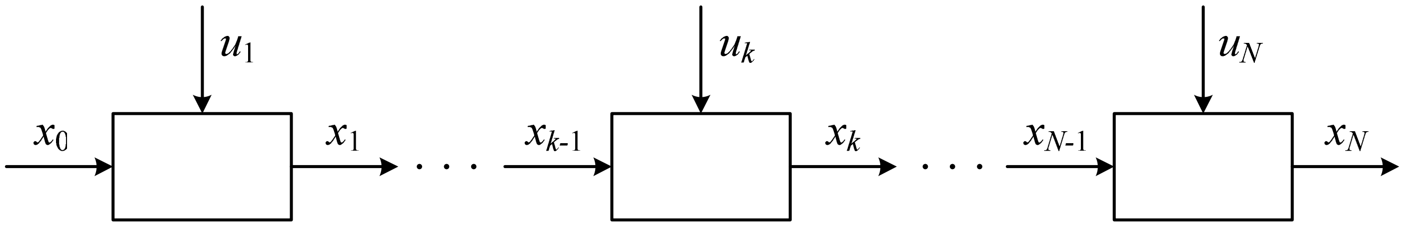
Metody dokładne rozwiązywania problemów dyskretnych

- algorytmy specjalizowane
- metody przeglądu
  - metoda podziału i oszacowań (*branch & bound*)
  - *branch & cut, branch & price* i inne odmiany
  - programowanie w logice ograniczeń
- metody odcięć
- **programowanie dynamiczne**
- modele sieci przepływowych

Wieloetapowy problem (proces) decyzyjny



$$x_k = f_k\,(x_{k\text{-}1}, u_k)\ , \quad k = 1, 2, \dots , N$$

Cel:

$$\text{min (lub max)}\ \ g(x_0, u_1, u_2, ..., u_N)$$

$$x_k \in X_k\ , \quad k = 1, 2, \dots , N$$

$$u_k \in U_k\ , \quad k = 1, 2, \dots , N$$

Programowanie dynamiczne – założenia

- Dotyczy **wieloetapowego problemu decyzyjnego**
- Spełniona jest **własność Markowa**
  Wartość funkcji celu w etapach *k*, ... ,*N* zależy tylko od stanu na początku *k*-tego etapu i od decyzji podjętych w tych etapach
- Funkcja celu **separowalna** (np. addytywna)

$$g(x_0, u_1, u_2, ..., u_N) = g_1(x_0, u_1) \circ ... \circ g_N(x_{N-1}, u_N) \circ g_{N+1}(x_N)$$

Zasada optymalności Bellmana

Jeżeli ciąg decyzji $(u_1^*, u_2^*, ..., u_k^*, ..., u_N^*)$ jest optymalną strategią przejścia od stanu x_0 *do* x_N w procesie N-etapowym, to ciąg decyzji $(u_k^*, ... , u_N^*)$ jest optymalną strategią przejścia od stanu $x_{k\text{-}1^*}$ do stanu x_N

Algorytm Programowania Dynamicznego

- Dla każdego $x_N \in X_N$ przyjmujemy
- $$J_N(x_N) = g_{N+1}(x_N)$$
- Zmniejszając $k = N, \dots, 2, 1$ wyznaczamy dla każdego $x_{k-1} \in X_{k-1}$ takie $u_k \in U_k$, że

$$J_{k-1}(x_{k-1}) = \min \{ g_k (x_{k-1}, \mathrm{u}_k) \circ J_k [f_k (x_{k-1}, u_k)] \}$$

Optymalna wartość funkcji celu:

$$\min\ g(x_0, u_1, u_2, ..., u_N) = J_0(x_0)$$

Optymalne decyzje: u_k^* takie, że

$$J_{k-1}(x_{k-1}) = g_k(x_{k-1}, u_k^*) \circ J_k(x_k) \quad \text{dla } k = 1, 2, \dots, N$$

Przykład 1: Planowanie produkcji (przypadek nieliniowy)

- Produkcja pewnego wyrobu jest planowana z horyzontem 4-tygodniowym.
- Na koniec poszczególnych tygodni należy dostarczyć odpowiednio 1, 4, 2 oraz 2 sztuki wyrobu.
- W każdym tygodniu można wyprodukować 1, 2 lub 3 sztuki tego wyrobu. Koszty produkcji $c(u)$ w zależności od wielkości produkcji u są następujące:

