BADANIA OPERACYJNE - ĆWICZENIA

1 Wstęp 3

1.1 Kilka słów o przedmiocie 3

1.2 Problemy i modele decyzyjne (Siudak) 3

1.3 Zadanie programowania matematycznego 4

2 Zadanie programowania liniowego (Gass, Siudak, Błażewicz) 5

2.1 Wprowadzenie 5

2.2 Interpretacja geometryczna zadania programowania liniowego 6

2.3 Własności zadania programowania liniowego 6

2.4 Metoda Sympleks (MINIMALIZACJA) 8

2.5 Znajdowanie początkowego bazowego rozwiązania dopuszczalnego 12

2.6 Postępowanie w przypadku degeneracji 16

2.7 Dualizm w programowaniu liniowym 17

Rozwiązywanie zadań PL. za pomocą programu FRACTIONS 21

3 Optymalizacja dyskretna - całkowitoliczbowe programowanie liniowe 22

3.1 Wprowadzenie 22

3.2 Metoda płaszczyzn tnących - algorytm Gomory’ego 22

3.3 Algorytm 23

3.4 Przykład 23

4 Optymalizacja dyskretna – wstęp (under construction) 25

4.1 Wstęp 25

4.2 Złożoność obliczeniowa 25

4.3 Klasy złożonościowe problemów 25

5 Optymalizacja dyskretna - metoda podziału i ograniczeń 28

5.1 Wprowadzenie 28

5.2 Algorytm 28

5.3 Przykład 29

5.4 Problem komiwojażera - Algorytm Little’a 29

5.5 Problem przydziału – metoda węgierska 25

6 Optymalizacja dyskretna - metaheurystyki 33

6.1 Wprowadzenie 33

6.2 Metaheurystyki 33

6.3 Algorytm Tabu (ang. Tabu Search - TS) 33

6.4 Programy ewolucyjne 34

6.5 Algorytm symulowanego odprężania 35

7 Metoda programowania dynamicznego (Błażewicz) 37

7.1 Wprowadzenie 37

7.2 Wieloetapowe procesy decyzyjne 37

7.3 Metoda programowania dynamicznego 38

7.4 Jednowymiarowy proces alokacji 39

Zaliczenie zajęć na podstawie zdobytej ilości punktów:

w ciągu semestru odbędą się dwa kolokwia (zapowiedziane z conajmniej 2 tygodniowym wyprzedzeniem). Zaliczenie pozytywne - uzyskanie powyżej 50% punktów. W przypadku niezaliczenia kolokwium, w terminie do dwóch tygodni od ogłoszenia wyników odbędzie się kolokwium poprawkowe. By zaliczyć kolokwium poprawkowe trzeba zdobyć co najmniej 60% punktów (przy niezmienionym poziomie trudności). Otrzymuje się wtedy (potrzebne do końcowej oceny) dokładnie 50% punktów właściwego kolokwium.

dodatkowe punkty za aktywność na zajęciach

punkty karne za nieusprawiedliwiona nieobecność

na końcu semestru istnieje możliwość podwyższenia oceny (maksymalnie o 1 stopień w górę) po rozmowie ustnej (zaliczenie całego materiału); w wyniku rozmowy ocena może też zostać obniżona

Egzamin - wymagane wiadomości z ćwiczeń i wykładów

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Problem programowania liniowego** |  | **programowanie dynamiczne** | |  |
| Definicja problemu | TAK | | WPD z własnością markowa | TAK |
| Metoda sympleks + przykład | TAK | | zasada optymalności | TAK |
| Odmiany metody sympleks | NIE | | Algorytm | TAK |
| Unikanie degeneracji | NIE | | zastosowanie dla problemów ciągłych i dyskretnych |  |
| Oprogramowanie | TAK | |  |  |
| Dualność i interpretacja | TAK | |  |  |
| Niesimpleksowe podejścia | TAK | |  |  |
|  |  | |  |  |
| **Problem programowania dyskretnego** |  | | **problem programowania nieliniowego** |  |
| Metoda płaszczyzn odcinajacych | TAK | | TW. Kuhna-Tuckera | TAK |
| Metoda podziału i ograniczeń | TAK | | TW. Lagrange’a | TAK |
| Metaheurystyki | TAK | | przegląd i klasyfikacja metod | TAK |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |

Znalezione błędy (zaznaczono na czerwono)

# Wstęp

## Kilka słów o przedmiocie

Tematem zajęć będą Metody Optymalizacji a nie same Badania Operacyjne. Ponieważ większość z tych metod powstała na potrzeby Badań Operacyjnych, pozostaniemy przy tej drugiej nazwie.

Zajmować będziemy się **metodami optymalizacji**, służącymi do rozwiązywania **problemów decyzyjnych**. Program zajęć obejmuje zapoznanie się z metodami rozwiązywania następujących problemów

Problemy dynamiczne

Problemy statyczne

Problemy liniowe Problemy nieliniowe

Programowanie liniowe

(metoda sympleks, sztuczna baza,

dualność)

Programowanie nieliniowe

Problemy

ciągłe

Problemy

dyskretne

Programowanie liniowe

całkowitoliczbowe

(algorytm Gomory'ego)

Wieloetapowe procesy decyzyjne

Jednowymiarowy proces alokacji

(programowanie dynamiczne)

## Problemy i modele decyzyjne (Siudak)

Każdy system działania rodzi tzw. **sytuacje decyzyjne**. Sytuacją decyzyjną nazywać będziemy ogół czynników, które wyznaczają postępowanie decyzyjne podmiotu podejmującego decyzję, zwanego na ogół **decydentem**.

Następstwem prawie każdej sytuacji decyzyjnej jest pewien problem decyzyjny, którego rozwiązanie wyznacza racjonalną decyzję. Problem decyzyjny można zdefiniować następująco:

istnieje cel lub zbiór celi które należy osiągnąć

istnieją alternatywne sposoby dojścia do celu

optymalny sposób lub kombinacja sposobów nie jest oczywista.

**Badania operacyjne** (ang. operational research) należą do tych dziedzin wiedzy, które zajmują się metodami rozwiązywania problemów decyzyjnych, wynikających z potrzeb racjonalnej działalności człowieka.

**Rozwiązanie** problemu decyzyjnego za pomocą badań operacyjnych jest procedurą składającą się z następujących etapów:

rozpoznanie sytuacji decyzyjnej i wynikającego z niej problemu decyzyjnego

budowa modelu decyzyjnego

rozwiązanie problemu decyzyjnego (metodą badań operacyjnych lub programowania matematycznego)

ocena poprawności i realności uzyskanych rozwiązań oraz ewentualna weryfikacja modelu decyzyjnego

przedstawienie rozwiązań decydentowi i ostateczne przygotowanie decyzji.

Nie każdy problem decyzyjny może być rozwiązany za pomocą badań operacyjnych. Rozwiązać można jedynie problemy dobrze ustrukturalizowane, dające się przedstawić w postaci modelu matematycznego. Nie da się rozwiązać metodami badań operacyjnych problemów nieustrukturalizowanych i słabo ustrukturalizowanych.

Budowa modelu decyzyjnego jest jednym z trudniejszych etapów w procedurze podejmowania decyzji. W praktyce najwięcej problemów sprawia:

wyróżnienie istotnych cech sytuacji decyzyjnej i ujęcie ich w modelu

modelowanie - tę samą sytuację często można odwzorować za pomocą kilku modeli.

Od postaci sformułowanego modelu zależą szanse jego efektywnego rozwiązania - stąd przy formułowaniu modelu należy mieć rozeznanie co do metod, którymi można go rozwiązać.

**Model decyzyjny** jest konstrukcją formalną, odwzorowującą istotne cechy rzeczywistej sytuacji decyzyjnej. Model taki może być sformułowany w różnej postaci. **Matematyczna postać modelu decyzyjnego** jest następująca:

z=f(x1,x2,...,xn) (1.1)

gdzie:

zmienne x1,x2,...,xn (zapisywane też jako x) noszą nazwę zmiennych decyzyjnych i są przedmiotem sterowania przez podejmowanie decyzji; określają one alternatywne sposoby działania

z jest miarą oceny podjętej decyzji

f jest funkcją odwzorowującą zależność między zmiennymi decyzyjnymi a miarą oceny z; funkcja f nazywana jest **funkcją celu** lub **funkcją kryterium**.

Na ogół zbiór alternatywnych sposobów działania nie jest dowolny i wtedy podejmowanie decyzji przebiega w warunkach pewnych ograniczeń. Uwzględnienie tych warunków w modelu decyzyjnym polega na określeniu zbioru **dopuszczalnych decyzji (rozwiązań)** dla konkretnej sytuacji decyzyjnej. Ogólnie warunki ograniczające można przedstawić następująco:

gi(x)<= 0 (i=1,2,...,m) (1.2)

Zagadnienie wyboru decyzji za pomocą modelu decyzyjnego polega na określeniu wartości zmiennych decyzyjnych ze zbioru dopuszczalnych sposobów działania opisanych ograniczeniami tak, aby uzyskać najkorzystniejszą wartość miary podjętej decyzji - optimum funkcji celu.

Operację poszukiwania wartości zmiennych decyzyjnych x1,x2,...,xn które maksymalizują lub minimalizują funkcję celu będziemy zapisywać symbolicznie w sposób następujący:

(max) z=f(x1,x2,...,xn) lub

(min) z=f(x1,x2,...,xn) (1.3)

Tak więc ogólna, matematyczna postać modelu decyzyjnego jest następująca:

(max lub min) z=f(x1,x2,...,xn)

przy ograniczeniach: gi(x)<= 0 (i=1,2,...,m) (1.4)

**Przykład:** Zakład produkcyjny produkuje dwa typy wyrobów: krzesła i stoły. Każdy z tych produktów musi być złożony z części a następnie wykończony i zapakowany. Czas potrzebny na złożenie krzesła i stołu wynosi odpowiednio 3 i 4 jednostki czasu. Wykończenie i zapakowanie krzesła i stołu wynosi odpowiednio 6 i 2 jednostki czasu. Producent dysponuje 60 jednostkami czasu na składanie wyrobów i 32 jednostkami czasu na wykończenie i zapakowanie. Każde krzesło przynosi zysk wielkości 20 jednostek a stół - 24 jednostki. Ile krzeseł i ile stołów powinien zakład wyprodukować dla maksymalizacji zysku?

Matematyczna postać problemu decyzyjnego wygląda następująco:

(max) zysk = 20\*krzesła+24\*stoły

3\*krzesła+4\*stoły <= 60 - ograniczenie czasu składania wyrobów

6\*krzesła+2\*stoły <= 32 - ograniczenie czasu wykończenia i pakowania

krzesła >= 0 - ograniczenie wielkości produkcji krzeseł

stoły >= 0 - ograniczenie wielkości produkcji stołów

Modele decyzyjne a co za tym idzie typ problematyki, można podzielić według różnych kryteriów:

Postać funkcji celu

* programowanie liniowe - funkcja celu i ograniczenia są funkcjami liniowymi
* programowanie nieliniowe - w przeciwnym razie

Postać zmiennych decyzyjnych

* zmienne ciągłe
* zmienne dyskretne

Liczba etapów procesu decyzyjnego

* programowanie statyczne - jeden etap
* programowanie dynamiczne - wiele etapów

Liczba kryteriów (funkcji celu)

* model jednokryterialny
* model wielokryterialny

## Zadanie[[1]](#footnote-1) programowania matematycznego

**Ogólna postać zadania programowania matematycznego** jest następująca:

funkcja celu: (max lub min) z = f(x) (1.5)

ograniczenia: gi(x) <= 0 (i=1,2,...,m) (1.6)

gdzie zR, a xRn jest wektorem **zmiennych decyzyjnych**. Funkcja *f* nosi nazwę **funkcji celu** lub funkcji kryterium. Funkcje gi(x) nazywa się **funkcjami ograniczeń** (warunki ograniczające, ograniczenia).

Dowolne rozwiązanie xRn spełniające układ równań i nierówności (1.6) nosi nazwę **rozwiązania dopuszczalnego**. Zbiór wszystkich rozwiązań dopuszczalnych, będziemy oznaczać przez X. Jeżeli zbiór rozwiązań dopuszczalnych jest zbiorem pustym (X=), to zadanie (1.6) nazywa się **sprzecznym**.

**DEFINICJA:** Rozwiązanie xoX nazywamy **rozwiązaniem optymalnym** zadania programowania matematycznego (1.5-6), jeżeli dla dowolnego xX spełniony jest warunek:

(minimalizacja funkcji celu) f(xo)f(x) (1.7a)

(maksymalizacja funkcji celu) f(xo)≥f(x) (1.7b)

W przypadku gdy zadanie (1.5-6) nie jest sprzeczne i przy tym nie istnieje xo spełniające warunek (1.7a-b), to mówimy że zadanie nie ma **skończonego rozwiązania optymalnego**. Zadanie może mieć również więcej niż jedno rozwiązanie optymalne, jeżeli warunek (1.7a-b) spełniony jest dla więcej niż jednego xoX.

**TWIERDZENIE:** Dla dowolnej funkcji z = f(x) spełnione są równości

(min) f(x) = - (max) [-f(x)]

(max) f(x) = - (min) [-f(x)] (1.8)

# Zadanie programowania liniowego (Gass, Siudak, Błażewicz)

## Wprowadzenie

Jeżeli w zadaniu (1.5-6) funkcja celu oraz warunki ograniczające są liniowe, to zadanie takie nazywamy zadaniem programowania liniowego.  **Postać ogólna zadania programowania liniowego** jest zatem następująca:

funkcja celu: (min albo max) z=cTx

ograniczenia: Ax b

x ≥ 0

zR wartość funkcji celu

cRn współczynniki kosztu funkcji celu

xRn zmienne decyzyjne

bRm wektor wyrazów wolnych

A - macierz m x n współczynniki technologiczne

Każdy liniowy model decyzyjny odwzorowujący określoną sytuację decyzyjną zawiera się w ogólnej postaci zadania programowania liniowego. Postać ta nie jest wygodna do określenia różnych własności zadań. Z tego powodu sformułowana zostanie inna postać zadania programowania liniowego.

**Zadaniem programowania liniowego o postaci standardowej[[2]](#footnote-2)** nazywamy zadanie:

funkcja celu[[3]](#footnote-3): (min) z=cTx (F1)

ograniczenia: Ax = b (b 0 W

x 0 (W2)

☝ Istnieje jeszcze jedno ważne założenie, że m<n (Gass)[[4]](#footnote-4)

Zadanie o postaci ogólnej można przekształcić do zadania o postaci standardowej w następujący sposób:

zamiana rodzaju ekstremum - zadanie na minimum (maksimum) można przekształcić w zadanie na maksimum (minimum) zmieniając funkcję celu na przeciwną (mnożąc ją przez -1)

zamiana zmiennych decyzyjnych dowolnych co do znaku na zmienne nieujemne. Jeżeli x jest zmienną dowolną co do znaku, to podstawiając: x=x+-x-, gdzie x+=max{0,x}, x-=max{0,-x}, otrzymujemy przedstawienie tej zmiennej za pomocą nieujemnych zmiennych x+ i x-

zamiana nierówności na równość:

aTxb lub aTxb można zastąpić odpowiednio: aTx+xd=b lub aTx-xd=b.

Nowo wprowadzona zmienna xd nosi nazwę zmiennej dodatkowej (osłabiającej).☝Zmienne te nie występują w funkcji celu.

