktu przynosi pewien zysk

PROD1	PROD2	PROD3	PROD4	PROD5
550	600	350	400	200

- Każdy produkt wymaga pewnego czasu przygotowania (– oznacza że dany proces nie jest potrzebny)

	PROD1	PROD2	PROD3	PROD4	PROD5
Szlifowanie	12	20	–	25	15
Wiercenie	10	8	16	–	–

Przykład 2: optymalizacja produkcji

- Każda jednostka każdego produktu przynosi pewien zysk

PROD1	PROD2	PROD3	PROD4	PROD5
550	600	350	400	200

- Każdy produkt wymaga pewnego czasu przygotowania (– oznacza że dany proces nie jest potrzebny)

	PROD1	PROD2	PROD3	PROD4	PROD5
Szlifowanie	12	20	–	25	15
Wiercenie	10	8	16	–	–

- Ponadto każda jednostka każdego produktu wymaga 20 godzin pracy technika.

Przykład 2: optymalizacja produkcji

Przykład 2: optymalizacja produkcji

- Fabryka posiada cztery urządzenia szlifujące i dwa wiercące.

Przykład 2: optymalizacja produkcji

- Fabryka posiada cztery urządzenia szlifujące i dwa wiercące.

- Tydzień pracy trwa sześć dni, z 8 godzinami roboczymi każdego dnia.

Przykład 2: optymalizacja produkcji

- Fabryka posiada cztery urządzenia szlifujące i dwa wiercące.

- Tydzień pracy trwa sześć dni, z 8 godzinami roboczymi każdego dnia.
- Fabryka zatrudnia 8 techników.

Przykład 2: optymalizacja produkcji

- Fabryka posiada cztery urządzenia szlifujące i dwa wiercące.

- Tydzień pracy trwa sześć dni, z 8 godzinami roboczymi każdego dnia.
- Fabryka zatrudnia 8 techników.
- Jak powinna być zaplanowana produkcja, by zmaksymalizować zysk fabryki?

Przykład 2: optymalizacja produkcji

- Fabryka posiada cztery urządzenia szlifujące i dwa wiercące.

- Tydzień pracy trwa sześć dni, z 8 godzinami roboczymi każdego dnia.
- Fabryka zatrudnia 8 techników.
- Jak powinna być zaplanowana produkcja, by zmaksymalizować zysk fabryki?
- To pierwszy przykład zastosowania programowania liniowego do **problemu optymalizacji produkcji**.

Optymalizacja produkcji jako zagadnienie PL

- Wprowadzamy zmienne $x_1, \ldots, x_5$, które opisują liczbę jednostek każdego z produktów `PROD1, ..., PROD5`, produkowanych w ciągu tygodnia.

Optymalizacja produkcji jako zagadnienie PL

- Wprowadzamy zmienne $x_1, \ldots, x_5$, które opisują liczbę jednostek każdego z produktów `PROD1`, ..., `PROD5`, produkowanych w ciągu tygodnia.
- Każda jednostka `PROD1` przynosi 550 zysku, każda jednostka `PROD2` przynosi 600 zł zysku itd.

Optymalizacja produkcji jako zagadnienie PL

- Wprowadzamy zmienne $x_1, \ldots, x_5$, które opisują liczbę jednostek każdego z produktów PROD1, ..., PROD5, produkowanych w ciągu tygodnia.
- Każda jednostka PROD1 przynosi 550 zysku, każda jednostka PROD2 przynosi 600 zł zysku itd.
- Zatem całkowity zysk możemy opisać wyrażeniem

$$550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5$$

Optymalizacja produkcji jako zagadnienie PL

- Wprowadzamy zmienne $x_1, \ldots, x_5$, które opisują liczbę jednostek każdego z produktów PROD1, ..., PROD5, produkowanych w ciągu tygodnia.
- Każda jednostka PROD1 przynosi 550 zysku, każda jednostka PROD2 przynosi 600 zł zysku itd.
- Zatem całkowity zysk możemy opisać wyrażeniem

$$550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5$$

- Celem jest wybranie wartości $x_1, \ldots, x_5$ w taki sposób, by całkowity zysk był największy.

