METODY ILOŚCIOWE W BADANIACH EKONOMICZNYCH
Tom XVII/2, 2016, s. 134 – 148

INTERAKTYWNA METODA SATYSFAKCJONUJĄCYCH POZIOMÓW KRYTERIÓW W WIELOKRYTERIALNYM PROGRAMOWANIU DYNAMICZNYM[1]

Tadeusz Trzaskalik
Katedra Badań Operacyjnych, Uniwersytet Ekonomiczny w Katowicach
e-mail: tadeusz.trzaskalik@ue.katowice.pl

Streszczenie: Celem pracy jest zaproponowanie metody pozwalającej na znajdowania rozwiązania końcowego zadania wielokryterialnego dyskretnego programowania dynamicznego z wykorzystaniem odpowiednio zmodyfikowanego podejścia interaktywnego satysfakcjonującego poziomu kryteriów. Procedura w pierwszej fazie wykorzystuje jednokryterialny algorytm programowania dynamicznego oraz algorytm generowania kolejnych realizacji procesu w zadaniu jednokryterialnym. W dalszej części proponowanej metody operujemy na skończonym zbiorze realizacji, zapisanym w postaci listy.

Słowa kluczowe: MCDM, wielokryterialne dyskretne programowanie dynamiczne, metoda interaktywna

WPROWADZENIE

Przedmiotem zainteresowania niniejszej pracy są wieloetapowe, dyskretne procesy decyzyjne, podzielone na skończoną liczbę etapów. Decyzja nie jest podejmowana jednorazowo, lecz wielokrotnie, na początku każdego etapu.

Przedstawimy dynamikę rozpatrywanych procesów.[Trzaskalik 1990, 1998]. Na początku każdego etapów proces znajduje się w jednym ze stanów dopuszczalnych. Decydent podejmuje decyzję, co skutkuje przejściem procesu do stanu początkowego następnego etapu. W rozpatrywanych w pracy procesach deterministycznych przejście to określone jest poprzez funkcję przejścia.

[1] Praca została sfinansowana ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji numer DEC 2013/11/B/HS4/01471.

W zależności od dokonanych wyborów skutki wcześniejszych decyzji ograniczają lub – przeciwnie – rozszerzają możliwości decyzyjne w następnych etapach.

W rozpatrywanych w niniejszej pracy procesach deterministycznych istotną rolę odgrywa pojęcie realizacji procesu. Etapową realizacją procesu jest para, złożona ze stanu procesu na początku rozpatrywanego etapu oraz podjętej decyzji. Realizacją procesu jest ciąg realizacji etapowych, powiązanych ze sobą poprzez funkcje przejścia dla kolejnych etapów.

Rozpatrywać będziemy procesy wielokryterialne. Każda etapowa realizacja procesu oceniana jest przy pomocy kryteriów etapowych. Realizacje wieloetapowe oceniane są z wykorzystaniem addytywnych kryteriów wieloetapowych. Zadanie wielokryterialnego programowania dynamicznego polega na znalezieniu zbioru wszystkich rozwiązań niezdominowanych oraz zbioru wszystkich sprawnych realizacji procesu.

W zadaniach o dużych rozmiarach rozwiązanie powyższego zadania jest trudno, często niewykonalne. Wygenerowanie obszernego zbioru rozwiązań nie jest również praktycznie użyteczne dla decydenta, który oczekuje konkretnego wsparcia zgodnego ze swymi preferencjami, które często nie są w pełni ujawnione w trakcie formułowaniu zadania. Dlatego też, by całościowo uwzględnić preferencje decydenta, wykorzystuje się rożne formy skalaryzacji problemu wektorowej maksymalizacji [Steuer 1986, Galas i in. 1987], jak również podejście interaktywne [Nowak 2008]. Przedmiotem zainteresowania niniejszej pracy jest ta ostatnia możliwość.

Podejście interaktywne jest jednym z często wykorzystywanych sposobów wspomagania decydenta w rozwiązywaniu problemów wielokryterialnych. Jego popularność związana jest z przyjaznością w stosunku do użytkownika oraz przejrzystą ideą. Pozyskiwanie informacji na temat preferencji decydenta odbywa się tu stopniowo, w miarę jak proces rozwiązywania problemu postępuje naprzód. Fazy dialogu z decydentem oraz obliczeniowa są wielokrotnie powtarzane, a decydent każdorazowo ma możliwość zapoznania się z uzyskiwanymi wynikami pośrednimi. W fazie dialogu analityk, wspomagany przez odpowiednio skonstruowany program komputerowy, prosi decydenta o ocenę proponowanego rozwiązania oraz wskazanie kierunków jego poprawy, a tym samym o dalsze ujawnienie swoich preferencji. Proces ten jest zwykle kontynuowany do momentu uzyskania rozwiązania uznawanego przez decydenta za satysfakcjonujące [Nowak 2008].

Jedną z najwcześniejszych procedur interaktywnych jest interaktywne programowanie celowe [Spronk 1981]. Procedura ta pozwala w uporządkowany sposób modelować poziomy aspiracji decydenta, mając na celu precyzyjne określenie jego dodatkowych preferencji względem rozpatrywanych kryteriów. Punktem wyjścia są wartości optymalne, uzyskane w wyniku rozwiązanie zadań jednokryterialnych, czyli współrzędne wektora idealnego w przestrzeni kryterialnej. W kolejnych krokach dołączane są dodatkowe warunki ograniczające określające poziomy kryteriów akceptowane przez decydenta w kolejnych iteracjach.