**PRZYKŁAD:**

Postać wyjściowa:

funkcja celu: (max) z=2x1-x2+5x3-2x4

ograniczenia: x1+x2+2x3+5x4<=-20

2x1-x2-x4>=15

3x2+x3-2x4=14

x1,x3,x4>=0, x2 dowolne

Postać standardowa:

funkcja celu: (min) z’=-2x1+(x2+-x2-)-5x3+2x4

ograniczenia: -x1-(x2+-x2-)-2x3-5x4-x5=20

2x1-(x2+-x2-)-x4-x6=15

3(x2+-x2-)+x3-2x4=14

x1,x2+,x2-,x3,x4,x5,x6>=0

**🗹 PYTANIA:**

Czym się różni zadanie programowania liniowego od zadania programowania matematycznego ?

Czym się różni postać standardowa zadania programowania liniowego od jego postaci ogólnej ?

Czy zmienne osłabiające występują w funkcji celu ?

Czy zmienne decyzyjne lub osłabiające mogą być ujemne ?

## Interpretacja geometryczna zadania programowania liniowego

Zadanie programowania liniowego ma prostą interpretację geometryczna. Z konieczności ograniczymy się do przypadku dwóch zmiennych decyzyjnych.

PRZYKŁAD :

funkcja celu: (max) z=x1+x2

ograniczenia: -x1+2x2<=6 (równanie prostej: 2x2=6+x1)

2x1-x2<=6 (równanie prostej: x2=2x1-6)

x1,x2>=0

Zadanie to możemy rozwiązać w sposób graficzny:

x2=2x1-6

1. Znajdujemy obszar rozwiązań dopuszczalnych - X

a. rysujemy proste w oparciu o ograniczenia

x2

b. zaznaczamy półpłaszczyzny spełniające nierówności

2x2=6+x1

c. część wspólna jest obszarem rozwiązań dopuszczalnych

6

2. Znajdujemy rozwiązanie optymalne – x\*

a. rysujemy prostą przechodząca przez rozwiązania

z=4

o tej samej wartości funkcji celu

b. przesuwamy równolegle tę prostą w kierunku

z=12

polepszania wartości funkcji celu

z=2

c. skrajne rozwiązanie jest rozwiązaniem optymalnym

Dla naszego zadania:

x 1

x1\*=6, x2\*=6 z\*=x1\*+x2\*=12

6

☝ Graficzna prezentacja następujących sytuacji:

nie ma skończonego rozwiązania - rozwiązanie jest nieograniczone gdy X nie jest zwarte

jest jedno skończone rozwiązanie

jest wiele skończonych rozwiązań

nie ma żadnego rozwiązania - X jest puste.

## Własności zadania programowania liniowego

### Kombinacja liniowa wektorów, wektory liniowo niezależne (cel: baza)

Wektor x∈Rn nazywamy ***kombinacją liniową wektorów*** x1,x2,...,xk∈Rn jeżeli:

 (2.1)

Przykład: wektor  jest kombinacją liniową wektorów  ponieważ: 

Wektor x∈Rn nazywamy ***wypukłą kombinacją liniową wektorów*** x1,x2,...,xk∈Rn jeżeli:

 przy czym  (2.2)

Zbiór wektorów x1,x2,...,xn nazywamy ***liniowo niezależnym***, jeżeli równość:

 (2.3)

zachodzi tylko wtedy, gdy wszystkie λi=0 (i=1,...,k). W przeciwnym razie zbiór jest ***liniowo zależny***.

W n-wymiarowej przestrzeni euklidesowej Rn istnieje układ n liniowo niezależnych wektorów, ale każdy układ n+1 wektorów w tej przestrzeni jest układem liniowo zależnym.

### Baza, rozwiązanie bazowe (cel: zadanie PL a baza)

***Bazą*** zbioru S ⊂ Rn jest zbiór liniowo niezależnych wektorów, takich że wszystkie inne wektory należące do S da się przedstawić jako kombinację liniową wektorów bazy.

Dowolny zbiór n liniowo niezależnych wektorów należących do Rn jest bazą w przestrzeni Rn .

Zbiór n wektorów jednostkowych [1,0,...,0],[0,1,...,0],...,[0,0,...,1] jest bazą w Rn. !!!

Dany jest układ równań liniowych Ax=b. Macierz A jest typu m x n i **zakładamy że m<=n**. Niech AB jest bazą zbioru kolumn macierzy A, xB niech będzie wektorem zmiennych stojących przy kolumnach bazowych (zmienne bazowe), xR wektorem zmiennych stojących przy kolumnach niebazowych (zmienne niebazowe lub wtórne). Układ równań liniowych możemy przedstawić teraz w postaci ABxB+ARxR=b.

### Zbiory wypukłe, sympleks

***Wypukłą kombinacją punktów*** U1,U2,...,Uk (Ui∈ Rn i=1..k) nazywamy punkt:

 (2.4)

**TWIERDZENIE**: Dowolny punkt leżący na odcinku łączącym dwa punkty w Rn może być wyrażony jako kombinacja wypukła tych dwóch punktów.

W interpretacji geometrycznej, zbiór wypukły jest to taki zbiór, który składa się z odcinków łączących każde dwa punkty zbioru.

Punkt U zbioru wypukłego nazywamy ***wierzchołkiem***, jeśli nie może być on wyrażony jako kombinacja wypukła dwóch różnych punktów należących do tego zbioru.

***Sympleks*** jest n-wymiarowym wielościanem wypukłym mającym dokładnie n+1 wierzchołków. Brzeg sympleksu składa się z sympleksów niższych wymiarów zwanych powierzchniami sympleksu. Ilość takich płaszczyzn o wymiarze „i” wynosi . Sympleksem w przestrzeni zerowymiarowej jest punkt, jednowymiarowej - prosta, dwuwymiarowej -trójkąt, trójwymiarowej - czworościan.

### Związek między rozwiązaniami optymalnymi a punktami wierzchołkowymi (cel: wierzchołek a baza)

**Rozwiązaniem dopuszczalnym** zadania programowania liniowego jest wektor x spełniający warunki (W1) i (W2)

**Rozwiązaniem bazowym** układu równań (W1) nazywamy rozwiązanie układu powstałe z (W1) przez przyrównanie do zera *n-m* zmiennych przy założeniu, że wyznacznik współczynników tych *m* zmiennych jest niezerowy. Te *m* zmiennych nazywamy zmiennymi bazowymi.

**Rozwiązaniem bazowym dopuszczalnym** nazywamy rozwiązanie bazowe, które spełnia warunek (W2), czyli wszystkie zmienne bazowe są nieujemne.

**Niezdegenerowanym rozwiązaniem bazowym dopuszczalnym** nazywamy bazowe rozwiązanie dopuszczalne, w którym wszystkie zmienne bazowe są dodatnie (>0).

**WŁASNOŚĆ**: Zbiór wszystkich rozwiązań dopuszczalnych zadania programowania liniowego jest zbiorem wypukłym (sympleksem).

**WŁASNOŚĆ**: Funkcja celu zadania programowania liniowego przyjmuje wartość optymalną w punkcie wierzchołkowym zbioru wypukłego *X*. Jeśli przyjmuje wartość optymalna w więcej niż jednym punkcie wierzchołkowym, to tę samą wartość przyjmuje dla każdej kombinacji wypukłej tych punktów.

### Związek między punktami wierzchołkowymi a wektorami liniowo niezależnymi.

Warunek ograniczający (W1) można przedstawić także w następującej postaci:

P1\*x1+P2\*x2+ ...+Pn\*xn=Po (W1’)

gdzie PiRm i=0..n

**WŁASNOŚĆ:** Jeżeli można znaleźć k<=m wektorów P1,...,Pk liniowo niezależnych takich że:

P1\*x1+P2\*x2+ ...+Pk\*xk=Po oraz wszystkie xj>=0

to punkt o współrzędnych x=[x1,...,xk,0,...,0]T jest punktem wierzchołkowym zbioru *X*.

**WŁASNOŚĆ:** Jeśli x=[x1,...,xn]T jest punktem wierzchołkowym zbioru X, to wektory Pj odpowiadające dodatnim xj są liniowo niezależne.

Z obu własności wynika, że dodatnich xj jest co najwyżej *m*.

**WNIOSEK**: Każdemu punktowi wierzchołkowemu zbioru *X* odpowiada zbiór *m* wektorów liniowo niezależnych z danego zbioru P1,...,Pn. Tych *m* wektorów tworzy bazę m-wymiarowej przestrzeni wektorowej, a odpowiadający im punkt wierzchołkowy reprezentuje bazowe rozwiązanie dopuszczalne zadania programowania liniowego.

Uogólnienie dotychczasowych 4 własności:

**TWIERDZENIE:**  Punkt x jest punktem wierzchołkowym zbioru *X* wtedy i tylko wtedy, gdy w kombinacji liniowej wektorów niezależnych Pj

 (2.5)

współczynniki xj są dodatnie.

WNIOSKI:

Każde bazowe rozwiązanie dopuszczalne jest punktem wierzchołkowym zbioru *X*

Istnieje punkt wierzchołkowy zbioru *X*, w którym funkcja celu przyjmuje optimum

Trzeba więc badać tylko rozwiązania w punktach wierzchołkowych. Takich punktów wierzchołkowych jest  - tyle ile wektorów liniowo niezależnych z danego zbioru *n* wektorów. Idea rozwiązania może polegać więc na wygenerowaniu wszystkich kombinacji i porównaniu funkcji celu. Dla dużych *m* i *n* byłoby niemożliwe obliczenie wszystkich rozwiązań. Potrzebna jest metoda, pozwalająca utworzyć ciąg rozwiązań dopuszczalnych, zbieżny do rozwiązania optymalnego. Taką metodą jest metoda sympleks (sympleksów) podana przez G.B.Dantziga. Dla znalezienie minimalnego rozwiązania wystarczy zwykle *m* do *2m* kroków (skończona i ograniczona liczba kroków).

## Metoda Sympleks (MINIMALIZACJA)

### Idea

Z wcześniejszych rozważań[[5]](#footnote-5) wiemy, że jeżeli zadanie programowania liniowego ma skończone rozwiązanie optymalne to jest nim punkt wierzchołkowy tego obszaru. Ogólna idea metody sympleks polega na przechodzeniu pomiędzy sąsiednimi punktami wierzchołkowymi obszaru rozwiązań dopuszczalnych w celu polepszenia funkcji celu. W przypadku niemożności polepszenia wartości funkcji celu, dane rozwiązanie jest rozwiązaniem optymalnym.

Z matematycznego punktu widzenia każdemu punktowi wierzchołkowemu odpowiada bazowe rozwiązanie dopuszczalne. Przejściu pomiędzy sąsiednimi wierzchołkami odpowiada zaś zamiana danej bazy (danego dopuszczalnego rozwiązania bazowego) na sąsiednią (sąsiednie dopuszczalne rozwiązanie bazowe). Sąsiednia baza jest konstruowana przez zamianę jednego wektora bazowego innym - niebazowym. Wektory te są tak dobierane, by polepszyć wartość funkcji bazowego rozwiązania dopuszczalnego. Problemem pozostaje jedynie dobór początkowego (startowego) bazowego rozwiązania dopuszczalnego.

**☝Zakładać będziemy, że każde dopuszczalne rozwiązanie bazowe jest niezdegenerowane.**

### Początkowe bazowe rozwiązanie dopuszczalne

Metoda sympleks startuje od początkowego bazowego rozwiązania dopuszczalnego. Dla pewnej klasy problemów, rozwiązanie to można znaleźć natychmiast. Spójrzmy na zadanie programowania liniowego:

funkcja celu: (min) z=cTx

ograniczenia: Ax≤b (b0

x0

Należy sprowadzić to zadanie do postaci standardowej - dodanie **zmiennych osłabiających** utworzy początkową bazę.

Przykład:

funkcja celu: (min) z=-x1-4x2

ograniczenia: x1+x2≤1

4x1+2x2≤3

x1,x2≥0

Po dodaniu zmiennych osłabiających otrzymamy następujące zadanie (zadanie PL. o postaci standardowej):

funkcja celu: (min) z=-x1-4x2

ograniczenia: x1+x2+x3=1

4x1+2x2+x4=3

x1,x2,x3,x4≥0

Współczynniki przy zmiennych osłabiających (wektory jednostkowe) tworzą bazę. Zmienne te są więc zmiennymi bazowymi. Wartości tych zmiennych są równe wartościom elementów wektora b. Pozostałe zmienne (zmienne decyzyjne) są zmiennymi niebazowymi i ich wartości są równe 0. Funkcja celu ma zatem wartość równą 0.

### Tablica Sympleks

Dla ułatwienia obliczeń, metoda Sympleks bazuje na tablicy sympleks. Oto jej wygląd:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i\j |  |  | 0 | 1 | 2 | ... | n |
|  | Baza |  |  | <c1> | <c2> | ... | <cn> |
|  | **(MIN)** | cB | P0 | P1 | P2 | ... | Pn |
| 1 | <P1B> | <c1B> | **<x1B>** | **<x1,1>** |  |  | **<x1,n>** |
| ... | ... | ... | **...** | **...** |  | **...** | **...** |
| m | <PmB> | <cmB> | **<xmB>** | **<xm,1>** |  |  | **<xm,n>** |
|  | Wskaźniki | optymalności | **z0** | **z1-c1** | **z2-c2** | **...** | **zn-cn** |

1) numery wektorów bazowych

5) wskaźniki optymalności

3) wartość funkcji celu

4) wartości zmiennych bazowych

2) współczynniki

przy zmiennych bazowych w funkcji celu

Właściwości tablicy sympleks:

w komórce **z0** znajduje się aktualna wartość funkcji celu. Można ją obliczyć ze wzoru z0= cTx=. Ponieważ zmienne niebazowe mają wartość równą 0, wartość **z** można obliczyć z prostszego wzoru: z=, czyli mnożąc dwie kolumny w tablicy sympleks - cB i P0

w kolumnie P0 znajdują się wartości zmiennych bazowych. Pozostałe zmienne mają wartość 0.

w tablicy sympleks wskaźniki optymalności (zj-cj) (gdzie) odpowiadają zmniejszeniu wartości funkcji celu przez zwiększenie wartości zmiennej xj o jednostkę. Zmienne bazowe mają zawsze wskaźnik optymalności równy 0.

**Przykład cd. :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -1 | -4 | 0 | 0 |
| **(MIN)** |  |  | P1 | P2 | P3 | P4 |
| P3 | 0 | **x1B=b1=1** | **1** | **1** | **1** | **0** |
| P4 | 0 | **x2B=b2=3** | **4** | **2** | **0** | **1** |
| Wskaźniki | Optymalności | **z0=0** | **1** | **4** | **0** | **0** |

### Kryterium wejścia i optymalności

**TWIERDZENIE**: Jeżeli dla ustalonego „i” spełniony jest warunek zi-ci>0, to można skonstruować zbiór rozwiązań dopuszczalnych taki, że dla każdego elementu tego zbioru spełniona jest nierówność z<z0, gdzie dolna granica z jest albo skończona albo nieskończona.

Jeśli dolna granica jest skończona, nowe rozwiązanie ma dokładnie m dodatnich zmiennych. Jeśli dolna granica jest nieskończona - nowe rozwiązanie dopuszczalne ma m+1 dodatnich zmiennych a wartość funkcji celu może być dowolnie mała.

Komentarz

Jeśli wskaźnik optymalności jest dodatni dla pewnej zmiennej niebazowej, wtedy funkcja celu może być zmniejszona przez zwiększenie wartości tej zmiennej niebazowej. Ponieważ zmienna ta jest niebazowa, więc jej wartość jest równa zeru i zwiększenie jej wartości jest możliwe jedynie przez dodanie do bazy odpowiadającego jej wektora. W naszym przykładzie wektory P1 i P2 mogą być wprowadzone do bazy i polepszyć wartość funkcji celu. Ogólnie wybiera się ten wektor, którego wskaźnik optymalności jest największy. Dodanie do bazy innego wektora niebazowego o dodatnim wskaźniku optymalności także doprowadzi do rozwiązania optymalnego, ale **może** to zająć więcej czasu. Dodanie do bazy wektora niebazowego o ujemnym wskaźniku optymalności pogorszy wartość funkcji celu. Dodanie do bazy wektora niebazowego o zerowym wskaźniku optymalności nie zmieni wartości funkcji celu.

