ACTA UNIVERSITATIS LODZIENSIS
FOLIA OECONOMICA 103, 1990

Jerzy Tymiński*

DYNAMICZNY ALGORYTM OPTYMALIZACJI HIPOTETYCZNEJ - MOŻLIWOŚCI ZASTOSOWANIA W EKSPLOATACJI OBIEKTÓW MIESZKALNYCH

Olbrzymie zapotrzebowanie na metody optymalizacyjne w ostatnich latach spowodowało opracowanie całego szeregu algorytmów, które z równym powodzeniem to zapotrzebowanie zaspokajają[1].

Różnorodność podejścia (tj. wykorzystanie różnych zasad postępowania w procesach ekstremalizacji) powoduje ograniczoność stosowania tych metod do pewnej klasy problemów[2]. Znane metody programowania matematycznego bazują na wielu założeniach (głównie w odniesieniu do funkcji celu), przez co są mało uniwersalne[3]. Zasięg ich stosowalności jest ograniczony jedynie do pewnej grupy problemów gospodarczych[4]. Oczywiście takiej sytuacji nie ma, jeżeli chodzi o modele liniowe. Znane są metody programowania liniowego (szczególnie różne warianty metody simpleks), pozwalające na rozwiązanie teoretycznie wszelkich problemów techniczno-ekonomicznych, byleby były

* Dr, adiunkt w Zakładzie Ekonomiki Budownictwa i Inwestycji UŁ.

[1] Mikroekonomiczne problemy badań operacyjnych, Warszawa 1977, s. 7-12.

[2] W. Grabowski, Programowanie matematyczne, Warszawa 1980; P. Jędrzejewicz, Wybrane modele dycyzyjne w produkcji i eksploatacji, Warszawa 1980; S. Kryński. Minimalizacja funkcji wklęsłej na wielościanie wypukłym, "Matematyka Stosowana" 1978; W. I. Zangwill, Programowanie nieliniowe, Warszawa 1974.

[3] Jędrzejewicz, Wybrane modele ..., s. 207-224. Zangwill, Programowanie nieliniowe..., s. 51-52.

[4] Jędrzejewicz, Wybrane modele..., s. 69; Podstawy organizacji zarządzania i technologii w budownictwie, Warszawa 1985, s. 40.

one scharakteryzowane za pomocą układu zmiennych decyzyjnych pozostających w zależnościach liniowych[5]. Sprawa się jednakże komplikuje, jeżeli mamy do czynienia z problemami nieliniowymi. Szczególne problemy, np. zadanie programowania kwadratowego, można rozwiązać m. in. także metodą simpleks, opracowaną przez E. M. L. Beale'a[6]. W innych przypadkach należy sięgnąć do różnych - w zależności od rodzaju zagadnień optymalizacyjnych - metod programowania matematycznego[7]. W problemach, w których zmienne decyzyjne przyjmują postać liczb całkowitych, ma zastosowanie metoda podziału i ograniczeń (można tutaj też wymienić metodę płaszczyzn odcinających, metodę drzew decyzyjnych, metody heurystyczne i inne)[8].

Jeżeli chodzi ogólnie o teorię programowania nieliniowego, nie jest ona zbyt bogata. W szczególności nie istnieją w zasadzie ogólne metody programowania nieliniowego dla zagadnień, które nie są problemami programowania wypukłego[9]. W szczególności nie znane są metody, które w skończonej liczbie kroków - pozwoliłyby uzyskać rozwiązanie optymalne, chociaż są one zbieżne w nieskończonej liczbie iteracji[10]. Stosowanie zaś metod programowania wypukłego do problemów programowania nieliniowego niewypukłego jest co najmniej ryzykowne (nie zapewniają zbieżności w skończonej liczbie iteracji). Jednak wszystkie prawie znane metody programowania nieliniowego zakładają różniczkowalność (spełnienie warunków Kuhna-Tuckera) i ciągłość funkcji celu[11]. Istnieje co prawda dział programowania matematycznego - programowanie dynamiczne, który obejmuje wiele metod rozwiązania konkretnych problemów zarówno wklęsłych, jak i wypuk-

---

[5] Grabowski, Programowanie matematyczne..., s. 21; Kryński, Minimalizacja funkcji..., s. 35.

[6] Grabowski, Programowanie matematyczne..., s. 251.

[7] Zangwill, Programowanie nieliniowe..., s. 110, 134, 163, 195, 219, 250, 298.

[8] Grabowski, Programowanie matematyczne..., s. 332; Mikroekonomiczne problemy badań..., s. 94.

[9] Grabowski, Programowanie matematyczne..., s. 290; Kryński, Minimalizacja funkcji...

[10] Grabowski, Programowanie matematyczne..., s. 223.

[11] Ibidem.

łych. Niestety, programowanie dynamiczne może być stosowane tylko do pewnej klasy problemów, z małą liczbą zmiennych i ograniczeń[12]. Jak dotychczas nie opracowano metod w miarę uniwersalnych, tj. takich, które umożliwiałyby rozwiązywanie różnej klasy problemów.

Prezentowany w tej pracy Dynamiczny Algorytm Optymalizacji Hipotetycznej DAOH ze względu na jego uniwersalność może być przydatny w szczególności do rozwiązywania wielu nietypowych zadań charakteryzujących się dużą liczbą ograniczeń.

Algorytmem DAOH mogą być rozwiązywane różnego typu problemy optymalizacyjne programowania nieliniowego i liniowego, tj. problemy:

a) o dowolnej konstrukcji funkcji celu;

b) o dowolnej konstrukcji ograniczeń i ich liczbie (jeżeli mają postać (1b) i (1c);

c) o co najmniej trzech zmiennych decyzyjnych, jeżeli problem posiada jedynie tylko dwie zmienne, to można wprowadzić trzecią sztuczną (zarówno do funkcji celu, jak i ograniczeń);

d) o co najmniej jednym ograniczeniu modelowym lewostronnym (wzór 4a) (jeżeli takiego ograniczenia rozwiązywany problem nie posiada, to należy wprowadzić warunek sztuczny, zakładając dostatecznie dużą wartość M_0).