Proponowana dalej procedura polega na rozwiązaniu ciągu jednokryterialnych zadań programowania dynamicznego [Bellman 1957, Bellman, Dryfus 1963, Trzaskalik 1990]. Dołączenie dodatkowych ograniczeń powoduje, że konieczne staje się wykorzystanie procedury generowania prawie optymalnych realizacji procesu [Elmaghraby 1970, Trzaskalik 2015].

Celem niniejszej pracy jest prezentacja interaktywnej metody satysfakcjonujących poziomów kryteriów dla programowania dynamicznego. Praca składa się z pięciu części. Po wprowadzeniu, w rozdziale - dyskretne programowanie dynamiczne - omówione zostały wybrane zagadnienie z zakresu jedno i wielokryterialnego programowania dynamicznego. Modelowanie poziomów aspiracji decydenta oraz metoda satysfakcjonujących poziomów kryteriów dla zadania wielokryterialnego dyskretnego programowania dynamicznego przedstawione zostały w rozdziale – modelowanie poziomów aspiracji decydenta. Kolejny rozdział to ilustracja proponowanej metody. Posumowanie pracy znajduje się w ostatnim rozdziale.

DYSKRETNE PROGRAMOWANIE DYNAMICZNE

Oznaczenia

Wykorzystamy następującą notację [Trzaskalik 1990, 1998]:

T – liczba rozpatrywanych etapów procesu wieloetapowego

y_t –stan procesu na początku etapu t (t=1,…,T),

$\mathbf{Y}_t$ – skończony zbiór stanów procesu na początku etapu t,

$\mathbf{Y}_{T+1}$ – skończony zbiór stanów procesu na końcu procesu,

x_t – stan dopuszczalny na początku etapu t,

$\mathbf{X}_t(y_t)$ – skończony zbiór stanów dopuszczalnych w etapie t, gdy proces był w stanie $y_t \in \mathbf{Y}_t$ na początku tego etapu,

d_t – realizacja etapowa w etapie t. Mamy:

$$d_t = (y_t, x_t) \tag{1}$$

$\mathbf{D}_t$ – zbiór realizacji etapowych w etapie t,

$\Omega_t(y_t, x_t)$ – funkcja przejścia. Mamy:

$$y_{t+1} = \Omega_t(y_t, x_t) \tag{2}$$

d – realizacja procesu. Mamy:

$$d = ((y_1, x_1), (y_2, x_2), ..., (y_T, x_T)) \tag{3}$$

gdzie :

$$\begin{array}{ll} y_1 \in \mathbf{Y}_1, & x_1 \in \mathbf{X}_1(y_1) \\ y_2 = \Omega_1(y_1, x_1) & x_2 \in \mathbf{X}_2(y_2) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ y_T = \Omega_{T-1}(y_{T-1}, x_{T-1}) & x_T \in \mathbf{X}_T(y_T) \\ y_{T+1} = \Omega_t(y_T, x_T) & \end{array}$$

$\mathbf{D}$ – zbiór wszystkich realizacji procesu,

$d_{\overline{t,T}}(y_t)$ – realizacja skrócona, rozpoczynająca się w stanie y_t i obejmująca etapy od t do T. Mamy:

$$d_{\overline{t,T}}(y_t) = [(y_t, x_t), (y_{t+1}, x_{t+1}), \ldots, (y_T, x_T)]. \qquad (4)$$

$\mathbf{D}_{\overline{t,T}}(y_t)$ – zbiór realizacji skróconych, rozpoczynających się w stanie y i obejmujących stany od t do T,

$F_t(d_t)$ – etapowa funkcja kryterialna,

$F(d)$ – funkcja kryterium, oceniająca realizację procesu d. Mamy:

$$F(d) = \sum_{t=1}^{T} F(d_t). \qquad (5)$$

Równania optymalności

Naszym celem jest znalezienie maksymalnej wartości funkcji kryterium oraz tych realizacji procesu, które pozwalają na jej osiągnięcie. W tym celu wykorzystamy dekompozycję zadania, zasadę optymalności Bellmana oraz wynikające z niej równania optymalności [Bellman 1957, Bellman, Dreyfus 1964].

Algorytm 1

1. Dla każdego $y_T \in \mathbf{Y}_T$ obliczamy wartość optymalną

$$G_T(y_T) = \max_{x_T \in \mathbf{X}_T(y_T)} F_T(y_T, x_T) \qquad (6)$$

   i znajdujemy decyzję optymalną dla tego stanu, którą oznaczamy jako $x_T^*(y_T)$.
2. Dla każdego $y_t \in \mathbf{Y}_t$ obliczamy wartość optymalną

$$G_t(y_t) = \max_{x_t \in \mathbf{X}_t(y_t)} \{F_t(y_t, x_t) + G_{t+1}(\Omega_t(y_t, x_t))\} \qquad (7)$$

   i znajdujemy decyzję optymalną dla tego stanu, którą oznaczamy jako $x_t^*(y_t)$.
3. Optymalną wartość funkcji kryterium obliczamy ze wzoru:

$$G^* = \max_{y_1 \in \mathbf{Y}_1} G_1(y_1) \qquad (8)$$

Optymalny stan początkowy y_1^* znajdujemy z zależności:

$$F(y_1^*) = G^* \qquad (9)$$

4. Optymalną realizację procesu otrzymujemy następująco:

y_1^* - optymalny stan początkowy	$x_1^* = x_1^*(y_1^*)$
$y_2^* = \Omega_1(y_1^*, x_1^*)$	$x_2 = x_2^*(y_t)$
………………….	
$y_T^* = \Omega_{T-1}(y_{T-1}^*, x_{T-1}^*)$	$x_T^* = x_2^*(y_t^*)$
$y_{T+1}^* = \Omega_t(y_T, x_T)$	