**Kryterium wejścia**

Do bazy wchodzi wektor Pwe (zmienna niebazowa xwe) dla którego (dla której):

 (2.6)

**Kryterium optymalności**

Jeśli wszystkie wskaźniki optymalności są mniejsze lub równe zeru dane rozwiązanie jest optymalne.

**Przykład cd. :**

Wybieramy wektor niebazowy P2 jako wchodzący do bazy.

### Kryterium wyjścia

Do bazy wchodzi zmienna xwe. Będzie ona zwiększana tak bardzo, jak jest to możliwe, by zmniejszyć wartość funkcji celu. Nowe rozwiązanie bazowe musi być jednak rozwiązaniem dopuszczalnym - musi być punktem wierzchołkowym obszaru rozwiązań dopuszczalnych. Należy więc znaleźć punkt przecięcia prostej (płaszczyzny) ograniczającej obszar rozwiązań z osią 0Xwe, najbliższy punktowi 0 na tej osi.

rozw. niedopuszczalne

**Przykład cd. :**

Interpretacja graficzna zadania

rozw. dopuszczalne

a) aktualne rozwiązanie bazowe: x1=0, x2=0

wzrost x2

1

b) do bazy wchodzi P2 (zwiększamy x2)

c) możliwe dwa nowe rozwiązania bazowe (tylko jedno dopuszczalne):

1

dopuszczalne x1=0, x2=1 niedopuszczalne x1=0, x2=3/2

Współrzędną punktu przecięcia osi 0Xwe przez prostą (płaszczyznę) ograniczającą można obliczyć z tablicy sympleks w sposób następujący:

 (2.7)

**Kryterium wyjścia**

Z bazy jest usuwany wektor Pwy dla którego:

 (2.8)

Nowa wartość funkcji celu będzie równa:

znew=z-Θ(zj-cj) (2.9)

Komentarz

Θ oznacza nową wartość zmiennej bazowej xwe i musi być ona dodatnia. Licznik ułamka jest zawsze dodatni (wartość dotychczasowej zmiennej bazowej). Mianownik -współczynnik xi,we musi być też dodatni**. Jeżeli wszystkie xi,we będą mniejsze lub równe zeru, to nie ma skończonego rozwiązania minimalnego**. Θ nie jest ograniczone z góry i funkcja celu ma dolną granicę równą -∞. Jest to drugi przypadek twierdzenia Gass’a. Rozwiązanie dopuszczalne zawiera dokładnie m+1 zmiennych.

Minimalne Θ = 0 oznacza, że wartość funkcji celu pozostaje bez zmian.

**Przykład cd. :**

Wartość Θ dla P3 = 1 Wartość Θ dla P4 = 3/2 Z bazy usuwany jest wektor P3.

### Obliczanie nowego bazowego rozwiązania dopuszczalnego.

Tablicę sympleks należy przekształcić w sposób następujący:

element xwy, we zostaje **elementem centralnym.**

kolumna Pwe staje się wektorem bazowym (jeden element równy 1 w wierszu k=we, a reszta elementów = 0)

wartości wiersza zawierającego element centralny należy podzielić przez tę wartość (element centralny)

wartości w pozostałych wierszach należy przekształcić w sposób następujący:

 (2.10)

ręcznie uaktualnić kolumnę Baza i cB

**Przykład cd. :**

Tablica sympleks:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -1 | -4 | 0 | 0 |
| **(MIN)** |  |  | P1 | P2 | P3 | P4 |
| P3 | **0** | **1** | **1** | **1** | **1** | **0** |
| P4 | **0** | **3** | **4** | **2** | **0** | **1** |
| Wskaźniki | optymalności | **0** | **1** | **4** | **0** | **0** |

Zostaje przekształcona w nową tablicę sympleks:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -1 | -4 | 0 | 0 |
| **(MIN)** |  |  | P1 | P2 | P3 | P4 |
| P2 | **-4** | **1** | **1** | **1** | **1** | **0** |
| P4 | **0** | **1** | **2** | **0** | **-2** | **1** |
| Wskaźniki | optymalności | **-4** | **-3** | **0** | **-4** | **0** |

**W wyniku stosowania algorytmu sympleks uzyskamy zbiór bazowych rozwiązań dopuszczalnych zbieżny do rozwiązania minimalnego lub stwierdzimy że nie istnieje rozwiązanie skończone.**

### Przykład

funkcja celu: max z=2x1+x2

ograniczenia 3x1-4x2>=-12

3x1-2x2<=12

1/2x1+8/10x2<=4

x1,x2>=0

Postać standardowa:

funkcja celu: min z’=-2x1-x2

ograniczenia: -3x1+4x2+x3=12

3x1-2x2+x4=12

1/2x1+8/10x2+x5=4

x1,x2,x3,x4,x5>=0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -2 | -1 | 0 | 0 | 0 |
|  |  |  | **we=P1** | P2 | P3 | P4 | P5 |
| P3 | 0 | 12 | -3 | 4 | 1 | 0 | 0 |
| **wy=P4** | 0 | 12 | 3 | -2 | 0 | 1 | 0 |
| P5 | 0 | 4 | 1/2 | 8/10 | 0 | 0 | 1 |
| Wskaźniki | optymalności | 0 | 2 | 1 | 0 | 0 | 0 |

1. we=1 (do bazy wchodzi wektor P1)
2. =min {12/3, 4/1/2} = 4 - wy=4 (z bazy usuwamy wektor P4)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -2 | -1 | 0 | 0 | 0 |
|  |  |  | P1 | **we=P2** | P3 | P4 | P5 |
| P3 | 0 | 12-(-3\*12)/3=24 | 0 | 4-(-2\*-3)/3=2 | 1-(0\*-3)/3=1 | 0-(1\*-3)/3=1 | 0-(0\*-3)/3=0 |
| P1 | -2 | 12/3=4 | 1 | -2/3 | 0/3=0 | 1/3=1/3 | 0/3=0 |
| **wy=P5** | 0 | 4-(12\*1/2)/3=2 | 0 | 8/10-(-2\*1/2)/3=  17/15 | 0-(0\*1/2)/3=0 | 0-(1\*1/2)/3=-1/6 | 1-(0\*1/2)/3=1 |
| Wskaźniki | optymalności | 0-12\*2/3=-8 | 0 | 1-(2\*-2)/3=7/3 | 0-(0\*2)/3=0 | 0-(1\*2)/3=-2/3 | 0-(0\*2)/3=0 |

1. we=2 (do bazy wchodzi wektor P2)
2. =min {24/2, 2/(17/15)} = 2/(17/15) wy=5 (z bazy usuwamy wektor P5)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -2 | -1 | 0 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 | P5 |
| P3 | 0 | 348/17 | 0 | 0 | 1 | 22/17 | -30/17 |
| P1 | -2 | 88/17 | 1 | 0 | 0 | 4/17 | 10/17 |
| P2 | -1 | 30/17 | 0 | 1 | 0 | -5/34 | 15/17 |
| Wskaźniki | optymalności | -206/17 | 0 | 0 | 0 | -11/34 | -35/17 |

Koniec: z’\*=-206/17 => z\*=206/17; x1\*=88/17; x2\*=30/17; x3\*=348/17; x4\*=x5\*=0

### Rozwiązywanie problemu maksymalizacji

Algorytm metody sympleks dla problemu maksymalizacji różni się od wersji dla problemu minimalizacji wyłącznie kryterium wejścia i optymalności:

do bazy wchodzi wektor o minimalnej (i ujemnej) wartości wskaźnika optymalności

jeżeli wszystke wskażniki optymalności są wieksze lub równe zeru, rozwiazanie jest optymalne

### Inne metody rozwiązywania problemu PL

Dotychczas omawiana metoda sympleks (nazywana też prostą metodą sympleks) posiada liczne odmiany (ulepszenia). Można tu wspomnieć o zrewidowanej metodzie sympleks, czy też o całkowicie odmiennym podejściu - metodzie Chaczijana. W ogólności metoda sympleks jest wykładnicza. Chaczijan podał wielomianowy algorytm rozwiązywania problemu programowania liniowego wykorzystujący ideę malejących elipsoid.

### Zadania

funkcja celu: (max) z=6x1+4x2+8x3

ograniczenia: x1+x2+5x3<=1

3x1+x2+x3<=1

x1,x2,x3>=0

Rozwiązanie: x1\*=0, x2\*=1, x3\*=x4\*=x5\*=0 z\*=4

## Znajdowanie początkowego bazowego rozwiązania dopuszczalnego

### Wprowadzenie

Do tej pory zakładaliśmy, że zagadnienie programowania liniowego ma **początkowe bazowe rozwiązanie dopuszczalne** (jest macierz jednostkową, która może być wykorzystana jako baza początkowa). Istnieje jednak wiele zagadnień, które nie mają początkowego bazowego rozwiązania dopuszczalnego. By je uzyskać należy dodać do wyjściowego problemu **zmienne sztuczne** (ang. artificial variables), które pozwolą utworzyć **początkowe rozwiązanie bazowe dopuszczalne**, ale dla innego niż oryginalne zadania. Dla takiego zadania trzeba znaleźć rozwiązanie optymalne. Ponieważ zmienne sztuczne nie należą do problemu wyjściowego, należy wyeliminować je z tego rozwiązania, by otrzymać **rozwiązanie bazowe dopuszczalne** zadania oryginalnego. Można to uzyskać sprowadzając ich wartość do zera stosując metodę **dwu-fazową** (ang. Two-Phase method ) lub metodę **dużego współczynnika M** (ang. Big-M Method).

### Metoda dwu-fazowa

Tworzymy nową funkcję celu z' równą sumie zmiennych sztucznych, która będzie MINIMALIZOWANA

1. sprowadzamy do zera (metodą sympleks) sztuczną funkcję celu (zmienne sztuczne będą równe 0)
2. wyznaczamy rozwiązanie optymalne zadania wyjściowego rozpoczynając od dopuszczalnego rozwiązania bazowego znalezionego w kroku a) dla zerowej wartości funkcji celu z’.

**Przykład:**

funkcja celu: min z=2x1+x2

ograniczenia: x1<=1

x2<=2

x1+x2>=2

x1,x2>=0

Postać standardowa:

funkcja celu: min z=2x1+x2

ograniczenia: x1+x3=1

x2+x4=2

x1+x2-x5=2

x1,x2,x3,x4,x5>=0

Do trzeciego równania trzeba wprowadzić zmienną sztuczną x6, która w końcowym rozwiązaniu zadania wyjściowego musi być równa zeru, by otrzymać rozwiązanie dopuszczalne. Nowe zadanie:

funkcja celu: min z’=x6

ograniczenia: x1+x3=1

x2+x4=2

x1+x2-x5+x6=2

x1,x2,x3,x4,x5,x6>=0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 |
| P6 | 1 | 2 | 1 | 1 | 0 | 0 | -1 | 1 |
| Wsk. | Opt. | 2 | 1 | 1 | 0 | 0 | -1 | 0 |

* we=1 (do bazy wchodzi wektor P1, choć możliwe jest wybranie wektora P2)
* =min {1/1, 2/1} = 1 - z bazy usuwamy wektor P3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 |
| P6 | 1 | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opt. | 1 | 0 | 1 | -1 | 0 | -1 | 0 |

* we=2 (do bazy wchodzi wektor P2)
* =min {2/1, 1/1} = 1 - z bazy usuwamy wektor P6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | -1 |
| P2 | 0 | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opt. | 0 | 0 | 0 | 0 | 0 | 0 | -1 |

Rozwiązaliśmy zadanie rozszerzone, zmienne sztuczne mają wartość równą zeru, sztuczna funkcja celu też jest równa zeru. Mamy więc dopuszczalne rozwiązanie bazowe zadania wyjściowego i minimalizujemy je.

z poprzedniej tablicy sympleks wykreślamy kolumny związane ze zmiennymi sztucznymi

zmieniamy wartości wektora „cB” i uaktualniamy wskaźniki optymalności (powrót do oryginalnej funkcji celu)

postępujemy zgodnie z metodą sympleks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 | P5 |
| P1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 |
| P4 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| P2 | 1 | 1 | 0 | 1 | -1 | 0 | -1 |
| Wsk. | Opt. | 3 | 0 | 0 | 1 | 0 | -1 |

* we=3 (do bazy wchodzi wektor P3)
* =min {1/1, 1/1} = 1 - z bazy usuwamy wektor P1 (można też usunąć P4)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 | P5 |
| P3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| P4 | 0 | 0 | -1 | 0 | 0 | 1 | 1 |
| P2 | 1 | 2 | 1 | 1 | 0 | 0 | -1 |
| Wsk. | Opt. | 2 | -1 | 0 | 0 | 0 | -1 |

Koniec: z\*=2; x1\*=0; x2\*=2; x3\*=1; x4\*=0; x5\*=0. Rozwiązanie zdegenerowane – zmienna bazowa = 0.

### Metoda dużego współczynnika M

Rozszerzenie zadania polega na wprowadzeniu zmiennych sztucznych do ograniczeń, co pozwoli utworzyć bazę jednostkową. Wprowadza się je ponadto do funkcji celu z dużym dodatnim współczynnikiem M (współczynnik kary). Ponieważ minimalizujemy funkcję celu, algorytm metody sympleks usunie zmienne sztuczne z bazy, przez co otrzymamy **rozwiązanie bazowe dopuszczalne** dla problemu wyjściowego. Dla problemu maksymalizacji, współczynnik kary powinien mieć dużą wartość ujemną.

Przykład (po wprowadzeniu zmiennej sztucznej do funkcji celu):

funkcja celu: min z’=2x1+x2+Mx6

ograniczenia: x1+x3=1

x2+x4=2

x1+x2-x5+x6=2

x1,x2,x3,x4,x5,x6>=0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 |
| P6 | M | 2 | 1 | 1 | 0 | 0 | -1 | 1 |
| Wsk. | Opt. | 0  +2M | -2  +M | -1  +M | 0 | 0 | 0  -M | 0 |

* we=1 (do bazy wchodzi wektor P1, można też wybrać P2)
* =min {1/1, 2/1} = 1 - z bazy usuwamy wektor P3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 |
| P6 | M | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opty. | 2  +M | 0 | -1  +M | 2  -M | 0 | 0  -M | 0 |

* we=2 (do bazy wchodzi wektor P2)
* =min {2/1, 1/1} = 1 - z bazy usuwamy wektor P6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | -1 |
| P2 | 1 | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opt. | 3 | 0 | 0 | 1 | 0 | -1 | 1  -M |

* we=3 (do bazy wchodzi wektor P3)
* =min {1/1, 1/1} = 1 - z bazy usuwamy wektor P1 (można też usunąć P4)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M. |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 0 | -1 | 0 | 0 | 1 | 1 | -1 |
| P2 | 1 | 2 | 1 | 1 | 0 | 0 | -1 | 1 |
| Wsk. | Opt. | 2 | -1 | 0 | 0 | 0 | -1 | 1  -M |

Koniec: z\*=2; x1\*=0; x2\*=2; x3\*=1; x4\*=0; x5\*=0; Rozwiązanie zdegenerowane – zmienna bazowa = 0.

