Programowanie liniowe

Algorytm sympleks III

- Otrzymywanie postaci kanonicznej

- Tworzenie macierzy jednostkowej

- Technika podproblemów

- Zastosowanie obrotów do tworzenia podproblemu

OTRZYMYWANIE POSTACI KANONICZNEJ Z POSTACI STANDARDOWEJ

Wcześniej pokazaliśmy, że pewnych sytuacjach (postać sprzeczna) tablicy sympleksowej nie można przekształcić z postaci standardowej do postaci kanonicznej. Teraz pokażemy jak to zrobić wtedy, gdy jest to możliwe.

Proces przekształcania programu liniowego z postaci standardowej do postaci kanonicznej (ewentualnie stwierdzenia, że program jest sprzeczny) jest często nazywany *pierwszą fazą* algorytmu sympleks.

Drugą fazą algorytmu sympleks nazywa się zwykle proces otrzymywania postaci optymalnej (lub postaci nieograniczonej) za pomocą serii obrotów tablicy sympleksowej, zgodnych z zasadą sympleks.

TWORZENIE MACIERZY JEDNOSTKOWEJ

Na początku postaramy się (za pomocą serii obrotów) przekształcić tablicę sympleksową tak, aby macierz ograniczeń zawierała kolumny jednostkowe tworzące macierz jednostkową, a współczynniki funkcji celu w tych kolumnach były równe zero.

Po każdym obrocie zmienne decyzyjne pozostają nieujemne, natomiast nie będziemy (na razie!) dążyć do tego, aby wyrazy wolne były nieujemne.

Rozważmy następującą tablicę:

0	1	1	0	0	-2
1	$\underline{1}$	0	-1	2	-2
4	-1	-1	2	-1	1
5	0	-1	1	1	-1

Po obrocie wokół zaznaczonego elementu kolumna x_1 staje się pierwszą kolumną jednostkową oraz $c_1 = 0$:

-1	0	1	1	-2	0
1	1	0	-1	2	-2
5	0	$\underline{-1}$	1	1	-1
5	0	-1	1	1	-1

Uwaga 1. Obroty, które wykonujemy w tej fazie, nie są zgodne z zasadą sympleks! Jako oś kolejnego obrotu przyjmujemy pierwszy od lewej, niezerowy element w kolejnym wierszu. Element ten może być ujemny.

Po kolejnym obrocie wokół zaznaczonego elementu kolumna x_2 staje się drugą kolumną jednostkową oraz $c_2 = 0$:

4	0	0	2	-1	-1
1	1	0	-1	2	-2
-5	0	1	-1	-1	1
0	0	0	0	0	0

Wiersz złożony z samych zer, który pojawił się w ostatniej tablicy świadczy o tym, że ostatni warunek ograniczający (ostatnie równanie) jest zbędny, bo jest zawsze spełniony (każde rozwiązanie spełniające pozostałe warunki będzie też spełniać ten).

Zatem ten ostatni warunek może być wyeliminowany, co skutkuje tablicą w równoważnej postaci:

4	0	0	2	-1	-1
1	1	0	-1	2	-2
-5	0	1	-1	-1	1

Jest to postać jaką chcieliśmy otrzymać, ponieważ mamy macierz jednostkową z zerowymi wagami funkcji celu dla kolumn jednostkowych. Nie jest to jeszcze postać kanoniczna, ponieważ nie wszystkie wyrazy wolne sa nieujemne.

Uwaga 2. Powyższy algorytm może być powtórzony w każdym innym przypadku (za wyjątkiem przypadku sprzecznego) dając w efekcie tablicę sympleksową w postaci „prawie kanonicznej”, tzn. z ewentualnymi ujemnymi wyrazami wolnymi.

Uwaga 3. Proces otrzymywania macierzy jednostkowej jest nazywany *fazą zerową* algorytmu sympleks.

TECHNIKA PODPROBLEMÓW

(Uzyskiwanie nieujemnych wyrazów wolnych)

W wyniku działania fazy „0'' algorytmu sympleks może się zdarzyć, że otrzymamy tablicę w postaci kanonicznej - jeżeli wszystkie wyrazy wolne otrzymanej tablicy są nieujemne.

Może się też zdarzyć, że otrzymamy tablicę sprzeczną I lub II rodzaju.

Albo otrzymamy tablicę w postaci „prawie kanonicznej'', jeżeli niektóre wyrazy wolne są ujemne. Pokażemy jak postępować, gdy zajdzie ostatni przypadek.