Optymalizacja produkcji jako zagadnienie PL

- Wprowadzamy zmienne $x_1, \ldots, x_5$, które opisują liczbę jednostek każdego z produktów `PROD1, ..., PROD5`, produkowanych w ciągu tygodnia.
- Każda jednostka `PROD1` przynosi 550 zysku, każda jednostka `PROD2` przynosi 600 zł zysku itd.
- Zatem całkowity zysk możemy opisać wyrażeniem

$$550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5$$

- Celem jest wybranie wartości $x_1, \ldots, x_5$ w taki sposób, by całkowity zysk był największy.
- Zatem zysk jest funkcją celu, którą należy zmaksymalizować.

Optymalizacja produkcji jako zagadnienie PL

- Wprowadzamy zmienne $x_1, \ldots, x_5$, które opisują liczbę jednostek każdego z produktów `PROD1, ..., PROD5`, produkowanych w ciągu tygodnia.
- Każda jednostka `PROD1` przynosi 550 zysku, każda jednostka `PROD2` przynosi 600 zł zysku itd.
- Zatem całkowity zysk możemy opisać wyrażeniem

$$550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5$$

- Celem jest wybranie wartości $x_1, \ldots, x_5$ w taki sposób, by całkowity zysk był największy.
- Zatem zysk jest funkcją celu, którą należy zmaksymalizować.
- Ograniczenia czasu pracy maszyn i ludzi ograniczają wartości, jakie mogą przyjąć zmienne $x_1, \ldots, x_5$.

- Szlifowanie:

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.

Optymalizacja produkcji jako zagadnienie PL

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.
  - Każda jednostka `PROD1` wymaga 12 godzin szlifowania.

Optymalizacja produkcji jako zagadnienie PL

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.
  - Każda jednostka `PROD1` wymaga 12 godzin szlifowania.
  - Zatem x_1 jednostek wymaga $12x_1$ godzin szlifowania.

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.
  - Każda jednostka `PROD1` wymaga 12 godzin szlifowania.
  - Zatem x_1 jednostek wymaga $12x_1$ godzin szlifowania.
  - Analogicznie x_2 jednostek `PROD2` wymaga $20x_2$ godzin szlifowania.

Optymalizacja produkcji jako zagadnienie PL

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.
  - Każda jednostka PROD1 wymaga 12 godzin szlifowania.
  - Zatem x_1 jednostek wymaga $12x_1$ godzin szlifowania.
  - Analogicznie x_2 jednostek PROD2 wymaga $20x_2$ godzin szlifowania.
  - Całkowity czas szlifowania jest zatem ograniczony przez maksymalny czas pracy maszyn.

$$12x_1 + 20x_2 + 25x_4 + 15x_5 \leq 144$$

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.
  - Każda jednostka `PROD1` wymaga 12 godzin szlifowania.
  - Zatem x_1 jednostek wymaga $12x_1$ godzin szlifowania.
  - Analogicznie x_2 jednostek `PROD2` wymaga $20x_2$ godzin szlifowania.
  - Całkowity czas szlifowania jest zatem ograniczony przez maksymalny czas pracy maszyn.

$$12x_1 + 20x_2 + 25x_4 + 15x_5 \leq 144$$

- Takie nierówności w modelu matematycznym nazywamy **ograniczeniami**.

Optymalizacja produkcji jako zagadnienie PL

- Szlifowanie:
  - Mamy 3 maszyny szlifujące.
  - Każda maszyna pracuje przez 48 godzin w tygodniu.
  - Zatem łączny czas szlifowania nie może przekroczyć 144 h.
  - Każda jednostka `PROD1` wymaga 12 godzin szlifowania.
  - Zatem x_1 jednostek wymaga $12x_1$ godzin szlifowania.
  - Analogicznie x_2 jednostek `PROD2` wymaga $20x_2$ godzin szlifowania.
  - Całkowity czas szlifowania jest zatem ograniczony przez maksymalny czas pracy maszyn.

$$12x_1 + 20x_2 + 25x_4 + 15x_5 \leq 144$$

- Takie nierówności w modelu matematycznym nazywamy **ograniczeniami**.
- **Ograniczają** one wartości, jakie mogą przyjąć zmienne $x_1, \ldots, x_5$.

Optymalizacja produkcji jako zagadnienie PL

- Wiercenie:

Optymalizacja produkcji jako zagadnienie PL

- Wiercenie:
  - Mamy 2 maszyny wiercące.

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.

