



POLSKA AKADEMIA NAUK
Instytut Badań Systemowych

BADANIA SYSTEMOWE
Inżynieria Środowiska

**KOMPUTEROWA SYMULACJA
I OPTIMALIZACJA MODELU
OCZYSZCZALNI ŚCIEKÓW**

Marcin Stachura

Warszawa 2008



**KOMPUTEROWA SYMULACJA
I OPTIMALIZACJA MODELU
OCZYSZCZALNI ŚCIEKÓW**

**POLSKA AKADEMIA NAUK
INSTYTUT BADAŃ SYSTEMOWYCH**

Seria: BADANIA SYSTEMOWE, tom 59

Redaktor naukowy: prof. Jakub Gutenbaum

Podseria: Inżynieria Środowiska

Warszawa 2008

**KOMPUTEROWA SYMULACJA
I OPTYMALIZACJA MODELU
OCZYSZCZALNI ŚCIEKÓW**

Marcin Stachura

Publikacja wydana ze środków projektu badawczego MINISTERSTWA NAUKI i SZKOLNICTWA WYŻSZEGO: nr R11 001 01.

W pracy omówiono sposób konstruowania modelu matematycznego dla oczyszczalni ścieków z osadem czynnym z wykorzystaniem bilansowych równań różniczkowych zwyczajnych, wynikających z zasad zachowania masy i podstawowych zależności kinetycznych i stechiometrycznych zachodzących przemian i procesów fizycznych w obiektach technicznych oczyszczalni. Równania różniczkowe opisują dynamikę procesu a występujące w równaniach współczynniki mają interpretację fizyczną. Koncepcja przedstawionego sposobu konstrukcji modelu matematycznego polega na opracowaniu modelu mogącego być pożytecznym narzędziem wspomagającym pracę operatora procesu technologicznego. Wobec tego opracowywany model opisuje konkretną i ograniczoną grupę obiektów a proces modelowania uwzględnia również kalibrację modelu na podstawie rzeczywistych pomiarów. Dzięki takiemu podejściu utworzony model matematyczny staje się przybliżeniem konkretnego obiektu i może być użyty do jego badania, co jest niewątpliwie celem nadrzędnym modelowania matematycznego. Praca ma również na celu prezentację techniki */fast--prototyping/*, czyli szybkiego prototypowania przy pomocy komputera wielowymiarowych procesów przemysłowych na przykładzie procesów zachodzących w mechaniczno--biologicznych oczyszczalniach ścieków. Pod pojęciem modelowania w pracy rozumie się zespół czynności obejmujących takie zagadnienia, jak: opracowanie modelu procesu w postaci układu równań różniczkowych (model fizyczny), implementację modelu w odpowiednim algorytmie komputerowym, kalibrację wraz z optymalizacją nieznanymi współczynników występujących w równaniach opisujących proces oraz analizę otrzymanych wyników.

Recenzenci:

Prof. dr hab. inż. Krzysztof Janiszowski
Dr hab. inż. Janusz Łomotowski

Semi
Bibl. podręczna

45905

Komputerowa edycja tekstu: Anna Gostyńska

© Instytut Badań Systemowych PAN, Warszawa 2008

Instytut Badań Systemowych PAN
Newelska 6, PL 01-447 Warsaw

Sekcja Informacji Naukowej i Wydawnictw IBS PAN
e-mail: biblioteka@ibspan.waw.pl

ISBN 978-83-89475-15-2

ISSN 0208-8029

Druk i oprawa: ARTPRESS, tel. 052 354 95 10

7. Optymalizacja modelu

7.1. Zadanie optymalizacji

Zadanie optymalizacji w omawianym przypadku będzie rozumiane jako efekt automatycznego wyznaczenia nieznanymi współczynników występujących w równaniach opisujących model procesu w sensie osiągnięcia minimum pewnego wskaźnika (Luenberger, 1974):

$$f(\hat{\mathbf{x}}) = \min_{\mathbf{x} \in R^n} f(\mathbf{x}) \quad (7.1)$$

gdzie:

f – funkcja celu,

\mathbf{x} – zmienna decyzyjna w optymalizacji statycznej,

$\hat{\mathbf{x}}$ – rozwiązanie optymalne,

$f: R^n \rightarrow R$, przy czym zakłada się, że funkcja f jest ograniczona od dołu.

W optymalizacji modelu wykorzystywane są algorytmy definiowane jako proces iteracyjny składający się z ciągu odwzorowań punktu w zbiór:

$$A: V \rightarrow V \quad (7.2)$$

rozumiany jako przyporządkowanie każdemu punktowi $\mathbf{x} \in V$ zbioru $A(\mathbf{x}) \subset V$.

Mając punkt \mathbf{x}^1 , ciąg punktów $\{\mathbf{x}^k\}_{k=1}^{\infty}$ generujemy rekurencyjnie w myśl formuły:

$$\mathbf{x}^{k+1} \in A(\mathbf{x}^k) \quad (7.3)$$

tzn. dowolny punkt należący do zbioru $A(\mathbf{x}^k)$ może być wybrany jako punkt następny \mathbf{x}^{k+1} .



7.2. Opis wybranych algorytmów optymalizacji

7.2.1. Podział i wybór metod optymalizacji do zastosowania w omawianym zadaniu

Metody numeryczne poszukiwań minimum funkcji (bez ograniczeń) można zasadniczo podzielić na dwie grupy:

- metody przypadkowe,
- metody zdeterminowane.

W omawianym zagadnieniu optymalizacji modelu mechaniczno-biologicznego oczyszczania ścieków najbardziej przydatne były metody należące do drugiej grupy, przeszukujące przestrzeń R^n wzdłuż prostych, zwanych dalej kierunkami poszukiwań. Metody te różnią się między sobą i ze względu na podstawowe cechy algorytmu można podzielić je, rozważając:

- sposób tworzenia kierunków poszukiwań,
- sposób znajdowania kolejnego punktu wzdłuż danego kierunku poszukiwań.

Oczywiście, istnieją metody mieszane, które wykorzystują obydwa przedstawione powyżej mechanizmy. Ze względu na kryterium pierwsze można wyróżnić dwa typy algorytmów:

- metody o modyfikowanej bazie, zwane *bezgradientowymi*,
- metody o modyfikowanym kierunku, zwane *gradientowymi*.

Metody pierwszej grupy dokonują poszukiwań w n niezależnych kierunkach, po czym następuje modyfikacja bazy, która realizowana jest jedynie na podstawie informacji o zmianach wartości funkcji f . W metodach drugiej grupy korzysta się z dodatkowej informacji o wartości i zmianach gradientu w punktach będących rezultatem poszukiwania wzdłuż jednego kierunku.

Ze względu na przeprowadzoną kalibrację modelu można wnioskować, iż wektor wartości parametrów modelu procesu jest w okolicach punktu optymalnego. Wobec tego postanowiono w zadaniu optymalizacji porównać wybrane metody bezgradientowe oraz dwie wybrane metody gradientowe. Metody bezgradientowe są zasadniczo wolniejsze (tj. wymagają większej ilości obliczeń funkcji celu) od metod gradientowych, jednakże w okolicach punktu optymalnego różnice te nie powinny być bardzo duże.

Istnieje możliwość, iż znaleziony przy pomocy kalibracji punkt (uznawany za startowy dla wszystkich metod optymalizacji) nie będzie leżał w okolicach najlepszego punktu optymalnego danej funkcji. W celu przeszukania innych punktów

optymalnych postanowiono więc zastosować metodę typu przypadkowego *Metropolis*, która bazując na pewnych zmiennych losowych może znaleźć minimum globalne, nawet jeśli punkt startowy leży w okolicach pewnego minimum lokalnego.

Zastosowane metody należą do różnych grup, biorąc pod uwagę klasyfikację podaną powyżej. Ponieważ optymalizacja modelu wykonywana była w trybie *off-line*, to czas znalezienia optimum nie był pierwszorzędym ograniczeniem, jakim kierowano się przy doborze metod.