Algorytm DAOH pozwala na rozwiązywanie problemów programowania matematycznego:

a) całoliczbowych;

b) ciągłych problemów optymalizacji liniowej i nieliniowej wiążących się zarówno z maksymalizacją, jak i minimalizacją; należy w przypadku minimalizacji przekształcić funkcję celu (por. wzór 1), poprzez przemnożenie jej prze -1, nie jest przy tym istotne czy funkcje te dla F(X) i Gj(X) (por. wzory 1a i 1b) posiadają ciągłe pochodne cząstkowe pierwszego i drugiego rzędu);

c) problemów z ujemnymi wartościami zmiennych decyzyjnych ($X_i \leq 0$), należy wówczas dokonać pewnych drobnych weryfikacji algo-

---

[12] R. E. Bellman, S. E. Dreyfus, Programowanie dynamiczne, Warszawa 1967, s. 242. Grabowski, Programowanie matematyczne..., s. 139.

rytmu polegających na wprowadzeniu równoległym (z ciągiem kroków dodatnich) wektora kroków ujemnych, zaś w procedurze ekstremalizacji wprowadzić wartości modułowe.

Struktura matematyczna algorytmu nie jest złożona, daje się on łatwo oprogramować (np. program emc w dwóch opcjach: całoliczbowej i ciągłej został opracowany w Ośrodku Informatyki Uniwersytetu Łódzkiego i był wielokrotnie z powodzeniem użytkowany).

Algorytmem DAOH mogą być rozwiązywane problemy optymalizacyjne ze stosunkowo małą liczbą zmiennych decyzyjnych (liczba zmiennych X wpływa na czas obliczeń), ale dużą liczbą ograniczeń. Stąd szczególnie przydatny może być dla problemów produkcyjno-eksploatacyjnych charakteryzujących się dużą liczbą ograniczeń nie wymagających dużych dokładności.

Algorytm DAOH nie zakłada żadnych istotnych założeń na funkcję celu, a także na warunki ograniczające, jeżeli są one postaci:

$$F\ (x_1, x_2, \ldots, x_n) \qquad (1a)$$

$$G_j(x_1, x_2, \ldots, x_n) \geqslant\ ,\ \leqslant c_j \qquad (j = 1, 2, \ldots, U) \qquad (1b)$$

$$x_1, x_2, \ldots, x_n \geqslant b_i \qquad (i = 1, 2, \ldots, n) \qquad (1c)$$

Zależności (1a), (1b), (1c) oznaczają odpowiednio funkcję celu wielu zmiennych ($n \geqslant 3$), ograniczenia modelowe oraz zmienne decyzyjne przyjmują wartości dowolne (także całoliczbowe), przy czym funkcje F i G_j mogą być zarówno liniowe, jak i nieliniowe.

Konstrukcja algorytmu oparta jest na wieloetapowej rekurencji i wykorzystuje elementy teorii programowania dynamicznego[13]. Rozwiązanie problemu uzyskuje się pośrednio poprzez ekstremalizację zbieżnej - do funkcji celu (1a) - funkcji kryterialnej rozwiązywanego problemu, będącego hipotetyczną funkcją celu problemu, a zwanej w dalszym ciągu rozważań funkcjonałem kryterialnym.

---

[13] B e l l m a n, D r e y f u s, Programowanie dynamiczne, s. 20-21, J. T y m i ń s k i, Optymalizacja programu rozwoju produkcji towarowej w oparciu o czynniki wzrostu wydajności pracy, Uniwersytet Łódzki 1979, s. 141-150 (prca doktorska).

DAOH składa się z pięciu procedur obliczeniowych (por. rys. 1).

1. Procedura transformacji modelu wyjściowego (1a, 1b, 1c,) - rys. 1(1). Ma ona na celu utworzenie funkcjonału kryterialnego i sprowadza się do następujących procesów:

a) utworzenie n-tych członów funkcjonału kryterialnego wg zasady:

$$h_1 = \frac{F\ (T_l + S_1\ ,\ (T_l\ ,\ \ldots,\ T_l)}{|F\ (T_l,\ \ T_l,\ \ldots,\ T_l)|}$$

$$h_2 = \frac{F\ (T_l,\ T_l + S_2,\ \ldots,\ (T_l)}{|F\ (T_l,\ T_l,\ \ldots\ T_l)|}$$

.

.

.

$$h_n = \frac{F\ (T_l,\ T_l,\ \ldots,\ T_l + S_n)}{|F\ (T_l,\ T_l,\ \ldots,\ T_l)|} \tag{2}$$

gdze T_l oznacza parametr stabilizujący przybierający wartości liczbowe wg zasad:

T_o = N (liczba bardzo mała, np. 0,000001)

$T_1 = H_o = N$,

$T_2 = H_1$,

$T_3 = H_2$,

itd.; ogólnie: $T_l = H_{l-1}$ (l = 1, 2, ..., k = liczba kroków obliczeniowych), gdzie:

S_i - zmienna sterująca (w procesie warunkowego i bezwarunkowego sterowania optymalnego),

h_i - "hipotetyczna" zmienna decyzyjna,

H_l - wartość l-tego kroku (ustalona wg drugiej procedury).

b) utworzenie funkcjonału kryterialnego ogólnej postaci:

$$F_h = h_1,\ h_2,\ \ldots,\ h_n \longrightarrow \max \tag{3}$$

Algorytm DAOH realizuje poszukiwania największej wartości funkcji (3) w wielowymiarowym obszarze n zmiennych rzeczywistych zdefiniowanym poprzez:

- warunki modelowe:

lewostronne $L_g(X_1, X_2, \ldots, X_n) \leq M_g$[14], $(g = 1, 2, \ldots, G)$ (4a)

pozostałe $P_v(X_1, X_2 \ldots, X_n) \geq Z_v$, $(v = 1, 2, \ldots, V)$ (4b)

- warunki brzegowe:

$X_i \geq\ \leq b_i$ (5)

2. Procedura utalenia zmiennych (czynników hipotetycznych) h_1 - rys. 1(2a). Realizacja procedury rozpoczyna się od ustalenia wartości kroku wstępnego H_w, będącego rozwiązaniem bazowym. Mogą być tutaj zastosowane różne zasady ustalenia H_w, np.:

$$H_w = \min (b_1, b_2, \ldots, b_n) + a, \text{ (tylko dla } x_i \geq 0) \quad (6)$$

gdzie a oznacza stały parametr (np: 1,0, 0,5, 0,1 itd.).

