Programowanie liniowe, część 1

Iwo Błądek

Politechnika Poznańska

24 lutego 2019

Optymalizacja

Typowe problemy w informatyce

Problemy w informatyce dzielimy na:

- **decyzyjne**
  - czy w grafie istnieje między dwoma punktami droga o długości < 20 km?
  - czy można zakupić wymagane produkty przy zadanych ograniczeniach?
- **optymalizacyjne**
  - jaka jest najkrótsza droga między dwoma punktami w grafie?
  - ile maksymalnie produktów można kupić przy zadanych ograniczeniach?
- **przeszukiwania**
  - wskaż w grafie drogę o długości < 20 km.
  - wskaż taką liczbę zakupionych towarów, by ograniczenia były spełnione.
  - dla danej listy liczb zwróć listę z elementami ułożonymi malejąco.

Na czym polega optymalizacja?

Intuicyjnie: chcemy uczynić coś tak dobrym, jak to tylko możliwe w danej sytuacji.

Wartość optymalna to taka wartość, której nie można w żaden sposób poprawić w danej sytuacji/problemie.

Przykłady optymalizacji:

- przydział pracowników do zadań,
- decyzje dotyczące produkcji,
- zakupy,
- dystrybucja towarów,
- czas ładowania strony internetowej,
- ...

Podstawowe pojęcia związane z optymalizacją

- **Funkcja celu**
- **Zmienne decyzyjne**
- **Ograniczenia**
- **Przestrzeń rozwiązań**

Funkcja celu

- Zapisywana zwykle w postaci $f(x_1, \ldots, x_n)$.
- Wyraża liczbowo to, co chcemy zoptymalizować. Możemy chcieć, by wartość funkcji celu była jak najmniejsza (problem **minimalizacji**), albo jak największa (problem **maksymalizacji**).
- Jako że jest to funkcja, to przyjmuje pewne argumenty, tzw. **zmienne decyzyjne** $(x_1, \ldots, x_n)$.

Funkcją celu może być np. zysk z produkcji wyrażony w PLN, łączny koszt zakupionych produktów albo czas przejazdu wyrażony w godzinach.

Zmienne decyzyjne

- $x_1, \ldots, x_n$.
- Reprezentują to, na co mamy w problemie wpływ.
- Dla każdej kombinacji wartości zmiennych funkcja celu zwróci pewną wartość. Naszym celem jest znalezienie takich wartości zmiennych decyzyjnych, dla których funkcja celu zwróci optymalną wartość.
- Jeżeli przyjmują tylko wartości $\{0, 1\}$ to nazywamy je **zmiennymi binarnymi**.

Zmiennymi decyzyjnymi może być np. liczba produktów danego typu do zakupienia albo to, czy przydzielamy danego pracownika do jakiegoś zadania czy nie (wtedy zmienna binarna).

Ograniczenia

- Mówią nam o tym, które rozwiązania są **dopuszczalne** w danym problemie.
- Często obrazują ograniczenia na liczbę posiadanych zasobów jakiegoś typu.

Przykłady:

- dysponujemy budżetem 1000 zł i tylko tyle możemy sumarycznie wydać na pakiety akcji. Niedopuszczalne będą wszystkie przydziały wartości do zmiennych decyzyjnych prowadzące do przekroczenia tej sumy.

- niektórzy pracownicy potrafią wykonywać tylko niektóre zadania i nie można ich przydzielić do innych.