Można przypuszczać, że funkcja celu w zadaniu optymalizacji modelu procesu posiada wiele minimów lokalnych. Z tego względu istotne było zastosowanie kilku odmiennych algorytmów, które miałyby szanse znalezienia różnych punktów optymalnych z ich punktów widzenia (choć niektóre metody mogły być znacznie szybsze od pozostałych).

Jednocześnie ważnym zamierzeniem niniejszej pracy było zaproponowanie nieskomplikowanych metod, które będą mogły być zastosowane w normalnej praktyce eksploatacyjnej przez operatorów procesu. Z tego względu metody zostały tak dobrane, aby były w miarę proste, tzn. nie wymagały znacznej wiedzy z zakresu teorii optymalizacji oraz definiowania dużej liczby współczynników występujących w samych algorytmach.

Poniżej przeprowadzono krótki przegląd wybranych metod wraz z opisem podstawowych kroków algorytmu numerycznego dla każdej z nich. Przegląd taki był potrzebny, ponieważ porównywane algorytmy dostępne były w różnych źródłach literaturowych a dostęp do tej informacji jest ograniczony (biblioteki uniwersyteckie), zwłaszcza dla operatorów badanego procesu. Należy także pamiętać o specyfice zadania – nie wszystkie rozwiązania są technologicznie akceptowalne, zatem dobrze jest wyjaśnić sposób działania algorytmu.

7.2.2. Algorytm Hooka i Jeevesa - HJ

W metodzie tej w kolejnej iteracji występują dwa sposoby poruszania się: *próbny* oraz *roboczy*. Pierwszy sposób służy do zbadania lokalnego zachowania się funkcji w niewielkim obszarze, poprzez wykonanie kroków próbnych wzdłuż wszystkich kierunków ortogonalnej bazy. Sposób roboczy polega na przejściu w ściśle zdeterminowany sposób do następnego obszaru, w którym powtarzany jest pierwszy etap, lecz tylko w przypadku, gdy przynajmniej jeden z wykonanych kroków próbnych był pomyślny. W przeciwnym razie powraca się do poprzednio badanego obszaru, w którym fakt taki miał miejsce, i cykl poszukiwania rozpoczynany jest od nowa przy zmniejszonej długości kroku.

Informacje wejściowe

x^0 – dowolnie wybrany punkt startowy,

$[d^1, d^2, \dots, d^n]$ – baza wyjściowa utworzona z wzajemnie ortogonalnych wektorów,

- τ – poszukiwana długość kroku,
 α – współczynnik korekcyjny zwiększający krok; $\alpha > 1$,
 β – współczynnik korekcyjny zmniejszający krok; $0 < \beta < 1$,
 ε – wymagana dokładność obliczeń minimum,
 n – liczba zmiennych niezależnych,
 ε_{\min} – założona dokładność warunkująca zakończenie algorytmu,
 τ_{\min} – założona minimalna długość kroku.

Warunkiem zakończenia działania procedury jest spełnienie nierówności:

$$|f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| < \varepsilon \quad (7.4)$$

Algorytm obliczeń

Etap próbny

- (1) Podstaw $k=1$ oraz oblicz w punkcie \mathbf{x}^0 wartość funkcji $q_0 \Leftarrow f(\mathbf{x}^0)$.

- (2) Wzdłuż kierunku \mathbf{d}^k wykonaj krok próbny:

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \tau \mathbf{d}^k \quad (7.5)$$

oraz oblicz wartość funkcji w tym punkcie.

- (3) Zbadaj, czy krok był pomyślny, tzn. czy $q < q_0$. Jeżeli tak, to podstaw q w miejsce q_0 oraz przejdź do wykonywania punktu (6), natomiast jeśli nie, to (4).

- (4) Wykonaj krok próbny w przeciwnym kierunku

$$\mathbf{x}^k = \mathbf{x}^{k-1} - 2\tau \mathbf{d}^k \quad (7.6)$$

oraz oblicz wartość funkcji w tym nowym punkcie.

- (5) Zbadaj, czy ten krok był pomyślny. Jeżeli tak, to podstaw q w miejsce q_0 oraz przejdź do wykonania punktu (6), natomiast w przeciwnym razie pozostaw bieżący punkt bez zmian, tzn. podstaw \mathbf{x}^{k-1} w miejsce \mathbf{x}^k .

- (6) Zbadaj, czy wykonano kroki we wszystkich kierunkach ortogonalnej bazy, tzn. czy $k=n$. Jeżeli nie, to podstaw $k=k+1$ oraz powtórz czynności od punktu (2), natomiast jeżeli tak, to (7).

- (7) Zbadaj, czy w wykonanym cyklu poszukiwania wystąpiły kroki pomyślne, tzn. czy $f(\mathbf{x}_{B_0}) > f(\mathbf{x}^k)$, przy czym w pierwszej iteracji $\mathbf{x}_{B_0} = \mathbf{x}_0$. Jeśli tak, to podstaw \mathbf{x}_k w miejsce \mathbf{x}_B , który nazywany jest punktem bazowym, oraz przejdź do wykonania etapu roboczego, w przeciwnym razie (8).
- (8) O ile nie zostało spełnione kryterium na minimum (7.4), zbadaj, czy realizowana iteracja jest pierwszą iteracją. Jeśli tak, to zmień punkt startowy \mathbf{x}_0 i powtórz czynności od punktu (1), jeśli natomiast nie, to powrót do poprzednio poszukiwanego obszaru w myśl zasady $\mathbf{x}_0 = \mathbf{x}_{B_0}$, zmniejsz długość kroku współczynnikiem β , tzn. $\tau = \tau \beta$, oraz rozpocznij wykonanie procedury od punktu (1).

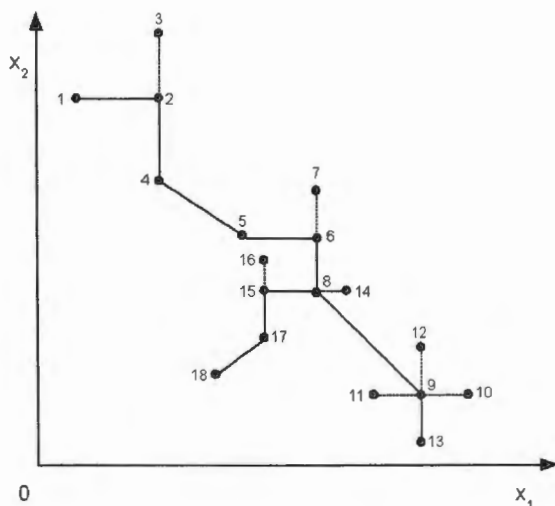
Etap roboczy

- (1) Wykonaj krok roboczy według reguły:

$$\mathbf{x}^0 = \mathbf{x}_B + (\mathbf{x}_B - \mathbf{x}_{B_0}) = 2\mathbf{x}_B - \mathbf{x}_{B_0} \quad (7.7)$$

- (2) Podstaw \mathbf{x}_B w miejsce \mathbf{x}_{B_0} oraz wróć do realizacji kroku próbnego.