Otrzymaną wartość H_w wprowadzamy (jako wstępną, bazową, zmienną decyzyjną) do lewostronnych warunków modelowych (4a), badając ich dopuszczalność dla bazowego rozwiązania, przy $H_w = X_1$, $H_w = X_2$, ..., $H_w = X_n$. Jeżeli zachodzi zależność (4a), to wtedy:

$$H_w = H_1^{\langle 1 \rangle} \quad (7)$$

gdzie $H_1^{\langle 1 \rangle}$ oznacza pierwszy krok w pierwszym cyklu optymalizacyjnym. Ogólnie $H_l^{\langle m \rangle}$ (m = 1, 2, ..., M = numer cyklu optymalizacyjnego). Zależność (7) jest konsekwencją ustalenia wartości wstępnego (bazowego), dopuszczalnego rozwiązania problemu.

---

[14] Do lewostronnych warunków modelowych należy zaliczyć jedynie te ograniczenia lewostronne, które w zapisie przy zmiennych decyzyjnych X_i nie posiadają znaku "-". Te wszystkie ograniczenia, które ten znak posiadają, należy przenieść do warunków pozostałych, a równocześnie (w problemach ciągłych) pożądane byłoby do warunków brzegowych. Jeżeli jest np.: $X_1 + X_2 - X_3 + \ldots + X_n \leq M_g$ to po przekształceniu $- X_1 - X_2 + X_3 - \ldots - X_n - M_g$ wprowadzamy tę nierówność do bloku ograniczeń pozostałych, zaś do warunków brzegowych możemy wprowadzić <u>dodatkowo</u> X_3 w postaci $X_3 \geq b_3 = -M_g + X_1 + X_2 \ldots + X_n$.

Jeżeli natomiast zajdzie zależność:

$$L_g(X_1, X_2, \ldots, X_n) > M_g \tag{8}$$

to H_w zmniejszamy, np. dwukrotnie, powtarzając zabieg przez wracanie do sprawdzenia zależności (4a). Jeżeli w dalszym ciągu otrzymamy zależność (8), czynność tę należy powtórzyć, zmniejszając, np. czterokrotnie H_w, aż do uzyskania zależności (4a). Jeżeli rozwiązanie istnieje, to po skończonej liczbie takich kroków, uzyskujemy (4a). Jeżeli nie uzyskamy (4a), to rozwiązanie nie istnieje i proces się kończy.

Dla uzyskania następnych l-tych kroków (l = 0,1, 2, ..., k) możemy zastosować ciąg liczb jednostajnych (rys. 1(2c)), np.:

$$H_1^{(1)} = H_w \cdot 1, \quad H_2^{(1)} = H_w \cdot 2, \quad H_3^{(1)} = H_w \cdot 3 \text{ itd.} \tag{9}$$

bądź też wykorzystując szybko rosnące ciągi liczb, jak Fibonacciego czy wykładniczy, np.:

$$\begin{aligned} H_1^{(1)} &= H_w \\ H_2^{(2)} &= H_w \cdot 1 \\ H_3^{(1)} &= H_w \cdot 2 \\ H_4^{(3)} &= H_w \cdot 3 \\ H_5^{(1)} &= H_w \cdot 5 \end{aligned} \qquad \text{itd.} \tag{10a}$$

$$\begin{aligned} H_1^{(1)} &= H_w \cdot 2^0 \\ H_2^{(2)} &= H_w \cdot 2^1 \\ H_3^{(3)} &= H_w \cdot 2^2 \\ H_4^{(1)} &= H_w \cdot 2^3 \end{aligned} \qquad \text{itd.} \tag{10b}$$

Do ustalonej w ten sposób wartości kroku dodajemy krok zerowy ustalony arbitralnie, dla którego przyjmujemy wartość dowolnie małą, bliską zeru (np. $H_0^{(1)} = 0{,}000001$). Wprowadzenie kroku zerowego jest zabiegiem pozwalającym na uzyskanie (w dalszych procedurach algorytmicznych) wariantu zerowego dla każdej zmiennej decyzyjnej (X_i).

Dla rozwiązania całoliczbowego wartości całego zbioru kroków ustalamy arbitralnie:

$H_0^{(1)}$ = N (np. równe 0,000001)

$H_1^{(1)}$ = 1

$H_2^{(1)}$ = 2

$H_3^{(1)}$ = 3

.

.

.

$$H_k^{(1)} = k \tag{11}$$

(k - wartość ostatniego kroku rozwiązania dopuszczalnego).
Po ustaleniu ciągu kroków dla pierwszego cyklu optymalizacyjnego $H_k^{(1)}(1 = 0, 1, 2, \ldots, k)$ przechodzimy do następnej procedury.

3. Procedura optymalnego sterowania. Trzecia procedura ekstremalizacji oparta jest na elementach teorii programowania dynamicznego[15] i składa się z następnych procesów obliczeniowych:

a) ustalenie warunkowego sterowania optymalnego,

b) ustalenie bezwarunkowego sterowania optymalnego.