Realizacje prawie optymalne

Skończony zbiór **D** realizacji procesu może być podzielony na N klas w taki sposób, że

$$\mathbf{D} = \mathbf{D}^1 \cup \mathbf{D}^2 \cup \ldots \mathbf{D}^N \tag{10}$$

gdzie:

$$\mathbf{D}^i \cap \mathbf{D}^j = \emptyset \text{ dla } i \neq j,\ i,j = 1,\ldots,N, \tag{11}$$

$$\forall_{i=1,\ldots,N} \forall_{d^j, d^k \in \mathbf{D}^i} F(d^j) = F(d^k), \tag{12}$$

$$\forall_{i<j} \forall_{d^i \in \mathbf{D}^i} \forall_{d^j \in \mathbf{D}^j} F(d^i) > F(d^j). \tag{13}$$

Wartość $F(d^i)$ dla $i=1,\ldots,N$ nazywamy i-tą wartością procesu. Każda realizacja ze zbioru $\mathbf{D}^i$ nazywana jest i-tą realizacją procesu. Przy poszukiwaniu i-tej wartości procesu wykorzystamy następującą notację:

$$\max\nolimits_i F(\mathbf{D}) = F(d^i) \tag{14}$$

Sposób określenia i-tej realizacji procesu opisuje algorytm 2 [Trzaskalik 2015].

Algorytm 2

1. Rozpoczynając od i=1 dla każdego $y_T \in \mathbf{Y}_T$ obliczamy i-tą wartość

$$G_T^i(y_T) = \max_{x_T \in \mathbf{X}_T(y_T)}{}_i\ F_T(y_T, x_T) \tag{15}$$

   i znajdujemy zbiór realizacji skróconych procesu $\mathbf{D}_{\overline{T,T}}(y_T)$, dla których wartość ta została osiągnięta.

2. Rozpoczynając od i = 1 dla etapu t, $t \in \overline{T-1,1}$ i każdego stanu $y_t \in \mathbf{Y}_t$ obliczamy i-tą wartość

$$G_t^i(y_t) = \max_{x_t \in \mathbf{X}(y_t)}{}_i \left\{F_t(y_t, x_t) + G_{t+1}^j(\Omega_t(y_t, x_t)) : j = 1,\ldots,i\right\} \tag{16}$$

   oraz znajdujemy zbiór realizacji skróconych procesu $\mathbf{D}_t(y_t)$, dla których wartość ta została osiągnięta.

3. i-ta wartość procesu obliczona jest ze wzoru:

$$G^i = \max_{y_1 \in \mathbf{Y}_1}{}_i \left\{G_1^j(y_1) : j = 1,\ldots,i\right\} \ . \tag{17}$$

4. Zbiór wszystkich i-tych realizacji procesu obliczamy ze związku:

$$\mathrm{D}^i = \bigcup_{y_1 \in \mathrm{Y}_1} \left\{\mathrm{D}^j(y_1) : G^j(y_1) = G^i,\ j = 1,\ldots,i\right\}. \tag{18}$$

Realizację d^o nazywamy realizacją prawie optymalną, jeżeli jej wartość kryterialna różni się od wartości kryterialnej realizacji optymalnej d^* co najwyżej o zadaną wartość α, czyli

$$F(d^*) - F(d^o) \le \alpha. \tag{19}$$

Oznacza to, że decydenta interesują realizacje, dla których $F(d^o) \ge z$, przy czym

$$z = F(d^*) - \alpha \tag{20}$$

Przypuścimy, że decydent zainteresowany jest znalezieniem realizacji prawie optymalnych i określił wartość α. Wykorzystując algorytm 2 znajdujemy realizacje prawie optymalne następująco:

Algorytm 3

1. Start.
2. Podstaw i=1.
3. Wykorzystując algorytm 2, znajdź i-tą realizacje procesu d^i.
4. Sprawdź, czy $F(d^i) \geq z$. Jeżeli tak, podstaw $i=i+1$ i wróć do kroku 2.
5. Koniec procedury.

Wielokryterialne programowanie dynamiczne

Wykorzystamy dodatkowe oznaczenia:
K – liczba rozpatrywanych kryteriów,
$F_t^k(d_t)$ – k-te kryterium etapowe w etapie t (k=1,…,K),
$F^k(d)$ – k-te kryterium wieloetapowe,
$F = [F^1,\ldots, F^K]$ – wektorowa funkcja kryterium,
$\geq$ **-** relacja dominacji, zdefiniowana następująco**:** realizacja d^o dominuje realizację d, jeżeli spełniony jest warunek:

$$\forall_{k=1,\ldots,K}\, F^k(d^o) \geq F^k(d) \wedge \exists_{\,i=1,\ldots,K}\, F^i(d^o) > F^i(d), \tag{21}$$

d^* - realizacja sprawna, zdefiniowana następująco:

$$\sim\exists_{\,d\in D}\, F(d) \geq F(d^*), \tag{22}$$

$\mathbf{D}^*$ - zbiór wszystkich realizacji sprawnych.

Rozwiązanie problemu wektorowej optymalizacji dynamicznej polega na znalezieniu zbioru wszystkich realizacji sprawnych $\mathbf{D}^*$ oraz zbioru $F(\mathbf{D}^*)$ wektorów niezdominowanych w przestrzeni ocen. Rozwiązanie tego problemu możemy uzyskać stosując wektorową zasadę optymalności Bellmana [Brown i Strauch 1965, Trzaskalik 1990].