Działanie omówionego algorytmu zostało przedstawione na przykładzie funkcji dwu zmiennych (rys. 7.1). Na rysunku tym poszczególne punkty \mathbf{x}^k oznaczono liczbami w kolejności, w jakiej zostały wyliczane. Tak więc, startując z punktu \mathbf{x}^1 , rozpoczyna się realizacja etapu próbnego, po zakończeniu którego otrzymano punkt \mathbf{x}^4 . Punkt ten przyjęty został jako bazowy, bowiem w czasie przebiegu tego etapu miały miejsce dwa pomyślne kroki próbne \mathbf{x}^2 i \mathbf{x}^4 oraz tylko jeden niepomyślny \mathbf{x}^3 , a przy tym $f(\mathbf{x}^4) < f(\mathbf{x}^1)$. Kolejno wykonany został etap roboczy, w rezultacie otrzymujemy punkt \mathbf{x}^5 . Z punktu tego rozpoczęto nowy cykl próbny kończący się znalezieniem nowego punktu bazowego \mathbf{x}^8 , przy czym $f(\mathbf{x}^8) < f(\mathbf{x}^4)$. Postępowanie kolejne przeprowadzone zostało w analogiczny sposób, dochodząc do sytuacji, gdy warunek (7) nie jest spełniony, bowiem $f(\mathbf{x}^{13}) > f(\mathbf{x}^8)$. Wobec tego powrócono do punktu bazowego \mathbf{x}^8 , zmniejszając długość kroku τ i cykl poszukiwań zostanie rozpoczęty od nowa. Jako kryterium zbieżności zakończenia działania procedury przyjęto warunek, że aktualna długość kroku τ będzie mniejsza od założonej liczby ε (7.4) (Findeisen, Szymanowski, Wierzbicki, 1977).



Rysunek 7.1. Przykład działania metody Hooka-Jeevesa

Należy zauważyć, że algorytm może zmniejszać krok lub go zwiększać, jeżeli przyrosty x następują w kilku osiach, ale tylko do wartości $\tau \sqrt{n}$.

7.2.3. Algorytm Rosenbrocka – R

W metodzie tej minimum również poszukiwane jest w n wzajemnie ortogonalnych kierunkach. W odróżnieniu do metody Hooka-Jeevesa kierunki bazowe nie pozostają stałe, lecz w szczególnych przypadkach w wyniku obrotu ulegają zmianom. Baza wyjściowa $\mathbf{d}_0^1, \mathbf{d}_0^2, \dots, \mathbf{d}_0^n$ tworzona jest zazwyczaj z wektorów układu współrzędnych kartezjańskich. Na wstępie w każdym z tych kierunków wykonywane są kolejno po jednym kroku o długości τ . Jeśli eksperyment taki kończy się powodzeniem, to w następnym kroku w danym kierunku wartość τ zwiększana jest α razy ($\alpha > 1$), natomiast w przeciwnym razie τ mnożona jest przez β , przy czym $0 < \beta < 1$. Tego rodzaju tryb postępowania powtarzany jest aż do momentu, gdy wykonanie kroku we wszystkich kierunkach daje niepomyślny wynik, to znaczy $f(\mathbf{x}^{k-1}) < f(\mathbf{x}^k)$.

W takiej sytuacji, jeśli jest spełnione kryterium na ekstremum, to procedura kończy swoje działanie, jeśli zaś nie, to wykonywany jest *algorytm obrotu współrzędnych* i działanie procedury rozpoczyna się od początku (Jastriebow, Wciślik, 2004).

Informacje wejściowe

\mathbf{x}^0 – dowolnie wybrany punkt startowy,

$\mathbf{D}_0 = [\mathbf{d}_0^1, \mathbf{d}_0^2, \dots, \mathbf{d}_0^n]$ – baza wyjściowa utworzona z wzajemnie ortogonalnych wektorów,

- τ – poszukiwana długość kroku,
 α – współczynnik korekcyjny zwiększający τ ; $\alpha > 1$;
 β – współczynnik korekcyjny zmniejszający τ ; $0 < \beta < 1$,
 ε – wymagana dokładność obliczeń minimum,
 n – liczba zmiennych niezależnych.

Algorytm obliczeń

- (1) Podstaw $s \leftarrow 0$.
- (2) Dla $k = 1, 2, \dots, n$ oblicz $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1} + \tau \mathbf{d}^k$ oraz podstaw $q \leftarrow f(\mathbf{x}^k)$. Jeżeli $q < q_0$, to zwiększ sumę pomyślnych kroków w k -tym kierunku: $s_k \leftarrow s_k + \tau_k$, oraz podstaw $q \leftarrow q_0$ i $\tau_k \leftarrow \alpha \tau_k$. W przeciwnym razie podstaw $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1} - \tau \mathbf{d}^k$ oraz $\tau_k \leftarrow -\beta \tau_k$.
- (3) Jeżeli przynajmniej w jednym kierunku \mathbf{d}_0^k wystąpił pomyślny krok, tzn. $f(\mathbf{x}^k) < f(\mathbf{x}^{k-1})$, to $\mathbf{x}^0 \leftarrow \mathbf{x}^k$ i powtórz (2). W przeciwnym razie (4).
- (4) Jeżeli $\mathbf{s} = \mathbf{0}$, to zmień punkt startowy \mathbf{x}^0 , podstaw $q \leftarrow f(\mathbf{x}^k)$ i powtórz (2). Jeżeli $\mathbf{s} \neq \mathbf{0}$ i nie zostało spełnione kryterium zbieżności, to (5).
- (5) Wykonaj algorytm obrotu ortogonalnego układu współrzędnych \mathbf{D}_0 i powróć do wykonania kroku (1), przy czym podstaw $\mathbf{x}^0 \leftarrow \mathbf{x}^k$.

Algorytm obrotu współrzędnych

- (1) Wyznacz układ wektorów:

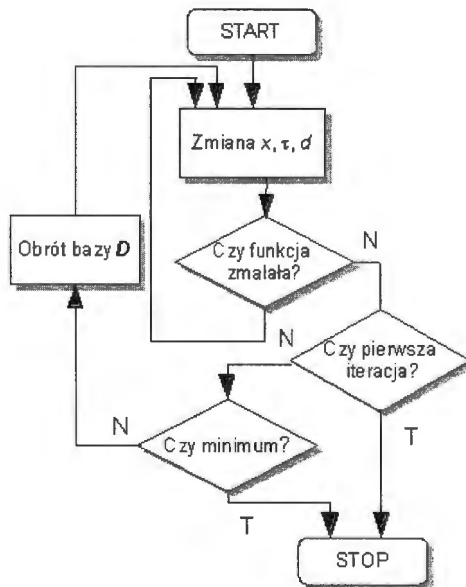
$$\begin{aligned}
 \mathbf{w}^1 &= s_1 \mathbf{d}_0^1 + s_2 \mathbf{d}_0^2 + \dots + s_n \mathbf{d}_0^n \\
 \mathbf{w}^2 &= \quad \quad s_2 \mathbf{d}_0^2 + \dots + s_n \mathbf{d}_0^n \\
 \mathbf{w}^n &= \quad \quad \quad \quad \quad s_n \mathbf{d}_0^n
 \end{aligned} \tag{7.8}$$

- (2) Dokonaj ortonormalizacji tego układu stosując procedurę *Gramma-Schmidta*:

$$\begin{aligned}
 \mathbf{v}^1 &= \mathbf{w}^1, \\
 \mathbf{d}^1 &= \frac{\mathbf{v}^1}{\|\mathbf{v}^1\|}, \\
 \mathbf{v}^2 &= \mathbf{w}^2 - \langle \mathbf{w}^2, \mathbf{d}^1 \rangle \mathbf{d}^1, \\
 \mathbf{d}^2 &= \frac{\mathbf{v}^2}{\|\mathbf{v}^2\|}, \\
 &\dots \\
 \mathbf{v}^n &= \mathbf{w}^n - \sum_{k=1}^{n-1} \langle \mathbf{w}^n, \mathbf{d}^k \rangle \mathbf{d}^k, \\
 \mathbf{d}^n &= \frac{\mathbf{v}^n}{\|\mathbf{v}^n\|},
 \end{aligned} \tag{7.9}$$

- (3) Podstaw znaleziony zbiór wektorów bazy $\mathbf{D} = [\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^n]$ w miejsce bazy \mathbf{D}_0 .