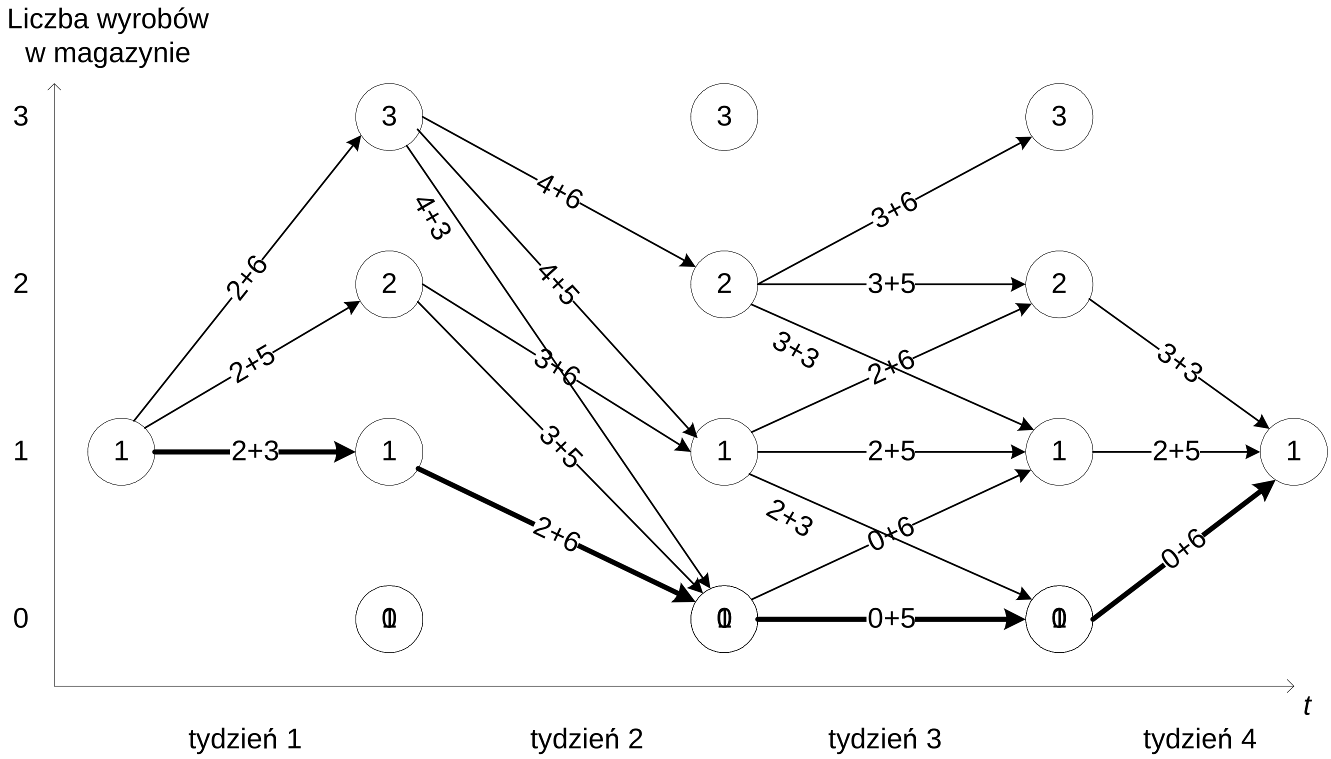
u	1	2	3
$c(u)$	3	5	6

- Wyprodukowane wyroby mogą być przechowywane w magazynie zdolnym pomieścić maksymalnie 3 sztuki wyrobu. Na początku zapas wyrobu jest równy 1 i tyle ma pozostać w magazynie po czterech tygodniach. Koszt $h(x)$ przechowywania x sztuk wyrobu przez każdy kolejny tydzień wynosi:

x	0	1	2	3
$h(x)$	0	2	3	4

- Celem jest określenie takiego planu produkcji wyrobów w poszczególnych tygodniach, aby łączne koszty produkcji i magazynowania były jak najmniejsze.

Przykład 1: Planowanie produkcji



Przykład 2: Binarne zadanie plecakowe

Które spośród 5 przedmiotów posiadających dla nas wartość ocenianą jako 13, 10, 18, 22, 24 i ważących odpowiednio 2, 3, 5, 6 i 7 kg należy zapakować do plecaka o ładowności 11 kg, aby łączna wartość zapakowanych przedmiotów była jak największa?

Przykład 2: Binarne zadanie plecakowe

w_k – waga przedmiotu k,

v_k – wartość przedmiotu k

$J_k(i)$ – maksymalna wartość przedmiotów, które można zapakować do w etapach k, ... , 5 do plecaka o wadze i

Równanie Bellmana (wariant „wstecz")

$J_k(i) = \max(J_{k+1}(i) , J_{k+1}(i+w_k) + v_k)$

$J_k(i)$	1	2	3	4	5	6
0	**45**	34	40	24	24	0
1	41	34	24	24	24	0
2	37	**32**	24	24	24	0
3	35	28	24	24	24	0
4	31	24	24	24	24	0
5	23	22	**22**	**22**	0	0
6	23	18	18	0	0	0
7	13	10	0	0	0	0
8	13	10	0	0	0	0
9	13	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	**0**	**0**

Przykład 2: Binarne zadanie plecakowe

$V_k(i)$ – maksymalna wartość przedmiotów o łącznej wadze nie większej niż i, które można zapakować w etapach 1,…,k

Równanie Bellmana (wariant „w przód")

$V_k(i) = \max(\ V_{k\text{-}1}(i),\ V_{k\text{-}1}(i - w_k) + v_k\ )$

$V_k(i)$	0	1	2	3	4	5
0	**0**	0	0	0	0	0
1	0	0	0	0	0	0
2	0	**13**	13	13	13	13
3	0	13	13	13	13	13
4	0	13	13	13	13	13
5	0	13	**23**	**23**	23	23
6	0	13	23	23	23	23
7	0	13	23	31	31	31
8	0	13	23	31	35	35
9	0	13	23	31	35	37
10	0	13	23	41	41	41
11	0	13	23	41	**45**	**45**