Rozważmy następujący przykład:

$$T_1 \quad = \quad \begin{array}{|r|rrrrrrrrr|}
\hline
81 & 0 & 0 & 0 & 2 & 14 & -42 & 13 & 0 & -12 \\
\hline
-55 & 0 & 0 & 0 & 0 & -5 & 25 & -6 & 1 & 5 \\
-4 & 1 & 0 & 0 & -1 & 0 & 10 & 0 & 0 & 1 \\
2 & 0 & 1 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \\
1 & 0 & 0 & 1 & -1 & 2 & -11 & 2 & 0 & -1 \\
\hline
\end{array}$$

Widzimy, że ta tablica nie jest w postaci kanonicznej, ponieważ wyrazy wolne b_1 i b_2 są ujemne.

Możemy teraz wykonać obrót tej tablicy, co z pewnością nie pogorszy sytuacji - tablica w dalszym ciągu będzie zawierać macierz jednostkową z odpowiadającymi jej zerowymi współczynnikami funkcji celu. Ponadto jeżeli dobrze wybierzemy oś obrotu - możemy zwiększyć wartość ujemnych wyrazów wolnych (np. b_2).

Właściwy wybór osi obrotu podpowiada metoda nazywana *techniką podproblemów*.

Tworzenie podproblemu

1. Niech ograniczeniami w podproblemie będą te ograniczenia oryginalnej tablicy, które mają nieujemne wyrazy wolne.
2. Niech funkcją celu podproblemu będzie to ograniczenie oryginalnej tablicy, które ma ujemny wyraz wolny.

Załóżmy, że w naszym przykładzie chcemy wykonać obrót tablicy T_1 zwiększający wartość wyrazu wolnego b_2 (tutaj równie dobrze moglibyśmy wybrać wyraz b_1, ponieważ on też jest ujemny).

Przyjmujemy wiersz zawierający b_2 (drugi wiersz macierzy ograniczeń) jaki wiersz funkcji celu podproblemu, natomiast wiersze trzeci i czwarty jako ograniczenia podproblemu.

Wydzielamy podproblem z oryginalnej tablicy:

$$
T_1 \quad = \quad
\begin{array}{|r|rrrrrrrrr|}
\hline
81 & 0 & 0 & 0 & 2 & 14 & -42 & 13 & 0 & -12 \\
\hline
-55 & 0 & 0 & 0 & 0 & -5 & 25 & -6 & 1 & 5 \\
\hline
\hline
-4 & 1 & 0 & 0 & -1 & 0 & 10 & 0 & 0 & 1 \\
\hline
2 & 0 & 1 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \\
1 & 0 & 0 & 1 & -1 & 2 & -11 & 2 & 0 & -1 \\
\hline
\end{array}
$$

Dzięki swojej konstrukcji jest to program w postaci kanonicznej. Można zatem, za pomocą serii obrotów, doprowadzić go do postaci optymalnej lub nieograniczonej.

Osie obrotów wyznaczamy stosując zasadę sympleks do podproblemu, ale obrót wykonujemy dla całej tablicy.

$$T_1 = \begin{array}{|r|rrrrrrrrr|}
\hline
81 & 0 & 0 & 0 & 2 & 14 & -42 & 13 & 0 & -12 \\
\hline
-55 & 0 & 0 & 0 & 0 & -5 & 25 & -6 & 1 & 5 \\
\hline
\hline
-4 & 1 & 0 & 0 & -1 & 0 & 10 & 0 & 0 & 1 \\
\hline
2 & 0 & 1 & 0 & \underline{1} & -1 & 1 & -1 & 0 & 0 \\
1 & 0 & 0 & 1 & -1 & 2 & -11 & 2 & 0 & -1 \\
\hline
\end{array}$$

$$T_2 = \begin{array}{|r|rrrrrrrrr|}
\hline
77 & 0 & -2 & 0 & 0 & 16 & -44 & 15 & 0 & -12 \\
\hline
-55 & 0 & 0 & 0 & 0 & -5 & 25 & -6 & 1 & 5 \\
\hline
\hline
-2 & 1 & 1 & 0 & 0 & -1 & 11 & -1 & 0 & 1 \\
\hline
2 & 0 & 1 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \\
3 & 0 & 1 & 1 & 0 & \underline{1} & -10 & 1 & 0 & -1 \\
\hline
\end{array}$$

$$T_3 = \begin{array}{|r|rrrrrrrrr|}
\hline
29 & 0 & -18 & -16 & 0 & 0 & 116 & -1 & 0 & 4 \\
\hline
-40 & 0 & 5 & 5 & 0 & 0 & -25 & -1 & 1 & 0 \\
\hline
\hline
1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
5 & 0 & 2 & 1 & 1 & 0 & -9 & 0 & 0 & -1 \\
3 & 0 & 1 & 1 & 0 & 1 & -10 & 1 & 0 & -1 \\
\hline
\end{array}$$

W tym przypadku optymalizacja podproblemu doprowadziła do uzyskania dodatniej wartości wyrazu b_2.