Optymalizacja produkcji jako zagadnienie PL

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.
  - Stąd następujące ograniczenie:

$$10x_1 + 8x_2 + 16x_3 \leq 96$$

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.
  - Stąd następujące ograniczenie:

$$10x_1 + 8x_2 + 16x_3 \leq 96$$

- Praca techników:

Optymalizacja produkcji jako zagadnienie PL

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.
  - Stąd następujące ograniczenie:

$$10x_1 + 8x_2 + 16x_3 \leq 96$$

- Praca techników:
  - Mamy 8 techników, pracujących po 48 godzin w tygodniu.

Optymalizacja produkcji jako zagadnienie PL

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.
  - Stąd następujące ograniczenie:

$$10x_1 + 8x_2 + 16x_3 \leq 96$$

- Praca techników:
  - Mamy 8 techników, pracujących po 48 godzin w tygodniu.
  - Zatem łączny czas pracy nie może przekroczyć 384 h.

Optymalizacja produkcji jako zagadnienie PL

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.
  - Stąd następujące ograniczenie:
  $$10x_1 + 8x_2 + 16x_3 \leq 96$$
- Praca techników:
  - Mamy 8 techników, pracujących po 48 godzin w tygodniu.
  - Zatem łączny czas pracy nie może przekroczyć 384 h.
  - Ponieważ każda jednostka każdego produktu wymaga 20 h pracy technika, otrzymujemy następujące ograniczenie:
  $$20x_1 + 20x_2 + 20x_3 + 20x_4 + 20x_5 \leq 384$$

- Wiercenie:
  - Mamy 2 maszyny wiercące.
  - Zatem łączny czas wiercenia nie może przekroczyć 96 h.
  - Stąd następujące ograniczenie:

$$10x_1 + 8x_2 + 16x_3 \leq 96$$

- Praca techników:
  - Mamy 8 techników, pracujących po 48 godzin w tygodniu.
  - Zatem łączny czas pracy nie może przekroczyć 384 h.
  - Ponieważ każda jednostka każdego produktu wymaga 20 h pracy technika, otrzymujemy następujące ograniczenie:

$$20x_1 + 20x_2 + 20x_3 + 20x_4 + 20x_5 \leq 384$$

- Tym sposobem przekształciliśmy wyjściowy problem praktyczny do modelu matematycznego:

$$\begin{array}{rrcrcrcrcr}
\max & 550x_1 & + & 600x_2 & + & 350x_3 & + & 400x_4 & + & 200x_5 \\
\text{przy ogr.} & 12x_1 & + & 20x_2 & & & + & 25x_4 & + & 15x_5 \\
 & 10x_1 & + & 8x_2 & + & 10x_3 & & & & \\
 & 22x_1 & + & 20x_2 & + & 20x_3 & + & 20x_4 & + & 20x_5
\end{array}$$

Kilka uwag odnośnie modelu

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:
  $$x_1^2, x_1 x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$
- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$

  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \dots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:
$$x_1^2, x_1x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$
- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń
$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1 x_2, \tfrac{x_1}{x_2}, \log(x), \sin(x)$$

  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.

Kilka uwag odnośnie modelu

- Chcemy dobrać dla zmiennych $x_1, \ldots, x_5$ wartości, które zmaksymalizują funkcję celu przy zachowaniu ograniczeń.
- Ten model **modelem programowania liniowego (PL)**.
  - W funkcji celu i wszystkich ograniczeniach zmienne pojawiają się tylko w sumach we stałymi współczynnikami.
  - Nie ma zatem wyrażeń takich jak:

$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

- W modelu poczyniliśmy pewne niejawne założenia.
  - Musimy założyć, że zmienne nie mogą mieć wartości ujemnych.
  - Można to jawnie wyrazić za pomocą dodatkowych ograniczeń

$$x_1, x_2, x_3, x_4, x_5, \geq 0$$
$$x_1^2, x_1 x_2, \frac{x_1}{x_2}, \log(x), \sin(x)$$

  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.

Kilka uwag odnośnie modelu

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.
  - Jeśli zmienna opisuje dyskretne jednostki produktu (np. liczba wyprodukowanych silników), ułamkowe wartości nie mają sensu.

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.
  - Jeśli zmienna opisuje dyskretne jednostki produktu (np. liczba wyprodukowanych silników), ułamkowe wartości nie mają sensu.
  - Jeśli liczby są duże, możemy je zaokrąglić do najbliższej liczby całkowitej.