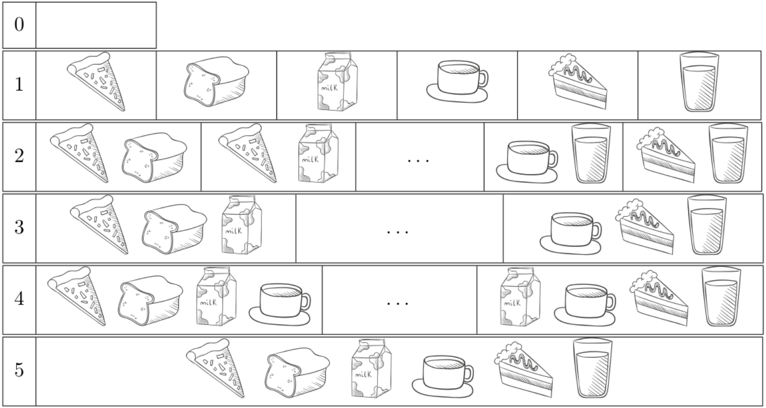
- w sklepie są różne ilości towarów i nie można kupić danego towaru więcej niż liczba jego sztuk na stanie (fizycznie niemożliwe).

Przestrzeń rozwiązań

- Jest to zbiór **wszystkich możliwych** dozwolonych kombinacji wartości zmiennych decyzyjnych.

Przykład: Jeżeli funkcja celu jest określona na dwóch binarnych zmiennych decyzyjnych a i b, to przestrzeń przeszukiwania złożona jest z czterech rozwiązań: $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$. Przedstawia ona wszystkie możliwe kombinacje przypisań wartości do tych dwóch zmiennych.

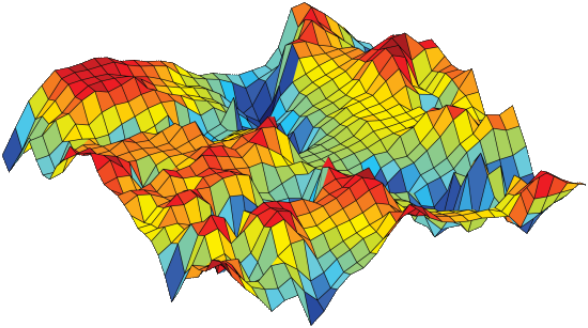
Może być **dyskretna**:



Obecność produktu w koszyku mówi o tym, czy w danym wypadku został zakupiony czy też nie (zmienna binarna). Z każdym takim koszykiem można powiązać pewną wartość funkcji celu.

Przestrzeń rozwiązań

Albo **ciągła**:



W tym wypadku zmienne decyzyjne (na osiach X i Y) mogą przyjąć dowolne rzeczywiste wartości. Wartości funkcji celu (na osi Z oraz kolor) na rysunku również są liczbami rzeczywistymi, ale w ogólności nie jest to konieczne.

Rodzaje algorytmów optymalizacyjnych

- **Algorytmy dokładne** – sprawdzają całą przestrzeń rozwiązań. Algorytmy te dają gwarancję, że otrzymany wynik rzeczywiście jest najlepszy z możliwych. Często potrzebują bardzo dużo czasu na uzyskanie wyniku.
- **Algorytmy heurystyczne** – nie dają gwarancji, że zwrócony wynik jest optymalny, jednak potrafią go uzyskać relatywnie szybko.

To, którego użyjemy w praktyce, będzie zależeć od naszych potrzeb i ograniczeń, w szczególności czasowych.

Programowanie liniowe

Programowanie liniowe

- Szczególny problem optymalizacji, w którym funkcja celu i ograniczenia mają postać **funkcji liniowej**, czyli zmienne decyzyjne mogą być mnożone tylko przez pewne stałe (więcej o tym za chwilę).
- Jak to często bywa w informatyce, nałożenie takich dodatkowych ograniczeń na strukturę problemu pozwala użyć wyspecjalizowanych algorytmów, które potrafią zwrócić wynik szybciej.
- Bardzo wiele problemów w informatyce można sprowadzić właśnie do zadania programowania liniowego.

Przykładowe zadanie dla programowania liniowego

Rolnik postanowił zasadzić sadzonki trzech typów – A, B i C. Każda z sadzonek zajmuje odpowiednią ilość miejsca: A – $2m^2$, B – $1.5m^2$, C – $2.5m^2$. Dodatkowo, każde drzewo potrzebuje odpowiedniej ilości nawozu: A – 10 jednostek, B – 15 jednostek, C – 20 jednostek. Po pewnym czasie nasz rolnik będzie miał z sadzonek zysk: A – 500zł, B – 400zł, C – 700zł. Mając ograniczony obszar pola ($500m^2$) i ograniczoną ilość nawozu (2000 jednostek), wybierz sadzonki tak, aby rolnik zarobił najwięcej.