MODELOWANIE POZIOMÓW ASPIRACJI DECYDENTA

Metoda satysfakcjonujących poziomów kryteriów

Jednym ze sposobów generowania realizacji sprawnych w zadaniach wielokryterialnych jest wykorzystanie metody satysfakcjonujących decydenta poziomów kryteriów (*ε- constraint method*). Aby wykorzystać to podejście w zadaniach dyskretnego programowania dynamicznego, tworzymy ciąg zadań jednokryterialnych:

$$\text{Max}\,\{F^k(d):\, d\in\mathbf{D}\}. \tag{23}$$

Oznaczymy symbolem F^{*k} wartość optymalną funkcji celu k-tego zadania w postaci (23). Wektor $F^* = [F^{*1}, \ldots, F^{*K}]$ jest wektorem idealnym w przestrzeni kryterialnej. W realnych problemach decyzyjnych zazwyczaj nie istnieje dopuszczalna realizacja procesu d^* taka, że $F(d^*) = F^*$. Wektor idealny może jednak służyć jako punkt odniesienia w modelowaniu preferencji decydenta względem jego oczekiwań w stosunku do wartości kryteriów wieloetapowych, które można uzyskać.

Przypuśćmy, że decydent ustalił satysfakcjonujące go poziomy osiągnięcia celów wieloetapowych, opisane wektorem $\mathbf{z} = [z^1,...,z^K]$, którego składowe określają poziomy aspiracji decydenta względem kolejnych kryteriów. Dokonując wyboru wartości satysfakcjonujących decydent musi uwzględnić to, że dla każdego k ($k=1,..,K$)

$$z^k \leq F^{*k}. \tag{24}$$

Podając wartości satysfakcjonujące decydent zakłada, że zajmuje się tylko takimi realizacjami procesu, dla których zachodzi warunek:

$$F^k(d) \geq z^k. \tag{25}$$

Możemy sprawdzić, czy istnieje możliwość zrealizowania tych oczekiwań. Jeżeli istnieje, możemy zapytać, czy istniałaby możliwość poprawy wybranego przez decydenta kryterium o numerze i ($i=1,...,K$). Tworzymy zadanie:

$$\{\text{Max } F^i(d): d \in \mathbf{D}, F^k(d) \geq z^k: k=1,...,i-1, i+1,...,K\}. \tag{26}$$

Twierdzenie 1 [Galas i in. 1987, Trzaskalik 1990]

Załóżmy, że wektor $\mathbf{z}$ określony został w taki sposób, że każde zadanie, określone wzorem (26) ma rozwiązanie[2]. Jeżeli istnieje dokładnie jedno rozwiązanie zadania (26), to jest to realizacja sprawna, spełniająca warunek (22), czyli element zbioru $\mathbf{D}^*$. W przypadku, gdy rozwiązań jest więcej, z otrzymanego zbioru rozwiązań eliminujemy realizacje zdominowane. Pozostałe realizacje są realizacjami sprawnymi zadania (26).

Interaktywna metoda satysfakcjonujących poziomów kryteriów

Opiszemy teraz możliwość interaktywnego modelowania poziomów aspiracji decydenta. Poniżej opisane zostały kolejne kroki proponowanej procedury.

Algorytm 4

1. Przyjmujemy $m := 1$, $\mathbf{D}(m) := \mathbf{D}$ oraz wektor poziomów dokładności
$$\beta = [\beta^1,\ldots, \beta^K].$$
2. Dla $k = 1,\ldots, K$ rozwiązujemy ciąg zadań w postaci:
$$\text{Max } \{F^k(d): d \in \mathbf{D}(m)\}.$$
W pierwszej iteracji (dla $m = 1$) wykorzystujemy algorytm 1, w kolejnych iteracjach - dokonujemy przeglądu list realizacji wygenerowanych uprzednio.
3. Otrzymane w poprzednim kroku wartości optymalne oznaczamy odpowiednio jako $F^{1*}(m),\ldots,F^{K*}(m)$. Zbiór realizacji optymalnych otrzymanych w wyniku rozwiązania zadania o numerze k w iteracji m oznaczamy jako $\mathbf{D}^{k*}(m)$.

---

[2] Może się okazać, że zbiór rozwiązań dopuszczalnych zadania (26) jest pusty. Wtedy należałoby zalecić decydentowi obniżenie poziomów aspiracji.

4. Tworzymy zbiór:

$$\mathbf{D}^{*}(m) = \bigcup_{k=1}^{K} \mathbf{D}^{k*}(m) \tag{27}$$

zawierający $l(m)$ elementów.

5. Tworzymy macierz wartości kryteriów $\mathbf{W}(m)$, której elementy $w^{ki}(m)$ dla $i = 1,\ldots,l(m)$, $k=1,\ldots,K$ określone są następująco:

$$w^{ki}(m) = F^{k}(d^{i*}(m)),$$

przy czym $d^{i*}(m) \in \mathbf{D}^{*}(m)$.

6. Dla $k=1,\ldots,K$ znajdujemy wartości

$$z^{k}(m) = \min \{F^{k}(d^{i*}(m): d^{i*}(m)) \in \mathbf{D}^{*}(m)\}. \tag{28}$$

7. Tworzymy macierz możliwości $\mathbf{M}(m)$ o wymiarach $2\times K$. Jej strukturę opisuje tabela 1.

Tabela 1. Macierz możliwości

Wartości	F^1	…	F^k	…	F^K
Akceptowane	$z^1(m)$	…	$z^k(m)$	…	$z^K(m)$
Optymistyczne	$F^{1*}(m)$	…	$F^{k*}(m)$	…	$F^{K*}(m)$

Źródło: opracowanie własne

Pierwszy wiersz tej macierzy nazywany jest dalej wektorem ocen akceptowanych w iteracji m, drugi – wektorem ocen optymistycznych w iteracji m.