Sieć działań przedstawionego algorytmu przedstawiono na rysunku 7.2:



Rysunek 7.2. Schemat metody *Rosenbrocka*

7.2.4. Bezgradientowa metoda Powella (1) kierunków sprzężonych – P1

Metoda *Powella* należy do grupy metod poszukiwania metod na kierunku, więc na wstępnie zdefiniowana zostanie minimalizacja kierunkowa, polegająca na określeniu minimum na kierunku w myśl zasady:

$$\mathbf{x}_i^k \in T(\mathbf{x}_{i-1}^k, \mathbf{d}^i) \quad (7.10)$$

gdzie odwzorowanie T ma postać:

$$T(\mathbf{x}, \mathbf{d}) = \{y: y = \mathbf{x} + \hat{\alpha}\mathbf{d}, f(y) = \min f(\mathbf{x} + \alpha\mathbf{d})\} \quad (7.11)$$

W pierwszym etapie tej metody dokonywana jest minimalizacja wzdłuż n ortogonalnych kierunków $\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^n$. Następnie po zakończeniu tego obiegu wyznaczany jest nowy sprzężony kierunek \mathbf{d}^{n+1} , w myśl zasady:

$$\mathbf{d}^{n+1} = \frac{\mathbf{x}^n - \mathbf{x}^0}{\|\mathbf{x}^n - \mathbf{x}^0\|} \quad (7.12)$$

oraz nowy punkt startowy \mathbf{x}^0 w rezultacie minimalizacji funkcji wzdłuż kierunku \mathbf{d}^{n+1} . Z kolei dokonywana jest modyfikacja kierunków w ten sposób, że z bazy usunięty zostaje kierunek \mathbf{d}^1 a na jego miejsce zostaje włączony \mathbf{d}^{n+1} , przy jednoczesnej zmianie kolejności kierunków według reguły $\mathbf{d}^i \leftarrow \mathbf{d}^{i+1}$ dla $i = 1, \dots, n$. Opisany algorytm powtarzany jest tak długo, aż zostanie spełnione kryterium zbieżności procedury.

Informacje wejściowe

\mathbf{x}^0 – dowolnie wybrany punkt startowy,

$\mathbf{D}_0 = [\mathbf{d}_0^1, \mathbf{d}_0^2, \dots, \mathbf{d}_0^n]$ – baza wyjściowa utworzona z wzajemnie ortogonalnych wektorów,

ε – wymagana dokładność obliczeń minimum,

n – liczba zmiennych niezależnych.

Algorytm obliczeń. Etap wstępny: oblicz $\mathbf{x}^0 \in T(\mathbf{x}^n, \mathbf{d}^n)$.

Iteracja k -ta. Podstaw $k \leftarrow 1$ oraz $\mathbf{x}_0^k \leftarrow \mathbf{x}^0$.

(1) Dla $i = 1, 2, \dots, n$ oblicz kolejno:

$$\mathbf{x}_i^k \in T(\mathbf{x}_{i-1}^k, \mathbf{d}^i) \quad (7.13)$$

(2) Wyznacz kierunek sprzężony

$$\mathbf{d}^{n+1} = \frac{\mathbf{x}_n^k - \mathbf{x}_0^k}{\|\mathbf{x}_n^k - \mathbf{x}_0^k\|} \quad (7.14)$$

oraz oblicz $\mathbf{x}_{n+1}^0 \in T(\mathbf{x}^n, \mathbf{d}^{n+1})$.

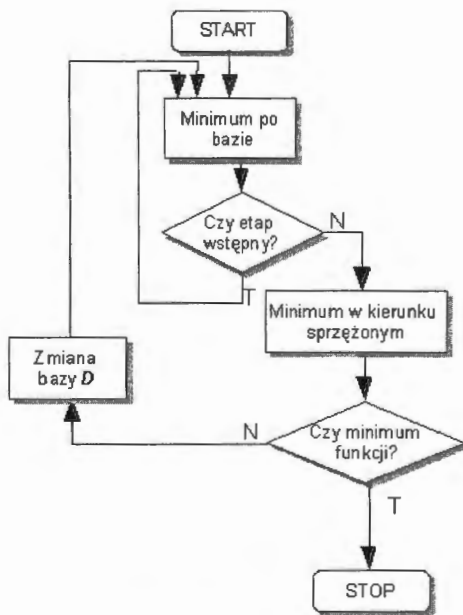
(3) Zbadaj, czy zostało spełnione kryterium zbieżności. Jeśli tak, to stop, jeśli nie, to podstaw $\mathbf{x}_0^k \leftarrow \mathbf{x}_{n+1}^k$.

(4) Dokonaj modyfikacji bazy

$$\mathbf{d}^i \leftarrow \mathbf{d}^{i+1}, \text{ dla } i=1, \dots, n, \quad (7.15)$$

(5) Podstaw $k \leftarrow k+1$ i powtórz (1).

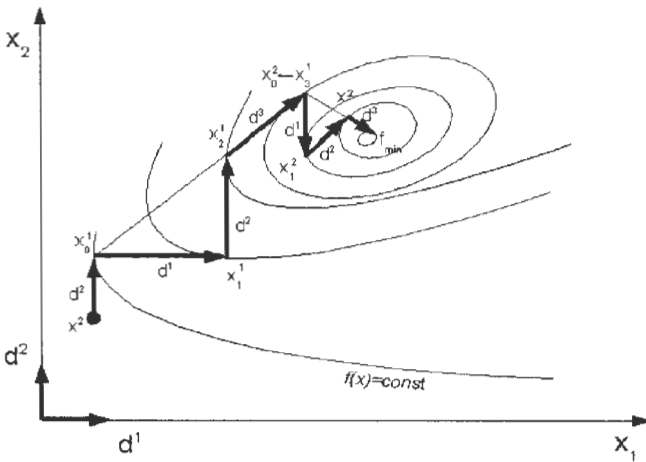
Ogólną sieć działań przytoczonego algorytmu przedstawiono na rysunku 7.3:



Rysunek 7.3. Ogólna sieć działań metody Powella P1

Działanie omówionej metody na przedstawiono przykładowo dla funkcji dwu zmiennych, której poziomice przedstawiono na rysunku 7.4. W etapie wstępnym określono punkt \mathbf{x}_0^1 tak, aby stanowił minimum na kierunku \mathbf{d}^2 . W pierwszym kro-

ku, startując z tak znalezionej punktu \mathbf{x}_0^1 , dokonano minimalizacji funkcji $f(x)$ wzdłuż kierunku \mathbf{d}^1 . Poszukiwanym punktem jest punkt \mathbf{x}_1^1 . Następnie powtórzono te same czynności wzdłuż kierunku \mathbf{d}^2 i w punkcie \mathbf{x}_2^1 osiągnięto szukane minimum. W drugim kroku wyznaczono składowe kierunku sprzężonego, przy czym w omawianym przypadku $\mathbf{x}_n^1 = \mathbf{x}_2^1$. Kierunek ten na rysunku 7.4 oznaczono przez \mathbf{d}^3 . W trzecim kroku wzdłuż tego nowego kierunku dokonano minimalizacji funkcji celu. Niech tym nowym punktem będzie \mathbf{x}_3^1 . Przeprowadzono teraz modyfikację kierunków w myśl przyjętej zasady, tzn. $\mathbf{d}^1 \leftarrow \mathbf{d}^2$ oraz $\mathbf{d}^2 \leftarrow \mathbf{d}^3$. Startując następnie z punktu $\mathbf{x}_0^2 \leftarrow \mathbf{x}_3^1$ powtórzono cały cykl od początku.



Rysunek 7.4. Ilustracja zasady działania metody Powella – P1

7.2.5. Metoda Sympleksu Neldera i Meada – N

N – wymiarowym sympleksem o $n+1$ wierzchołkach nazywamy zbiór wszystkich punktów określonych przez wektory:

$$\mathbf{x} = \sum_{i=1}^{n+1} x_i \mathbf{d}^i, \mathbf{d}^i \in R^n \tag{7.16}$$

przy czym:

$$\sum_{i=1}^{n+1} x_i = 1, \quad x_i \geq 0 \tag{7.17}$$

gdzie:

- \mathbf{d}^i – wektory sympleksu,
- x_i – współrzędne punktów sympleksu.