Gdyby okazało się, że optymalizacja podproblemu daje $b_2 < 0$, to oznaczałoby to, że oryginalny problem jest sprzeczny - sprzeczność drugiego rodzaju.

W uzyskanej tablicy jeszcze wyraz b_1 pozostał ujemny. Tworzymy nowy podproblem:

$T_3 =$

29	0	-18	-16	0	0	116	-1	0	4
-40	0	5	5	0	0	-25	-1	1	0
1	1	2	1	0	0	1	0	0	0
5	0	2	1	1	0	-9	0	0	-1
3	0	1	1	0	1	-10	1	0	-1

i powtarzamy powyższe rozumowanie.

$$T_3 = \begin{array}{|r|rrrrrrrrr|}
\hline
29 & 0 & -18 & -16 & 0 & 0 & 116 & -1 & 0 & 4 \\
\hline
\hline
-40 & 0 & 5 & 5 & 0 & 0 & -25 & -1 & 1 & 0 \\
\hline
1 & 1 & 2 & 1 & 0 & 0 & \underline{1} & 0 & 0 & 0 \\
5 & 0 & 2 & 1 & 1 & 0 & -9 & 0 & 0 & -1 \\
3 & 0 & 1 & 1 & 0 & 1 & -10 & 1 & 0 & -1 \\
\hline
\end{array}$$

$$T_4 = \begin{array}{|r|rrrrrrrrr|}
\hline
-87 & -116 & -250 & -132 & 0 & 0 & 0 & -1 & 0 & 4 \\
\hline
\hline
-15 & 25 & 55 & 30 & 0 & 0 & 0 & -1 & 1 & 0 \\
\hline
1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
14 & 9 & 20 & 10 & 1 & 0 & 0 & 0 & 0 & -1 \\
13 & 10 & 21 & 11 & 0 & 1 & 0 & \underline{1} & 0 & -1 \\
\hline
\end{array}$$

$$T_5 = \begin{array}{|r|rrrrrrrrr|}
\hline
-74 & -106 & -229 & -121 & 0 & 1 & 0 & 0 & 0 & 3 \\
\hline
\hline
-2 & 35 & 76 & 41 & 0 & 1 & 0 & 0 & 1 & -1 \\
\hline
1 & 1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
14 & 9 & 20 & 10 & 1 & 0 & 0 & 0 & 0 & -1 \\
13 & 10 & 21 & 11 & 0 & 1 & 0 & 1 & 0 & -1 \\
\hline
\end{array}$$

Po dwóch obrotach tablica T_5 podproblemu jest w postaci nieograniczonej, a wyraz wolny b_1 jest nadal ujemny. Teraz jednak wartość funkcji celu dla podproblemu możemy zmiejszać nieograniczenie (!) - a tym samym zwiększać wartość b_1. W tym celu wykonujemy obrót wokół podkreślonego elementu.

T_5 =

-74	-106	-229	-121	0	1	0	0	0	3
-2	35	76	41	0	1	0	0	1	$\underline{-1}$
1	1	2	1	0	0	1	0	0	0
14	9	20	10	1	0	0	0	0	-1
13	10	21	11	0	1	0	1	0	-1

T_6 =

-80	-1	-1	2	0	4	0	0	3	0
2	-35	-76	-41	0	-1	0	0	-1	1
1	1	2	1	0	0	1	0	0	0
16	-26	-56	-31	1	-1	0	0	-1	0
15	-25	-55	-30	0	0	0	1	-1	0

Zauważmy, że w ostatniej tablicy podproblem nie jest w postaci optymalnej, jednak nie musimy się nim dalej zajmować! Cel został osiągnięty - wyraz wolny b_1 jest większy od zera, a cała tablica jest w postaci kanonicznej:

-80	-1	-1	2	0	4	0	0	3	0	
2	-35	-76	-41	0	-1	0	0	-1	1	
1	$\underline{1}$	2	1	0	0	1	0	0	0	
16	-26	-56	-31	1	-1	0	0	-1	0	
15	-25	-55	-30	0	0	0	1	-1	0	

Kolejny obrót oryginalnego problemu, wokół zaznaczonego elementu prowadzi do postaci optymalnej i rozwiązania.