**UWAGA**: Zobaczmy jak wyglądałoby zadanie, gdybyśmy w ostatniej iteracji usunęli wektor P4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | -1 |
| P2 | 1 | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opt. | 3 | 0 | 0 | 1 | 0 | -1 | 1  -M |

* we=3 (do bazy wchodzi wektor P3)
* =min {1/1, 1/1} = 1 - z bazy usuwamy wektor P4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 0 | 1 | 0 | 0 | -1 | -1 | 1 |
| P3 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | -1 |
| P2 | 1 | 2 | 0 | 1 | 0 | 1 | 0 | 1 |
| Wsk. | Opt. | 2 | 0 | 0 | 0 | -1 | -2 | 2  -M |

Koniec: z\*=2; x1\*=0; x2\*=2; x3\*=1; x4\*=0; x5\*=0; Rozwiązanie zdegenerowane – zmienna bazowa = 0.

**Przykład:**

funkcja celu: min z=2x1+x2

ograniczenia: x1<=1

**2**x2<=2

x1+x2>=2

x1,x2>=0

Postać standardowa:

funkcja celu: min z=2x1+x2

ograniczenia: x1+x3=1

**2**x2+x4=2

x1+x2-x5=2

x1,x2,x3,x4,x5>=0

Po dokonaniu zmian w funkcji celu:

funkcja celu: min z’=2x1+x2+Mx6

ograniczenia: x1+x3=1

**2**x2+x4=2

x1+x2-x5+x6=2

x1,x2,x3,x4,x5,x6>=0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 2 | 0 | 2 | 0 | 1 | 0 | 0 |
| P6 | M | 2 | 1 | 1 | 0 | 0 | -1 | 1 |
| Wsk. | Opt. | 0  +2M | -2  +M | -1  +M | 0 | 0 | 0  -M | 0 |

* we=1 (do bazy wchodzi wektor P1, można też wybrać P2)
* =min {1/1, 2/1} = 1 - z bazy usuwamy wektor P3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 2 | 0 | 2 | 0 | 1 | 0 | 0 |
| P6 | M | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opt. | 2  +M | 0 | -1  +M | 2  -M | 0 | 0  -M | 0 |

* we=2 (do bazy wchodzi wektor P2)
* =min {2/2, 1/1} = 1 - z bazy usuwamy wektor P6, choć można też usunąć P4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| P4 | 0 | 0 | 0 | 0 | 2 | 1 | 2 | -2 |
| P2 | 1 | 1 | 0 | 1 | -1 | 0 | -1 | 1 |
| Wsk. | Opt. | 3 | 0 | 0 | 1 | 0 | -1 | 1  -M |

Otrzymaliśmy rozwiązanie **zdegenerowane** – jedna ze zmiennych bazowych jest równa zeru.

* we=3 (do bazy wchodzi wektor P3)
* =min {1/1, 0/2} = 0 - z bazy usuwamy wektor P4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 2 | 1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | 2 | 1 | 1 | 0 | 0 | -½ | -1 | 1 |
| P3 | 0 | 0 | 0 | 0 | 1 | ½ | 1 | 0 |
| P2 | 1 | 1 | 0 | 1 | 0 | ½ | 0 | 1 |
| Wsk. | Opty. | 3 | 0 | 0 | 0 | -½ | -2 | 3 |

Koniec: z\*=2; x1\*=1; x2\*=1; x3\*=0; x4\*=0; x5\*=0; Rozwiązanie zdegenerowane!

## Postępowanie w przypadku degeneracji

Zdegenerowane rozwiązanie dopuszczalne posiada pewne rozwiązania bazowe równe zeru. Dla rozwiązania dopuszczalnego zdegenerowanego istnieje możliwość obliczenia „tety” równej zeru:

 (2.11)

Wprowadzenie nowego wektora do bazy nie zmieni więc wartości funkcji celu. Teoretycznie można wejść w cykl rozwiązań dopuszczalnych. Jest to jednak możliwe gdy dopuszczalne rozwiązanie bazowe posiada co najmniej dwie zmienne bazowe o wartości równej zeru. Ponieważ w praktyce nie spotkano żadnych cyklicznych rozwiązań, pominiemy ten problem, wspominając jedynie że idea rozwiązania (metoda peturbacji) jest opisana w książce [Gass80].

## Dualizm w programowaniu liniowym

### Wprowadzenie

Dla każdego zadania programowania liniowego (dalej nazywanym zadaniem pierwotnym) można zbudować tzw. zadanie dualne, które też jest zadaniem programowania liniowego. Pomiędzy tymi zadaniami występują interesujące zależności.

Dane jest zadanie programowania liniowego w następującej postaci:

Zadaniem dualnym sprzężonym z zadaniem pierwotnym nazywamy następujące zadanie (D):

(min) zd=bTy (D)

ogr.: ATy≥c

y0

y∈Rm

Zadanie pierwotne (P):

(max) zp=cTx (P)

ogr,: Ax≤b

x0

x∈Rn

**TWIERDZENIE:**

Zadanie pierwotne jest zadaniem dualnym do swojego zadania dualnego.

**WNIOSEK:**

Zadania (P) i (D) są dualne względem siebie.

Powyższą pierwotną-dualną parę zadań można przedstawić w zapisie macierzowym:

y

bT

(min) zd  =

(max) zp =

x

cT

c

AT

y

A

x

b

≥

≤

≥

0

y

x

0

≥

Przejście między zadaniami jest w dwie strony i jest sprawą umowną, które z zadań jest zadaniem pierwotnym. Dozwolone są następujące przejścia (i tylko te!):

|  |  |
| --- | --- |
| **zadanie pierwotne** | **zadanie dualne** |
| max zp  ... <= ... | min zd  ... >= ... |

Uwaga - nie ma ograniczenia by wektor b był >=0. W przypadku, gdy ograniczenia są dane w postaci równań, należy je zastąpić przez odpowiednie pary nierówności:

ograniczenie: Aix = bi

zamieniamy na ograniczenia: Aix >= bi oraz Aix <= bi

**PRZYKŁAD:** Zbudować zadanie dualne do zadania:

Zadanie dualne :

(min) zd=6y1+5(y2-y3)-7y4

ogr. 3y1+2(y2-y3)-y4 >= 1

2y1+(y2-y3)+3y4>=4

y1,y2,y3,y4>=0

(max) zp=x1+4x2

przekształcamy do wymaganej postaci

(max) zp=x1+4x2

ogr. 3x1+2x2<=6

2x1+x2<=5

-2x1-x2<=-5

-x1+3x2<=-7

x1,x2>=0

ogr. 3x1+2x2<=6

2x1+x2=5

x1-3x2>=7

x1,x2>=0

**TWIERDZENIE**: Jeżeli w zadaniu pierwotnym i-te ograniczenie ma postać **równania**, to i-ta zmienna w zadaniu dualnym jest **dowolna co do znaku**. Jeżeli j-ta zmienna zadania pierwotnego jest **dowolna co do znaku**, to j-te ograniczenie zadania dualnego ma postać **równania**.

**TWIERDZENIE Dantziga i Ordena:**

Jeżeli dla optymalnych rozwiązań dopuszczalnych zadania pierwotnego i dualnego k-te ograniczenie dowolnego zadania jest spełnione jako nierówność (odpowiednia zmienna osłabiająca jest dodatnia) to k-ta zmienna w zadaniu dualnym jest równa 0.

Jeśli k-ta zmienna jest dodatnia w dowolnym zadaniu, to k-te ograniczenie w jego zadaniu dualnym jest równością (odpowiednia zmienna osłabiająca jest równa 0).

Twierdzenie to pozwala ustalić rozwiązanie zadania pierwotnego znając rozwiązanie zadania dualnego ale można to zrobić prościej odczytując z ostatniej tablicy sympleksów.

### Własności rozwiązań zadań dualnych

**TWIERDZENIE:** Jeżeli jedno z zadań dualnych ma skończone rozwiązanie optymalne, to drugie z tych zadań ma także skończone rozwiązanie optymalne oraz zachodzi równość:



**TWIERDZENIE**: Jeżeli w jednym z zadań dualnych rozwiązanie optymalne jest nieograniczone, to drugie z tych zadań jest sprzeczne.

**TWIERDZENIE**: Jeżeli jedno z zadań dualnych jest sprzeczne, to drugie z tych zadań jest sprzeczne albo nie ma skończonego rozwiązania optymalnego.

Z wcześniejszego twierdzenia wiemy, że istnieje związek między i-tą zmienną w jednym zadaniu a i-tym ograniczeniem w drugim zadaniu. Ponadto można wskazać inny związek: zmienne pierwotne i dualne grupują się we wzajemnie sprzężone pary:

xj oraz ym+j i xn+j oraz yj

W każdej z par jeżeli jedna ze zmiennych jest zmienną decyzyjną, to druga jest zmienną osłabiającą (dodatkową) i na odwrót. Oprócz tego, jeśli jedna ze zmiennych jest zmienną bazową, to druga jest zmienną niebazową i na odwrót.

**TWIERDZENIE:** Jeżeli para zadań dualnych ma skończone rozwiązania optymalne to w rozwiązaniu optymalnym zadania dualnego zmienna decyzyjna yj jest równa wskaźnikowi optymalności zn+j-cn+j zmiennej dodatkowej (osłabiającej) xn+j w rozwiązaniu optymalnym zadania pierwotnego. Natomiast zmienna dodatkowa ym+j jest równa wskaźnikowi optymalności zj-cj zmiennej decyzyjnej xj w rozwiązaniu optymalnym zadania pierwotnego.

**Wskaźniki optymalności zadania pierwotnego - maksymalizacja (rozwiązanie optymalne; m<n):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| zmienne decyzyjne | | | zmienne osłabiające | | |
| M+1 | **z** | **z1-c1** | **...** | **zn-cn** | **zn+1 - cn+1** | **...** | **zn+m-cn+m** |

yj = xm+1 , n+j (czyli zn+j - cn+j)

ym+j = xm+1 , j (czyli zj - cj)

**Wskaźniki optymalności zadania dualnego - minimalizacja (rozwiązanie optymalne; m>n):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| zmienne decyzyjne | | | zmienne osłabiające | | |
| n+1 | **z** | **z1-c1** | **...** | **zm-cm** | **zm+1 - cm+1** | **...** | **zm+n - cm+n** |

xj = -yn+1 , m+j (czyli zm+j - cm+j)

xn+j = -yn+1 , j (czyli zj-cj)

Znak minus wynika, że w zadaniu pierwotnym odejmuje się zmienne osłabiające i aby rozpocząć obliczenia z dodatnią macierzą jednostkową, należy pomnożyć układ ograniczeń przez -1.

**WNIOSEK:** Jeżeli zadanie pierwotne ma wymiary m x n takie, że m > n, to wtedy wygodniej jest wyznaczyć rozwiązanie optymalne zadania dualnego a dopiero na jego podstawie rozwiązanie optymalne zadania pierwotnego.

Korzyści rozwiązywania zadania dualnego

|  |  |
| --- | --- |
| zadanie pierwotne | zadanie dualne |
| duża liczba ograniczeń i niewielka liczba zmiennych | baza mniejsza - obliczenia szybsze |
| mamy rozwiązanie optymalne i dodajemy ograniczenie (algorytmy programowania całkowitoliczbowego) i trzeba obliczenia prowadzić od początku | tutaj dodajemy zmienną z wartością 0, baza się nie zmienia, zadanie optymalne nadal jest dopuszczalne i można kontynuować obliczenia. |

### Interpretacja ekonomiczna zadania dualnego

|  |  |
| --- | --- |
| W jakich ilościach produkować wyroby by maksymalizować zysk?  Ograniczenia – zużycie zasobu „i” musi być mniejsze lub równe dostępnej ilości tego zasobu.  (max) zp=cTx  ogr.: Ax≤b  x0  x∈Rn  **funkcja celu:**  zp – dochód ze sprzedaży wszystkich wyrobów (max)  xj – produkcja wyrobu „j”  cj - cena jednostki wyrobu „j”  **ograniczenia:**  aij - zużycie zasobu i do produkcji jednostki wyrobu „j”  bi - dostępna ilość jednostek zasobu „i” | Jakie są minimalne ceny sprzedaży zasobów, by zysk był większy lub równy niż przy wykorzystaniu tych zasobów do produkcji i sprzedaży wyrobów ?  Ograniczenia – ceny ilości zasobów potrzebne na wyprodukowanie jednostki wyrobu „j” muszą być większe lub równe cenie jednostki tego wyrobu  (min) zd=bTy  ogr.: ATy≥c  y0  y∈Rm  **funkcja celu:**  zd – dochód ze sprzedaży wszystkich surowców (min)  yi – cena sprzedaży zasobu „i”  bi - dostępna ilość jednostek zasobu „i”  **ograniczenia:**  aji – zużycie zasobu i do produkcji jednostki wyrobu „j”  cj - cena jednostki wyrobu „j” |

### Przykład

funkcja celu: min z=2x1+x2

ograniczenia: x1<=1

x2<=2

x1+x2>=2

x1,x2>=0

Zadania tego nie da się rozwiązać stosując poznane metody (nie znając idei zmiennych sztucznych). Można je natomiast rozwiązać przekształcając zadanie pierwotne do dualnego. To przekształcenie można wykonać na dwa sposoby, ale rezultat końcowy będzie taki sam:

|  |  |  |  |
| --- | --- | --- | --- |
| Sposób 1:  min zp=2x1+x2  x1<=1  x2<=2  x1+x2>=2  x1,x2>=0 | max z’p=-2x1-x2  x1<=1  x2<=2  -x1-x2<=-2  x1,x2>=0 | min z’d=y1+2y2-2y3  y1-y3 >=-2  y2-y3 >=-1  y1,y2,y3>=0 | do postaci standardowej  min z’d=y1+2y2-2y3  -y1+y3+y4 =2  -y2+y3+y5 =1  y1,y2,y3, y4,y5>=0 |
| Sposób 2:  min zp=2x1+x2  x1<=1  x2<=2  x1+x2>=2  x1,x2>=0 | min zp=2x1+x2  -x1 >= -1  -x2 >= -2  x1+x2 >= 2  x1,x2>=0 | max zd=-y1-2y2+2y3  -y1+y3 <=2  -y2+y3 <=1  y1,y2,y3>=0 | do postaci standardowej  tak jak wyżej |

Teraz możemy rozwiązać zadanie dualne stosując metodę sympleks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 1 | 2 | -2 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 | P5 |
| P4 | 0 | 2 | -1 | 0 | 1 | 1 | 0 |
| P5 | 0 | 1 | 0 | -1 | 1 | 0 | 1 |
| Wsk. | Opt. | 0 | -1 | -2 | 2 | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | 1 | 2 | -2 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 | P5 |
| P4 | 0 | 1 | -1 | 1 | 0 | 1 | -1 |
| P3 | -2 | 1 | 0 | -1 | 1 | 0 | 1 |
| Wsk. | Opt. | -2 | -1 | 0 | 0 | 0 | -2 |

Jest to rozwiązanie optymalne:

Zadanie dualne: (min) z’d=-2

zmienne decyzyjne: y1\*=0 y2\*=0 y3\*=1

zmienne sztuczne: y4\*=1 y5\*=0

Zadanie pierwotne: (min) zp=- z’d =2

zmienne decyzyjne: x1\*= -yn+1,4\*=0 x2\*=- y n+1,5\*=2

zmienne sztuczne: x3\*=- y n+1,1\*=1 x4\*=- y n+1,2\*=0 x5\*=- y n+1,3\*=0

### Zadania

Dane jest zadanie:

Funkcja celu: min 2x1+2x2

Ograniczenia: 2x1+4x2>=1

x1+x2>=2

2x1+x2>=3

x1,x2>=0

Rozwiązać zadanie dualne do tego zadania i odczytać rozwiązanie optymalne zadania pierwotnego.