Inaczej mówiąc, n -wymiarowy sympleks jest wielościanem o $n+1$ wierzchołkach rozpiętych na $n+1$ wektorach bazowych (sympleksem jednowymiarowym jest odcinek, dwuwymiarowym jest trójkąt itd.).

Metoda sympleksu polega na utworzeniu nowej przestrzeni R^{n+1} , która zawiera n -wymiarowy sympleks o $n+1$ wierzchołkach w taki sposób, aby na nich była rozpięta powierzchnia określona przez badaną funkcję celu $f(x)$. Wylicza się więc na wstępie współrzędne punktów wierzchołkowych sympleksu P_i (dla $i = 1, 2, \dots, n+1$), zakładając przy tym pewną odległość między tymi wierzchołkami zwaną „krokiem”. W następnych iteracjach dokonuje się przekształceń sympleksu w sposób przedstawiony w poniższym algorytmie tak długo, dopóki odległość między jego wierzchołkami w pobliżu szukanego minimum będzie nie większa od założonej dokładności ϵ .

Wprowadzając oznaczenia:

P_h – wybrany punkt wierzchołkowy sympleksu spośród $n+1$ wierzchołków P_i , w którym wartość funkcji celu osiąga maksimum,

P_l – wybrany punkt wierzchołkowy sympleksu spośród $n+1$ wierzchołków P_i , w którym wartość funkcji celu osiąga minimum, oraz

\bar{P} – środek symetrii sympleksu, z wyłączeniem punktu P_h , zdefiniowany jako:

$$\bar{P} = \frac{\sum_{i=1}^{n+1} P_i}{n}, \text{ dla } i \neq h, \quad (7.18)$$

można zdefiniować następujące operacje stosowane w algorytmie sympleksu:

(1) operację *odbicia* punktu P_h względem \bar{P} , określoną przez:

$$P^* = (1 + \alpha)\bar{P} - \alpha P_h \quad (7.19)$$

(2) operację *ekspansji* punktu P^* względem \bar{P} , określoną przez:

$$P^{**} = (1 + \gamma)P^* - \gamma\bar{P} \quad (7.20)$$

(3) operację *kontrakcji* punktu P_h względem \bar{P} , określoną przez:

$$P^{***} = \beta P_h + (1 - \beta)\bar{P} \quad (7.21)$$

(4) operację *redukcji* sympleksu względem punktu P_l , określoną przez:

$$P^{****} = \delta P_l + (1 - \delta)P_i \quad (7.22)$$

gdzie:

P^* – operacja odbicia punktu P_h względem \bar{P} ,

P^{**} – operacja ekspansji punktu P^* względem \bar{P} ,

P^{***} – operacja kontrakcji punktu P_h względem \bar{P} ,

P^{****} – operacja skurczenia sympleksu względem punktu P_i ,

α – współczynnik odbicia, $\alpha > 0$,

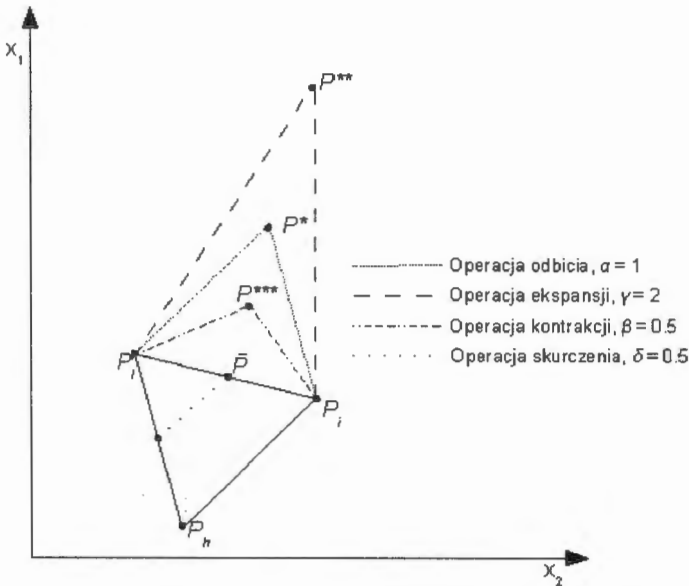
β – współczynnik kontrakcji, $0 < \beta < 1$,

γ – współczynnik ekspansji, $\gamma > 1$,

δ – współczynnik redukcji, $0 < \delta < 1$.

Wartości liczbowe współczynników α , β , γ , δ należy dobierać eksperymentalnie i są one parametrami postępowania. W przypadkach rozwiązywanych przez Neldera i Meada jako optymalna strategię przyjęto $\alpha=1$; $\beta=0,5$; $\gamma=2$; $\delta=0,5$.

Rysunek 7.5 przedstawia interpretację powyżej wymienionych operacji na przykładzie funkcji dwu zmiennych:



Rysunek 7.5. Operacje stosowane w algorytmie *Sympleksu Neldera i Meada* (Press i in., 1992)

Informacje wejściowe

- x_0 – dowolnie wybrany punkt startowy,
 s – początkowa odległość pomiędzy wierzchołkami wejściowego sympleksu,
 α – współczynnik odbicia, $\alpha > 0$,
 β – współczynnik kontrakcji, $0 < \beta < 1$,
 γ – współczynnik ekspansji, $\gamma > 1$,
 δ – współczynnik redukcji, $0 < \delta < 1$,
 ε – wymagana dokładność,
 n – liczba zmiennych niezależnych.

Algorytm obliczeń

- (1) Oblicz wartość funkcji celu w punktach wierzchołkowych sympleksu $q_i = f(P_i)$ dla $i = 1, 2, \dots, n+1$. Wyznacz h i l takie, że $f(P_h) = \max_i f(P_i)$ oraz $f(P_l) = \min_i f(P_i)$.

- (2) Oblicz środek symetrii sympleksu $\bar{P} = \frac{\sum_{i=1}^{n+1} P_i}{n}$, $i \neq h$. Wykonaj odbicie P^* punktu P_h względem \bar{P} . Oblicz $f(\bar{P}) = q_s$ oraz $f(P^*) = q_0$. Jeżeli $q_0 < \min_i q_i$, to (3).

- (3) Oblicz $P^{**} = (1 + \gamma)P^* - \gamma\bar{P}$ oraz $f(P^{**}) = q_e$. Gdy $q_e < \max_i q_i$, wtedy podstaw $P_h = P^{**}$, w innym przypadku podstaw $P_h = P^*$.

- (4) Jeżeli nie zostało spełnione kryterium na minimum, powtórz procedurę od kroku (1). Jeżeli $q_0 > \min_i q_i$, to (5).