Tablica optymalna jest postaci:

-79	0	1	3	0	4	1	0	3	0
37	0	-1	-6	0	-1	35	0	-1	1
1	1	2	1	0	0	1	0	0	0
42	0	-4	-5	1	-1	26	0	-1	0
40	0	-5	-5	0	0	25	1	-1	0

Wektorem optymalnym jest:

$$\widehat{x} = [1, 0, 0, 42, 0, 0, 40, 0, 37]^T$$

z wartością optymalną:

$$z(\widehat{x}) = 79.$$

ZASTOSOWANIE OBROTÓW DO TWORZENIA PODPROBLEMU

Pokazaliśmy jak, wykorzystując technikę podproblemów, wyjściową tablicę, otrzymaną w fazie „0", przekształcić do postaci kanonicznej lub stwierdzić, że rozważany problem jest sprzeczny.

Pozostaje jeszcze jedno pytanie: *„Co zrobić, gdy wszystkie wyrazy wolne tej tablicy są mniejsze od zera i nie można utworzyć żadnego podproblemu?"*

Prosta odpowiedź: *zmienić znaki w jednym wierszu* - nie jest dobra, ponieważ „psuje się" wtedy macierz jednostkowa!

Rozważmy następujący przykład:

75	0	0	5	-1	2	0
-10	1	0	$\underline{-1}$	0	0	0
-5	0	1	-1	2	-2	0
-15	0	0	-1	1	-1	1

Aby problem nie był sprzeczny, w każdym wierszu, oprócz ujemnego wyrazu wolnego, musi być co najmniej jeden element ujemny!

Dokonując obrotu wokół takiego elementu otrzymamy tablicę z co najmniej jednym wyrazem wolnym dodatnim i można będzie rozpocząć tworzenie podproblemów.

$$T_1 \quad = \quad \begin{array}{|r|rrrrrr|}
\hline
75 & 0 & 0 & 5 & -1 & 2 & 0 \\
\hline
-10 & 1 & 0 & \underline{-1} & 0 & 0 & 0 \\
-5 & 0 & 1 & -1 & 2 & -2 & 0 \\
-15 & 0 & 0 & -1 & 1 & -1 & 1 \\
\hline
\end{array}$$

$$T_2 \quad = \quad \begin{array}{|r|rrrrrr|}
\hline
25 & 5 & 0 & 0 & -1 & 2 & 0 \\
\hline
10 & -1 & 0 & 1 & 0 & 0 & 0 \\
5 & -1 & 1 & 0 & 2 & -2 & 0 \\
-5 & -1 & 0 & 0 & 1 & -1 & 1 \\
\hline
\end{array}$$

Po wykonaniu obrotu widzimy, że b_1 i b_2 są nieujemne, więc możemy utworzyć podproblem używając wiersza pierwszego i drugiego jako ograniczeń, a wiersza trzeciego jako funkcji celu.

Mamy

$$
T_2 \quad = \quad \begin{array}{|r|rrrrrr|}
\hline
25 & 5 & 0 & 0 & -1 & 2 & 0 \\
\hline
\hline
10 & -1 & 0 & 1 & 0 & 0 & 0 \\
5 & -1 & 1 & 0 & 2 & -2 & 0 \\
\hline
-5 & \underline{-1} & 0 & 0 & 1 & -1 & 1 \\
\hline
\end{array}
$$

Otrzymany podproblem jest w postaci nieograniczonej, więc dokonując obrotu wokół zaznaczonego elementu, dostajemy całą tablicę w postaci kanonicznej:

$$
T_3 \quad = \quad \begin{array}{|r|rrrrrr|}
\hline
0 & 0 & 0 & 0 & 4 & -3 & 5 \\
\hline
15 & 0 & 0 & 1 & -1 & 1 & -1 \\
10 & 0 & 1 & 0 & 1 & -1 & -1 \\
5 & 1 & 0 & 0 & -1 & \underline{1} & -1 \\
\hline
\end{array}
$$

Po jeszcze jednym obrocie, zgodnie z zasadą sympleks, wokół zaznaczonego elementu, otrzymujemy tablicę w postaci optymalnej:

15	3	0	0	1	0	2
10	-1	0	1	0	0	0
15	1	1	0	0	0	-2
5	1	0	0	-1	1	-1

Wektorem optymalnym jest:

