

Marcin SKOWRONEK

Ewa STARZEWSKA-KARWAN

ALGORYTMY OPTYMALIZACJI W ZADANIACH MODELOWANIA

Streszczenie. W artykule przedstawiono opis procedur umożliwiających rozwiązywanie zadań optymalizacji parametrycznej i sposób ich wykorzystania w zadaniach modelowania układów dynamicznych ciągłych z poziomu pakietu symulacyjnego DarP. Umieszczono również wyniki badań przydatności opracowanych algorytmów do rozwiązywania typowych zadań optymalizacji parametrycznej.

THE ALGORITHMS OF OPTIMIZATION IN THE MODELLING PROBLEMS

Summary. Seven algorithms of optimization realized in DarP packet are presented. These algorithms for calculating the unconstrained optimization and constrained optimization are addressed. The efficiency of optimization procedures is briefly discussed.

DIE ALGORITHMEN DER OPTIMIERUNG IN DEN MODELLIERUNGS AUFGABEN

Zusammenfassung. Im Artikel wurden sieben Algorithmen der Optimierung dargestellt, die im Paket DarP implementiert wurden. Diese Algorithmen ermöglichen, die Aufgaben der Parameteroptimierung sowohl mit den Begrenzungen als auch ohne sie zu lösen. Den Algorithmen wurden ebenfalls die Ergebnisse der Experimente hinzugefügt, die ihre Effizienz bewerten.

1. Wprowadzenie

Badania układu dynamicznego drogą modelowania prowadzi się przede wszystkim w celu:

- poznania właściwości modelu,
- doboru parametrów modelu.

Właściwości modelu, a tym samym właściwości układu, określone są przez jego odpowiedzi na typowe wymuszenia lub przez określone charakterystyki, będące funkcją tych odpowiedzi. Wpływ na odpowiedzi modelu mają takie elementy, jak: konfiguracja modelu, reguły działania, postać wymuszeń, warunki początkowe zmiennych stanu i parametry układu. Wszystkie te elementy nazwiemy parametrami modelu. Poznanie właściwości modelu to również określenie wpływu zmian parametrów modelu na jego odpowiedzi, jak i określenie zakresów zmian parametrów modelu, przy których odpowiedzi modelu zachowują pożądane cechy.

Drugi z celów modelowania sprowadza się do wyznaczenia takich parametrów modelu, przy których model będzie posiadać określone właściwości. Właściwości układu w tym przypadku określone są przez charakterystyki nazywane wskaźnikami jakości. Wskaźnikami jakości są zazwyczaj takie wielkości, jak czas wykonania określonej operacji lub zadania, wydatkowana energia, koszt wykonania operacji lub wielkość spodziewanego zysku. Celem modelowania jest więc taki dobór parametrów modelu, przy którym uzyskuje się żądaną wartość lub ekstremum wskaźnika jakości.

Realizacja tak sformułowanych celów modelowania wymaga posiadania odpowiedniego pakietu procedur optymalizacji bądź dużego nakładu pracy na przygotowanie takich procedur w celu zapisu wymaganych sekwencji badań symulacyjnych. W pracy przedstawiony jest pakiet procedur optymalizacyjnych dołączonych do pakietu symulacyjnego DarP [1]. Pakiet DarP udostępnia mechanizmy do opisu modeli układów dynamicznych ciągłych, opisu badań modelowych oraz do interakcyjnej realizacji badań modelowych i prezentacji wyników modelowania. Rozszerzenie pakietu DarP o zestaw procedur optymalizacji i interakcyjny interfejs do ich obsługi znakomicie skraca czas potrzebny na przygotowanie algorytmu i zapis sekwencji badań modelowych koniecznych do znalezienia parametrów modelu spełniających cel modelowania.

2. Zadanie optymalizacji parametrycznej

Dany jest układ dynamiczny opisany równaniem stanu:

$$\frac{dx}{dt} = f(x, t, \alpha), \quad (1)$$

gdzie:

$$x^T = [x_1, x_2, \dots, x_N] \quad - \quad \text{wektor stanu,}$$

$$\alpha^T = [\alpha_1, \alpha_2, \dots, \alpha_m] \quad - \quad \text{wektor parametrów,}$$

obserwowany w przedziale czasu $t \in (t_0, t_{\max})$. Warunki początkowe wektora stanu mogą być określone lub mogą być zależne od składowych wektora α , co zapisujemy równaniem

$$x(t_0) = \varphi(\alpha). \quad (2)$$

Przyjmujemy więc, że składowe wektora α mogą oznaczać parametry modelu, jeżeli występują jawnie w równaniu stanu (1) lub mogą oznaczać, bądź definiować wartości początkowe składowych wektora stanu, gdy występują jawnie w równaniach (2). Niezależnie od znaczenia składowych wektora α przyjmuje się, że nie zmieniają one wartości w przedziale obserwacji (t_0, t_{\max}) .

Rozwiązanie równania stanu (1) jest funkcją zmiennej niezależnej t oraz funkcją składowych wektora parametrów α , co podkreślamy zapisując rozwiązanie w postaci

$$x = x(t, \alpha). \quad (3)$$

W zadaniach modelowania, których celem modelowania jest dobór parametrów modelu, wprowadza się zazwyczaj pewną charakterystykę – funkcjonal, który przypisuje każdemu rozwiązaniu (3) liczbę, określającą przydatność tego rozwiązania. Charakterystyki tego typu nazywane są wskaźnikami jakości i oznaczane zapisem

$$Q(\alpha) = \phi(x(t, \alpha)), \quad (4)$$

gdzie funkcjonal ϕ jest określony analitycznie (jawnie).

Funkcja $Q(\alpha)$ oznacza tu zależność wskaźnika jakości od wektora parametrów α . Funkcja ta nie jest znana analitycznie, gdyż nie jest znana zazwyczaj analityczna postać rozwiąza-

nia (3). Wyznaczenie wartości wskaźnika jakości Q , dla zadanego wektora $\underline{\alpha}$, wymaga każdorazowo rozwiązania układu równań stanu (1).

Zadanie optymalizacji parametrycznej [2] sprowadza się do doboru wartości składowych wektora $\underline{\alpha}$ ze zbioru dopuszczalnych wartości \mathbf{A} , ($\underline{\alpha} \in \mathbf{A}$), tak by wskaźnik jakości (4) przyjmował minimalną wartość.

Przykłady typowych zadań optymalizacji parametrycznej przedstawiono poniżej.

Dobór parametrów układu regulacji. W zadaniach tego typu składowe wektora $\underline{\alpha}$ występują tylko w równaniach stanu. Jeżeli przez $y(t)$ oznaczymy zmienną wyjściową modelu (rozpatrujemy układ jednowymiarowy), a przez $w(t)$ pożądaną postać sygnału wyjściowego, to wskaźnik jakości określony jest wyrażeniem

$$Q(\underline{\alpha}) = \int_{t_0}^{t_{\max}} (y(t) - w(t))^2 dt . \quad (5)$$

Strojenie modelu (identyfikacja). W zadaniach tego typu przyjmuje się, że model układu znany jest z *dokładnością do współczynników*, będących w tym przypadku składowymi wektora $\underline{\alpha}$. Przyjmuje się dodatkowo, że znana jest odpowiedź układu $y(t)$ na pobudzenie sygnałem wejściowym $u(t)$. Odpowiedzią modelu pobudzanego sygnałem wejściowym $u(t)$ jest natomiast sygnał $y_M(t)$. Celem modelowania jest tu taki dobór składowych wektora $\underline{\alpha}$, by różnica między odpowiedzią układu i odpowiedzią modelu była minimalna. Miarą tej różnicy może być wskaźnik jakości

$$Q(\underline{\alpha}) = \int_{t_0}^{t_{\max}} (y(t) - y_M(t))^2 dt . \quad (6)$$