Funkcja celu

- Każdej zmiennej decyzyjnej x_j przydzielona jest pewna **waga** c_j.
- Funkcja celu ma postać **sumy** wartości wszystkich zmiennych decyzyjnych **pomnożonych** przez ich wagi.
- Może być maksymalizowana lub minimalizowana.

$$\begin{aligned}(max)/(min)\ z &= \sum_{j=1}^{n} c_j \cdot x_j \\ &= c_1 \cdot x_1 + \cdots + c_n \cdot x_n\end{aligned}$$

Przykład

- "Za każdy sprzedany towar A sprzedawca zarobi 5 zł, a za każdy towar B 3 zł."
  Funkcja celu: $(max)\ z = 5 \cdot x_A + 3 \cdot x_B$

Funkcja celu

Uwagi:

- W funkcji celu *nie można* mnożyć między sobą zmiennych decyzyjnych, czyli np. zabronione jest $x_A \cdot x_B$. Taka operacja czyni problem nieliniowym (i trudniejszym do rozwiązania).

Ograniczenia

- Ograniczają dopuszczalne wartości zmiennych decyzyjnych.
- Mają ogólną postać:

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \; \{\leqslant, =, \geqslant\} \; b_1 \\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \; \{\leqslant, =, \geqslant\} \; b_2 \\
&\ldots \\
&a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \; \{\leqslant, =, \geqslant\} \; b_m
\end{aligned}
$$

- Zakładamy, że wszystkie x_i są nieujemne.

Przykład

- "Mamy 3 rodzaje części i można ich sprowadzić sumarycznie nie więcej niż 600."
  Ograniczenie: $x_A + x_B + x_C \leqslant 600$

Metody rozwiązania

- Metoda graficzna. (następne zajęcia?)
- Korzystając z solvera wbudowanego w MS Excela/Libre Office Calc. (te zajęcia?)
- Korzystając z zoptymalizowanego ze względu na wydajność zewnętrznego solvera, np. LPSolve, Gurobi, CPLEX.

Brak rozwiązań dopuszczalnych.

$$
\begin{aligned}
(max)\ z &= x_1 \\
x_1 &\geqslant 2 \\
x_1 &\leqslant 1
\end{aligned}
$$

Nie można przydzielić wartości do x_1 tak, by wszystkie ograniczenia były spełnione. Dla tej instancji problemu nie ma żadnego rozwiązania dopuszczalnego.

Brak rozwiązania optymalnego.

$$
\begin{aligned}
(max)\ z &= x_1 \\
x_1 &\geqslant 2
\end{aligned}
$$

Możemy podstawiać w nieskończoność coraz to większe wartości pod x_1, tak więc żadnego konkretnego rozwiązania nie możemy nazwać optymalnym.

Nieskończenie wiele rozwiązań optymalnych.

$$(max)\ z = x_1 + x_2$$
$$x_1 + x_2 \leqslant 10$$

W tym wypadku optymalne są rozwiązania $(x_1 = 1, x_2 = 9)$, $(x_1 = 1.5, x_2 = 8.5)$, itp. Takich par liczb rzeczywistych sumujących się do 10 jest nieskończenie wiele i wszystkie produkują tę samą wartość funkcji celu.

Programowanie liniowe całkowitoliczbowe

- Zakładamy w nim, że zmienne są **liczbami całkowitymi** a nie rzeczywistymi.
- Jest trudniejszy niż standardowe programowanie liniowe.
- Jego szczególnym wariantem jest **programowanie liniowe binarne**, w którym zmienne decyzyjne przyjmują wartości z dziedziny $\{0, 1\}$.