8. Jeżeli dla każdego $k= 1,\ldots,K$ zachodzi związek:

$$\mathrm{F}^{k*}(\mathrm{m}) - \mathrm{z}^{k}(\mathrm{m}) < \beta^{k} \tag{29}$$

wówczas kończymy procedurę. Jako rozwiązanie końcowe decydent wybiera realizację ze zbioru $\mathbf{D}^{*}(m)$.

9. Na podstawie oceny macierzy możliwości decydent:
   a. wybiera jedno lub więcej kryteriów, względem których podwyższa poziomy aspiracji, określając aspiracje dla pozostałych kryteriów na poziomie akceptowanym w iteracji m. Zbiór $\mathbf{P}$ zawiera wszystkie te kryteria, względem których nastąpiło podwyższenie poziomu aspiracji. Nowe poziomy aspiracji oznaczamy jako z^{p+} dla $p\in\mathbf{P}$. Przyjmujemy, że dla $p\in\mathbf{P}$ mamy

$$z^{p+}(m) > z^{p}(m). \tag{30}$$

   Przechodzimy do kroku 10.

   b. rezygnuje z możliwości podniesienia poziomu aspiracji względem jednego kryterium lub grupy kryteriów i wybiera rozwiązanie końcowe spośród rozwiązań $\mathbf{D}^{*}(m)$ rozpatrywanych w iteracji m uznając, że uzyskane poziomy kryteriów w wybranym rozwiązaniu są dla niego satysfakcjonujące. Przechodzimy do kroku 14.
   c. odstępuje od kontynuowania procedury i decyduje się rozpocząć ją od

początku lub wybrać rozwiązanie końcowe w inny sposób. Przechodzimy do kroku 14.

10. Dla $k = 1,\ldots,K$ określamy zbiory realizacji procesu, spełniające warunki:

    dla kryteriów numerach $p \in \mathbf{P}$

$$\mathbf{D}^p(m+1) = \{d \in \mathbf{D}(m): F^p(d) \geq z^{p+}\}, \tag{31}$$

    dla pozostałych kryteriów $i \notin \mathbf{P}$

$$\mathbf{D}^i(m+1) = \{d \in \mathbf{D}(m): F^i(d) \geq z^i(m)\}. \tag{32}$$

    Dla $m = 1$ wykorzystujemy w tym celu algorytm 3, w kolejnych iteracjach (dla $m>1$} dokonujemy przeglądu list realizacji wygenerowanych uprzednio.

11. Tworzymy zbiór

$$\mathbf{D}(m+1) = \bigcup_{k=1}^{K} \mathbf{D}^k(m+1) \tag{33}$$

12. Jeżeli D(m+1) = ∅, informujemy o tym decydenta. Wracamy do punktu 9, zalecając mu obniżenie poziomów aspiracji w stosunku do tych wartości, które zostały określone uprzednio.
13. Przyjmujemy m := m+1 i przechodzimy do punktu 2.
14. Koniec procedury.

ILUSTRACJA NUMERYCZNA

Rozpatrujemy dwuetapowy proces decyzyjny. Zbiory stanów na początku kolejnych etapów są następujące:

$\mathbf{Y}_1 = \{1, 2, 3\}$, $\mathbf{Y}_2 = \{4, 5, 6\}$.

Zbiór stanów końcowych procesu ma postać:

$\mathbf{Y}_3 = \{7, 8, 9\}$.

Zbiory decyzji dopuszczalnych są następujące:

$\mathbf{X}_1(1) = \{A, B, C\}$, $\mathbf{X}_1(2) = \{D, E, F\}$, $\mathbf{X}_2(3) = \{G, H, I\}$,
$\mathbf{X}_2(4) = \{J, K, L\}$, $\mathbf{X}_3(5) = \{M, N, O\}$, $\mathbf{X}_3\{6\} = \{P, R, S\}$.

Mamy następujące wartości funkcji przejścia (tabela 2):

Tabela 2. Wartości funkcji przejścia

Etap 1		
$\Omega_1(1, A) = 4$	$\Omega_1(1, B) = 5$	$\Omega_1(1, C) = 6$
$\Omega_1(2, D) = 4$	$\Omega_1(2, E) = 5$	$\Omega_1(2, F) = 6$
$\Omega_1(3, G) = 4$	$\Omega_1(3, H) = 5$	$\Omega_1(3, I) = 6$

Etap 2		
$\Omega_2(4, J) = 7$	$\Omega_2(4, K) = 8$	$\Omega_2(4, L) = 9$
$\Omega_2(5, M) = 7$	$\Omega_2(5, N) = 8$	$\Omega_2(5, O) = 9$
$\Omega_2(6, Q) = 7$	$\Omega_2(6, R) = 8$	$\Omega_2(6, R) = 9$

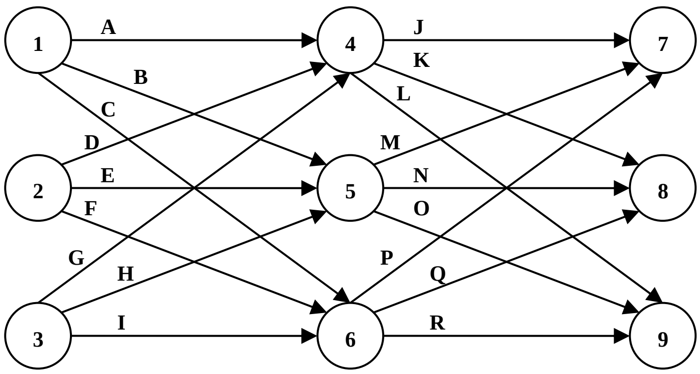
Źródło: obliczenia własne

Graf procesu przedstawiony jest na rysunku 1.

Ze względu na niewielkie rozmiary tego ilustracyjnego zadania, wszystkie realizacje procesu możemy dla lepszej przejrzystości wypisać i ponumerować od 1 do 27. Numerację tę przedstawia tabela 3.