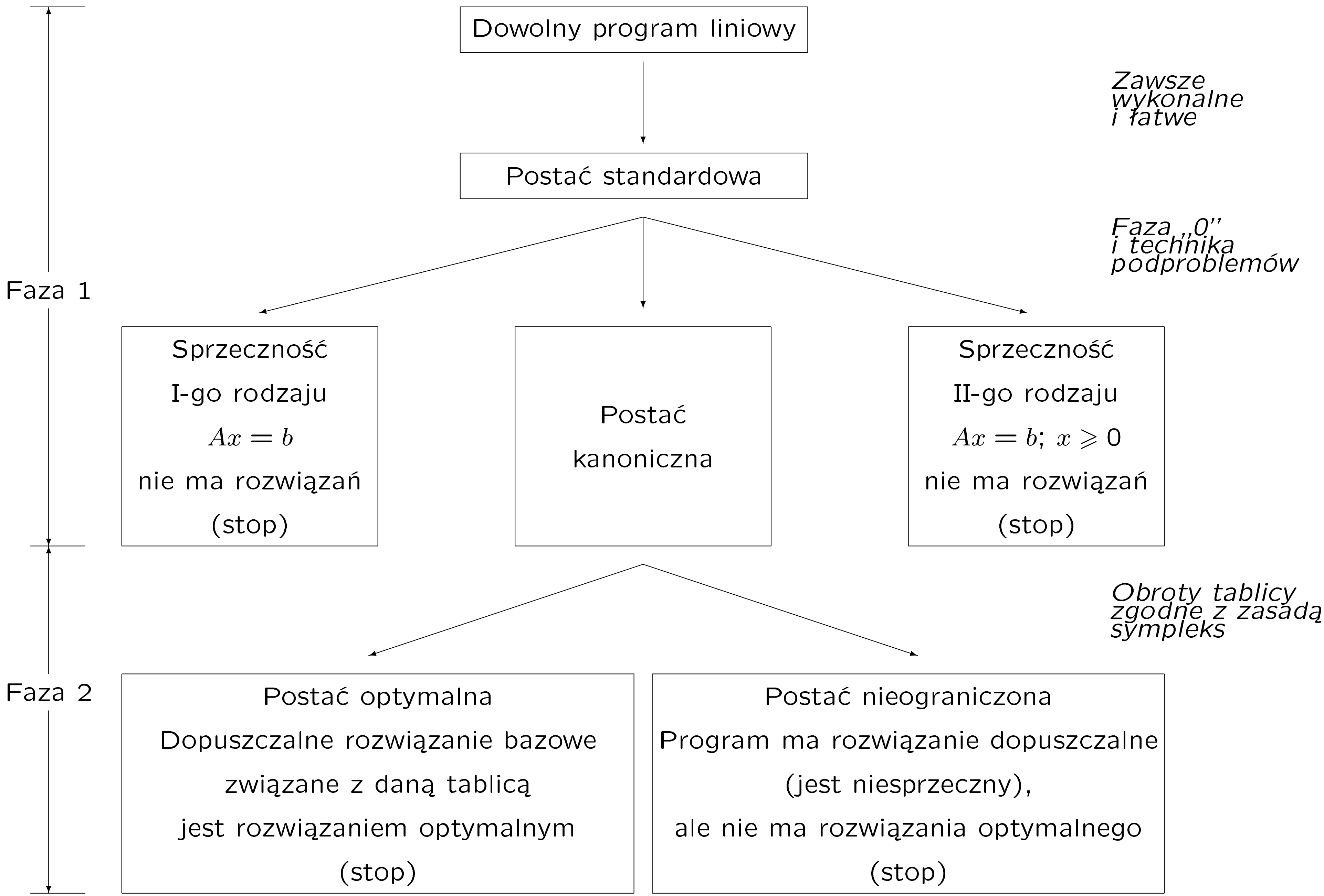
$$\widehat{x} = [0, 15, 10, 0, 5, 0]^T$$

z wartością optymalną:

$$z(\widehat{x}) = -15.$$

Uwaga 4. Procedurę optymalizacji podproblemu można zawsze przerwać (po kolejnym obrotcie) w momencie, gdy pojawi się dodatkowy wyraz wolny większy od zera i zacząć tworzyć nowy podproblem (jeśli to konieczne).

Poniższy diagram przedstawia pełną procedurę zastosowania algorytmu sympleks do dowolnego programu liniowego.


Dowolny program liniowy
Postać standardowa
Zawsze wykonalne i łatwe
Faza „0'' i technika podproblemów
Faza 1
Sprzeczność I-go rodzaju $Ax = b$ nie ma rozwiązań (stop)
Postać kanoniczna
Sprzeczność II-go rodzaju $Ax = b$; $x \geqslant 0$ nie ma rozwiązań (stop)
Obroty tablicy zgodne z zasadą sympleks
Faza 2
Postać optymalna Dopuszczalne rozwiązanie bazowe związane z daną tablicą jest rozwiązaniem optymalnym (stop)
Postać nieograniczona Program ma rozwiązanie dopuszczalne (jest niesprzeczny), ale nie ma rozwiązania optymalnego (stop)