Rozwiązanie:

???

## Rozwiązywanie zadań PL. za pomocą programu FRACTIONS

### Wprowadzenie

Program Fractions jest programem edukacyjnym wspomagającym naukę programowania liniowego. Pozwala na rozwiązywanie zadań programowania liniowego bez i ze zmiennymi sztucznymi, pozwalana dołączanie ograniczeń w trakcie obliczeń, co można zastosować przy rozwiązywaniu zadań programowania liniowego całkowitoliczbowego.

### Zadanie programowania liniowego bez zmiennych sztucznych

Dane jest następujące zadanie:

(max) zysk = 20\*krzesła+24\*stoły

3\*krzesła+4\*stoły <= 60 - ograniczenie czasu składania wyrobów

6\*krzesła+2\*stoły <= 32 - ograniczenie czasu wykończenia i pakowania

krzesła >= 0 - ograniczenie wielkości produkcji krzeseł

stoły >= 0 - ograniczenie wielkości produkcji stołów

Wprowadzić zadanie do programu (pamiętając o minimalizacji i wierszu wskaźnikowym - POSITIVE VALUES).

Wyjaśnić opisy w wierszu stanu (PHASE, RHS) i wskazywanie elementu centralnego.

Wykonać kroki algorytmu simplex (wchodzi 2 a potem 1). Pokazać, że ten sam wynik można uzyskać wybierając na początku zmienną 1 wchodzącą do bazy.

Niech do bazy wejdzie element, dla którego wskaźnik optymalności jest mniejszy lub równy 0 (pogorszenie lub zmian wartości funkcji celu).

Pokazać co się stanie, gdy wybierzemy element wychodzącego z bazy dla którego iloraz, nie będzie najmniejszą wartością z pośród wszystkich wartości (wartości zmiennych bazowych będą ujemne).

Pokazać, co się stanie, gdy do bazy wejdzie element, w którego ilorazie mianownik był ujemny

### Zadanie programowania liniowego ze zmiennymi sztucznymi

Dane jest następujące zadanie:

(max) zysk = 20\*krzesła+24\*stoły

3\*krzesła+4\*stoły <= 60 - ograniczenie czasu składania wyrobów

6\*krzesła+2\*stoły <= 32 - ograniczenie czasu wykończenia i pakowania

1\*krzesła+0\*stoły >= 5 - ograniczenie na min liczbę wyrobów

krzesła >= 0 - ograniczenie wielkości produkcji krzeseł

stoły >= 0 - ograniczenie wielkości produkcji stołów

Wprowadzić zadanie do programu (pamiętając o minimalizacji i wierszu wskaźnikowym - POSITIVE VALUES).

Pokazać, że system sam wprowadził zmienne sztuczne i będzie rozwiązywał zadanie metodą dużego współczynnika w

Wprowadzić zmienne sztuczne do funkcji celu - COMPUTE / ENTER ARTIFICIALS

W pierwszym kroku usunięta zostanie z bazy zmienna sztuczna

Odczytać końcowe rozwiązanie

### Zadanie programowania liniowego całkowitoliczbowego

Dane jest następujące zadanie:

min z=-x1-x2

2x1+x2<=6

4x1+5x2<=20

x1,x2>=0 i całkowite

Wprowadzić zadanie do programu (pamiętając o minimalizacji i wierszu wskaźnikowym - POSITIVE VALUES).

Rozwiązać zadanie PL. i sprawdzić, czy istnieją zmienne o wartościach ułamkowych

Zgodnie z algorytmem dodać ograniczenie 1/3x3+1/3x4>=2/3 (INSERT / NEW ROW podświetlony wiersz wskaźnikowy)

Dokonać edycji współczynników w dodanym ograniczeniu

Pokazać, że system sam wprowadził zmienne sztuczne i będzie rozwiązywał zadanie metodą dużego współczynnika w

Wprowadzić zmienne sztuczne do funkcji celu - COMPUTE / ENTER ARTIFICIALS

Odczytać wynik z tablicy

# Optymalizacja dyskretna - całkowitoliczbowe programowanie liniowe

## Wprowadzenie

Z dyskretnym modelem decyzyjnym mamy do czynienia wówczas, gdy co najmniej jedna zmienna decyzyjna musi przyjmować wartości z dyskretnego zbioru izolowanych punktów. Szczególnym przypadkiem modeli dyskretnych są modele całkowitoliczbowe, w których część lub wszystkie zmienne decyzyjne muszą przyjmować wartości całkowite. Modele dyskretne mogą być zarówno modelami liniowymi jak i nieliniowymi.



Przykład dyskretności zbioru rozwiązań dopuszczalnych pewnego zadania programowania liniowego. Warunek całkowitoliczbowości sprawia, że do zbioru rozwiązań dopuszczalnych należą izolowane punkty o współrzędnych całkowitych.

Źródła dyskretności:

Rozpatrywanie w modelu obiektów fizycznie niepodzielnych. Przykładem tego rodzaju obiektów są wszelkiego rodzaju środki transportowe.

Kombinatoryczna struktura modelowanej sytuacji decyzyjnej. W problemach tego typu poszczególne punkty zbioru rozwiązań dopuszczalnych reprezentują permutację lub kombinację pewnego zbioru obiektów.

Wprowadzenie dyskretnych zmiennych decyzyjnych umożliwia niekiedy względne łatwe rozwiązanie zadań, w których zbiór rozwiązań dyskretnych nie jest zbiorem wypukłym lub też jest zbiorem niespójnym.

Metody rozwiązania zadań optymalizacji dyskretnej można podzielić na trzy grupy:

metody płaszczyzn tnących,

metody drzew decyzyjnych,

metody heurystyczne.

Pierwsze dwie metody są metodami dokładnymi, metody heurystyczne dostarczają rozwiązań przybliżonych.

W ramach zajęć poświecimy uwagę metodom płaszczyzn tnących. Ograniczymy się tylko **do całkowitoliczbowych modeli liniowych.**:

## Metoda płaszczyzn tnących - algorytm Gomory’ego

Problem całkowitoliczbowego programowania liniowego ma następującą postać:

(min) z=cTx

przy warunkach ograniczających

Ax<=b

x0

xj - liczba całkowita dla j ∈ J ⊂ N gdzie N= {1,2,...,n}

Jeśli J=N (gdy wszystkie zmienne są całkowitoliczbowe), to zadanie powyższe nazywamy kompletnym. Natomiast gdy nie wszystkie zmienne są całkowitoliczbowe, to mamy do czynienia z zadaniem mieszanym.

Oznaczmy przez Xc zbiór rozwiązań zadania (powyższego), a przez X zbiór rozwiązań zadania (bez ost.ograniczenia) - czyli zadania bez warunku całkowitoliczbowości. Wtedy oczywista jest następująca zależność

Xc∈X

Jeśli zadanie ogólne ma skończone rozwiązanie optymalne, to z powyższej zależności wynika następująca nierówność



**TWIERDZENIE:** Jeżeli xo jest rozwiązaniem optymalnym zadania ogólnego, takim, że x0∈Xc to xo jest również rozwiązaniem optymalny zadania całkowitoliczbowego.

Oznaczmy przez P(Xc) wypukłą powłokę zbioru Xc, tzn. Najmniejszy zbiór wypukły zawierający w sobie zbiór Xc.



Ogólna idea **metody płaszczyzn odcinających** (tnących) jest następująca:

jeżeli w wyniku rozwiązania zadania bez warunku dyskretności otrzymano rozwiązanie niecałkowitoliczbowe, to do ograniczeń zadania wyjściowego należy dołączyć nowe ograniczenie liniowe spełniające warunki

1. otrzymane rozwiązanie niecałkowitoliczbowe nie spełnia tego ograniczenia
2. spełnia je dowolne rozwiązanie całkowitoliczbowe.

Następnie rozwiązuje się otrzymane zadanie i jeśli to konieczne dodaje nowe ograniczenie.

Geometrycznie dołączenie nowego ograniczenia liniowego odpowiada odcięciu od wielościanu rozwiązań poprzedni punkt optymalny o współrzędnych ułamkowych, bez odcinania punktów o współrzędnych całkowitych.

Pierwszy realizowalny numerycznie algorytm dla tej metody podał R.E.Gomory.

## Algorytm

1. Znaleźć rozwiązanie optymalne zadania programowania liniowego (bez warunku całkowitoliczbowości).
2. Jeżeli rozwiązanie to jest całkowite, to problem został rozwiązany (koniec metody) w przeciwnym wypadku idź do 3.

Konstruujemy dodatkowe ograniczenia (odcięcia):

3. Wybierz zmienną xk która ma wartość niecałkowitą w ostatnim rozwiązaniu.

4. W l-tym równaniu współczynnik przy tej zmiennej wynosi „1”. Zastąp współczynniki i stałą w l-tym równaniu ich częściami ułamkowymi: -2 zastąp przez 0, 24/7 przez 4/7, -2/3 przez -2/3, -31/6 przez -1/6.

5. Dodaj 1 do każdego ujemnego ułamka wynikającego z kroku 4 i zapisz otrzymane równanie jako ograniczenie nierównościowe ze znakiem „>=”

6. Zamień ograniczenie nierównościowe na równość (wprowadź zmienną osłabiającą i sztuczną). Dołącz równanie na koniec układu równań z ostatniej tablicy sympleksowej i przydziel nowej zmiennej dowolnie duży dodatni współczynnik kosztu M (czyli wprowadź ją do funkcji celu ze współczynnikiem M)

1. Wykonaj dodatkowe iteracje metody sympleks dla nowej tablicy i wróć do kroku 2.

## Przykład

min z=-x1-x2

2x1+x2<=6

4x1+5x2<=20

x1,x2>=0 i całkowite

Postać standardowa:

min z=-x1-x2

2x1+x2+x3=6

4x1+5x2+x4=20

x1,x2>=0 i całkowite

x3,x4>=0

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -1 | -1 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 |
| P3 | 0 | 6 | 2 | 1 | 1 | 0 |
| P4 | 0 | 20 | 4 | 5 | 0 | 1 |
| Wskaźniki | optymalności | 0 | 1 | 1 | 0 | 0 |
| Baza | cB | P0 | -1 | -1 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 |
| P1 | -1 | 3 | 1 | 1/2 | 1/2 | 0 |
| P4 | 0 | 8 | 0 | 3 | -2 | 1 |
| Wskaźniki | optymalności | -3 | 0 | 1/2 | -1/2 | 0 |
| Baza | cB | P0 | -1 | -1 | 0 | 0 |
|  |  |  | P1 | P2 | P3 | P4 |
| P1 | -1 | 5/3 | 1 | 0 | 5/6 | -1/6 |
| P2 | -1 | 8/3 | 0 | 1 | -2/3 | 1/3 |
| Wskaźniki | optymalności | -13/3 | 0 | 0 | -1/6 | -1/6 |

Niech xk = x2; l=2 - modyfikujemy więc równanie 2: 0\*x1+1\*x2-2/3\*x3+1/3\*x4=8/3

Po przekształceniu mamy ograniczenie nierównościowe: 1/3\*x3+1/3\*x4>=2/3

które przekształcamy do postaci równości: 1/3\*x3+1/3\*x4-x5+x6=2/3

Stosujemy algorytm sympleks dla znalezienia rozwiązania optymalnego dla naszego zadania:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -1 | -1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | -1 | 5/3 | 1 | 0 | 5/6 | -1/6 | 0 | 0 |
| P2 | -1 | 8/3 | 0 | 1 | -2/3 | 1/3 | 0 | 0 |
| P6 | M | **2/3** | **0** | **0** | **1/3** | **1/3** | **-1** | **1** |
| Wskaźniki  optymalności |  | **-13/3**  **+2/3M** | **0** | **0** | **-1/6**  **+1/3M** | **-1/6**  **+1/3M** | **-M** | **0** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Baza | cB | P0 | -1 | -1 | 0 | 0 | 0 | M |
|  |  |  | P1 | P2 | P3 | P4 | P5 | P6 |
| P1 | -1 | 0 | 1 | 0 | 0 | -1 | 5/2 | -5/2 |
| P2 | -1 | 4 | 0 | 1 | 0 | 1 | -2 | 2 |
| P3 | 0 | 2 | 0 | 0 | 1 | 1 | -3 | 3 |
| Wskaźniki  optymalności |  | -4 | 0 | 0 | 0 | 0 | -1/2 | 1/2  -M |

Koniec zadania - po jednym odcięciu wszystkie rozwiązania są całkowitoliczbowe. Jedna ze zmiennych bazowych jest równa zeru - otrzymaliśmy więc rozwiązanie zdegenerowane.

Interpretacja geometryczna odcięcia: odcięcie 1/3\*x3+1/3\*x4>=2/3 możemy przedstawić jako x3+x4>=2; zmienne osłabiające x3 i x4 możemy wyrazić w funkcji x1 i x2 ( z ograniczeń) i otrzymamy nowe ograniczenie x1+x2<=4



# Optymalizacja dyskretna – wstęp (under construction)

## Wstęp

**Algorytmem** nazywa się procedurę, która w skończonej liczbie dobrze zdefiniowanych operacji znajduje rozwiązanie danego problemu.

Problemy dzielimy na decyzyjne i optymalizacyjne

**Problem optymalizacyjny** jest to problem w których minimalizowana (maksymalizowana) jest pewna zadana funkcja celu.

**Optymalizacyjny problem kombinatoryczny** jest to problem optymalizacyjny, w którym funkcja celu jest określona na skończonym zbiorze potencjalnych rozwiązań problemu (tzw. zbiorze rozwiązań dopuszczalnych).

Algorytm jest **algorytmem dokładnym** jeśli gwarantuje uzyskanie rozwiązania optymalnego. Przykładem algorytmu dokładnego dla optymalizacyjnych problemów kombinatorycznych jest **algorytm pełnego przeglądu**.

## Złożoność obliczeniowa

Wymagania czasowe danego algorytmu, wyrażone w funkcji rozmiaru instancji problemu, nazywa się **złożonością czasową algorytmu**.

Algorytm o złożoności czasowej wielomianowej (**algorytm wielomianowy**), to taki którego złożoność czasowa jest O(w(n)), przy czym w(n) jest wielomianem, a n oznacza rozmiar instancji problemu. Jeżeli powyższy warunek nie jest spełniony, to algorytm nazywa się algorytmem o złożoności czasowej wykładniczej lub algorytmem wykładniczym.

## Klasy złożonościowe problemów

Złożoność czasowa algorytmów dokładnych, rozwiązujących optymalizacyjne problemy kombinatoryczne, prowadzi do zdefiniowania klas złożonościowych tych problemów.