Zadanie dwugraniczne. W zadaniach tego typu wartości niektórych składowych wektora stanu nie są znane w chwili początkowej. Wartości początkowe tych zmiennych stanu przyjmuje się jako składowe wektora $\underline{\alpha}$ bądź też jako funkcje składowych tego wektora. Poszukiwane rozwiązanie równania stanu (1) powinno w chwili końcowej spełniać warunek

$$G(\underline{x}(t_{\max})) = 0 . \quad (7)$$

Jako wskaźnik jakości przyjmuje się zazwyczaj wyrażenie

$$Q(\underline{\alpha}) = G^2(\underline{x}(t_{\max})) . \quad (8)$$

Zadanie wartości własnych. W zadaniach tego typu składowe wektora $\underline{\alpha}$ występują tylko w równaniach stanu. Poszukuje się takich wartości składowych wektora $\underline{\alpha}$, by rozwiązanie równania stanu (1) spełniało w chwili końcowej warunek

$$G(\underline{x}(t_{\max})) = 0 . \quad (9)$$

Podobnie jak dla zadania dwugranicznego jako wskaźnik jakości przyjmuje się wyrażenie

$$Q(\underline{\alpha}) = G^2(\underline{x}(t_{\max})) . \quad (10)$$

3. Opis algorytmów optymalizacji

3.1. Sformułowanie problemu i klasyfikacja algorytmów

Zadanie *optymalizacji parametrycznej* sprowadza się do znalezienia takiej wartości wektora $\hat{\underline{\alpha}}$ ze zbioru dopuszczalnych wartości A ($\hat{\underline{\alpha}} \in A$), by dla każdego $\underline{\alpha} \in A$ spełniona była nierówność $Q(\hat{\underline{\alpha}}) \leq Q(\underline{\alpha})$, co zapisujemy:

$$\min_{\underline{\alpha} \in A} Q(\underline{\alpha}) = Q(\hat{\underline{\alpha}}) . \quad (11)$$

Jest to zadanie minimalizacji wskaźnika jakości $Q(\underline{\alpha})$ na zbiorze A .

Zbiór dopuszczalnych wartości A wyznaczony jest zazwyczaj przez układ nierówności

$$g_i(\underline{\alpha}) \leq 0 \quad i = 1, 2, \dots, k_1 \quad (12)$$

i/lub układ równań

$$h_i(\underline{\alpha}) = 0 \quad i = 1, 2, \dots, k_2 . \quad (13)$$

Jeśli zbiór A jest całą przestrzenią R^m (nie ma ograniczeń), to jest to zadanie *optymalizacji bez ograniczeń*.

Przedstawione w pracy algorytmy optymalizacji są algorytmami numerycznymi. Numeryczne rozwiązanie zadania optymalizacji polega na znalezieniu w sposób rekurencyjny takiego ciągu wartości parametrów $\underline{\alpha}_1, \underline{\alpha}_2, \dots, \underline{\alpha}_n$, by

$$\lim_{n \rightarrow \infty} \alpha_n = \hat{\alpha} . \quad (14)$$

Dalej przedstawiono algorytmy optymalizacji bez ograniczeń. Algorytmy te po pewnej modyfikacji wskaźnika jakości można stosować też w przypadku optymalizacji z ograniczeniami (metoda optymalizacji z funkcją kary).

Wszystkie prezentowane dalej algorytmy przeszukują przestrzeń R^m wzdłuż prostych zwanych *kierunkami poszukiwań*. W realizowaniu poszukiwań pojawiają się dwa problemy:

- 1) ustalania kierunków poszukiwań,
- 2) znajdowania kolejnego punktu (przybliżenia minimum) wzdłuż danego kierunku poszukiwań.

Te dwa problemy stanowią kryteria podziału większości metod optymalizacji.

Kierunki poszukiwań ustalać można opierając się tylko na wartościach wskaźnika jakości bądź na wartości jego gradientu. Stąd podział metod optymalizacji na *metody bezgradientowe* i *metody gradientowe*. W metodach bezgradientowych poszukuje się minimum w m niezależnych kierunkach tworzących bazę. Bazę tę modyfikuje się na podstawie informacji o zmianach wartości wskaźnika jakości w poprzednich krokach. W metodach gradientowych wartość gradientu służy do wyznaczenia kierunku poszukiwań, który to kierunek w kolejnych krokach można korygować.

W przypadku drugiego kryterium metody optymalizacji dzieli się na *metody poszukiwań prostych* oraz *metody kierunków poprawy*. W metodach poszukiwań prostych bada się wartość wskaźnika jakości w 1 lub 2 punktach leżących na kierunku poszukiwań, przy czym sposób wyboru tych punktów jest ustalony na początku każdej iteracji. W metodach kierunków poprawy wyznacza się minimum wzdłuż kierunku poszukiwań. Efektywność tych metod zależy od przyjętego algorytmu poszukiwań minimum w kierunku.

Dalej przedstawiono trzy metody poszukiwań w kierunku: *metodę złotego podziału*, *metodę interpolacji kwadratowej* i *metodę Berndta*, dwie metody poszukiwań prostych, które są też metodami bezgradientowymi: *metodę Hooke'a-Jeevesa* i *metodę Neldera-Meada*, dwie bezgradientowe metody poprawy: *metodę Gaussa-Seidela* i *metodę Powella* oraz dwie gradientowe metody poprawy: *metodę gradientu sprzężonego* i *metodę zmiennej metryki*.

3.2. Metody poszukiwania minimum bez ograniczeń

3.2.1. Metody poszukiwania minimum w kierunku

Zadaniem tej metody jest określenie minimum funkcji przy zadanym punkcie startowym α_0 i kierunku poszukiwań d , czyli obliczenie:

$$\min_{\tau \in \mathbb{R}} Q(\alpha_0 + \tau d) = Q(\alpha_0 + \tau_{\min} d). \quad (15)$$

Problem sprowadza się do wyznaczenia minimum funkcji jednej zmiennej. We wszystkich niżej prezentowanych metodach poszukiwania minimum w kierunku najpierw lokalizuje się położenie minimum, czyli dla zadanych α_0 i d określa się przedział $\langle a, b \rangle$ zawierający optymalną wartość τ_{\min} parametru τ .

Złoty podział

Metoda ta polega na zawężaniu w kolejnych iteracjach przedziału $\langle a, b \rangle$, tak by obejmował on minimum wg następującego algorytmu. Niech w i -tej iteracji przedział obejmujący minimum wynosi $\langle a^i, b^i \rangle$. Wyznaczmy punkty β^i i γ^i takie, że $\beta^i < \gamma^i$ oraz spełniona jest zależność:

$$\frac{\gamma^i - a^i}{b^i - a^i} = \frac{b^i - \beta^i}{b^i - a^i} = \xi, \quad \text{gdzie } \xi = \frac{\sqrt{5} - 1}{2}.$$

Następnie zawęża się przedział wg następującej reguły:

jeśli $Q(\beta^i) < Q(\gamma^i)$, to:

$$a^{i+1} = a^i, \quad b^{i+1} = \gamma^i, \quad \gamma^{i+1} = \beta^i, \quad \beta^{i+1} = a^i + (1 - \xi)(b^i - a^i),$$

jeśli $Q(\beta^i) \geq Q(\gamma^i)$, to:

$$a^{i+1} = \beta^i, \quad b^{i+1} = b^i, \quad \beta^{i+1} = \gamma^i, \quad \gamma^{i+1} = b^i - (1 - \xi)(b^i - a^i).$$

Obliczenia są kończone, jeśli $|b^{i+1} - a^{i+1}| < \varepsilon$, gdzie ε jest założoną dokładnością.