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.
  - Jeśli zmienna opisuje dyskretne jednostki produktu (np. liczba wyprodukowanych silników), ułamkowe wartości nie mają sensu.
  - Jeśli liczby są duże, możemy je zaokrąglić do najbliższej liczby całkowitej.
  - Np. jeśli zmienna wskazuje, że optymalnie jest wyprodukować 298 278,374 silniki, możemy po prostu zaokrąglić tę liczbę w dół do 298 278.

Kilka uwag odnośnie modelu

- W modelu poczyniliśmy pewne niejawne założenia.
  - Ponadto założyliśmy, że zmienne mogą mieć wartości niecałkowite.
  - Ale co to znaczy, że mamy wyprodukować 4.68 jednostek `PROD1`?
  - Jeśli `PROD1` jest produktem ciągłym (mleko, piwo, olej, zboże) ułamki mogą być dopuszczalne.
  - Jeśli zmienna opisuje dyskretne jednostki produktu (np. liczba wyprodukowanych silników), ułamkowe wartości nie mają sensu.
  - Jeśli liczby są duże, możemy je zaokrąglić do najbliższej liczby całkowitej.
  - Np. jeśli zmienna wskazuje, że optymalnie jest wyprodukować 298 278,374 silniki, możemy po prostu zaokrąglić tę liczbę w dół do 298 278.
  - Jeśli takie zaokrąglanie nie jest możliwe, należy zastosować metody **programowania całkowitoliczbowego**.

Kilka uwag odnośnie modelu

Kilka uwag odnośnie modelu

- Nasz pierwszy model obrazuje kilka kluczowych cech modeli programowania liniowego.

Kilka uwag odnośnie modelu

- Nasz pierwszy model obrazuje kilka kluczowych cech modeli programowania liniowego.
    - Istnieje jedno liniowe wyrażenie (**funkcja celu**), które chcemy zminimalizować lub zmaksymalizować.

Kilka uwag odnośnie modelu

- Nasz pierwszy model obrazuje kilka kluczowych cech modeli programowania liniowego.
    - Istnieje jedno liniowe wyrażenie (**funkcja celu**), które chcemy zminimalizować lub zmaksymalizować.
    - Ponadto dany jest szereg ograniczeń w postaci wyrażeń liniowych, które nie mogą przekraczać ($\leq$) pewnej zadanej wartości.

Kilka uwag odnośnie modelu

- Nasz pierwszy model obrazuje kilka kluczowych cech modeli programowania liniowego.
  - Istnieje jedno liniowe wyrażenie (**funkcja celu**), które chcemy zminimalizować lub zmaksymalizować.
  - Ponadto dany jest szereg ograniczeń w postaci wyrażeń liniowych, które nie mogą przekraczać ($\leq$) pewnej zadanej wartości.
  - Ograniczenia liniowe mogą być też typu $\geq$ lub $=$. Pokazują one, że wartość wyrażenia nie może być niższa niż pewna zadana wartość lub musi być jej równa.

Kilka uwag odnośnie modelu

- Nasz pierwszy model obrazuje kilka kluczowych cech modeli programowania liniowego.
  - Istnieje jedno liniowe wyrażenie (**funkcja celu**), które chcemy zminimalizować lub zmaksymalizować.
  - Ponadto dany jest szereg ograniczeń w postaci wyrażeń liniowych, które nie mogą przekraczać ($\leq$) pewnej zadanej wartości.
  - Ograniczenia liniowe mogą być też typu $\geq$ lub $=$. Pokazują one, że wartość wyrażenia nie może być niższa niż pewna zadana wartość lub musi być jej równa.
  - Współczynniki 144, 96, 384 są zazwyczaj nazywane **prawą stroną** zagadnienia PL.

Kilka uwag odnośnie modelu

- Nasz pierwszy model obrazuje kilka kluczowych cech modeli programowania liniowego.
  - Istnieje jedno liniowe wyrażenie (**funkcja celu**), które chcemy zminimalizować lub zmaksymalizować.
  - Ponadto dany jest szereg ograniczeń w postaci wyrażeń liniowych, które nie mogą przekraczać ($\leq$) pewnej zadanej wartości.
  - Ograniczenia liniowe mogą być też typu $\geq$ lub $=$. Pokazują one, że wartość wyrażenia nie może być niższa niż pewna zadana wartość lub musi być jej równa.
  - Współczynniki 144, 96, 384 są zazwyczaj nazywane **prawą stroną** zagadnienia PL.
- Rzeczywiste modele są zazwyczaj znacznie większe (mają więcej zmiennych i ograniczeń), ale zawsze mają te podstawowe cechy.