Rozpatrywane zadanie jest zadaniem trzykryterialnym. Przyjmujemy, że wszystkie kryteria są maksymalizowane. Wartości kryteriów dla kolejnych realizacji przedstawione są w tabeli 4.

Rysunek 1. Graf procesu



Źródło: opracowanie własne

Tabela 3. Lista realizacji procesu

Nr	Realizacja	Nr	Realizacja	Nr	Realizacja
1	(1, A, 4, J)	10	(2, D, 4, J)	19	(3, G, 4, J)
2	(1, A, 4, K)	11	(2, D, 4, K)	20	(3, G, 4, K)
3	(1, A, 4, L)	12	(2, D, 4, L)	21	(3, G, 4, L)
4	(1, B, 5, M)	13	(2, E, 5, M)	22	(3, H, 5, M)
5	(1, B, 5, N)	14	(2, E, 5, N)	23	(3, H, 5, N)
6	(1, B, 5, O)	15	(2, E, 5, O)	24	(3, H, 5, O)
7	(1, C, 6, P)	16	(2, F, 6, P)	25	(3, I, 6, P)
8	(1, C, 6, Q)	17	(2, F, 6, Q)	26	(3, I, 6, Q)
9	(1, C, 6, R)	18	(2, F, 6, R)	27	(3, I, 6, R)

Źródło: obliczenia własne

Tabela 4. Wartości kryteriów wieloetapowych

d^i	$F^1()$	$F^2()$	$F^3()$	d^i	$F^1()$	$F^2()$	$F^3()$	d^i	$F^1()$	$F^2()$	$F^3()$
d^1	11	80	237	d^{10}	11	79	249	d^{19}	12	81	259
d^2	14	83	235	d^{11}	14	82	247	d^{20}	15	84	257
d^3	12	77	241	d^{12}	12	76	253	d^{21}	13	78	263
d^4	13	77	230	d^{13}	11	84	235	d^{22}	12	78	249
d^5	14	76	229	d^{14}	12	83	234	d^{23}	13	77	248
d^6	16	80	233	d^{15}	14	87	238	d^{24}	15	81	252
d^7	13	81	249	d^{16}	12	72	245	d^{25}	14	80	260
d^8	18	84	247	d^{17}	17	75	243	d^{26}	19	83	258
d^9	17	84	244	d^{18}	16	75	240	d^{27}	18	83	255

Źródło: obliczenia własne

Opiszemy kolejne kroki algorytmu.

Iteracja 1

1. Przyjmujemy $m:=1$, $\mathbf{D}(1) = \mathbf{D}$ oraz $\beta_1 = \beta_2 = \beta_3 = 1$.
2. Stosując algorytm 1 dla k = 1, 2, 3 rozwiązujemy zadania:

$$\text{Max } \{F^1(d) \colon d \in \mathbf{D}(1),$$
$$\text{Max } \{F^2(d) \colon d \in \mathbf{D}(1),$$
$$\text{Max } \{F^3(d) \colon d \in \mathbf{D}(1).$$

3. Mamy:

$$F^{1*}(1) = 19, \qquad F^{2*}(1) = 87, \qquad F^{3*}(1) = 260,$$
$$\mathbf{D}^{1*}(1) = \{d^{26}\}, \quad \mathbf{D}^{2*}(1) = \{d^{15}\}, \quad \mathbf{D}^{3*}(1) = \{d^{25}\}.$$

4. Tworzymy zbiór

$$\mathbf{D}^*(1) = \mathbf{D}^{1*}(1) \cup \mathbf{D}^{2*}(1) \cup \mathbf{D}^{3*}(1) = \{d^{26}, d^{15}, d^{25}\}$$

5. Tworzymy macierz wartości kryteriów $\mathbf{W}(1)$:

$$\begin{bmatrix} F^1(d^{26}) & F^2(d^{26}) & F^3(d^{26}) \\ F^1(d^{15}) & F^2(d^{15}) & F^3(d^{15}) \\ F^1(d^{25}) & F^2(d^{25}) & F^3(d^{25}) \end{bmatrix} = \begin{bmatrix} 19 & 83 & 258 \\ 14 & 87 & 238 \\ 14 & 80 & 260 \end{bmatrix}$$

6. Znajdujemy wartości:

$$z^1(1) = \min\{19, 14, 14\} = 14, \quad z^2(1) = \min\{83, 87, 80\},$$
$$z^3(1) = \min\{258, 238, 260\}.$$

7. Tworzymy macierz $\mathbf{W}(1)$:

Wartości	F^1	F^2	F^3
Akceptowane	14	80	238
Optymistyczne	19	87	260

8. Mamy:

$$F^1{*}(1) - z^1(1) = 19 - 14 > 1, \qquad F^2{*}(1) - z^2(1) = 87 - 80 > 1,$$

$$F^{3*}(1) - z^3(1) = 260 - 238 > 1.$$

Postępowanie jest kontynuowane.