Proces ustalania warunkowego sterowania optymalnego (rys. 1(3a)) rozpoczyna się od zerowego kroku ($H_0^{(1)}$) dla pierwszego członu funkcjonału F_h, tj. pierwszej "hipotetycznej" zmiennej decyzyjnej h_1. Zgodnie z właściwym dla programowania dynamicznego postępowaniem korzystamy z równania rekurencyjnego, które dla pierwszego etapu odpowiadającego h_1 będzie następujące:

$$P_1^{*}(S_1^{*}) = \max_{\substack{S_1 \in \{0,\, U_1\} \\ U_1 \in \{0,\, H_0^{(1)}\}}} \frac{F\,(T_0 + S_1,\ T_0,\ \ldots,\ T_0)}{|F\,(T_0,\ T_0,\ \ldots,\ T_0)|} \tag{12}$$

gdzie:

---

15 B e l l m a n, D r e y f u s, Programowanie-dynamiczne...; T y m i ń s k i, Optymalizacja programu...

$P_1^{\circledast}(S_1^{\circledast})$ - warunkowo optymalna wartość funkcjonału kryterialnego;

$S_1^{\circledast}$ - warunkowo optymalna wartość zmiennej sterującej w pierwszym etapie;

U_1 - zapas "środków hipotetycznych", jakie mamy do dyspozycji w pierwszym etapie, które należy optymalnie podzielić;

$H_0^{(1)}$ - wartość zerowego kroku dla pierwszego cyklu optymalizacyjnego (równego liczbie b małej, bliskiej zera). Symbol $\circledast$ tu i w dalszym ciągu oznacza warunkowo optymalny. Dla obliczenia drugiego etapu zerowego kroku ($H_0^{(1)}$) posłużyć się należy następującą zależnością rekurencyjną:

$$P_2^{\circledast}(S_2^{\circledast}) = \max\ (h_2 \cdot P_1^{\circledast}(S_1)) =$$

$$= \max\left(\frac{F(T_0, T_0 + S_2, \ldots, T_0)}{|F(T_0, T_0, \ldots, T_0)|} \cdot P_1^{\circledast}(S_1)\right) \tag{13}$$

$$S_2 \in \{0, U_2\}$$

$$U_2 \in \{0, H_0^{(1)}\}$$

Zatem:

$$P_2^{\circledast}(S_2^{\circledast}) = \max\left\{\left(\frac{F(T_0, T_0 + 0, \ldots, T_0}{|F(T_0, T_0, \ldots, T_0)|} \cdot P_1(0)\right),\right.$$

$$\left(\frac{F(T_0, T + 0_0, \ldots, T_0}{|F(T_0, T_0, \ldots, T_0)|} \cdot P_1(H_0^{(1)})\right),$$

$$\ldots$$

$$\left.\left(\frac{F(T_0, T_0 + H_0^{(1)}, \ldots, T_0}{|F(T_0, T_0, \ldots, T_0)|} \cdot P_1(0)\right)\right\} \tag{14}$$

Wyróżnione w nawiasach $\{\ \}$ trzy zależności odpowiadające następującym wartościom S_2, U_2: ($S_2 = 0$; $U_2 = 0$), ($S_2 = 0$; $U_2 = H_0^{(1)}$), ($S_2 = U_2$; $U_2 = H_0^{(1)}$). Dla wszystkich pozostałych etapów (czyli h_3,

h_4, h_5, ..., aż do h_{n-1}) wartość funkcji $P^{\circledast}(S^{\circledast})$ oblicza się wg wzorów (13)-(14). Ostatni zaś etap (tj. h_n) obliczamy w sposób następujący:

$$P_n^*(S^*) = \max_{S_n \in \{0;\ U_n\}} ((h_n \cdot p^*(S)) \tag{15}$$

$$U_n = H_o^{(1)}$$

tj.

$$P_n^*(S_n^*) = \max\left\{\left(\frac{F(T_o, T_o, \ldots, T_o + 0)}{|F(T_o, T_o, \ldots, T_o)|} \cdot P_{n-1}(H_o^{(1)}\right),\right.$$

$$\left.\left(\frac{F(T_o, T_o, \ldots, T_o + H_o^{(1)}}{|F(T_o, T_o, \ldots T_o)|}\right) \cdot P_{n-1}(0)\right)\right\} \tag{16}$$

dla $S_n = 0$ i $S_n = H_o^{(1)}$ (odpowiednio). Gwiazdka (*) oznacza ostateczne rezultaty (bez warunków).

Dla ułatwienia procesu obliczeń można utworzyć dla każdego etapu tablicę rezultatów.

Dla pierwszego etapu warunkowa wartość optymalna będzie następująca (tab. 1);

T a b e l a 1

Sterowanie warunkowe - pierwszy etap

U_1 \ S_1	0	U_1	$P_1(S_1)$	$S_1(U_1)$	$P_1^{\circledast}(S_1^{\circledast})$	$S_1^{\circledast}(U_1^{\circledast})$
0	1	.	1	0	-	-
$H_o^{(1)}$	1	A_1	A_1	$H_o^{(1)}$	A_1	$H_o^{(1)}$

Ź r ó d ł o: Opracowanie własne.

$$P_1^{\circledast}(S_1^{\circledast}) = A_1, \qquad (A_1 > 1) \tag{17}$$

zaś wartość zmiennej sterującej:

$$S_1^{\circledast}(U_1^{\circledast}) = H_0^{(1)} \tag{18}$$

Wyniki obliczeń dla drugiego etapu przedstawia tab. 2. W tym etapie mamy, dla $A_2 > A_1$ (przy: $A_2 > 1$)

Tabela 2

Sterowanie warunkowe - drugi etap

U_2 \ S_2	0	U_2	$P_2(S_2)$	$S_2(U_2)$	$P_2^{\circledast}(S_2^{\circledast})$	$S_2^{\circledast}(U_2^{\circledast})$
0	$1 \cdot 1 = 1$	.	1	0	-	-
$H_0^{(1)}$	$1 \cdot A_1 = A_1$	$A_2 \cdot 1 = A_2$	$\frac{A_1}{A_2}$	$\frac{0}{H^{(1)}}$	$\frac{A_1}{A_2}$	$\frac{0}{H^{(1)}}$

Źródło: Opracowanie własne.

$$P_2^{\circledast}(S_2^{\circledast}) = A_2 \tag{19}$$

oraz

$$S_2^{\circledast}(U_2^{\circledast}) = H_0^{(1)} \tag{20}$$

Jeżeli zaś: $A_1 > A_2$ $(A_1 > 1)$, to

$$P_2^{\circledast}(S_2^{\circledast}) = A_1 \tag{21}$$

oraz

$$S_2^{\circledast}(U_1^{\circledast}) = 0 \tag{22}$$

W przypadku $A_1 = A_2$ mamy dwa alternatywne rozwiązania (o jednakowej warunkowej wartości optymalnej). Analogicznie, jak dla drugiego etapu, będzie dla etapów następnych aż do n -1, n-ty etap jest przedstawiony w tab. 3.