## Metoda węgierska – problem przydziału

Metoda węgierska znajduje rozwiązanie w czasie wielomianowym (minimalizacja dla wartości dodatnich)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *Task a* | *Task b* | *Task c* | *Task d* |  |
| *Worker 1* | 2 | 10 | 9 | 7 | -2 |
| *Worker 2* | 15 | 4 | 14 | 8 | -4 |
| *Worker 3* | 13 | 14 | 16 | 11 | -11 |
| *Worker 4* | 4 | 15 | 13 | 9 | -4 |
|  |  |  |  |  |  |

1. W każdym wierszu znajdź element minimalny i odejmij go od wszystkich elementów z danego wiersza.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | 0 | 8 | 7 | 5 |  |
| *2* | 11 | 0 | 10 | 4 |  |
| *3* | 2 | 3 | 5 | 0 |  |
| *4* | 0 | 11 | 9 | 5 |  |
|  |  |  | -5 |  |  |

2. W każdej kolumnie znajdź element minimalny i odejmij go od wszystkich elementów z danej kolumny.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | 0 | 8 | 2 | 5 |  |
| *2* | 11 | 0 | 5 | 4 |  |
| *3* | 2 | 3 | 0 | 0 |  |
| *4* | 0 | 11 | 4 | 5 |  |
|  |  |  |  |  |  |

3. W wierszu zawierającym dokładnie **jedno** **nienaznaczone** „0” – **naznacz** je( „**0**”). **Skreśl** („0”) pozostałe zera w kolumnie z naznaczonym zerem by nie powtórzyć przydziału innego pracownika do tego samego zadania . Powtarzaj do chwili gdy w wierszu będą same naznaczone „0” lub co najmniej 2 nienaznaczone.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | **0** | 8 | 2 | 5 |  |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | 0 | 0 |  |
| *4* | 0 | 11 | 4 | 5 |  |
|  |  |  |  |  |  |

4. Wykonaj punkt 3 dla kolumn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | **0** | 8 | 2 | 5 |  |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | **0** | 0 |  |
| *4* | 0 | 11 | 4 | 5 |  |
|  |  |  |  |  |  |

5. Powtarzaj 3 i 4 aż do momentu gdy zajdzie jeden z 3 przypadków:

1. każdy wiersz ma naznaczone „0”
2. w co najmniej 2 wierszach i kolumnach są co najmniej 2 nienaznaczone „0”
3. nie ma nienaznaczonych zer a kompletnego przydziału nie otrzymano

6. Jeśli 5a – to przydział jest kompletny i optymalny

Jeśli 5b naznacz dowolne zero, skreśl pozostałe zera w wierszu i kolumnie, go to 3

Jeśli 5c – go to 7

7. Oznacz wiersze dla których nie dokonano przydziału („v”)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | **0** | 8 | 2 | 5 |  |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | **0** | 0 |  |
| *4* | 0 | 11 | 4 | 5 | v |
|  |  |  |  |  |  |

8. Oznacz nieoznaczone kolumny, które mają „0” w oznaczonym wierszu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | **0** | 8 | 2 | 5 |  |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | **0** | 0 |  |
| *4* | 0 | 11 | 4 | 5 | v |
|  | v |  |  |  |  |

1. Oznacz nieoznaczone wiersze, które mają przydział w oznaczonych kolumnach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | **0** | 8 | 2 | 5 | v |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | **0** | 0 |  |
| *4* | 0 | 11 | 4 | 5 | v |
|  | v |  |  |  |  |

10. Powtarzaj 8 i 9 do zakończenia procesu oznaczania.

1. Pokryj wszystkie nieoznaczone wiersze i wszystkie oznaczone kolumny

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | **0** | 8 | 2 | 5 | v |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | **0** | 0 |  |
| *4* | 0 | 11 | 4 | 5 | v |
|  | v |  |  |  |  |

Znajdź minimalny element niepokryty i odejmij go od każdego elementu w niepokrytych wierszach.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | -2 | 6 | 0 | 3 | -2 |
| *2* | 11 | **0** | 5 | 4 |  |
| *3* | 2 | 3 | **0** | 0 |  |
| *4* | -2 | 9 | 2 | 3 | -2 |
|  |  |  |  |  |  |

Dodaj ten sam element do każdego elementu w pokrytych kolumnach.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | 0 | 6 | 0 | 3 |  |
| *2* | 13 | **0** | 5 | 4 |  |
| *3* | 4 | 3 | **0** | 0 |  |
| *4* | 0 | 9 | 2 | 3 |  |
|  | +2 |  |  |  |  |

Go to 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | 0 | 6 | 0 | 3 |  |
| *2* | 13 | 0 | 5 | 4 |  |
| *3* | 4 | 3 | 0 | 0 |  |
| *4* | 0 | 9 | 2 | 3 |  |
|  |  |  |  |  |  |

Naznaczamy i skreślamy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Min* | *a* | *b* | *c* | *d* |  |
| *1* | 0 | 6 | **0** | 3 |  |
| *2* | 13 | **0** | 5 | 4 |  |
| *3* | 4 | 3 | 0 | **0** |  |
| *4* | **0** | 9 | 2 | 3 |  |
|  |  |  |  |  |  |

Przydział optymalny: 1c, 2b, 3d, 4a

# Optymalizacja dyskretna - metoda podziału i ograniczeń

## Wprowadzenie

Ogólna idea metody podziału i ograniczeń polega na **uporządkowanym przeszukiwaniu zbioru rozwiązań** zadania optymalizacyjnego, o którym zakładamy że jest skończony. Zbiór wszystkich rozwiązań (D) dzieli się sukcesywnie na mniejsze podzbiory (**podział**) a dla każdego podzbioru oblicza się **kres dolny** czyli oszacowanie z dołu wartości funkcji celu (minimalizacja). W kolejnych krokach dokonujemy podziału tych zbiorów, które mają najmniejszą wartość kresu dolnego (minimalizacja). Podział postępuje do chwili znalezienia rozwiązania dopuszczalnego, dla którego wartość funkcji celu jest nie większa niż najmniejsze dolne ograniczenie wszystkich nie podzielonych podzbiorów (**ograniczenie**). W najgorszym przypadku należy sprawdzić wszystkie rozwiązania, co przesądza o wykładniczej złożoności obliczeniowej tej metody.

Dla wyliczenia kresów dolnych (minimalizacja) należy zdefiniować funkcję ograniczającą „w”. Musi ona spełniać następujące warunki (minimalizacja):

Efektywność metody podziału i ograniczeń zależy od jakości oszacowań kresów dla każdego podzbioru. Im te oszacowania są dokładniejsze, tym mniej potrzeba podziałów. Idealna funkcja ograniczająca jest następująca:



Metodę podziału i ograniczeń wykorzystuje się z powodzeniem do rozwiązywania szerokiej klasy zadań o charakterze kombinatorycznym: programowanie całkowitoliczbowe, szeregowanie zadań niepodzielnych, problem komiwojażera, problem lokalizacji, problem plecakowy, problem przydziału i inne.

## Algorytm

Niech D będzie skończonym zbiorem rozwiązań dopuszczalnych. Przez L(D’) oznaczymy dolną granicę wartości funkcji celu f na pewnym podzbiorze D’ zbioru rozwiązań D (w szczególności D’=D) a przez k numer kroku podziałowego.

**Krok 1.** Oblicz L(D). Jeśli uda się znaleźć takie rozwiązanie x, że f(x)=L(D), to x jest rozwiązaniem optymalnym. W przeciwnym razie podziel według określonego sposobu zbiór D na skończoną liczbę podzbiorów:



podstaw k:=1 i przejdź do kroku 2.

**Krok 2.** Oblicz. Jeśli przy tym uda się znaleźć takie rozwiązanie x, że dla danego r (1<=r<=rk) i , i=1..rk, to x jest rozwiązaniem optymalnym. W przeciwnym razie wybierz podzbiór według zasady:



i podziel go na skończoną liczbę (zwykle rozłącznych) podzbiorów



**Krok 3.** Wszystkie nie podzielone dotąd podzbiory



oznacz przez



podstaw k:=k+1 i wróć do kroku 2.

Do realizacji opisanego schematu metody podziału i ograniczeń jest konieczne skonkretyzowanie

* zasady podziału
* wyznaczania dolnych ograniczeń
* znajdowania rozwiązania

Należy zwrócić uwagę na sposób realizacji zasady wyboru podzbioru do podziału (krok 2). Zamiast sprawdzania dolnych ograniczeń wszystkich nie podzielonych podzbiorów, można ograniczyć się do podzbiorów wynikających z ostatniego podziału. Zaletą pierwszego sposobu (na ogół) jest mniejsza liczba podziałów, potrzebny jest jednak większy obszar pamięci do przechowywania informacji o wszystkich nie podzielonych podzbiorach. Drugi sposób to na ogół większa liczba podziałów lecz mniejsza zajętość pamięci.

## Przykład

Dany jest zbiór „m” różnych wyrobów, z których każdy może być produkowany na jednym z „m” stanowisk produkcyjnych. Każde ze stanowisk ma różną wydajność przy produkcji każdego z wyrobów. Wydajność i-tego stanowiska przy produkcji j-tego wyrobu jest równa cij, a ponadto każde ze stanowisk ma produkować dokładnie jeden wyrób i każdy z wyrobów ma być produkowany na dokładnie jednym stanowisku. Należy przydzielić wyroby do stanowisk, by łączna wydajność była maksymalizowana.

Problem ten nazywa się **problemem przydziału** i można zapisać go formalnie w sposób następujący:



Problem przydziału jest szczególnym przypadkiem problemu transportowego

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **WYRÓB / STANOWISKO** | **1** | **2** | **3** | **4** |
| **A** | 2 | 10 | 9 | 7 |
| **B** | 15 | 4 | 14 | 8 |
| **C** | 13 | 14 | 16 | 11 |
| **D** | 4 | 15 | 13 | 9 |

## Problem komiwojażera - Algorytm Little’a

Metoda podziału i ograniczeń może być zastosowana dla rozwiązania problemu komiwojażera. Algorytm został podany przez Little’a.

Zbiór rozwiązań (wierzchołek w drzewie poszukiwań) będziemy rozbijać na dwa podzbiory:

* zawierający wyróżniony łuk <i,j>
* nie zawierający łuku <i,j>

Podział będzie dokonywany z pewną zasadą heurystyczną, opisaną poniżej. Po wykonaniu podziału liczone są kresy dolne na drodze **redukcji** macierzy kosztów przejść. Podzbiory rozwiązań mające wartości kresów dolnych **większe lub równe** długości najkrótszego dotychczas znalezionego rozwiązania będą pomijane (ograniczamy w ten sposób przestrzeń poszukiwań).

Przykład:

Dana jest macierz kosztów przejść „C” („∞” oznacza koszt o nieskończonej wartości):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Macierz C* | *1* | *2* | *3* | *4* |  |
| *1* | ∞ | 2 | 7 | 3 |  |
| *2* | 7 | ∞ | 8 | 5 |  |
| *3* | 9 | 4 | ∞ | 6 |  |
| *4* | 3 | 8 | 5 | ∞ |  |
|  |  |  |  |  |  |

Niech G0 oznacza początkowy zbiór rozwiązań. W celu wyznaczenia kresu dolnego dla G0 dokonujemy redukcji macierzy C

* w każdym wierszu znajdujemy element minimalny i odejmujemy go od wszystkich elementów danego wiersza
* ten sam proces wykonujemy dla kolumn.

Po wykonaniu redukcji macierzy, w każdym wierszu i każdej kolumnie powinien znaleźć się element zerowy. Suma wielkości, które odjęliśmy (stopień redukcji macierzy R) **dodawana jest do dotychczasowej wartości dolnego ograniczenia** (wartość początkowa dolnego ograniczenia = 0):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Macierz C0* | *1* | *2* | *3* | *4* |  |
| *1* | ∞ | 0 | 3 | 1 | -2 |
| *2* | 2 | ∞ | 1 | 0 | -5 |
| *3* | 5 | 0 | ∞ | 2 | -4 |
| *4* | 0 | 5 | 0 | ∞ | -3 |
|  |  |  | -2 |  | R=16 |

Po redukcji: (dolne ograniczenie = 16). Kandydatami do wykonania podziału są następujące odcinki (mające w tej zredukowanej macierzy koszt równy 0): <1,2>,<2,4>,<3,2>,<4,1>,<4,3>. Wybieramy ten z nich, który posiada **najwyższy optymistyczny koszt wyłączenia**. Jeżeli wyłączymy odcinek <i,j>, to z miasta „i” musimy wyruszyć do miasta innego niż „j”, a do miasta „j” musimy przybyć z miasta różnego od „i”. Rozważmy dla przykładu odcinek <2,4> Jeżeli wyłączymy ten odcinek, to z miasta 2 musimy wyruszyć do miasta różnego od 4, a minimalny koszt z tym związany jest równy 1 (minimalna wartość w wierszu 2 różna od tej na pozycji 2,4). Z kolei do miasta 4 trzeba dotrzeć z miasta różnego od 2 i minimalny koszt z tym związany jest równy 1 (minimalna wartość w kolumnie 4 różna od tej na pozycji 2,4). **Suma tych wartości** (2) jest **optymistycznym kosztem** który na pewno musimy ponieść wyłączając odcinek <2,4>. Wartości tych kosztów podane są w postaci indeksów przy elementach zerowych:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *macierz C0* | *1* | *2* | *3* | *4* |  |
| *1* | ∞ | 01 | 3 | 1 | -2 |
| *2* | 2 | ∞ | 1 | 02 | -5 |
| *3* | 5 | 02 | ∞ | 2 | -4 |
| *4* | 02 | 5 | 01 | ∞ | -3 |
|  |  |  | -2 |  | R=16 |

G0 (16)

W naszym wypadku aż trzy odcinki mają optymistyczny koszt wyłączenia równy 2. Arbitralnie wybieramy odcinek <2,4> tworząc zbiory marszrut G1 i G2 zawierające lub nie zawierające tego odcinka.

Weźmy pod uwagę zbiór G1. Tworzymy macierz C1 usuwając z macierzy C0 wiersz 2 i kolumnę 4 oraz przyjmując c42=∞ (blokujemy drogę przeciwną). Zmodyfikowana i zredukowana macierz C1 znajduje się poniżej:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Macierz C1* | *1* | *2* | *3* |  |
| *1* | ∞ | 03 | 3 |  |
| *3* | 5 | 05 | ∞ |  |
| *4* | 05 | ∞ | 03 |  |
|  |  |  |  | R=0 |

Weźmy pod uwagę zbiór G2. Tworzymy macierz C2 z macierzy C0 przyjmując c24=∞. Zmodyfikowana i zredukowana macierz C2 znajduje się poniżej:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *macierz C2* | *1* | *2* | *3* | *4* |  |
| *1* | ∞ | 00 | 3 | 01 |  |
| *2* | 1 | ∞ | 01 | ∞ | -1 |
| *3* | 5 | 01 | ∞ | 1 |  |
| *4* | 01 | 5 | 00 | ∞ |  |
|  |  |  |  | -1 | R=2 |

G2 (18)

~<2,4>

G1 (16)

<2,4>

G0 (16)

Do dalszego podziału wybieramy wierzchołek o najmniejszej wartości kresu dolnego G1. Z tablicy C1 wynika, że tylko odcinki <3,2> oraz <4,1> mogą być rozpatrywane (zerowy koszt, maksymalna optymistyczna wartość kosztu wyłączenia 5). Arbitralnie wybieramy <3,2> tworząc zbiory G3 i G4.

Weźmy pod uwagę zbiór G3. Tworzymy macierz C3 usuwając z macierzy C1 wiersz 3 i kolumnę 2. Nie dokonujemy podstawienia c32=∞, gdyż ten element został już wyeliminowany. Zmodyfikowana i zredukowana macierz C3 znajduje się poniżej:

|  |  |  |  |
| --- | --- | --- | --- |
| *Macierz C3* | *1* | *3* |  |
| *1* | ∞ | 0∞ | -3 |
| *4* | 0∞ | 00 |  |
|  |  |  | R=3 |

Weźmy pod uwagę zbiór G4. Tworzymy macierz C4 z macierzy C1 przyjmując c32=∞. Zmodyfikowana i zredukowana macierz C4 znajduje się poniżej:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Macierz C4* | *1* | *2* | *3* |  |
| *1* | ∞ | 0∞ | 3 |  |
| *3* | 0∞ | ∞ | ∞ | -5 |
| *4* | 00 | ∞ | 03 |  |
|  |  |  |  | R=5 |

G0 (16)

G2 (18)

~<2,4>

G1 (16)

<2,4>

G4 (21)

~<3,2>

G3 (19)

<3,2>

Do dalszego podziału wybieramy wierzchołek o najmniejszej wartości kresu dolnego G2. Z tablicy C2 wynika, że tylko odcinki <1,4> <2,3> oraz <3,2> mogą być rozpatrywane (zerowy koszt, maksymalna optymistyczna wartość kosztu wyłączenia 1). Arbitralnie wybieramy <1,4> tworząc zbiory G5 i G6.