9. Decydent ocenia macierz możliwości **M**(1) i wybiera kryterium F^1, dla którego podwyższa poziom aspiracji do poziomu 15, określając poziomy aspiracji dla pozostałych kryteriów na poziomie akceptowanym w iteracji 1. Mamy **P** = {1}. Nowe poziomy aspiracji spełniają warunki (30).
10. Korzystając z algorytmu generowania realizacji prawie optymalnych (Algorytm 3) dla k =1, 2, 3 określamy zbiory realizacji procesu, spełniające warunki (31) i (32):

$$\mathbf{D}^1(2) = \{d\in\mathbf{D}(1);\ F^1(d) \geq 15\} = \{d^6, d^8, d^9, d^{17}, d^{18}, d^{20}, d^{24}, d^{25}, d^{26}, d^{27}\},$$

$$\mathbf{D}^2(2) = \{d\in\mathbf{D}(1);\ F^2(d) \geq 80\} = \{d^1, d^2, d^6, d^7, d^8, d^9, d^{11}, d^{13}, d^{14}, d^{15}, d^{19}, d^{20}, d^{21}, d^{22}, d^{24}, d^{25}, d^{26}, d^{27}\},$$

$$\mathbf{D}^3(2) = \{d\in\mathbf{D}(1);\ F^3(d) \geq 238\} = \{d^3, d^7, d^8, d^9, d^{10}, d^{11}, d^{12}, d^{15}, d^{16}, d^{17}, d^{18}, d^{19}, d^{20}, d^{21}, d^{22}, d^{23}, d^{24}, d^{25}, d^{26}, d^{27}\}.$$

11. Tworzymy zbiór

$$\mathbf{D}(2) = \mathbf{D}^1(2) \cap \mathbf{D}^2(2) \cap \mathbf{D}^3(2) = \{d^8, d^9, d^{20}, d^{24}, d^{26}, d^{27}\}.$$

12. Mamy $\mathbf{D}(2) \neq \varnothing$.
13. Przyjmujemy m =: m + 1 = 2.

<u>Iteracja 2</u>

2. Przeglądając odpowiednie listy realizacji, rozwiązujemy zadania:

$$\text{Max}\ \{F^1(d)\text{: } d\in\mathbf{D}(2)\},$$
$$\text{Max}\ \{F^2(d)\text{: } d\in\mathbf{D}(2)\},$$
$$\text{Max}\ \{F^3(d)\text{: } d\in\mathbf{D}(2)\}.$$

3. Mamy:

$$F^{1*}(2) = 19, \quad F^{2*}(2) = 84, \quad F^{3*}(2) = 258,$$

$$\mathbf{D}^{1*}(2) = \{\mathrm{d}^{26}\}, \quad \mathbf{D}^{2*}(2) = \{\mathrm{d}^8, \mathrm{d}^9, \mathrm{d}^{20}\}, \quad \mathbf{D}^{3*}(2) = \{\mathrm{d}^{26}\}$$

4. Tworzymy zbiór

$$\mathbf{D}^*(2) = \mathbf{D}^{1*}(2) \cup \mathbf{D}^{2*}(2) \cup \mathbf{D}^{3*}(2) = \{\mathrm{d}^{26}, \mathrm{d}^8, \mathrm{d}^9, \mathrm{d}^{20}\}$$

5. Tworzymy macierz wartości kryteriów **W**(2). Jest ona następująca:

$$\begin{bmatrix} F^1(d^{26}) & F^2(d^{26}) & F^3(d^{26}) \\ F^1(d^8) & F^2(d^8) & F^3(d^8) \\ F^1(d^9) & F^2(d^9) & F^3(d^9) \\ F^1(d^{20}) & F^2(d^{20}) & F^3(d^{20}) \end{bmatrix} = \begin{bmatrix} 19 & 83 & 258 \\ 18 & 84 & 247 \\ 17 & 84 & 244 \\ 15 & 84 & 257 \end{bmatrix}$$

6. Znajdujemy wartości:

$$z^1(2) = \min\ \{19, 18, 17, 15\} = 15, \quad z^2(2) = \min\ \{83, 84, 84, 84\} = 83,$$

$$z^3(2) = \min\ \{258, 247, 244, 257\} = 244.$$

7. Tworzymy macierz **W**(2):

Wartości	F^1	F^2	F^3
Akceptowane	15	83	244
Optymistyczne	19	84	258

8. Mamy:

$$F^{1*}(2) - z^1(2) = 19 - 15 > 1,` \qquad F^{2*}(2) - z^2(2) = 84 - 83 \le 1,$$
$$F^{3*}(2) - z^3(2) = 258 - 244 > 1.$$

Postępowanie jest kontynuowane.

9. Decydent ocenia macierz możliwości **M**(2) i wybiera kryterium F^3, dla którego podwyższa poziom aspiracji do poziomu 255, określając poziomy aspiracji dla pozostałych kryteriów na poziomie akceptowanym w iteracji 2. Mamy **P** = {3}. Nowe poziomy aspiracji spełniają warunki (30).
10. Dokonując przeglądu listy określamy zbiory realizacji procesu, spełniające warunki (31) i (32) otrzymujemy:

$$\mathbf{D}^1(3) = \{d \in \mathbf{D}(2);\, F^1(d) \ge 15\} = \{d^8, d^9, d^{20,} d^{24}, d^{26}, d^{27}\},$$
$$\mathbf{D}^2(3) = \{d \in \mathbf{D}(2);\, F^2(d) \ge 83\} = \{d^8, d^9, d^{20}, d^{26}, d^{27}\},$$
$$\mathbf{D}^3(3) = \{d \in \mathbf{D}(2);\, F^3(d) \ge 255\} = \{d^{20}, d^{26}, d^{27}\}.$$

11. Tworzymy zbiór

$$\mathbf{D}(3) = \mathbf{D}^1(2) \cap \mathbf{D}^2(2) \cap \mathbf{D}^3(2) = \{d^{20}, d^{26}, d^{27}\}$$

12. Mamy $\mathbf{D}(3) \ne \emptyset$.
13. Przyjmujemy $m =: m + 1 = 3$.

<u>Iteracja 3</u>

2. Rozwiązujemy ciąg zadań

$$\text{Max } \{F^1(d): d \in \mathbf{D}(3)\},$$
$$\text{Max } \{F^2(d): d \in \mathbf{D}(3)\},$$
$$\text{Max } \{F^3(d): d \in \mathbf{D}(3)\}.$$