Tabela 3

Sterowanie bezwarunkowe - n-etap

S_n / $U_n = H_o(1)$	0	$H_o^{(1)}$	$P_n^* S_n^*$	$S_n^*(U_n^*)$
$H_o^{(1)}$	$1 \cdot A_{n-1} = A_{n-1}$	$A_n \cdot 1 = A_n$	$\frac{A_{n-1}}{A_n}$	$\frac{0}{H_o(1)}$

Źródło: Opracowanie własne.

Jeżeli zatem wynik optymalny (w n-tym etapie bezwarunkowo optymalny) ustali się na:

$$P_n^*(S_n^*) = A_{n-1} \qquad (\text{przy } A_{n-1} > A_n) \tag{23}$$

to zmienna sterująca na n-tym etapie przyjmie wartość

$$S_n^*(0) = S_n^* = 0 \tag{24}$$

Jeżeli zaś:

$$P_n^*(S_n^*) = A_n \qquad (\text{gdy } A_n > A_{n-1}) \tag{25}$$

to zmienna sterująca przyjmie wartość:

$$S_n(H_o^{(1)}) = H_o^{(1)} \tag{26}$$

Wartości sterowania otymalnego ostatniego etapu są wartościami ostatecznymi dla zmiennej decyzyjnej X_n, zaś dla pozostałych zmiennych tj. X_{n-1}, X_{n-2}, ..., X_2, X_1 ich wartości znajdujemy w procesie sterowania bezwarunkowego.

Proces bezwarunkowego sterowania (rys. 1(3b)) dla przypadku kiedy $S_n^* = 0$ będzie przebiegał następująco:

$$S_n(U_n^*) = S_n^*(U_n^*) = S_n^* = 0$$

$$U_{n-1}^* = H_o^{(1)} - S_n^* = H_o^{(1)} - 0 = H_o^{(1)}$$

$$S_{n-1}^* = S_{n-1}^*(U_{n-1}^*) = S_n^*(H_o^{(1)}) = H_o^{(1)}$$[16]

$$U_{n-2}^* = U_{n-1}^* - S_{n-1}^*(H^{(1)}) = H_o^{(1)} - H_o^{(1)} = 0$$

.

.

.

$$U_2^* = U_3^* - S_3^*(0) = 0 - 0 = 0$$

$$S_2^*(U_2^*) = S_2^*(0) = 0$$

$$U_1^* = U_2^* - S_2^*(0) = 0 - 0 = 0$$

$$S_1^* = S_1^*(U_1^*) = S_1^*(0) = S_1^* = 0 \qquad (27)$$

Jeżeli prześledzimy przypadek, kiedy zmienna sterująca w ostatnim etapie sterowania warunkowego przyjmie wartość $S_n^* = H_o^{(1)}$, to wtedy proces sterowania bezwarunkowego będzie następujący:

$$S_n^*(U_n^*) = S_n^*(H_o^{(1)}) = S_n^* = H_o^{(1)}$$

$$U_{n-1}^* = U_n^* - S_n^*(H_o^{(1)}) = H_o^{(1)} - H_o^{(1)} = 0$$

---

[16] Jeżeli natomiast zajdzie przypadek $U_{n-1} = 0$, co będzie miało miejsce wtedy, kiedy $P_{n-1}(S_{n-1})$, przy $S_{n-1}(U_{n-1}) = S_{n-1}(0)$, to oczywiście $S_{n-1} = 0$.

$$S^*_{n-1} = S^*_{n-1}\,(U^*_{n-1}) = S^*_{n-1}(0) = 0$$

$$\cdot$$
$$\cdot$$
$$\cdot$$

$$U^*_1 = U^*_2 - S^*_2\,(U^*_2) = -\,S^*_2\,(0) = 0 - 0 = 0$$

$$S^*_1(U^*_1) = S^*_1(0) = S^*_1 = 0 \tag{28}$$

Jeżeli w drugim przykładowym przypadku wszystkie środki hipotetyczne otrzymała zmienna sterująca S_n (reprezentująca zmienną decyzyjną X_n), a pozostałe zero, to w pierwszym przypadku środki te otrzymywała zmienna sterująca S_{n-1} (zmienna decyzyjna X_{n-1}). W pierwszym zatem przypadku maksymalną wartość funkcjonału (3) uzyskujemy przy $h_1, h_2, \ldots, h_{n-2}, h_n = 1$ oraz $h_{n-1} = A_{n-1}$. Stąd

$$F_h = h_1 \cdot h_2 \cdot \;, \ldots, \;\cdot h_{n-1} \cdot h_n = 1 \cdot 1, \ldots, A_{n-1} \cdot 1 \longrightarrow \max \tag{29}$$

W drugim przypadku (tj. dla $h_n = A_n$)

$$F_h = h_1 \cdot h_2 \cdot, \ldots, \cdot h_{n-1} \cdot h_n = 1 \cdot 1, \ldots, \cdot 1 \cdot A_n \longrightarrow \max, \tag{30}$$

gdzie A_{n-1} oraz A_n oznaczają odpowiednio wartość n - 1-tego oraz n-tego członu (zmiennej hipotetycznej) funkcjonału F_k, uwzględniającego ogólnie wartości $S_i = H^m_1$ ($m = 1, 2, \ldots, M$; $1 = 1, 2, \ldots, k$; $i = 1, 2, \ldots, n$). Na tym kończy się proces ekstremalizacji dla kroku zerowego (H_0). W podobny sposób proces ten przebiega dla kroków $H_1^{(1)}, H_2^{(1)}, H_3^{(1)}, \ldots$, aż do kroku $H_k^{(1)}$ (gdzie k-ty krok oznacza dopuszczalną wartość zmiennej sterującej S_i, czyli dopuszczalnego rozwiązania, w pierwszym cyklu optymalizacyjnym) spełniających lewostronne warunki modelowe problemu. Ten ostatni fakt stwierdza się jednakże dopiero po przeprowadzeniu procedury retransformacji (oczywiście dla każdego kroku odrębnie).