Weźmy pod uwagę zbiór G5. Tworzymy macierz C5 usuwając z macierzy C2 wiersz 1 i kolumnę 4. Ponadto dokonujemy podstawienia c41=∞. Zmodyfikowana i zredukowana macierz C5 znajduje się poniżej:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *macierz C5* | *1* | *2* | *3* |  |
| *2* | 04 | ∞ | 00 |  |
| *3* | 4 | 09 | ∞ |  |
| *4* | ∞ | 5 | 05 |  |
|  | -1 |  |  | R=1 |

Weźmy pod uwagę zbiór G6. Tworzymy macierz C6 z macierzy C2 przyjmując c14=∞. Zmodyfikowana i zredukowana macierz C6 znajduje się poniżej:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *macierz C2* | *1* | *2* | *3* | *4* |  |
| *1* | ∞ | 03 | 3 | ∞ |  |
| *2* | 1 | ∞ | 01 | ∞ |  |
| *3* | 5 | 00 | ∞ | 0∞ |  |
| *4* | 01 | 5 | 00 | ∞ |  |
|  |  |  |  | -1 | R=1 |

G0 (16)

G2 (18)

~<2,4>

G1 (16)

<2,4>

G6 (19)

~<1,4>

G5 (19)

<1,4>

G4 (21)

~<3,2>

G3 (19)

<3,2>

Do dalszego podziału wybieramy wierzchołek o najmniejszej wartości kresu dolnego G3. Z tablicy C3 wynika, że tylko odcinki <1,3> oraz <4,1> mogą być rozpatrywane (zerowy koszt, maksymalna optymistyczna wartość kosztu wyłączenia ∞). Arbitralnie wybieramy <1,3> tworząc zbiory G7 i G8.

Weźmy pod uwagę zbiór G7. Tworzymy macierz C7 usuwając z macierzy C3 wiersz 1 i kolumnę 3. Macierz redukuje się do jednego elementu. Zmodyfikowana i zredukowana macierz C7 znajduje się poniżej:

|  |  |  |
| --- | --- | --- |
| *Macierz C7* | *1* |  |
| *4* | 0 |  |
|  |  | R=0 |

UWAGA – z tej macierzy wynika, że dołączając do wierzchołka G7 jedyną możliwą marszrutę <4,1> otrzymujemy rozwiązanie końcowe, dla którego wartość funkcji kosztu równa się kresowi dolnemu równemu 19. Oznaczmy to rozwiązanie przez wierzchołek G9.

Weźmy pod uwagę zbiór G8. Tworzymy macierz C8 z macierzy C3 . Podstawiamy za c13=∞. Zmodyfikowana macierz C8 znajduje się poniżej: UWAGA – koszt redukcji wynosi ∞ (patrz wiersz 1). Kres dolny byłby równy także ∞. Oznacza to, że wierzchołek ten nie będzie dalej dzielony – można dowieść, że nie odpowiada mu żadna marszruta. Pomijamy więc w dalszych rozważaniach ten wierzchołek.

|  |  |  |  |
| --- | --- | --- | --- |
| *Macierz C8* | *1* | *3* |  |
| *1* | ∞ | ∞ |  |
| *4* | 0∞ | 00 |  |
|  |  |  | R = ∞ (!) |

G0 (16)

G1 (16)

<2,4>

G2 (18)

~<2,4>

G3 (19)

<3,2>

G6 (19)

~<1,4>

G5 (19)

<1,4>

G4 (21)

~<3,2>

G7 (19)

<1,3>

G8 (\*)

~<1,3>

G9 (19)

<4,1>

Podział pozostałych zbiorów nie może dać rozwiązania lepszego. Otrzymane rozwiązanie jest optymalne, choć niekoniecznie jedyne. W celu stwierdzenia czy istnieją inne rozwiązania, należałoby dokonać podziału zbiorów G5 i G6.

# Optymalizacja dyskretna - metaheurystyki

## Wprowadzenie

**Złożoność obliczeniowa**

Dotychczas zostały omówione dwie metody problemów programowania dyskretnego: metoda płaszczyzn tnących i metoda podziału i ograniczeń. Metody te mają istotne wady, które często ograniczają ich praktyczne zastosowanie. Wadą tą jest **złożoność obliczeniowa algorytmów programowania dyskretnego**. Najczęściej **pojęcie złożoności obliczeniowej algorytmu** rozwiązywania danego problemu odnosi się do czasu obliczeń i wyrażone jest **jako pewna funkcja rozmiaru problemu** (długości danych wejściowych). Przykładowo, jeśli algorytm rozwiązuje problem o rozmiarze n w ciągu c\*n2 jednostek czasu, to mówimy wówczas, że algorytm ten ma złożoność obliczeniową O(n2). Algorytm, którego złożoność obliczeniowa jest ograniczona wielomianem zmiennej n (tzn. jest O(p(n)) gdzie p jest wielomianem a n rozmiarem konkretnego problemu) nazywamy **algorytmem wielomianowym** **lub algorytmem o wielomianowym czasie obliczeń**. Każdy algorytm, którego złożoność obliczeniowa nie może być tak ograniczona nosi nazwę **algorytmu wykładniczego**. Algorytmy wielomianowe przyjęło się nazywać algorytmami efektywnymi, zaś algorytmy wykładnicze - nieefektywnymi.

**Metody heurystyczne**

Okazuje się, że większość problemów programowania dyskretnego nie posiada algorytmów wielomianowych. W szczególności omówione algorytmy Gomory’ego (dla metody płaszczyzn tnących) i podziału i ograniczeń (dla metody drzew decyzyjnych) są algorytmami wykładniczymi. Z praktycznego punktu widzenia jest to istotna wada tych metod. Wielu praktycznych modeli decyzyjnych nie udaje się rozwiązać za ich pomocą, nawet przy wykorzystaniu bardzo szybkich komputerów. Stało się to przyczyną poszukiwania **heurystycznych metod rozwiązywania modeli dyskretnych** dostarczających rozwiązań suboptymalnych (bliskich optymalnym). Z tego też względu heurystyczne metody rozwiązywania modeli dyskretnych określa się niekiedy mianem **metod przybliżonych**.

**Podział metod heurystycznych**

Metody przybliżone konstruowane są zasadniczo dwoma sposobami. Pierwszy z nich jest poszukiwaniem losowym, a drugi jest poszukiwaniem deterministycznym, polegającym na opracowaniu heurystycznych zasad, wykorzystujących specyfikę modelu dyskretnego.

**Metody poszukiwania losowego** polegają albo na pełnym rozwiązaniu zadania przy zastosowaniu pewnego mechanizmu przypadkowego błądzenia albo też na połączeniu poszukiwania losowego (wyznaczanie punktu startowego) z lokalną optymalizacją (poszukiwanie lokalnego optimum funkcji celu). W tym drugim podejściu dokonuje się wielokrotnego losowego wyboru punktu startowego i dla odpowiednio dużej liczby optimów lokalnych wybiera się najlepsze rozwiązanie.

**Metody poszukiwania deterministycznego** odnoszą się do specjalnych typów zadań. Wykorzystują one strukturę i właściwości modelu, co pozwala określić pewne zasady poszukiwania rozwiązania optymalnego. Dają one lepsze rozwiązania niż metody losowe.

**Tworzenie metod heurystycznych**

Zasady tworzenia metod heurystycznych nie poddają się jakiejkolwiek formalizacji. Ogólnie metody te wykorzystują pewne reguły heurystyczne poszukiwania rozwiązań lepszych od aktualnie istniejącego. Oparciem do tworzenia takich metod są intuicyjne metody poszukiwania twórczych rozwiązań. Warto wspomnieć, że każda metoda podziały i ograniczeń, której postępowanie nie zostanie doprowadzone do końca, jest metodą heurystyczną.

## Metaheurystyki

Generalnie metody heurystyczne tworzone są dla ściśle określonej klasy problemów. Im węższy zakres tej klasy, tym więcej można znaleźć deterministycznych reguł poszukiwania lepszych rozwiązań i tworzone heurystyki dostarczają lepszych rezultatów. Istnieje jednak grupa metod heurystycznych, które można zastosować dla wielu klas problemów. Dopasowanie do poszczególnego problemu można dokonać przez zmianę cechujących je parametrów. Tę grupę metod heurystycznych nazywa się często **metaheurystykami**. Najważniejsze metaheurystyki to:

metoda przeszukiwania tabu (ang. tabu search)

algorytmy genetyczne (ang. genetic algorithms)

metoda symulowanego wyżarzania (ang. simulated annealing)

## Algorytm Tabu (ang. Tabu Search - TS)

Główną cechą tego algorytmu jest możliwość wybrania rozwiązania pogarszającego wartość funkcji celu oraz wprowadzenie listy TABU.

Jeżeli następuje przejście z rozwiązania *x* do *y*, to *x* wprowadza się na koniec listy tabu i przebywa ono przez k iteracji, gdzie k jest długością listy. Dowolne rozwiązanie *z* nie zostanie wybrane ponownie tak długo, jak długo przebywa ono na liście tabu. Lista tabu została wprowadzona dla uniknięcia cykli (związanych z wyjściem i powrotem do lokalnego optimum), co zmusza algorytmu do przeszukiwania nowych regionów rozwiązań.

## Programy ewolucyjne

### Wstęp

Program ewolucyjny (ang. evolution program) jest to

algorytm probabilistyczny

w którym występuje populacja osobników P(t) = {x1t,...,xnt} gdzie t- numer iteracji

każdy osobnik reprezentuje potencjalne rozwiązanie

istnieje funkcja szacująca przystosowanie osobników (nieujemna, maksymalizowana) (ang. fitness function)

w każdym kroku wybiera się najlepsze osobniki (selekcja rodziców) o największej wartości funkcji przystosowania (np. metoda ruletki)

a następnie dokonuje się transformacji na wybranych osobnikach:

1. krzyżowanie - dotyczy dwóch osobników
2. mutacja - dotyczy jednego osobnika

### Algorytm

begin

t<-0

inicjuj P(t)

oszacuj P(t)

while not koniec (ocena najlepszego osobnika, iteracje, czas) do

begin

t<- t+1

wybierz P(t) z P(t-1)

transformuj P(t)

oszacuj P(t)

end

end

Algorytm genetyczny jest specjalnym przypadkiem programu ewolucyjnego:

reprezentacja danych to BINARNE łańcuchy o stałej długości (chromosomy)

transformacja składa się tylko z BINARNEJ mutacji i krzyżowania

### Uwagi

Stosując algorytm ewolucyjny do rozwiązania dowolnego problemu, trzeba najpierw rozwiązać następujące podproblemy:

genetyczna reprezentacja rozwiązań

inicjacja początkowej populacji

postać funkcji dopasowania

postać operatora selekcji

operatory genetyczne (mutacja i krzyżowanie)

wartości parametrów (rozmiar populacji, prawdopodobieństwa zastosowania operatorów)

generowanie rozwiązań dopuszczalnych

### Funkcja dopasowania

Zbyt dobra definicja funkcji dopasowania (ang. fitness function) może prowadzić do zbyt szybkiej dominacji dobrych chromosomów i zmniejszenia znaczenia operatorów genetycznych (szybkie zakończenie algorytmu w otoczeniu najlepszego aktualnie chromosomu – ekstremum lokalne).

Zbyt słaba definicja może skutkować brakiem zbieżności do rozwiązania optymalnego.

### Operator selekcji

Poprawna definicja pozwala uniknąć przedwczesną zbieżność. Prawdopodobieństwo selekcji chromosomu k:



może być funkcją czasu (liczby iteracji):



gdzie: T – spodziewana liczba iteracji, q – mała liczba naturalna. Ma to na celu zmniejszenie prawdopodobieństwa selekcji dobrych osobników w początkowej fazie i dopuszczenie do krzyżowania gorszych osobników, które mogą jednak prowadzić do lepszych rozwiązań.

### Operatory genetyczne

Krzyżowanie w trakcie upływu czasu staje się coraz mniej efektywne, gdyż chromosomy są coraz bardziej podobne do siebie. Można ulepszyć proces krzyżowania:

wybierając miejsce do krzyżowania w miejscu, gdzie chromosomy się różnią między sobą

zmieniając prawdopodobieństwo krzyżowania w zależności od urozmaicenia populacji

### Generowanie rozwiązań dopuszczalnych

Problem generowania rozwiązań dopuszczalnych można rozwiązać na dwa sposoby:

generować tylko rozwiązania dopuszczalne (czasochłonne)

dodawać dużą karę za brak dopuszczalności

### Zastosowania algorytmu genetycznego

Problem komiwojażera

Problem komiwojażera (ang. traveling salesman problem - TSP) zdefiniowany jest następująco: komiwojażer startuje z miasta A i musi odwiedzić wszystkie pozostałe miasta dokładnie raz. Na koniec ma powrócić do miasta A. Zadaniem jest znalezienie drogi o najmniejszym koszcie.

Załóżmy że mamy miasta oznaczone cyframi 1,2,3,4,5. Komiwojażer startuje z miasta 1 i musi do niego wrócić.

genetyczna reprezentacja rozwiązań - wyjątkowo miasta będą kodowane nie binarnie ale liczbami całkowitoliczbowymi: (2,3,4,5). Miasto od którego zaczynamy i kończymy nie musi być kodowane

inicjacja początkowej populacji - trzeba wygenerować możliwe permutacje

postać funkcji dopasowania - określamy koszt odwiedzin miast w zadanej kolejności (nie zapominając o mieście będącym początkiem i końcem)

operatory genetyczne (mutacja i krzyżowanie) - omijamy mutację i stosować będziemy jedynie krzyżowanie w zmienionej postaci:

1. dane są przykładowo dwa chromosomy: (2,3,4,5,6,7,8,9,10)
2. oraz (3,5,6,2,9,10,4,8,7)
3. losujemy z pierwszego chromosomu fragment, który nie będzie zmieniany (2,3,**4,5,6,7**,8,9,10)
4. nowy chromosom będzie miał postać: (a,b,4,5,6,7,c,d,e)
5. analizując drugi chromosom wypełnimy brakujące miejsca: a:3, (5 nie - bo występuje w kopiowanym fragmencie), (6 nie - patrz 5) b:2, c:9,d:10, (4 nie - patrz 5) e:8, (7 nie - patrz 5)

wartości parametrów:

1. rozmiar populacji – im większy rozmiar, tym lepsze wyniki (kosztem czasu)
2. prawdopodobieństwa krzyżowania – dopasować samodzielnie

Znajdowanie ekstremum funkcji

Dana jest funkcja np. f(x)=x\*sin(10\*pi\*x)+1. Znaleźć ekstremum w przedziale [-1,2]

genetyczna reprezentacja rozwiązań - wartości liczbowe będziemy kodować binarnie. Załóżmy, że oczekujemy precyzji z dokładnością do 6 miejsc po przecinku (odcinek o długości 1 podzielony na milion przedziałów) w sumie mamy 3 odcinki jednostkowe \* 1 000 000 = 3 000 000. Przypomnijmy, że na n-bitach można zakodować liczby z zakresu od 0 do 2n-1. Żeby zakodować tę liczbę potrzebujemy 22 bity: 221=2097152, 222=4194304. Zamiana liczby binarnej x’ na dziesiętną x będzie następująca: x=-1 + x’\*3/(222-1). Trzeba pamiętać, że możemy wyjść poza zakres

inicjacja początkowej populacji - trzeba wygenerować możliwe permutacje

postać funkcji dopasowania – wartość funkcji dla danego x

operatory genetyczne - klasyczna mutacja i krzyżowanie

wartości parametrów:

1. rozmiar populacji – im większy rozmiar, tym lepsze wyniki (kosztem czasu)
2. prawdopodobieństwa krzyżowania – dopasować samodzielnie

## Algorytm symulowanego odprężania

### Wprowadzenie

Jednym z procesów opisywanym przez fizykę statyczną jest proces krystalizacji. Z doświadczenia wiadomo, że wybrany materiał w pewnej temperaturze może znajdować się w różnych stanach stabilnych, w zależności od jego historii. Przykładem może być wymrażanie krzemu do postaci ciała stałego. Gwałtowne obniżanie temperatury płynnego krzemu prowadzi do otrzymania substancji amorficznej o wysokim nieuporządkowaniu atomów. Z kolei mniej radykalne schłodzenie może prowadzić do struktury polikrystalicznej, tzn. zlepka mikrokryształków. Dopiero przy bardzo powolnym wymrażaniu, gwarantującym utrzymanie w dowolnym przedziale temperatury stanu równowagi, istnieje możliwość otrzymania doskonale uporządkowanego kryształu krzemu.