Interpolacja kwadratowa

W metodzie tej określa się trzy punkty β, δ, γ (należące do przedziału $\langle a, b \rangle$), takie by $\beta < \delta < \gamma$ oraz $Q(\delta) < Q(\beta)$ i $Q(\delta) < Q(\gamma)$. Należy określić w tych trzech punktach (węzłach) wielomian interpolacyjny Lagrange'a stopnia drugiego (parabolę) oraz znaleźć minimum tego wielomianu. Punkt, w którym wielomian osiąga minimum

$$x_{\min} = \delta - \frac{1}{2} \frac{(\delta - \beta)^2 [Q(\delta) - Q(\gamma)] - (\delta - \gamma)^2 [Q(\delta) - Q(\beta)]}{(\delta - \beta)Q(\delta) - Q(\gamma) - (\delta - \gamma)[Q(\delta) - Q(\beta)]} \quad (16)$$

wprowadza się w miejsce najgorszego węzła (w którym wskaźnik jakości osiąga wartość największą) i procedurę tę powtarza się, aż zostanie spełnione kryterium dokładności. Kryterium tym jest warunek, by bezwzględna wartość różnicy między minimum wielomianu interpolacyjnego a najlepszym węzłem była mniejsza od zadanej dokładności.

Może okazać się jednak, że znaleziony punkt nie jest szukanym minimum, bo parabola osiąga w nim maksimum. Również zamiana punktów może prowadzić do rozbieżności. Dlatego w metodzie tej należy wprowadzić szereg dodatkowych zabezpieczeń.

Metoda Brenta

Metoda Brenta stanowi połączenie metody interpolacji kwadratowej z metodą złotego podziału. W metodzie tej operuje się na sześciu punktach: a , b , u , v , w , x , o następującym znaczeniu: a , b wyznaczają przedział, w którym poszukuje się minimum, x jest punktem, w którym wskaźnik jakości w danej iteracji osiąga wartość najmniejszą, w jest drugim najlepszym punktem, v jest poprzednią wartością w , a u jest nowym punktem wewnątrz przedziału $\langle a, b \rangle$ wyznaczonym metodą interpolacji kwadratowej lub złotego podziału. W dwóch pierwszych iteracjach określa się początkowe wartości punktów x , w , v metodą złotego podziału.

Algorytm dla każdej iteracji:

1. Wyznacz minimum u paraboli przechodzącej przez punkty x , w , v .
2. Jeśli nowy punkt u leży poza przedziałem $\langle a, b \rangle$ lub jego odległość od punktu x (tzw. krok) jest większa od połowy kroku z poprzedniej iteracji, to nowy punkt u wyznacz metodą złotego podziału:

$$u = \begin{cases} x + (a - x)\tau & \text{gdy } x \geq \frac{a + b}{2} \\ x + (b - x)\tau & \text{gdy } x < \frac{a + b}{2} \end{cases}$$

gdzie $\tau = 0.3819660$.

3. Jeśli $Q(u) \geq Q(x)$, to wykonaj etap 4, w przeciwnym wypadku wykonaj $x = u$, $w = u$, $v = w$ (zamiana punktów), określ nowy przedział wg reguły:

$$\begin{cases} a = x & \text{gd}y \quad u \geq x \\ b = x & \text{gd}y \quad u < x \end{cases}$$

i wykonaj etap 6.

4. Określ nowy przedział wg reguły:

$$\begin{cases} a = u & \text{gd}y \quad u < x \\ b = u & \text{gd}y \quad u \geq x \end{cases}$$

Jeśli $Q(u) \leq Q(w)$ lub $w = x$, to dokonaj zamiany punktów $v = w$, $w = u$ i wykonaj etap 6, w przeciwnym wypadku wykonaj etap 5.

5. Jeśli $Q(u) \leq Q(v)$ lub $v = x$, lub $v = w$, to wykonaj $v = u$ i wykonaj etap 6, w przeciwnym razie wykonaj tylko etap 6.
6. Sprawdź warunek zbieżności, czyli czy krok jest mniejszy od zadanej dokładności. Jeśli tak, to stop, w przeciwnym razie wykonaj etap 1 (kolejna iteracja).

3.2.2. Metody poszukiwania prostych

W kolejnych iteracjach przeszukuje się w określony sposób otoczenie wektora α_j będącego i -tym przybliżeniem minimum. Niżej przedstawiono dwa algorytmy: Hooke'a-Jeevesa i Nelder-Meada (simplex).

Metoda Hooke'a-Jeevesa

Przed właściwymi obliczeniami określa się bazę złożoną z m wektorów wzajemnie ortogonalnych e_1, \dots, e_m , punkt startowy α_0 , długość kroku τ , współczynnik zmniejszenia kroku β oraz dokładność ϵ . Na początku przyjmuje się $\alpha_{bo} = \alpha_0$. Każda iteracja składa się z etapu próbnego i roboczego.

Etap próbny:

1. Oblicz $Q_0 = Q(\alpha_0)$, $j = 1$.
2. Wykonaj krok próbny w kierunku e_j , $\alpha_j = \alpha_{j-1} + \tau e_j$. Jeśli był to krok pomyślny, czyli gdy $Q_0 > Q(\alpha_j)$, to podstaw $Q_0 = Q(\alpha_j)$ i wykonaj etap 4, w przeciwnym przypadku wykonaj etap 3.
3. Wykonaj krok w przeciwnym kierunku tj. $\alpha_j = \alpha_{j-1} - \tau e_j$. Jeśli był to krok pomyślny, czyli gdy $Q_0 > Q(\alpha_j)$, to podstaw $Q_0 = Q(\alpha_j)$ i wykonaj etap 4, w przeciwnym przypadku podstaw $\alpha_j = \alpha_{j-1}$ i wykonaj etap 4.

4. Jeśli $j < m$, to podstaw $j = j + 1$ i wykonaj etap 2, w przeciwnym przypadku wykonaj etap 5.
5. Zbadaj, czy wykonano kroki pomyślne, czyli czy $Q(\alpha_0) > Q(\alpha_m)$. Jeśli tak, to wykonaj etap 7, w przeciwnym razie wykonaj etap 6.
6. Jeśli $\tau < \varepsilon$, to zakończ obliczenia, w przeciwnym razie zmniejsz krok $\tau = \beta \tau$ i wykonaj etap 1.

Etap roboczy:

7. Podstaw $\alpha_b = \alpha_m$. Oblicz $\alpha_0 = 2\alpha_b - \alpha_{b0}$ oraz podstaw $\alpha_{b0} = \alpha_b$ i wykonaj etap 1.

Metoda Neldera-Meada (simplex)

Metoda ta polega na przekształcaniu m -wymiarowego wielościanu (zwanego simplexem) o $m + 1$ wierzchołkach w punktach P_1, \dots, P_{m+1} , tak by odległości między wierzchołkami w pobliżu szukanego minimum były nie większe od zadanej dokładności ε .

W trakcie przekształcania wielościanu (simplexu) wyznaczane są 3 punkty:

- P_h – wierzchołek simplexu, w którym wskaźnik jakości ma wartość największą,
- P_l – wierzchołek simplexu, w którym wskaźnik jakości ma wartość najmniejszą,
- P_o – środek symetrii simplexu wyznaczonego wg wzoru

$$P_o = \frac{1}{m} \sum_{i=1}^{m+1} P_i, \text{ przy czym } i \neq h$$

Przez sumę punktów należy rozumieć sumę wektorów wodzących tych punktów.

Simplex przekształca się realizując 4 operacje:

- operację odbicia punktu P_h względem P_o realizowaną wg wzoru

$$P^* = (1 + \lambda)P_o - \lambda P_h, \quad \lambda > 0,$$

- operację ekspansji punktu P^* względem P_o realizowaną wg wzoru

$$P^{**} = (1 + \gamma)P^* - \gamma P_o, \quad \gamma > 1,$$

- operację kontrakcji punktu P_h względem P_o realizowaną wg wzoru

$$P^{***} = (1 - \beta)P_o + \beta P_h, \quad 0 < \beta < 1,$$

- operację redukcji simplexu polegającą na zbliżeniu wierzchołków do punktu P_l realizowaną wg wzoru $P_i = (1 + \delta)P_l + \delta P_i$, $i = 1, \dots, m+1$, $i \neq l$, $0 < \delta < 1$.