4. Procedura transformacji. Czwarta procedura algorytmu składa się z następujących procedur:

a) ustalenie wartości zmiennych decyzyjnych - l-tego rozwiązania;

b) ustalenie dopuszczalności zmiennych decyzyjnych z:

- warunkami brzegowymi,
- warunkami (ograniczeniami) lewostronnymi modelu.

Ustalenie wartości zmiennych decyzyjnych (rys. 1 (4a)) dokonujemy po zakończeniu procesu ekstremalizacji dla każdego l-tego kroku poprzez przyrównanie wartości zmiennych sterujących S_i do wartości zmiennych decyzyjnych X_i:

$$(S_i) = X_l \text{ (dla l-tego rozwiązania)} \tag{31}$$

Tak więc pierwsze rozwiązanie (dla kroku zerowego, tj. $H_o^{(1)}$) będzie wyglądało następująco, dla przedstawionego pierwszego przypadku rozwiązania - wzory (27) i (29):

$$X_1 = 0,\quad X_2 = 0,\ \ldots,\quad X_{n-1} = H_o^{(1)},\quad X_n = 0 \tag{32}$$

Dla drugiego przypadku - wzory (28) i (30):

$$X_1 = 0,\ X_2 = 0,\ \ldots,\quad X_{n-1} = 0,\ X_n = H_o. \tag{33}$$

Dokonując następnie sprawdzenia dopuszczalności uzyskanego l-tego rozwiązania z warunkami brzegowymi (rys. 1 (4b)) stwierdzić należy, czy rozwiązanie to daje założoną zgodność (1c). Jeżeli jedna ze zmiennych nie spełniłaby zależności (1c), należy przyjąć arbitralnie wartość b_i jako wartość X_i. Na przykład, jeżeli dla (33) byłoby:

$$X_n \geqslant b_n \tag{34}$$

(zgodnie z modelowymi założeniami), zaś z otrzymanego rozwiązania wynika:

$$H_o^{(1)} < b_n \tag{35}$$

wtedy należy przyjąć[17]