Wraz ze wzrostem uporządkowania układu maleje jego energia. Z powyższego przykładu wynika, że zależnie od procesu schładzania możemy doprowadzić układ, jaki stanowi rozważany płynny krzem, do różnych stanów odpowiadających różnym minimom lokalnym energii. Jedynie powolne obniżanie temperatury (odprężanie), zapewniające utrzymanie stanu równowagi, zapobiegające powstawaniu wewnętrznych naprężeń, pozwala na osiągnięcie globalnego minimum energii, odpowiadające idelanemu kryształowi.

Wyznaczenie stanu najniższej energii jest problemem optymalizacji, a zatem fizyczny proces odprężania może stanowić inspirację do budowy nowej metaheurystyki.

### Algorytm

Pierwszy algorytm podano dla symulacji ruchów termicznych atomów cieczy w ustalonej temperaturze T (Metropolis 1953). W każdym kroku dokonuje się próbnego, losowego przesunięcia jednej z cząstek i wylicza wynikającą z przesunięcia zmianę energii całego systemu ΔE. Jeżeli energia ulega zmniejszeniu - przesunięcie jest akceptowane. Zwiększenie energii także jest akceptowane z pewnym prawdopodobieństwem:

P(ΔE)=exp(-ΔE/T)

Gdzie T oznacza temperaturę. Jak widać, prawdopodobieństwo akceptacji gorszych rozwiązań maleje wraz ze spadkiem temperatury.

Algorytm jest następujący:

BEGIN

S:=S0

T:=T0

WHILE not koniec DO

BEGIN

S':= nowe rozwiązanie sąsiednie

ΔE:=E(S')-E(S)

P( ΔE):= exp(-ΔE/T)

IF (ΔE<0) or (RANDOM<P( ΔE)) THEN S:=S'

Aktualizacja T

END

END

Funkcja celu (minimalizacja) może być funkcją energii. Przy obniżaniu temperatury nie zaleca się stosowania stałego kroku. Dokładne badania wykazały, że najlepszym rozwiązaniem jest powiązanie temperatury T z numerem iteracji n:



# Metoda programowania dynamicznego (Błażewicz)

## Wprowadzenie

Mówiąc o programowaniu liniowym mówiliśmy o problemie programowania liniowego. Programowanie dynamiczne nie jest problemem lecz metodą rozwiązywania pewnej klasy problemów. Ponadto rozpatrywaliśmy dotychczas problemy optymalizacji statycznej, w którym kryterium optymalności jest funkcją n-zmiennych. W optymalizacji dynamicznej kryterium optymalności jest funkcjonałem a rozwiązanie - funkcjami czasu.

Niniejszy rozdział opisuje opracowaną przez R. Bellmana ogólną metodę rozwiązywania pewnej klasy modeli decyzyjnych. Zakładamy, że modele te:

są wieloetapowymi procesami decyzyjnymi

posiadają własność Markowa

## Wieloetapowe procesy decyzyjne

DEFINICJA Wieloetapowy proces decyzyjny

Wieloetapowy proces decyzyjny (WPD) jest to proces, którego stan na każdym etapie zależy od decyzji wybranej ze zbioru decyzji dopuszczalnych.

Oznaczmy przez Sk (k=1,2,...,N) zbiór możliwych w k-tym etapie wartości zmiennej stanu (zbiór możliwych stanów). Oznacza to, że zmienna sk może przyjmować jedynie wartości ze zbioru Sk. Przez Dk (k=1,2,...,N) oznaczymy zbiór możliwych decyzji w k-tym etapie - zmienna xk może przyjmować jedynie wartości ze zbioru Dk. Wieloetapowy proces decyzyjny może zostać wyrażony następującą zależnością rekurencyjną:

sk=T(sk-1,xk-1) (k=1,2,...,N, sk∈Sk, xk∈Dk) (6.1)

Z wieloetapowym procesem decyzyjnym jest związana funkcja celu:

f(s1, s2, ..., sN; x1 ,x2 ,..., xN)

Problem, który należy rozwiązać w każdym wieloetapowym procesie decyzyjnym polega na określeniu ciągu decyzji:

x1\*,x2\*,...,xN\* (xj\*∈Dj)

przy których ustalona funkcja celu dla całości procesu przebiegającego w N etapach osiąga optimum. Ciąg decyzji optymalnych nazywać będziemy ***strategią optymalną***.

DEFINICJA Własność Markowa

WPD ma własność Markowa jeżeli po dowolnej liczbie decyzji (k) wpływ pozostałych etapów procesu decyzyjnego na wartość funkcji celu zależy jedynie od stanu procesu po k-tym etapie i od decyzji następnych (dalszy przebieg procesu nie zależy od jego historii lecz tylko od stanu w danej chwili).

Własność Markowa mają na przykład WPD z funkcją celu:

z=f1(s1,x1)+ f2(s2,x2)+ ... + fN(sN,xN)

PRZYKŁAD:

Należy przetransportować ładunek z punktu A do E. Układ komunikacyjny między tymi punktami wraz odległościami przedstawiony jest na rysunku poniżej. Zadanie polega na wyborze najkrótszej trasy.

8

B1

3

3

11

4

D1

C1

1

6

5

7

E

B2

A

4

2

4

12

D2

C2

10

B3

7

Z analizy rysunku wynika następujący schemat trasy z A do E:



w którym można wyróżnić 4 **etapy** i proces transportu z A do E można traktować jako **wieloetapowy proces decyzyjny**. Stanem początkowy jest punkt A i **decyzja na pierwszym etapie** polega na wyborze trasy do jednego z trzech punktów: B1, B2, B3. Podobnie interpretujemy decyzje i osiągane w ich wyniku stany na kolejnych etapach trasy.

W dowolnym etapie wieloetapowego procesu decyzyjnego wyróżnić można następujące elementy:

stan wejściowy procesu danego etapu - jest to stan jaki osiągnął proces w wyniku realizacji etapu poprzedniego

decyzję podejmowaną na danym etapie

stan wyjściowy procesu z danego etapu - stan ten zależy od stanu wejściowego oraz podjętej decyzji na danym etapie.

## Metoda programowania dynamicznego

Podstawowym elementem koncepcji dynamicznego rozwiązania zadania WPD z własnością Markowa jest tzw. zasada optymalności sformułowana przez R.Bellmana:

DEFINICJA Zasady Optymalności Bellmana

Strategia optymalna ma tę własność, że niezależnie od stanu początkowego i decyzji początkowej, pozostałe decyzje muszą stanowić politykę optymalną ze względu na stan wynikający z pierwszej decyzji.

Interpretacja: Jeżeli x1\*,...,xn\* jest strategią optymalną procesu N-etapowego, to również strategią optymalną jest strategia x2\*,...,xN\* dla procesu N-1 etapowego przy stanie początkowym s2=T(s1,x1\*).

**Algorytm programowania dynamicznego:**

Niech

z=f1(s1,x1)+ f2(s2,x2)+ ... + fN(sN,xN)

będzie funkcją celu którą należy zminimalizować. Zgodnie z zasadą optymalności

min {f1(s1,x1)+ f2(s2,x2)+ ... + fN(sN,xN)}=

min {f1(s1,x1)+ min {f2(s2,x2)+ ... + fN(sN,xN)}}

Wprowadzając oznaczenie

q1(s1) = min {f1(s1,x1)+ f2(s2,x2)+ ... + fN(sN,xN)}

otrzymamy

qk(sk) = min {fk(sk,xk) + qk+1(sk+1)}

qN(sN) = min {fN(sN,xN)}

Ostatecznie otrzymamy **równanie rekurencyjne Bellmana** (RRB)

qk(sk) = min { fk(sk,xk)+ qk+1(T(sk,xk)) }

qN(sN) = min {fN(sN,xN)}

Rozpoczynamy od etapu N. Dla każdego stanu wejściowego etapu N (dla każdego s∈SN) oblicz:

qN(sN) = min{ fN(xN, sN) }

Przechodzimy do etapu k (k=N-1,..,1). Dla każdego stanu wejściowego etapu k (dla każdego s∈Sk) oblicz:

qk(sk) = min {fk(sk,xk)+ qk+1(sk+1)}

Otrzymujemy minimalną wartość funkcji kryterialnej z\*=g1(s1) oraz optymalną decyzję x1\*. Znając si oraz xi\* możemy określić si+1\*

si+1\*=T(si,xi\*)

PRZYKŁAD cd.:

Wprowadźmy oznaczenia:

FH oznaczać będzie najkrótszą drogę pomiędzy F i H

FH=L[G] oznaczać będzie, że najkrótsza droga z F do H wynosi L i następnikiem F jest G

Zadanie polega na znalezieniu najkrótszej drogi z A do E. Oznaczmy ją następująco: A-Bk-Cl-Dm-E. Z zasady optymalności Bellmana wynika, że:

najkrótszą drogą z Bk do E musi być Bk-Cl-Dm-E

a jeżeli to, to najkrótszą drogą z Cl-E musi być Cl-Dm-E

a jeżeli to, to najkrótszą drogą z Dm do E musi być Dm-E.

W oparciu o algorytm programowania dynamicznego należy:

znaleźć DiE - najkrótszą drogę z Di do E (dla każdego Di)

znaleźć CiE - najkrótszą drogę z Ci do E (dla każdego Ci) jako min{CiDj +DjE}

znaleźć BiE - najkrótszą drogę z Bi do E (dla każdego Bi) jako min{BiCj +CjE}

znaleźć AE - najkrótszą drogę z A do E jako min{ABj +BjE}

Będziemy wyznaczać odległości począwszy od ostatniego etapu, przechodząc do etapu pierwszego.

D1E = 4[E] D2E= 7[E]

C1E = min{C1Di +DiE} = min {5+4, 1+7} = 8[D2] C2E=min {3+4, 2+7} = 7[D1]

B1E = min{8+8,11+7} = 16[C1] B2E=min{3+8, 10+7}=11[C1] B3E=min{12+8,4+7}=11[C2]

AE =min{4+16,6+11,7+11}=17[B2]

Możemy określić najkrótszą drogę z A do E idąc od początku:

A - B2 - C1 - D2 - E

## Jednowymiarowy proces alokacji

Jednowymiarowe procesy alokacji są klasą problemów, które mogą być rozwiązane za pomocą programowania dynamicznego.

**Przykład jednowymiarowego procesu alokacji:**

Pięć unikatowych maszyn należy rozdysponować pomiędzy trzy rodzaje działalności. Efekt działalności poszczególnych rodzajów w zależności od liczby przyporządkowanych im maszyn przedstawia tabela:

|  |  |  |  |
| --- | --- | --- | --- |
| Liczba maszyn | Działalność A (3) | Działalność B (2) | Działalność C (1) |
| 0 | 0 | 0 | 0 |
| 1 | 5 | 4 | 3 |
| 2 | 14 | 8 | 6 |
| 3 | 16 | 12 | 11 |
| 4 | 16 | 16 | 23 |
| 5 | 16 | 20 | 23 |

Alokacji maszyn należy dokonać w taki sposób, by łączny wynik był maksymalny.

**Formalny zapis modelu jednowymiarowego procesu alokacji:**

Wprowadzimy następujące oznaczenia:

N - liczba działalności

M - ilość dostępnego zasobu

xk - liczba zasobu przydzielona do działalności k-tej

fk(xk) - efekt przydzielenia xk ilości zasobu do k-tej działalności

funkcja celu: (max) 

ograniczenia: 

xi >= 0 (i=1,2,...,N)

W dalszej części będziemy mówić o przydzielanym do działalności **zasobie**. Interpretacja jednowymiarowego procesu alokacji w kontekście wieloetapowych procesów decyzyjnych może być następująca. Etap możemy zdefiniować jako przydział zasobu do danej działalności, przy czym nie jest istotna kolejność etapów. W naszym przykładzie będziemy mieli 3 etapy. Przez stan procesu (x) będziemy rozumieli wielkość zasobu, który pozostał jeszcze do rozdzielenia między pozostałe działalności. W stanie początkowym, gdzie zasobu nie przydzielono jeszcze żadnej działalności, x=M (M - dostępna ilość zasobu).

**Idea rozwiązania:**

Wprowadźmy dodatkową funkcję:



gdzie: 

Funkcja ta określa **maksymalny przychód wynikający z przydziału x jednostek zasobu do działalności k .. N**. Dysponując **x** jednostkami zasobu i przydzielając k-tej działalności **xk** jednostek, dla pozostałych działalności zostaje **x-xk** jednostek. Efekt całkowity można zapisać wzorem:







Strategię optymalną wyznacza się w sposób następujący:



gdzie oznacza ilość zasobów przydzielonych działalności i-tej

**Przykład rozwiązania problemu:**

Rozwiążemy nasze obliczając rekurencyjnie wartość funkcji fi(s). Wyniki umieszczać będziemy w tabelce:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| s | q3(s) |  | q2(s) |  | q1(s) |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 5 | 1 | 5 | 0 | 5 | 0 |
| 2 | 14 | 2 | 14 | 0 | 14 | 0 |
| 3 | 16 | 3 | 18 | 1 | 18 | 0 |
| 4 | 16 | 4 | 22 | 2 | 23 | 4 |
| 5 | 16 | 5 | 26 | 3 | 28 | 4 |













Odczytanie wyników - zgodnie ze wzorem od końca:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| s | q3(s) |  | q2(s) |  | q1(s) |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 5 | 1 | 5 | 0 | 5 | 0 |
| 2 | 14 | 2 | 14 | 0 | 14 | 0 |
| 3 | 16 | 3 | 18 | 1 | 18 | 0 |
| 4 | 16 | 4 | 22 | 2 | 23 | 4 |
| 5 | 16 | 5 | 26 | 3 | 28 | 4 |



Łączny optymalny efekt wynosi 28 jednostek zysku - można to odczytać z tabeli w pozycji q1(M=5).

1. M.Siudak - zadanie, S.I.Gass - zagadnienie; spotykany też termin - problem [↑](#footnote-ref-1)
2. M.Siudak podaje jeszcze jedną postać – postać kanoniczną [↑](#footnote-ref-2)
3. M.Siudak - max, S.I.Gass - min [↑](#footnote-ref-3)
4. układy w których ilość niewiadomych jest większa niż ilość równań (m<n) nazywa się układami nieoznaczonymi i albo nie mają rozwiązania albo mają ich nieskończenie wiele – takimi układami zajmuje się programowanie liniowe [↑](#footnote-ref-4)
5. patrz – interpretacja geometryczna zadania programowania liniowego [↑](#footnote-ref-5)