Najczęściej przyjmuje się $\lambda = 1$, $\beta = 0.5$, $\gamma = 2$, $\delta = 0.5$.

Algorytm obliczeń

Dane są: dokładność ε , współczynniki α , β , γ , δ .

1. Określ $m+1$ wierzchołków simplexu P_1, \dots, P_{m+1} .
2. Oblicz wartość wskaźnika jakości w wierzchołkach simplexu $Q(P_i)$ $i = 1, \dots, m+1$.
3. Wyznacz punkty P_h, P_l, P_o .
4. Wykonaj operację odbicia. Jeśli $Q(P^*) < Q(P_l)$, to wykonaj etap 5, w przeciwnym razie wykonaj etap 6.
5. Wykonaj ekspansję. Jeśli $Q(P^{**}) < Q(P^*)$, to podstaw $P_h = P^{**}$ i wykonaj etap 8, w przeciwnym razie podstaw $P_h = P^*$ i wykonaj etap 8.
6. Jeśli $Q(P^*) \geq Q(P_i)$ dla $i = 1, \dots, m+1$, $i \neq h$, to wykonaj etap 7, w przeciwnym razie postaw $P_h = P^*$ i wykonaj etap 8.
7. Wykonaj kontrakcję. Jeśli $Q(P^{***}) \geq Q(P_i)$ dla $i = 1, \dots, m+1$, $i \neq h$, to dokonaj redukcji simplexu i wykonaj etap 2, w przeciwnym razie podstaw $P_h = P^{***}$ i wykonaj etap 8.
8. Jeśli $\max_k \|P_k - \frac{1}{m+1} \sum_{i=1}^{m+1} P_i\| < \varepsilon$ dla $k = 1, \dots, m+1$, to zakończ obliczenia, w przeciwnym razie wykonaj etap 2 (kolejna iteracja).

3.2.3. Bezgradientowe metody kierunków poprawy

W metodach tych minimalizuje się wskaźnik wzdłuż kolejnych kierunków, tworzących bazę w R^m . Niżej przedstawiono dwie metody: Gaussa-Seidela ze stałą bazą oraz metodę Powella z modyfikowaną bazą.

Metoda Gaussa-Seidela

Jako bazę przyjmuje się m wektorów wzajemnie ortogonalnych e_1, \dots, e_m w przestrzeni R^m . W każdej iteracji minimalizuje się wskaźnik jakości wzdłuż kolejnych kierunków e_j , $j = 1, \dots, m$. Obliczenia kończy się, gdy odległość punktów uzyskanych w kolejnych dwóch iteracjach jest mniejsza od zadanej dokładności.

Algorytm obliczeń

Dany jest dowolny punkt startowy α_0 i dokładność obliczeń ε . Dla kolejnej iteracji wykonaj:

1. $j = 1$.
2. Oblicz minimum funkcji celu w kierunku e_j , czyli:

$$\min_{\tau \in R} Q(\alpha_{j-1} + \tau e_j) = Q(\alpha_j)$$

3. Jeśli $j < m$, to $j = j + 1$ i wykonaj etap 2, w przeciwnym wypadku wykonaj etap 4.
4. Sprawdź, czy odległość punktów α_m i α_0 jest mniejsza od dokładności ε ($\|\alpha_m - \alpha_0\| < \varepsilon$).

Jeśli tak, to zakończ obliczenia, w przeciwnym wypadku podstaw $\alpha_0 = \alpha_m$ i wykonaj etap 1.

Metoda Powella

W metodzie tej ma miejsce modyfikacja kierunków poszukiwań. Modyfikacja ta polega na tym, że w każdej iteracji w miejsce jednego z wcześniejszych kierunków wprowadza się kierunek sprzężony [3]. W pierwszej iteracji bazę podstawową tworzą kierunki jak w metodzie Gaussa-Seidela.

Algorytm obliczeń

Dany dowolny punkt startowy α_0 i baza kierunków poszukiwań e_1, \dots, e_m , będąca zbiorem m wektorów wzajemnie ortogonalnych.

Etap wstępny: wykonaj minimalizację w kierunku e_m , tzn. oblicz

$$\min_{\tau \in R} Q(\alpha_0 + \tau e_m) = Q(\alpha_m)$$

i podstaw $\alpha_0 = \alpha_m$.

Dla każdej iteracji wykonaj:

1. Szukaj minimum (jak w metodzie Gaussa-Seidela) wzdłuż kierunków e_1, \dots, e_m , czyli wyznacz

$$\min_{\tau \in R} Q(\alpha_{j-1} + \tau e_j) = Q(\alpha_j) \quad j = 1, \dots, m$$

2. Wyznacz kierunek sprzężony:

$$e_{m+1} = \frac{\alpha_m - \alpha_0}{\|\alpha_m - \alpha_0\|}$$

i wyznacz minimum wzdłuż tego kierunku

$$\min_{\tau \in R} Q(\alpha_m + \tau e_{m+1}) = Q(\alpha_{m+1})$$

Zbadaj, czy spełnione zostało kryterium dokładności $|\alpha_{m+1} - \alpha_0| < \varepsilon$. Jeśli tak, to zakończ obliczenia, a jeśli nie, to wykonaj etap 3.

3. Dokonaj modyfikacji bazy $e_j = e_{j+1}$ $j = 1, \dots, m$ oraz przyjmij $\alpha_0 = \alpha_{m+1}$ i wykonaj etap 1.

W metodzie Powella w trakcie obliczania kierunku sprzężonego może dojść do liniowej zależności kierunków. Aby uchronić się przed taką ewentualnością, modyfikacja kierunku następuje wtedy, gdy spełniona jest zależność:

$$\frac{\tau_{\max} \Delta}{\alpha_c} \geq 0.8$$

- gdzie τ_{\max} – maksymalne przesunięcie wzdłuż jednego z m kierunków k -tej iteracji,
 α_c – całkowite przesunięcie w k -tej iteracji,
 Δ – wyznacznik macierzy utworzonej z wektorów e_1, \dots, e_m .

W przeciwnym przypadku następna iteracja przebiega bez modyfikacji bazy. Obowiązują poprzednio przyjęte kierunki.

3.2.4. Gradientowe metody kierunków poprawy

Metody te wymagają obliczania gradientu. Zalicza się je do mieszanych metod kierunku poprawy. W metodach tych zakłada się, że funkcja celu jest ograniczona od dołu funkcją:

$$q(\alpha) = a + B^T \alpha + \frac{1}{2} \alpha^T A \alpha \tag{17}$$

gdzie elementy macierzy A są drugimi pochodnymi cząstkowymi wskaźnika jakości.

Metoda gradientu sprzężonego

W metodzie tej gradient wyznacza kierunek poszukiwań minimum.

Algorytm obliczeń

Dany jest dowolny punkt startowy α_0 oraz dokładność ε .

Etap wstępny: podstaw $k = 1$, $\alpha_k = \alpha_0$. Oblicz gradient wskaźnika jakości $\nabla Q(\alpha_k)$. Wykonaj dla każdej iteracji.

1. Przyjmij kierunek poszukiwań $d_k = -\nabla Q(\alpha_k)$.

2. Zbadaj, czy zostało osiągnięte minimum, czyli czy $\langle \nabla Q(\alpha_k), \nabla Q(\alpha_k) \rangle < \varepsilon$, gdzie symbol \langle, \rangle oznacza iloczyn skalarny. Jeśli tak, to zakończ obliczenia, a jeśli nie, to wykonaj etap 3.