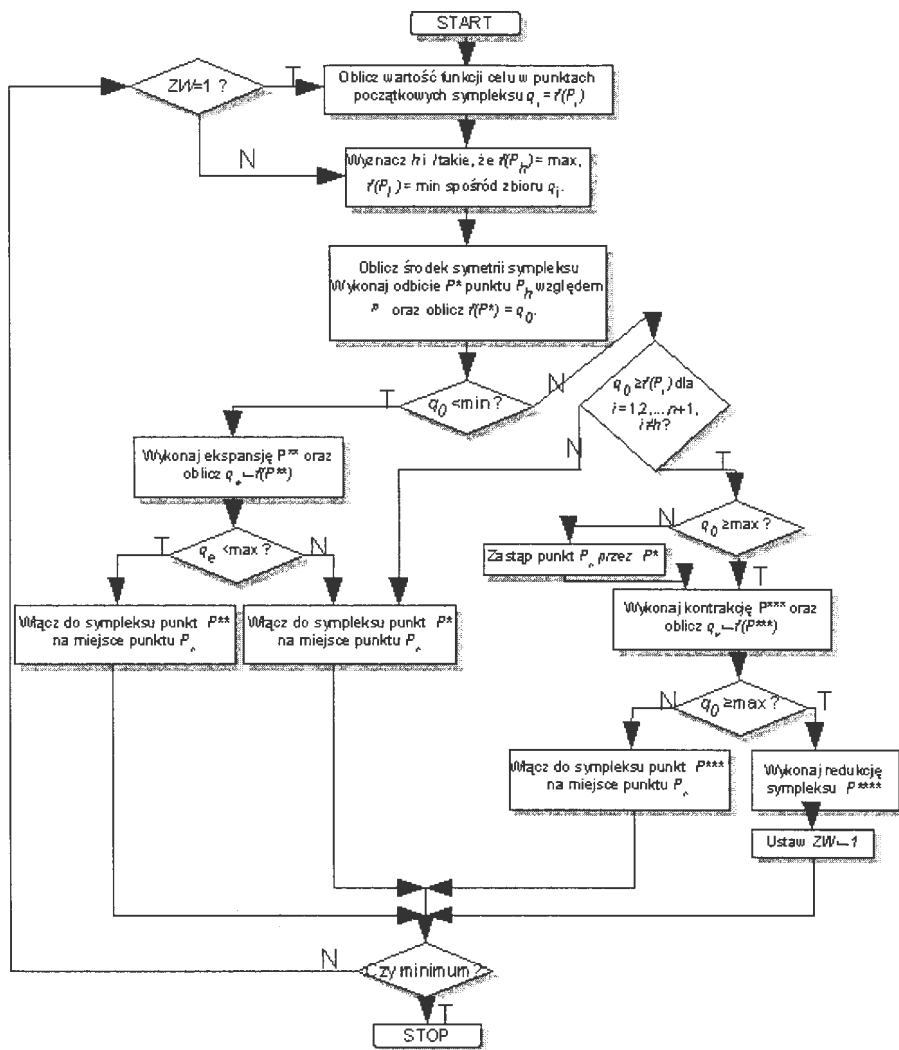
- (5) Jeżeli $q_0 \geq f(P_i)$ dla $i = 1, 2, \dots, n+1$, $i \neq h$ oraz $q_0 \geq \max_i q_i$, to przejdź do realizacji kroku następnego, natomiast jeśli $q_0 < \max_i q_i$, to podstaw uprzednio $P_h = P^*$. W przeciwnym razie (7).

- (6) Wykonaj kontrakcję P^{***} punktu P_h względem \bar{P} i oblicz $f(P^{***}) = q_k$; jeśli $q_k \geq \max_i q_i$, to wykonaj redukcję sympleksu P^{****} , natomiast gdy $q_k < \max_i q_i$, wówczas podstaw $P_h = P^{***}$ a następnie przejdź do realizacji kroku (8).

(7) Jeżeli $q_0 < f(P_i)$ dla $i = 1, 2, \dots, n+1, i \neq h$, to podstaw $P_h = P^*$.

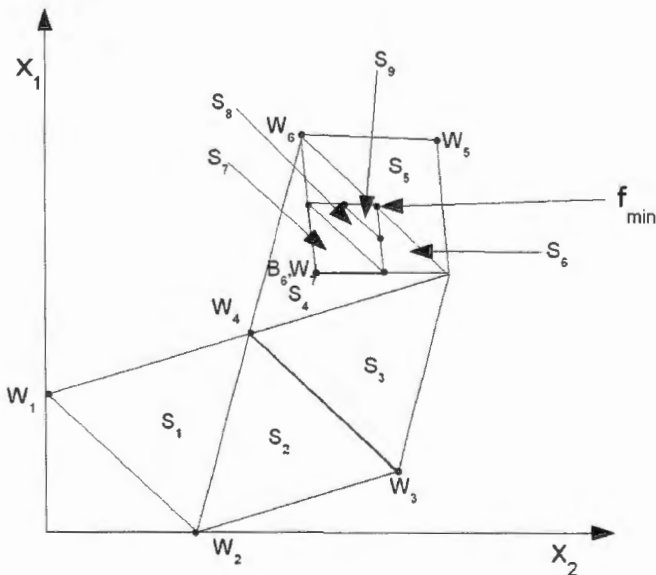
(8) Jeżeli nie zostało spełnione kryterium na minimum, powtórz procedurę począwszy od kroku (1).

Sieć działań przytoczonego algorytmu przedstawiono na rysunku 7.6, przy czym przed przystąpieniem do jego wykonywania, należy obliczyć współrzędne punktów wierzchołkowych sympleksu P_i (dla $i = 1, 2, \dots, n+1$).



Rysunek 7.6. Sieć działań metody Nelder i Meada

Działanie omówionej metody na przykładzie przedstawiono przykładowo dla funkcji dwu zmiennych. W czasie inicjalizacji metody utworzono dwuwymiarowy sympleks (w przestrzeni dwuwymiarowej jest to trójkąt) S_1 , w którym funkcja celu przyjmuje największą wartość w punkcie W_1 (punkty, gdzie funkcja celu przyjmuje największe wartości, nazwano dalej najgorszymi, te zaś, w których funkcja celu przyjmuje najmniejsze wartości - najlepszymi). W pierwszym kroku procedury dokonano odbicia punktu W_1 względem \bar{P} , uzyskując w ten sposób sympleks S_2 . Ekspansja nie przyniosła poprawy, więc do sympleksu dołączony został punkt W_2 . W tej sytuacji punktem najgorszym okazał się punkt W_2 . W drugim kroku dokonano kolejnego odbicia, tym razem punktu W_2 . Kolejnym najgorszym punktem okazał się punkt W_3 . W kolejnych krokach dokonywano kolejnych odbić (za każdym razem ekspansje nie przynosiły poprawy), aż odbicie punktu W_4 nie przyniosło poprawy, więc wykonano kontrakcję uzyskując sympleks S_5 . Najgorszym punktem okazał się punkt W_5 , który został kolejno odbity. Uzyskano w ten sposób sympleks S_6 . Operacje odbicia, ekspansji i kontrakcji nie dały pomyślnych rezultatów, więc wokół punktu B_6 dokonano redukcji sympleksu S_6 , uzyskując w ten sposób sympleks S_7 . Najgorszym w tym sympleksie okazał się punkt W_7 , od którego wykonano odbicie uzyskując sympleks S_8 . W tym sympleksie również operacje odbicia i kontrakcji nie przyniosły efektu, więc kolejny raz wykonana została redukcja sympleksu. Uzyskano w ten sposób sympleks S_9 . Ponieważ założona dokładność ϵ została przekroczona, algorytm zakończył działanie a poszukiwanym punktem okazał się najlepszy punkt sympleksu S_9 , oznaczony jako f_{min} . (Mathews, 1992).



Rysunek 7.7. Ilustracja działania metody Sympleksu Neldera i Meada

7.2.6. Metoda gradientów sprzężonych (Polaka - Riebery) - PR

Metoda gradientów sprzężonych wykorzystuje gradient funkcji celu $\nabla f(\mathbf{x})$. W kolejnych iteracjach metoda dokonuje minimalizacji funkcji w kierunku przeciwnym do jej gradientu, przy czym każdy nowy kierunek jest sprzężony z poprzednim w myśl reguły:

$$\mathbf{d}^{k+1} = -\nabla f(\mathbf{x}^{k+1}) + \beta_1 \mathbf{d}^k \quad (7.23)$$

Informacje wejściowe

\mathbf{x}^0 – dowolnie wybrany punkt startowy,

\mathbf{m}_k – wektor parametrów wejściowych koniecznych do wywołania minimalizacji kierunkowej (tj. początkowa długość kroku, minimalna długość kroku, współczynniki skrócenia i wydłużenia kroku),

ε – wymagana dokładność obliczeń,

n – liczba zmiennych niezależnych.