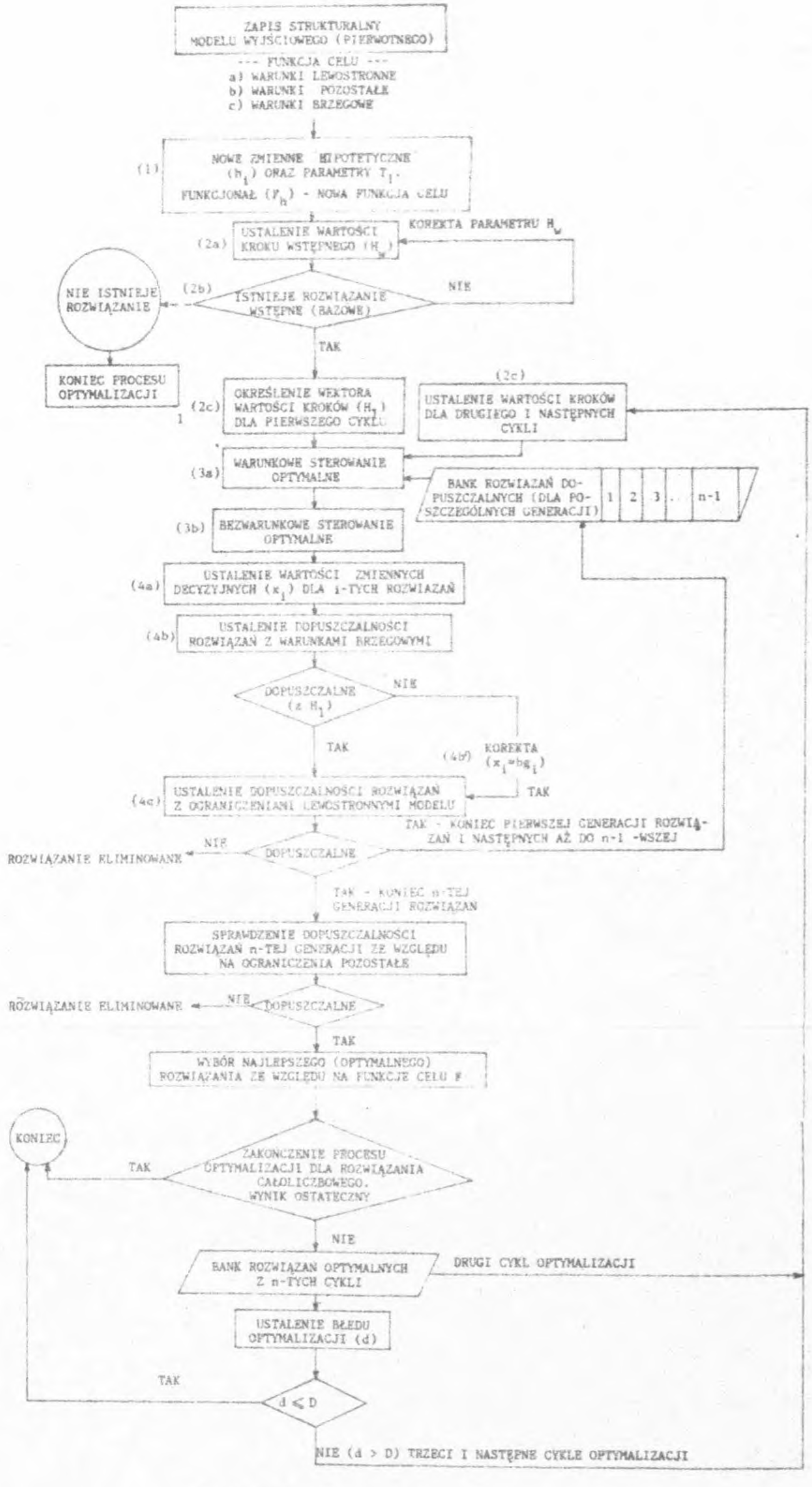
$$X_n = b_n \tag{36}$$

Ustalenie dopuszczalności rozwiązania ze względu na ograniczenia lewostronne (rys. 1 (4c)) sprowadza się do sprawdzenia, czy spełniony jest pierwszy z warunków modelowych (4) $L_g(X_1, X_2, \ldots, X_n) \leq M_g$. Jeżeli nierówność ta jest spełniona - rozwiązanie jest rozwiązaniem dopuszczalnym pierwszej generacji (rezultat pierwszego przebiegu realizacji trzeciej i czwartej procedury pozwalających na uzyskanie rozwiązania dopuszczalnego dla kroku $H_0^{(1)}$.) Jeżeli zaś nierówność (4a) nie jest spełniona - rozwiązanie jako niedopuszczalne jest odrzucane. Następuje dalej analogiczna realizacja procedur trzeciej i czwartej dla uzyskania rozwiązania dopuszczalnego, dla kroku $H_1^{(1)}$, a później $H_2^{(1)}$ itd., aż do $H_{k+1}^{(1)}$ ((k + 1) - e rozwiązanie niedopuszczalne). Uzyskane rozwiązania dopuszczalne dla $H_0^{(1)}$, $H_1^{(1)}$, ..., $H_k^{(1)}$ tworzą macierz rozwiązań dopuszczalnych pierwszej generacji, która stanowi podstawę wyjściową do realizacji procedury ekstremalizacji i - retransformacji (w następnej drugiej i dalej w i-tej, aż do n-tej generacji rozwiązań (i = 1, 2, ..., n = ilość generacji)).

Proces optymalizacji w drugiej generacji przebiega oddzielnie dla każdego rozwiązania dopuszczalnego z pierwszej generacji. Realizacja procesu tego wymaga pewnej korekty procedury ekstremalizacji (ściślej - podprocedury sterowania warunkowego). Zauważmy bowiem, iż ustalenie wartości zmiennych decyzyjnych w pierwszej generacji rozwiązań musi stanowić wartość stałą, która nie może być zmieniona w drugiej i następnych generacji rozwiązań. Rozważmy jeden z naszych przykładów (32). Zapis jego powinien być następujący:

---

[17] Jeżeli ogólnie w problemach ciągłych (dla $X_i \geq b_i$) struktura ograniczeń modelowych wymaga - celem skrócenia obliczeń - wprowadzenia (- poza $X_i \geq b_i$) całego wektora warunków brzegowych (por. wzór 4a), np.: $X_i \geq M_g - X_2 - X_{i-1} + \ldots = b_i'$, $X_i \geq M_g - X_1 - X_2 + \ldots = b_i''$, $X_i \geq M_g - X_1 - X_{i+1} + \ldots = b_i'''$, to wtedy będzie: X_i max (b_i', b_i'', b_i'''). Jeżeli zaś w jednym warunku wektora warunków brzegowych zamiast zależności " $\geq$ " wystąpi zależność "=", to ten warunek będzie stanowił podstawę zależności $X_i = b_i$ (bez konieczności wyboru warunku z maksymalną wartością).



Rys. 1. DAOH - schemat blokowy

$$
\begin{aligned}
X_1 &= 0 \\
X_2 &= 0 \\
&\ \vdots \\
X_{n-1} &= \boxed{H_o^{(1)}} \longrightarrow (S^{\circledast}_{n-1}(U^{\circledast}_{n-1}) = 0 \text{ oraz } P^{\circledast}_{n-1}(S^{\circledast}_{n-1}) = 1) \qquad (37) \\
X_n &= 0
\end{aligned}
$$

gdzie $\circledast$ oznacza wartość stałą (niezmienną w następnych generacjach). Oznacza to, iż w procesie optymalizacyjnym drugiej generacji, w procedurze ekstremalizacji, podprocedurze sterowania warunkowego (n - 1)-ty etap musi przebiegać nieco inaczej (por. tab. 4). Ten

T a b e l a 4

Sterowanie warunkowe (korekta) -(n - 1) etap

S_{n-1} / U_{n-1}	0	U_{n-1}	$P_{n-1}(S_{n-1})$	$S_{n-1}(U_{n-1})$
0 0	1 · 1 = 1	.	1	0
$H_o^{(1)}$	1 · 1 = 1	.	1	0

Ź r ó d ł o: Opracowanie własne.

zabieg (zastosowana zasada korekty procedury ekstremalizacji) wprowadzony jest po to, by wartości H_1 w następnych generacjach mogły się lokować na pozostałych zmiennych, z wyłączeniem zmiennych już ustalonych. Dotyczy to wszystkich i-tych zmiennych decyzyjnych (X_i), dla których zostały już ustalone wartości (S_i). Po zakończeniu procesu optymalizacji w drugiej generacji i ustaleniu macierzy rozwiązań dopuszczalnych rozpoczynamy proces optymalizacji w trzeciej generacji wg tej samej zasady (jak w generacji drugiej), a następnie czwartej aż do n-tej generacji. Zawsze - dla każdego rozwiązania z poprzedniej generacji - tworzony jest odrębny ciąg rozwiązań w następnej generacji. Po zakończeniu n-tej generacji wszystkie uzyskane rozwiązania dopuszczalne zostaną poddawane procesowi sprawdzania dopuszczalności ze względu na pozostałe ograniczenia (4a).

5. Sprawdzanie dopuszczalności rozwiązań ze względu na ograniczenia prawostronne (pozostałe) oraz wybór najlepszego rozwiązania ze względu na funkcje celu (1).

Na piątej procedurze kończy się pierwszy cykl optymalizacji, w wyniku którego uzyskujemy pierwsze przybliżenia rozwiązania optymalnego (rys. 1 (5b)). Drugi cykl i następne (m = 1, 2, ...,) różni się jedynie od pierwszego zasadą ustalania wartości ciągu kroków ($H_1^{(2)}$) - rys. 1 (2c). Zatem drugi cykl optymalizacji rozpoczynamy od ustalenia wartości wektora wartości kroków dla otrzymanego najlepszego rozwiązania (bądź - najlepszych, jeżeli wystąpiło kilka alternatywnych rozwiązań, ze względu na różne wartości zmiennych X_1, dla $F(X_1, X_2, ..., X_n) \longrightarrow max$) w pierwszym cyklu optymalizacyjnych. Nie korzystamy tutaj z ciągu liczb wykładniczych, czy - Fibonacciego. Możemy się posłużyć podziałem dziesiętnym. W takim razie druga procedura będzie przebiegała następująco:

$$H_s^{(2)} = H_1^{(2)}, H_2^{(2)}, ..., H_{10}^{(2)}, \quad (s = 1, 2, ..., 10) \tag{38}$$

gdzie:

$$H_1^{(2)} = H_{l-1}^{*(1)} + 0,1(H_{l+1}^{*(1)} - H_{l-1}^{*(1)}).$$

$$H_2^{(2)} = H_{l-1}^{*(1)} + 0,2(H_{l+1}^{*(1)} - H_{l-1}^{*(1)}).$$

.