3. Wykonaj minimalizację w kierunku d_k , czyli oblicz:

$$\min_{\tau \in \mathbb{R}} Q(\alpha_k + \tau d_k) = Q(\alpha_{k+1}).$$

4. Oblicz $\nabla Q(\alpha_{k+1})$ i wyznacz kierunek sprzężony $d_{k+1} = -\nabla Q(\alpha_{k+1}) + \beta_k d_k$, gdzie:

$$\beta_k = \frac{\langle \nabla Q(\alpha_{k+1}), \nabla Q(\alpha_{k+1}) \rangle}{\langle \nabla Q(\alpha_k), \nabla Q(\alpha_k) \rangle}.$$

5. Podstaw $k = k + 1$ i zbadaj, czy $k > m$. Jeśli tak, to $k = 1$, $\nabla Q(\alpha_k) = \nabla Q(\alpha_{m+1})$,

$\alpha_k = \alpha_{m+1}$ i wykonaj etap 1, w przeciwnym wypadku wykonaj etap 2.

Metoda zmiennej metryki

Również w tej metodzie gradient wyznacza kierunek poszukiwań wg wzoru $d_k = -V_k \nabla Q(\alpha_k)$, gdzie V_k jest aproksymacją macierzy A^{-1} (patrz wzór (17)). Przyjmuje się, że V_0 jest macierzą jednostkową.

Algorytm obliczeń

Dany jest dowolny punkt startowy α_0 oraz dokładność ε .

Etap wstępny: podstaw $k = 1$, $\alpha_k = \alpha_0$, $V_k = V_0$ i oblicz gradient wskaźnika jakości $\nabla Q(\alpha_k)$.

Wykonaj dla każdej iteracji:

1. Przyjmij kierunek poszukiwań $d_k = -V_k \nabla Q(\alpha_k)$.

2. Zbadaj, czy zostało osiągnięte minimum, czyli czy $\langle \nabla Q(\alpha_k), \nabla Q(\alpha_k) \rangle < \varepsilon$. Jeśli tak, to zakończ obliczenia, a jeśli nie, to wykonaj etap 3.

3. Wykonaj minimalizację w kierunku d_k , czyli oblicz:

$$\min_{\tau \in \mathbb{R}} Q(\alpha_k + \tau d_k) = Q(\alpha_{k+1}).$$

4. Oblicz $\nabla Q(\alpha_{k+1})$ i wyznacz V_{k+1}

$$Y_{k+1} = Y_k + \frac{S_k \langle S_k \rangle - Y_k Y_k \langle Y_k Y_k \rangle}{\langle S_k, Y_k \rangle - \langle Y_k, Y_k Y_k \rangle}$$

gdzie $Y_k = \nabla Q(\alpha_{k+1}) - \nabla Q(\alpha_k)$, $S_k = \alpha_{k+1} - \alpha_k$, symbol \langle, \rangle oznacza iloczyn skalarny, a symbol $\langle \rangle$ iloczyn zewnętrzny [3].

5. Podstaw $k = k + 1$ i zbadaj czy $k > m$. Jeśli tak to podstaw $k = 1$, $Y_k = Y_0$, $\alpha_k = \alpha_{m+1}$ i wykonaj etap 1, w przeciwnym wypadku wykonaj tylko etap 1.

3.3. Metody optymalizacji z ograniczeniami

3.3.1. Metoda funkcji kary

By móc wykorzystać metody optymalizacji bez ograniczeń do optymalizacji z ograniczeniami, należy odpowiednio przekształcić wskaźnik jakości. Można wykonać tę operację z zastosowaniem funkcji kary. Wskaźnik jakości $Q(\alpha)$ przekształca się do postaci

$$Q_z(\alpha) = Q(\alpha) + \gamma P(\alpha),$$

gdzie $Q_z(\alpha)$ jest zmodyfikowanym wskaźnikiem jakości, $P(\alpha)$ funkcją kary, γ współczynnikiem funkcji kary.

Problem

$$\min_{\alpha \in A} Q(\alpha) = Q(\hat{\alpha}) \tag{18}$$

sprowadza się do znajdowania

$$\min_{\alpha \in R^n} \{ Q_z(\alpha) = Q_z(\hat{\alpha}) \} \tag{19}$$

Fukcja kary powinna spełniać następujące warunki:

$$P(\alpha) = 0 \quad \text{dla } \alpha \in A$$

$$P(\alpha) > 0 \quad \text{dla } \alpha \notin A .$$

4. Implementacja algorytmów w pakiecie DarP

Zbudowanie modelu układu dynamicznego z wykorzystaniem pakietu DarP [1] sprowadza się do określenia nazw użytkownika: zmiennej niezależnej, zmiennych opisu modelu i parametrów modelu oraz zapisu wyrażeń określających wartości pochodnych zmiennych stanu i wartości zmiennych wyjściowych w przeznaczonych do tego celu procedurach. Wprowadzone nazwy użytkownika są odczytywane podczas inicjalizacji programu (modelu) i udostępniane w systemie menu w celu zwiększenia czytelności modelu.

Rozszerzenie możliwości pakietu DarP o procedury optymalizacji wymagało, oprócz przygotowania samych procedur, ustalenia:

- niezbędnych obiektów (typów i zmiennych),
- zasad obliczania wskaźnika jakości, jego gradientu i sposobu przekazywania optymalizowanych parametrów między procedurą optymalizacji a modelem,
- zasad zapisu warunków ograniczających (definiujących) obszar dopuszczalnych wartości parametrów,
- zbioru wyznaczanych parametrów optymalizacji,
- zasad inicjalizacji procedur optymalizacji z poziomu menu.

Elementami ustalającymi stan początkowy i sposób działania procedur optymalizacji są zmienne globalne i parametry procedur. Najważniejsze z tych elementów to:

- a) początkowe składowe wektora parametrów $\underline{\alpha}$ (tablica `_A0`),
- b) liczba składowych wektora $\underline{\alpha}$ (zmienna `_NA`), określająca wymiar zadania optymalizacji,
- c) dokładność (tolerancja) wyznaczania rozwiązania optymalnego (zmienna `_AEps`),
- d) nazwa zmiennej przechowującej nazwę procedury funkcyjnej do wyznaczania wartości wskaźnika jakości (zmienna `QFunc`),
- e) maksymalna dopuszczalna liczba iteracji (zmienna `_MaxIter`),
- f) tablica wektorów początkowych kierunków poszukiwań (tablica `_AD0`),
- g) flaga typu zadania (zmienna `_FRestr`),
- h) parametr *beta* określający początkowy krok poszukiwań w metodzie Hooke-Jeevesa bądź długości ramion simpleksu w metodzie Nelder-Meada (zmienna `_ABeta`),
- i) parametr *delta* określający przyrost stosowany przy wyznaczaniu gradientu metodą ilorazów różnicowych (zmienna `_ADelta`),
- j) nazwa zmiennej przechowującej nazwę procedury funkcyjnej do wyznaczania gradientu wskaźnika jakości (zmienna `DQFunc`),
- k) nazwa zmiennej przechowującej nazwę procedury funkcyjnej do wyznaczania wartości funkcji kary (zmienna `PQFunc`),

l) współczynnik wagowy funkcji kary w zastępczym wskaźniku jakości (zmienna `_AGamma`).

Elementy od a) do g) są wykorzystywane we wszystkich procedurach, niezależnie od typu zadania optymalizacji. Elementy k) i l) stosowane są we wszystkich procedurach dla zadań optymalizacji z ograniczeniami. Element j) występuje w procedurach, w których kierunek poszukiwań wyznaczany jest na podstawie gradientu, przy czym użytkownik nie korzysta ze standardowej procedury oferowanej w pakiecie.

Wynikiem optymalizacji są:

- kod określający przyczynę zakończenia procedury (znaleziono minimum, wykonano maksymalnie dopuszczalną liczbę iteracji, wystąpił błąd itp.),
- optymalne wartości wektora parametrów α ,
- optymalna wartość wskaźnika jakości,
- liczba wykonanych iteracji.