Algorytm obliczeń

Etap wstępny: Oblicz w punkcie \mathbf{x}^0 wartości $f(\mathbf{x}^0)$, $\nabla f(\mathbf{x}^0)$.

Iteracja k -ta. Podstaw $k \leftarrow 1, \mathbf{x}^k \leftarrow \mathbf{x}^0, \nabla f(\mathbf{x}^k) \leftarrow \nabla f(\mathbf{x}^0)$.

- (1) Wyznacz kierunek poszukiwań $\mathbf{d}^k = -\nabla f(\mathbf{x}^k)$.
- (2) Zbadaj, czy zostało spełnione kryterium zbieżności. Jeśli tak, to stop, jeśli nie, to (3).
- (3) Wykonaj minimalizację kierunkową (7.10):

$$\mathbf{x}^{k+1} \in T(\mathbf{x}^k, \mathbf{d}^k)$$

- (4) Oblicz $\nabla f(\mathbf{x}^{k+1})$ oraz wyznacz kierunek sprzężony:

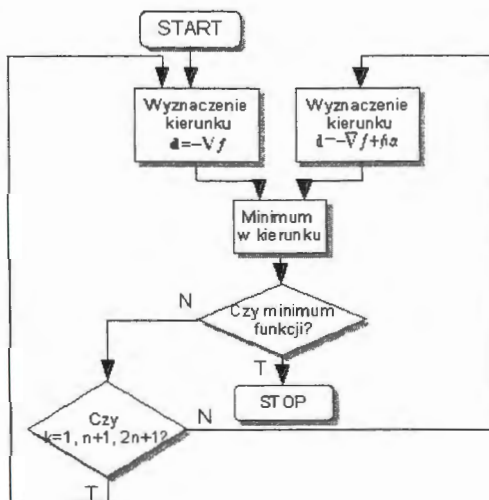
$$\mathbf{d}^{k+1} = -\nabla f(\mathbf{x}^{k+1}) + \beta_1 \mathbf{d}^k \quad (7.24)$$

gdzie:

$$\beta_k = \frac{\langle (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)), \nabla f(\mathbf{x}^{k+1}) \rangle}{\langle \nabla f(\mathbf{x}^k), \nabla f(\mathbf{x}^{k+1}) \rangle} - \nabla f(\mathbf{x}^k) \quad (7.25)$$

- (5) Podstaw $k \leftarrow k+1, \mathbf{x}^k \leftarrow \mathbf{x}^{k+1}$ oraz zbadaj, czy $k \leq n$. Jeśli tak, to podstaw $\mathbf{d}^k \leftarrow \mathbf{d}^{k+1}$ oraz powtórz (2). Jeśli nie, to $k \leftarrow 1, \nabla f(\mathbf{x}^k) \leftarrow \nabla f(\mathbf{x}^{k+1})$ i powtórz (1).

Ogólna sieć działań algorytmu została przedstawiona na rysunku 7.8.



Rysunek 7.8. Ogólna sieć działań metody gradientów sprzężonych

7.2.7. Metoda zmiennej metryki (Davidona - Fletchera - Powella) - DFP

W metodzie zmiennej metryki zakłada się, że funkcja celu $f(\mathbf{x})$ jest ograniczoną od dołu funkcją o postaci:

$$f(\mathbf{x}) = q + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (7.26)$$

gdzie \mathbf{A} oznacza macierz symetryczną dodatnio określoną, której elementami są drugie pochodnie cząstkowe funkcji $f(\mathbf{x})$.

Koncepcja metody DFP polega na takim sposobie tworzenia sprzężonych kierunków poszukiwań, aby począwszy od kierunku

$$\mathbf{d}^k = -\mathbf{V}_k \nabla f(\mathbf{x}^k) \quad (7.27)$$

dowolnie wybrana macierz \mathbf{V}_k w każdej następnej iteracji coraz lepiej aproksymowała macierz \mathbf{A}^{-1} . Z chwilą gdy macierz \mathbf{V}_k zostaje przekształcona w odwrotną ma-

cierz drugich pochodnych \mathbf{A}^{-1} , procedura kończy swoje działanie i bieżący punkt \mathbf{x} można wtedy uznać za poszukiwany punkt ekstremalny.

Informacje wejściowe

\mathbf{x}^0 – dowolnie wybrany punkt startowy,

\mathbf{V}^0 – macierz wyjściowych kierunków poszukiwań przyjmowana zazwyczaj jako macierz jednostkowa,

\mathbf{m}_k – wektor parametrów wejściowych koniecznych do wywołania minimalizacji kierunkowej (tj. początkowa długość kroku, minimalna długość kroku, współczynniki skrócenia i wydłużenia kroku),

ε – wymagana dokładność obliczeń,

n – liczba zmiennych niezależnych.

Algorytm obliczeń

Etap wstępny: Oblicz w punkcie \mathbf{x}^0 wartości $f(\mathbf{x}^0)$, $\nabla f(\mathbf{x}^0)$.

Iteracja k -ta. Podstaw $k \leftarrow 1, \mathbf{x}^k \leftarrow \mathbf{x}^0, \nabla f(\mathbf{x}^k) \leftarrow \nabla f(\mathbf{x}^0), \mathbf{V}_k \leftarrow \mathbf{V}_0$.

- (1) Wyznacz kierunek poszukiwań $\mathbf{d}^k = -\mathbf{V}_k \nabla f(\mathbf{x}^k)$.
- (2) Zbadaj, czy zostało spełnione kryterium zbieżności. Jeśli tak, to stop, jeśli nie, to (3).
- (3) Wykonaj minimalizację kierunkową:

$$\mathbf{x}^{k+1} \in T(\mathbf{x}^k, \mathbf{d}^k)$$

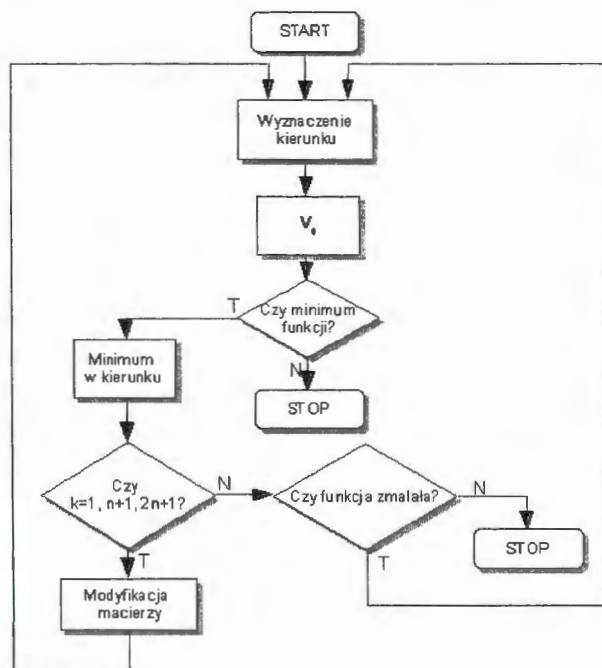
- (4) Oblicz $\nabla f(\mathbf{x}^{k+1})$ oraz wyznacz następną estymator macierzy \mathbf{V}_{k+1} :

$$\mathbf{V}_{k+1} = \mathbf{V}_k + \frac{\mathbf{s}^k \langle \mathbf{s}^k \rangle}{\langle \mathbf{s}^k, \mathbf{y}^k \rangle} - \frac{\mathbf{V}_k \mathbf{y}^k \langle \mathbf{V}_k \mathbf{y}^k \rangle}{\langle \mathbf{y}^k, \mathbf{V}_k \mathbf{y}^k \rangle} \quad (7.28)$$

gdzie: $\mathbf{s}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$ oraz $\mathbf{y}^k = \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$.