3. Mamy:

$$F^{1*}(3) = 19, \qquad F^{2*}(3) = 84, \qquad F^{3*}(3) = 258,$$
$$\mathbf{D}^{1*}(3) = \{d^{26}\}, \quad \mathbf{D}^{2*}(3) = \{d^{20}\}, \quad \mathbf{D}^{3*}(3) = \{d^{26}\}.$$

4. Tworzymy zbiór

$$\mathbf{D}^*(3) = \mathbf{D}^{1*}(3) \cup \mathbf{D}^{2*}(3) \cup \mathbf{D}^{3*}(3) = \{\{d^{26}\}, \{d^{20}\}\}.$$

5. Tworzymy macierz wartości kryteriów **W**(3):

$$\begin{bmatrix} F^1(d^{26}) & F^2(d^{26}) & F^3(d^{26}) \\ F^1(d^{20}) & F^2(d^{20}) & F^3(d^{20}) \end{bmatrix} = \begin{bmatrix} 19 & 83 & 258 \\ 15 & 84 & 257 \end{bmatrix}$$

6. Znajdujemy wartości:

$$z^1(3) = \min\{19, 15\} = 15, \qquad z^2(3) = \min\{83, 84\} = 83,$$
$$z^3(3) = \min\{258, 257\} = 257.$$

7. Tworzymy macierz **W**(3):

Wartości	F^1	F^2	F^3
Akceptowane	15	83	258
Optymistyczne	19	84	257

8. Mamy:

$$F^1*(1) - z^1(1) = 19 - 15 > 1, \qquad F^2*(1) - z^2(1) = 84 - 83 \leq 1,$$
$$F^3*(1) - z^3(1) = 258 - 257 \leq 1.$$

9b. Decydent rezygnuje z możliwości podniesienia poziomu aspiracji i wybiera rozwiązanie końcowe spośród rozwiązań $\mathbf{D}^*(3)$ rozpatrywanych w iteracji 3 (wykorzystując dodatkowe informacje, na przykład o wartościach kryteriów etapowych w poszczególnych rozwiązaniach).

14. Koniec procedury.

ZAKOŃCZENIE

Przedstawione w pracy podejście pozwala na efektywne włączenie decydenta w proces dochodzenia do rozwiązania końcowego. Należy podkreślić, że ważnym elementem proponowanego podejścia jest wykorzystanie możliwości generowania rozwiązań prawie optymalnych w zadaniach jednokryterialnych, istotnie wykorzystywanej w tej procedurze.

Zakres praktycznych zastosowań proponowanego algorytmu obejmuje wszystkie te problemy, w których występuje sekwencja powiązanych ze sobą decyzji. Z sytuacją taką mamy na przykład do czynienia w zarządzaniu portfelem projektów czy też w problemach z zakresu zarządzaniem zdolnością produkcyjną. W problemach tych decydenci mogą być zainteresowani poszukiwaniem takiego rozwiązania, które przykładowo oceniane jest nieco gorzej z punktu widzenia finansowego, jest jednak zdecydowanie bardziej atrakcyjne z punktu widzenia innych, ważnych celów organizacji.

Celem dalszych prac będzie rozwijanie podejścia interaktywnego, zarówno dla zadań deterministycznych, jak również stochastycznych i rozmytych. W rozważaniach tych powinno się uwzględnić strukturę etapową rozpatrywanych realizacji, czyli wartości kryteriów etapowych. Ich uwzględnienie powinno w istotny sposób wzbogacić prowadzoną analizę.

BIBLIOGRAFIA

Brown T. A., Strauch R. E. (1965) Dynamic Programming in Multiplicative Lattices. Journal of Mathematical Analysis and Applications, 12.

Bellman R. (1957) Dynamic Programming. Princeton University Press.

Bellman R., Dreyfus S. (1967) Programowania dynamiczne. Zastosowania. PWE, Warszawa.

Elmaghraby S. E. (1970) The Theory of Networks and Management Science. Management Science, 1, 17.

Galas Z., Nykowski I., Żółkiewski Z. (1987) Programowanie wielokryterialne. PWE, Warszawa.

Nowak M. (2008) Interaktywne wielokryterialne wspomaganie decyzji w warunkach ryzyka: metody i zastosowania. Wydawnictwo Akademii Ekonomicznej w Katowicach.

Spronk J. (1981) Interactive Multiple Goal Programming. Martinus Nijhoff, The Hague.

Steuer R. (1986) Multiple Criteria Optimization: Theory, Computation and Appication. John Willey and Sons.

Trzaskalik T. (2015) MCDM applications of near optimal solutions in dynamic programming. MCDM 10.

Trzaskalik T. (1998) Multiobjective Analysis in Dynamic Environmnent. The Karol Adamiecki University of Economics in Katowice Press.

Trzaskalik T. (1990) Wielokryterialne dyskretne programowanie dynamiczne. Teoria i zastosowania w praktyce gospodarczej. Akademia Ekonomiczna w Katowicach.

Trzaskalik T. (1986) Wybrane problemy programowania dynamicznego. Akademia Ekonomiczna w Katowicach.

INTERACTIVE MULTIPLE GOAL PROGRAMMING IN MULTIOBJECTIVE DISCRETE DYNAMIC PROGRAMMING

Abstract: The aim of the paper is to propose a method of finding a solution of the final tasks of multiple criteria discrete dynamic programming using suitably modified interactive ε - constraint approach. In the first phase single criterion dynamic programming algorithm is applied, as well the algorithm of generating near-optimal solutions. Next we operate on a finite set of sooutions, given as a list.

Keywords: MCDM, multiobjective disrete dynamic programming, IMGP, interactive method