4.1. Wyznaczanie wskaźnika jakości

Obligatoryjnym elementem przygotowywanym przez użytkownika podczas budowy modelu jest procedura funkcyjna wyznaczająca wartość wskaźnika jakości dla aktualnych wartości składowych wektora parametrów α . Nazwa tej procedury przypisywana jest w opisie badań modelowych zmiennej systemowej `QFunc`. Przykładowy nagłówek takiej procedury funkcyjnej i wykaz wykonywanych operacji przedstawiono poniżej

```
Function Wsk_Jak( AA: _TabValueA): otyp;
Begin
  {Przekazywanie parametrów optymalizacji (wektor AA) parametrom modelu
  i/lub warunkom początkowym zmiennych stanu}
  SetParam('<parametr>', aa[i]); {Ustawienie wartości parametru modelu}
  SetIC('<zmienna>', aa[i]); {Ustawienie początk. wartości zmiennej stanu}
  Run; {Wykonanie eksperymentu}
  Wsk_Jak := <wyrażenie określające wartość wskaźnika jakości>;
End;
```

W przedstawionym przykładzie występują typy:

`otyp` – typ zmiennoprzecinkowy (Real lub Double) wykorzystywany w pakiecie,

`_TabValueA` – typ tablicy przechowującej składowe wektora parametrów.

W opisie badań modelowych należy dodatkowo umieścić instrukcję przypisania

```
QFunc := Wsk_Jak;
```

4.2. Wyznaczanie gradientu wskaźnika jakości

Procedury optymalizacji wykorzystujące gradient wskaźnika jakości wywołują dla jego wyznaczenia procedurę funkcyjną o nazwie (wskaźniku) przechowywanej w zmiennej globalnej DQFunc. Standardowo zmienna ta przechowuje wskaźnik do systemowej procedury GradFunc wyznaczającej składowe gradientu jako ilorazy różnicowe

$$\frac{\partial Q(\underline{\alpha}_0)}{\partial \alpha_i} = \frac{Q(\underline{\alpha}_0 + \delta \underline{e}_i) - Q(\underline{\alpha}_0)}{\delta}, \quad (20)$$

gdzie:

- $\underline{\alpha}_0$ – wektor parametrów w punkcie wyznaczania gradientu,
- α_i – i-ta składowa wektora parametrów,
- \underline{e}_i – wektor jednostkowy z jedynką na i-tej pozycji,
- δ – przyrost (wartość zmiennej globalnej _ADelta).

Użytkownik może zdefiniować własną procedurę wyznaczania składowych gradientu, wykorzystując na przykład współczynniki wrażliwości. Nazwę tej procedury należy przypisać zmiennej globalnej DQFunc. Nagłówek takiej procedury ma postać:

```
Function Gradient( AA: _TabValueA; Var GradWskJak: _TabValueA ): otyp;
```

W opisie badań modelowych należy dodatkowo umieścić instrukcję przypisania

```
DQFunc := Gradient;
```

4.3. Definiowanie ograniczeń

Zadania optymalizacji z ograniczeniami mogą być rozwiązywane przez zastosowanie funkcji kary. W tym przypadku w procedurach optymalizacji wyliczana jest wartość zmodyfikowanego wskaźnika jakości

$$Q_z(\underline{\alpha}) = Q(\underline{\alpha}) + \gamma P(\underline{\alpha}), \quad (21)$$

gdzie:

- $Q_z(\underline{\alpha})$ – zmodyfikowany wskaźnik jakości,

$P(\underline{\alpha})$ – funkcja kary,

γ – współczynnik wagowy funkcji kary.

Funkcja kary określona jest ogólnie wyrażeniem

$$P(\underline{\alpha}) = \begin{cases} 0 & \text{dla } \underline{\alpha} \in A \\ P(\underline{\alpha}) > 0 & \text{dla } \underline{\alpha} \notin A \end{cases} \quad (22)$$

Zastosowanie takiego podejścia wymaga od użytkownika przygotowania procedury funkcyjnej wyznaczającej wartość funkcji kary. Nagłówek takiej procedury ma postać:

```
Function Funkcja_kary( AA: _TabValueA ): otyp;
```

W opisie badań modelowych należy dodatkowo umieścić instrukcję przypisania

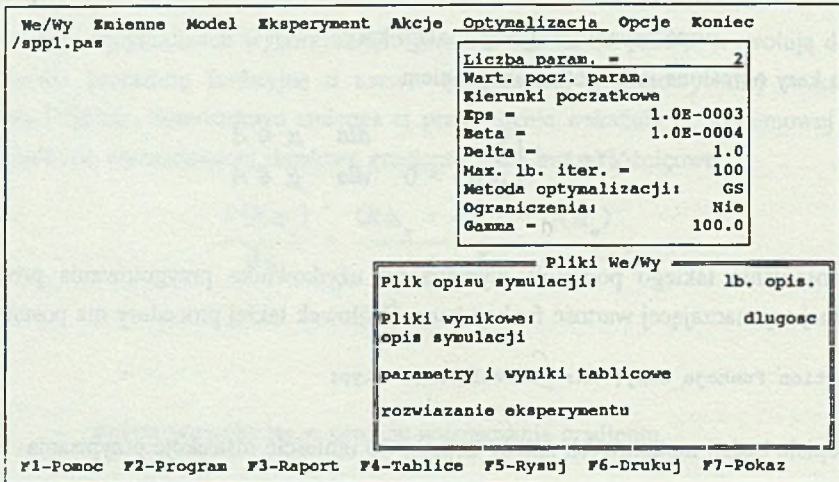
```
PQFunc := Funkcja_kary;
```

4.4. Inicjalizacja procedur optymalizacji

Rozpoczęcie optymalizacji następuje po wywołaniu procedury **RunOpt** wprost przez umieszczenie nazwy procedury w części głównej programu lub przez wybór pozycji menu w trybie pracy interakcyjnej. W tym drugim przypadku wystarczy w opisie badań modelowych umieścić instrukcje przypisujące zmiennym globalnym nazwy procedur zawierających opis modelu i nazwę procedury obliczania wskaźnika jakości oraz umieścić wywołanie procedury komunikacji z użytkownikiem. Przykładowy opis badań modelowych (część główna programu) przedstawiono poniżej.

```
Begin { **** Opis badań modelowych **** }
  Diffeq := R_Stan; { nazwa procedury zapisu równań stanu }
  Outeq := R_Wyjscia; { nazwa procedury zapisu równań wyjścia }
  QFunc := Wsk_Jak; { nazwa procedury obliczania wskaźnika jakości }
  [DQFunc := Gradient; { nazwa procedury oblicz. gradientu wskaźnika jakości }]
  [PQFunc := Kara; { nazwa procedury obliczania funkcji kary }]
  Interact; { wywołanie procedury komunikacji z użytkownikiem }
End.
```

W trybie pracy interakcyjnej, po aktualizacji parametrów modelu, parametrów eksperymentu i parametrów procedur optymalizacji, można zainicjować proces optymalizacji przez wybór pozycji **Optymalizacja** dostępnej po wybraniu pozycji **Akcje** z poziomu menu podstawowego. Dla ustawiania parametrów procedur optymalizacji przewidziano w menu podstawowym pozycję **Optymalizacja** (rys. 1).