.

.

$$H_{10}^{(1)} = H_{l-1}^{*(2)} 1,0(H_{l+1}^{*(1)} - H_{l-1}^{*(1)}).$$

zaś l^* krok stanowiący optymalną wartość H. Przy czym wartość kroków $H_s^{(2)}$ ustala się odrębnie dla każdej i-tej zmiennej decyzyjnej otrzymanego optymalnego rozwiązania w pierwszym cyklu.

Załóżmy np.: że optymalne rozwiązanie w pierwszym cyklu jest następujące:

$$X_1 = 0,\ X_2 = H_5^{(1)},\ \ldots,\ X_{n-1} = b_{n-1},\ X_n = H_2^{(1)} \tag{39}$$

Ustalenie wartości kroków dla drugiego cyklu przebiegać będzie następująco:

$$\begin{aligned}
\text{dla } X_1 \ - \ & H_o^{(2)} = 0{,}000001 \\
& H_1^{(2)} = 0 + 0{,}1\ (H_1^{(1)} - 0) \\
& H_2^{(2)} = 0 + 0{,}2\ (H_1^{(1)} - 0) - \text{itd. aż do } H_{10}^{(2)} \\
\text{dla } X_2 \ - \ & H_1^{(2)} = H_4^{(1)} + 0{,}1\ (H_6^{(1)} - H_4^{(1)}) \\
& H_2^{(2)} = H_4^{(1)} + 0{,}2\ (H_6^{(1)} - H_4^{(1)}) \text{ itd.} \\
& \cdot \\
& \cdot \\
& \cdot \\
\text{dla } X_{n-1} \ - \ & \text{wartości kroków dla wartości tej zmiennej } (b_{n-1}) \text{ pomijamy} \\
\text{dla } X_n \ - \ & H_1^{(2)} = H_1^{(2)} + 0{,}1\ (H_3^{(1)} - H_1^{(1)}) \\
& H_2^{(2)} = H_1^{(1)} + 0{,}2\ (H_3^{(1)} - H_1^{(1)}) - \text{itd.}
\end{aligned} \tag{40}$$

Następnie uszeregowujemy otrzymany zbiór kroków (dla wszystkich i-tych zmiennych) od wartości najmniejszej do największej i taki ciąg kroków wprowadzamy do procesu ekstremalizacji w drugim cyklu. Po przeprowadzeniu drugiego cyklu optymalizacji i uzyskaniu najlepszego rozwiązania przystępuje się do ustalenia błędu optymalizacji. Może to być, np.: błąd względny. Wtedy można posłużyć się następującym wzorem na błąd optymalizacji:

$$d = \frac{F^{(2)*} - F^{(1)*}}{F^{(2)*}} \tag{41}$$

gdzie:

$F^{(1)*}$ - optymalna wartość funkcji celu (1) z pierwszego cyklu,

$F^{(2)*}$ - optymalna wartość funkcji celu (1) z drugiego cyklu optymalizacji.

Jeżeli zachodzi zależność:

$$d \leqslant D \tag{42}$$

gdzie D oznacza wartość założonego błędu optymalizacji (np.: 0,001, 0,0005 itd.), to proces optymalizacji kończy się i otrzymane rozwiązanie (przy $F^{(2)*}$) w drugim cyklu jest rozwiązaniem ostatecznym.

Jeżeli natomiast:

$$d > D \tag{43}$$

to należy przystąpić do następnego cyklu optymalizacyjnego, ustalając zbiór kroków $H_s^{(3)}$ wg podanej zasady, uogólnionej wzorem (38) i ewentualnie do czwartego oraz dalszych, aż nie zostanie spełniona nierówność (42).

Prezentowany w opracowaniu algorytm DAOH był stosowany w wielu obliczeniach praktycznych i wykazał znaczną efektywność dla problemów programowania całoliczbowego, jest jednak pracochłonny w rozwiązywaniu problemów ciągłych (szczególnie dla dużych dokładności). Nie należy zatem zalecać stosowania go do rozwiązywania standardowych zadań, gdyż znane są metody o wiele szybsze. Jednakże gdy te klasyczne metody zawiodą ze względu na niespełnienie założeń, DAOH może okazać się szczególnie przydatny.

Jerzy Tymiński

DYNAMIC ALGORITHM OF HYPOTHETICAL OPTIMIZATION - POSSIBILITIES OF ITS APPLICATION IN EXPLOITATION OF DWELLINGS

The DAHO (Dynamic Algorithm of Hypothetical Optimization) is a combinations algorithm based on multistage recurrence and utilizing elements of the dynamic programming theory. It may be used for sol-

ving optimization problems with a relatively small number of decision-making variables (number of variables exerts its influence on time of calculations) but a big number of constraints. Hence, it may be useful especially for solving production-exploitation optimization problems (e. g. in exploitation of dwellings characterized by a big number of constraints). Due to the fact that it does not assume fundamental assumptions for functions of goal (as well as constraining conditions) it is quite a universal algorithm allowing to solve problems of both non-linear and linear programming.

The DAHO consists of five procedures:

1) transformation of starting model,

2) determining variables (hypothetical factors),

3) optimal steering (including the determining of conditional and nonconditional optimal steering),

4) retransformation (including the determining of values of decision-making variables and admissibility of values of decision-making variables with boundary conditions and left-sided constraints of the model),

5) checking admissibility of solutions due to remaining constraints and choosing the best solution with regard to the function of goal.