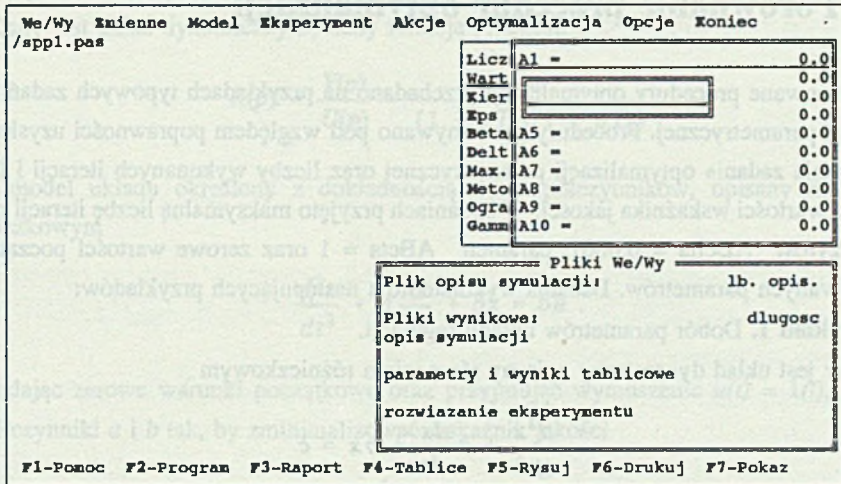


Rys. 1. Ekran z oknem ustawiania parametrów optymalizacji

Fig. 1. The screen with the window to setting the optimization parameters

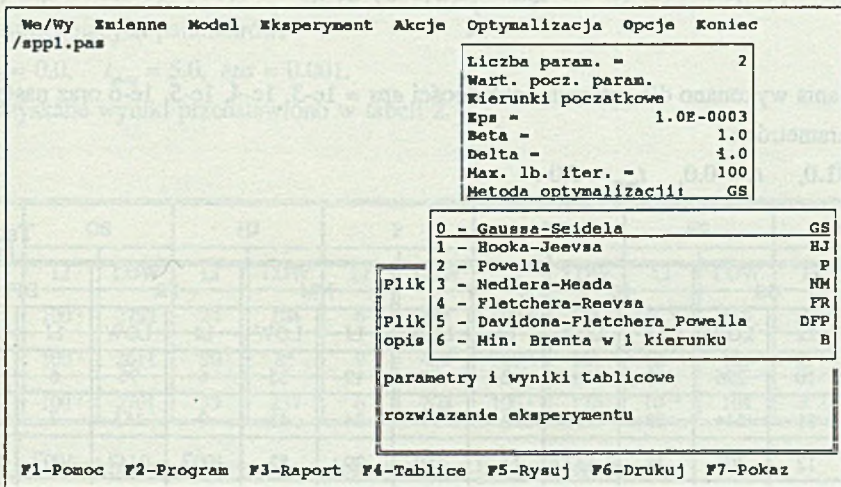
Poszczególne pozycje tego okna umożliwiają:

- Liczba parametrów** – ustawienie liczby parametrów optymalizacji,
- Wart. pocz. param.** – ustawianie początkowych wartości składowych wektora parametrów α (rys. 2),
- Kierunki początkowe** – wybór wektora kierunków i ustawianie wartości początkowych składowych wektora kierunku,
- Eps** – ustawienie dokładności wyznaczania rozwiązania optymalnego,
- Beta** – ustawienie początkowego kroku poszukiwań bądź długości ramion simpleksu,
- Delta** – ustawienie przyrostu stosowanego przy wyznaczaniu gradientu,
- Max. lb. iter.** – ustawienie maksymalnej dopuszczalnej liczby iteracji,
- Metoda optymalizacji** – wybór metody optymalizacji (rys. 3),
- Ograniczenia** – ustawienie flagi typu zadania (bez ograniczeń, z ograniczeniami),
- Gamma** – współczynnik wagowy funkcji kary.



Rys. 2. Ekran z oknem ustawiania początkowych wartości składowych wektora parametrów optymalizacji

Fig. 2. The screen with the window to setting the initial values of the optimization vector parameters



Rys. 3. Ekran z oknem wyboru metody optymalizacji

Fig. 3. The screen with the window of the method optimization selection

5. Porównanie procedur optymalizacji

Opracowane procedury optymalizacji przebadano na przykładach typowych zadań optymalizacji parametrycznej. Procedury porównywano pod względem poprawności uzyskanego rozwiązania zadania optymalizacji parametrycznej oraz liczby wykonanych iteracji i liczby obliczeń wartości wskaźnika jakości. W badaniach przyjęto maksymalną liczbę iteracji równą 100, przyrost $_ADelta = 0.0001$, parametr $_ABeta = 1$ oraz zerowe wartości początkowe poszukiwanych parametrów. Badania wykonano dla następujących przykładów:

Przykład 1. Dobór parametrów układu regulacji.

Dany jest układ dynamiczny opisany równaniem różniczkowym

$$\frac{d^2x}{dt^2} + a \frac{dx}{dt} + bx = c \quad (23)$$

z zerowymi warunkami początkowymi. Należy dobrać współczynniki a i b tak, by zminimalizować wskaźnik jakości

$$Q = \int_{t_0}^{t_{max}} (x - c)^2 dt \quad (24)$$

Badania wykonano dla czterech dokładności $eps = 1e-3, 1e-4, 1e-5, 1e-6$ oraz następujących parametrów:

$$c = 1.0, \quad t_0 = 0.0, \quad t_{max} = 5.0.$$

Tabela 1

Eps	GS		HJ		P		NM		FR		DFP	
	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW
1e-3	10	296	23	91	2	93	19	35	6	96	6	93
1e-4	11	344	29	115	2	97	24	43	9	153	7	118
1e-5	12	403	35	137	3	159	29	52	100 ^{*)}	2447	100 ^{*)}	2743
1e-6	12	416	41	158	3	170	31	56	100 ^{*)}	2421	100 ^{*)}	2820

^{*)} wykonano maksymalnie dopuszczalną liczbę iteracji

Uzyskane wyniki przedstawiono w tabeli 1. W tabelach prezentujących wyniki badań poszczególne algorytmy optymalizacji oznaczono skrótami literowymi, tak jak w oknie wyboru metody optymalizacji przedstawionym na rys. 3.

Przykład 2. Strojenie modelu.

Dany jest układ dynamiczny opisany funkcją przejścia

$$K(p) = \frac{Y(p)}{U(p)} = \frac{1}{(1 + pT_1)(1 + pT_2)} \quad (25)$$

oraz model układu określony z dokładnością do współczynników, opisany równaniem różniczkowym

$$\frac{d^2x}{dt^2} + a \frac{dx}{dt} + bx = bu . \quad (26)$$

Zakładając zerowe warunki początkowe oraz przyjmując wymuszenie $u(t) = 1(t)$, określić współczynniki a i b tak, by zminimalizować wskaźnik jakości

$$Q = \int_{t_0}^{t_{max}} (y - x)^2 dt . \quad (27)$$

Badania wykonano dla trzech układów o stałych czasowych:

- a) $T_1 = 0.5, T_2 = 0.5,$ b) $T_1 = 0.2, T_2 = 0.5,$ c) $T_1 = 0.1, T_2 = 1.0$

oraz następujących parametrów:

$$t_0 = 0.0, \quad t_{max} = 5.0, \quad eps = 0.001.$$

Uzyskane wyniki przedstawiono w tabeli 2.

Tabela 2

Układ	GS		HJ		P		NM		FR		DFP	
	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW
a)	100 ^{*)}	2782	24	104	8	341	100 ^{*)}	186	9 ^{*)}	145	10 ^{**)}	158
b)	100 ^{*)}	2611	20	85	9	453	100 ^{*)}	180	10 ^{**)}	155	10 ^{**)}	158
c)	100 ^{*)}	2701	53	237	6	208	100 ^{*)}	176	10 ^{**)}	164	8 ^{**)}	129

^{*)} wykonano maksymalnie dopuszczalną liczbę iteracji,
^{**)} obliczenia wykonano z dokładnością $eps = 0.0001$.

Przykład 3. Zadanie dwugraniczne.

Dany jest układ dynamiczny opisany układem równań stanu

$$\begin{aligned} \frac{dx_1}{dt} &= ax_1 + x_2 \\ \frac{dx_2}{dt} &= -2x_1 - bx_2 + c \end{aligned} \quad (28)$$

z zadanym warunkiem końcowym $x_1(t_{max})=x_2(t_{max})=1.0$. Korzystając z metody optymalizacji (najprostsze rozwiązanie uzyskuje się zmieniając kierunek całkowania) określić takie warunki początkowe, by spełnione zostały warunki końcowe.

Badania wykonano dla dwóch układów różniących się wartością parametru c :

- a) $c = 0.0$, b) $c = 1.0$

oraz następujących parametrów:

$$a = 0.5, \quad b = -1.5, \quad t_0 = 0.0, \quad t_{max} = 1.0, \quad eps = 0.001.$$

Uzyskane wyniki przedstawiono w tabeli 3.

Tabela 3

Układ	GS		HJ		P		NM		FR		DFP	
	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW
a)	33	799	18	72	4	155	100 ⁾	182	100 ⁾	1504	11	169
b)	31	757	21 ^{**)}	95	5	193	100 ⁾	187	100 ⁾	1511	12	191

- ⁾ wykonano maksymalnie dopuszczalną liczbę iteracji,
^{**)} wymagana była zmiana punktu startowego.

Przykład 4. Zadanie doboru wartości własnych

Rozpatrywane jest zadanie przesuwania masy m po płaskiej powierzchni (bez tarcia) z położenia początkowego $x(0)$, $\dot{x}(0)$ do początku układu współrzędnych, przy użyciu ograniczonej siły f ($-f_{max} \leq f \leq f_{max}$), w zadanym czasie t_{max} , minimalizując wydatkowaną energię [1].

Tak sformułowane zadanie da się sprowadzić do zadania doboru wartości własnych, opisanego układem równań stanu

$$\begin{aligned}
 p_2(t) &= a - bt \\
 \frac{dx_1}{dt} &= x_2 \\
 \frac{dx_2}{dt} &= \frac{g(p_2(t))}{m}
 \end{aligned} \tag{29}$$

z zadanymi warunkami początkowymi i końcowymi ($x=x_1, \dot{x}=x_2$), gdzie:

$$g(p_2) = \begin{cases} -f_{\max} & \text{gdy} & \frac{p_2}{m} \leq -f_{\max} \\ \frac{p_2}{m} & \text{gdy} & -f_{\max} < \frac{p_2}{m} < f_{\max} \\ f_{\max} & \text{gdy} & \frac{p_2}{m} \geq f_{\max} \end{cases} \tag{30}$$

Zadanie sprowadza się do wyznaczenia dla zadanych warunków początkowych takich wartości parametrów a i b , by spełnione były warunki końcowe.

Badania wykonano dla dwóch położzeń początkowych:

- a) $x_1(0) = -5.0, x_2(0) = 0.0,$ b) $x_1(0) = -5.0, x_2(0) = 5.0$

oraz następujących parametrów:

$$m = 1.0, \quad f_{\max} = 5.0, \quad t_0 = 0, \quad t_{\max} = 3.0, \quad eps = 0.001 .$$

Uzyskane wyniki przedstawiono w tabeli 4.

Tabela 4

Położenie	GS		HJ		P		NM		FR		DFP	
	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW	LI	LOW
a)	100 ⁾	2179	34	144	2	68	100 ⁾	183	100 ⁾	1370	15	214
b)	100 ⁾	2180	34	146	3	118	100 ⁾	187	100 ⁾	1312	51	213

⁾ wykonano maksymalnie dopuszczalną liczbę iteracji.

6. Wnioski końcowe

Przeprowadzone badania ilustrują przydatność pakietu wyposażonego w zestaw procedur do rozwiązywania zadań optymalizacji parametrycznej. Porównując uzyskane wyniki badań należy stwierdzić, że algorytm GS ma raczej znaczenie poglądowe niż praktyczne. Algorytm ten jest najwolniej zbieżny i jedynie w szczególnych przypadkach, gdy powierzchnie stałej wartości wskaźnika jakości są elipsoidami o osiach głównych zgodnych z osiami układu współrzędnych, algorytm ten jest efektywny. Najmniej wartościowa wskaźnika jakości wymagały metody bezgradientowe: HJ, NM, i P. Liczba wykonanych iteracji dla metod HJ i NM zależy od wartości parametru $_ADelta$. Algorytm HJ ma tylko wbudowany mechanizm zmniejszania wartości parametru $_ADelta$, stąd bezpieczniejsze jest przyjęcie większej wartości tego parametru, niż z góry ograniczanie szybkości zbieżności metody przez przyjęcie zbyt małej wartości początkowej tego parametru. Dla metody NM początkowa wartość parametru $_ADelta$ nie jest tak istotna dzięki zastosowanemu mechanizmowi (operacji) ekspansji. Metoda HJ wymagała w jednym przypadku zmiany punktu startowego lub zmniejszenia parametru $_ADelta$. Istotny wpływ na działanie metody HJ może mieć również wybór kierunków poszukiwań. W przeprowadzanych badaniach były to osie układu współrzędnych przyjmowane jako kierunki domyślne.

Z przeprowadzonych badań wynika, że metody gradientowe FR i DFP są bardzo czułe na zmniejszanie parametru określającego dokładności poszukiwań (parametr $_AEps$), co ilustrują wyniki przykładu 1. Takie zachowanie się tych metod wynika z przyjętego kryterium zakończenia poszukiwania wartości minimalnej i z właściwości stosowanej metody całkowania (metoda całkowania o zmiennym kroku). W metodach gradientowych przyjęto, że minimum jest osiągnięte, gdy norma gradientu jest mniejsza od zadanej tolerancji. Z drugiej strony stosowanie metody całkowania o zmiennym kroku oznacza, że nawet niewielka zmiana wartości parametrów modelu może oznaczać inny ciąg kroków całkowania w ramach eksperymentu, a to może być przyczyną stosunkowo dużej (w granicach przyjętej dokładności całkowania) zmiany wartości wskaźnika jakości i jego gradientu. Sytuacja taka jest szczególnie krytyczna, gdy przyjmie się dużą dokładność rozwiązania zadania optymalizacji, przyjmując równocześnie małą dokładność całkowania.

Pewnym utrudnieniem w interpretacji uzyskanych wyników badań są różne kryteria zakończenia poszukiwania minimum, stosowane w poszczególnych metodach. Ujednoczenie tych kryteriów nie wydaje się jednak celowe, gdyż poszczególne algorytmy oparte są na całkowicie różnych zasadach, a ponadto trudno wskazać uniwersalne kryterium zakończenia poszukiwania. Rozwiązaniem bezpiecznym jest tu ograniczenie maksymalnej dopuszczalnej liczby iteracji.

Wyniki przeprowadzonych badań potwierdzają również fakt, że nie można wskazać uniwersalnego algorytmu optymalizacji. Możliwość wyboru algorytmu dla rozwiązania określonego zadania optymalizacji jest więc cechą pożądaną.

LITERATURA

- [1] Skowronek M.: Modelowanie cyfrowe układów dynamicznych. Wyd. II, Skrypt Pol. Śl. nr 1776, Gliwice 1993.
- [2] Bekey G. A., Karplus W. J.: Obliczenia hybrydowe. WNT, Warszawa 1976.
- [3] Findeisen W., Szymanowski J., Wierzbicki A.: Teoria i metody obliczeniowe optymalizacji. PWN, Warszawa 1977.
- [4] Kręglewski T. i inni: Metody optymalizacji w języku Fortran. PWN, Warszawa 1984.

Recenzent: Dr hab. inż. Stanisław Wołek

Wpłynęło do Redakcji 20 lutego 1994 r.

Abstract

The paper presents the optimization procedures packet connected with the DarP packet.

First, the parameter optimization problem is defined. Then some typical parameter optimization examples are presented. Seven procedures for calculating the unconstrained optimization and constrained optimization problems are described. An algorithm for constrained optimization bases on the penalty function concept.

The second part of the paper is devoted to illustrative applications of parameter optimisation by using DarP packet. Then follows a brief discussion of quality factor, its gradient, definitions of constraints and the starting point of optimization. Some of the interactive approaches for optimisation are presented. Finally, the computational efficiency for presented methods is discussed.