- (5) Podstaw $\mathbf{x}^k \leftarrow \mathbf{x}^{k+1}$, $\nabla f(\mathbf{x}^k) \leftarrow \nabla f(\mathbf{x}^{k+1}), k \leftarrow k+1$, oraz zbadaj, czy $k \leq n$. Jeśli tak, to podstaw $\mathbf{V}_k \leftarrow \mathbf{V}_{k+1}$ oraz powtórz (2), natomiast jeśli nie, to $k \leftarrow 1, \mathbf{V}_k \leftarrow \mathbf{V}_{k+1}$ i powtórz (1).

Ogólny schemat działania metody zmiennej metryki przedstawiono na rysunku 7.9.



Rysunek 7.9. Sieć działań metody zmiennej metryki DFP

7.2.8. Symulowane wyżarzanie metodą Metropolis - M

Metoda symulowanego wyżarzania, to rodzaj algorytmu przeszukującego przestrzeń alternatywnych rozwiązań problemu w celu wyszukania optimum globalnego. Sposób działania symulowanego wyżarzania przypomina zjawisko wyżarzania w metalurgii (stąd nazwa).

Istotą metody symulowanego wyżarzania jest analogia do procesów termodynamicznych, w szczególności do procesów krzepnięcia i krystalizacji metali. W wysokich temperaturach molekuly metalu poruszają się dowolnie względem siebie. Wraz ze zmniejszaniem temperatury ruch termiczny ustaje a atomy tworzą uporządkowane struktury krystaliczne (atomy zajmują ściśle określone miejsca w sieci przestrzennej i mogą jedynie drgać w obrębie zajmowanych przez siebie miejsc, które to miejsca rozlokowane są zgodnie z symetrią translacyjną). Można więc powiedzieć, że stan, w którym atomy tworzą sieć krystaliczną, jest minimum energetycznym układu termodynamicznego. Warto podkreślić fakt, iż wolno schładzany metal samoczynnie znajduje taki stan minimum energetycznego, podczas gdy atomy

metalowo szybko chłodzonego czy np. hartowanego tworzą polikryształy lub struktury amorficzne (Encyclopedia Britannica; Weise, 2007).

Poprzez analogię do procesów termodynamicznych można powiedzieć, że przedstawione w poprzednich rozdziałach algorytmy optymalizacji odpowiadały „szybkemu chłodzeniu” czy też hartowaniu stopu. Ich zasada działania polegała na jak najszybszym minimalizowaniu funkcji celu (odpowiednika energii). Taka strategia prowadzi do znalezienia minimum lokalnego, które nie musi być globalnym. Zgodnie z rozkładem Boltzmanna:

$$\text{Prob}(E) \sim \exp(-E/kT) \quad (7.29)$$

gdzie:

T – temperatura,

E – stan energetyczny,

k – stała Boltzmanna,

energia w układzie znajdującym się w równowadze termicznej jest rozłożona probabilistycznie na wszystkie możliwe stany energetyczne. Więc nawet w niskiej temperaturze istnieje prawdopodobieństwo (bardzo małe), iż układ będzie znajdował się w wysokim stanie energetycznym. Co za tym idzie, jest szansa, że układ znajdujący się w stanie lokalnego minimum energetycznego opuści ten stan i przejdzie w stan innego minimum (lepszego) – być może globalnego.

Powyżej przedstawione zależności można wykorzystać w zadaniach optymalizacji (Press i in., 1992). Przyjęcie schematu optymalizacji, w którym kroki wykonywane są zazwyczaj w kierunku minimalizacji funkcji celu, lecz w szczególnych przypadkach również w kierunku jej maksymalizacji, nazywane jest algorytmem *Metropolis*.

Aby zastosowanie algorytmu *Metropolis* do zagadnień nie związanych z procesami termicznymi było możliwe, należy zdefiniować:

- opis możliwych stanów układu,
- generator losowych zmian w stanach układu (zmiany są „opcjami” dostępnymi dla układu),
- funkcję E (analogiczną do energii), której minimalizacja jest celem procedury,
- parametr kontrolny T (analogiczny do temperatury) oraz *harmonogram wyżarzania*, czyli charakterystykę spadku T .

W zagadnieniach optymalizacji powyższe zagadnienia definiowane są następująco:

- funkcja celu f jest odpowiednikiem energii E ,
- punkt x jest odpowiednikiem stanu układu termodynamicznego,

- parametr T jest pewnym odwzorowaniem temperatury układu,
- jako losowe zmiany w stanach układu interpretuje się przemieszczenie punktu x do $x + \Delta x$.

W niniejszym zastosowaniu przyjęto algorytm oparty na metodzie *Sympleksu Neldera i Meada*. Procedura działa analogicznie, jak opisana w punkcie 7.2.4 z następującymi różnicami:

- do wartości funkcji celu w każdym wierzchołku utworzonego sympleksu dodaje się pewną zmienną losową proporcjonalną do temperatury T ,
- od wartości funkcji celu w każdym nowym punkcie wierzchołkowym sympleksu odejmuje się pewną zmienną losową proporcjonalną do temperatury T .

W ten sposób algorytm porusza się w kierunku minimalizacji funkcji celu, jednakże czasami wykonuje kroki w kierunku odwrotnym. Jeśli temperatura będzie malała stosunkowo wolno (dla danego zagadnienia), możliwe jest, że algorytm znajdzie minimum lokalne (jeżeli oczywiście będzie ono możliwe do osiągnięcia przy założonej temperaturze początkowej T_0).

Jako harmonogram zmian temperatury T przyjęto następującą zależność:

$$T = T_0(1 - k/K)^\alpha \quad (7.30)$$

gdzie:

T – temperatura aktualna,

T_0 – temperatura początkowa,

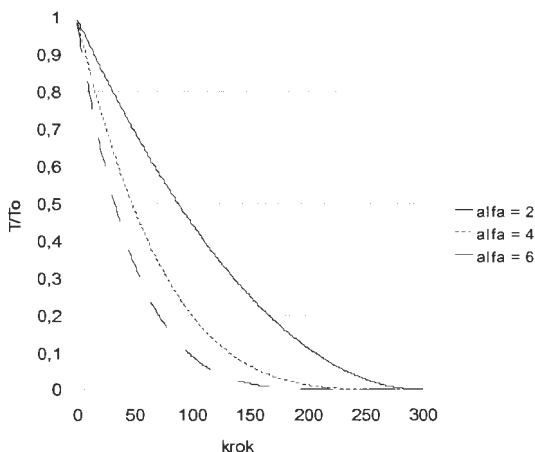
k – aktualny krok algorytmu,

K – założona liczba kroków, po których temperatura jest równa zero,

α – wykładnik funkcji temperatury $\alpha \geq 1$.

Dobór współczynników T_0 , K oraz α jest sprawą arbitralną i zależy od znajomości zagadnienia osoby przeprowadzającej optymalizację. Intuicyjnie można zauważyć, iż im większe α , tym więcej iteracji algorytm spędzi w niższych temperaturach.

Wykres zmian temperatury w stosunku do temperatury początkowej dla $K=300$ oraz różnych α przedstawia rysunek 7.10.

Rysunek 7.10. Wpływ współczynnika α na zmiany T/T_0 *Informacje wejściowe*

x_0 – dowolnie wybrany punkt startowy,

T_0 – początkowa temperatura stopu,

K – założona liczba kroków, po których temperatura jest równa zeru,

α_T – wykładnik funkcji temperatury $\alpha_T \geq 1$.

s – początkowa odległość pomiędzy wierzchołkami wejściowego sympleksu,

α – współczynnik odbicia, $\alpha > 0$,

β – współczynnik kontrakcji, $0 < \beta < 1$,

γ – współczynnik ekspansji, $\gamma > 1$,

δ – współczynnik redukcji, $0 < \delta < 1$,

ε – wymagana dokładność,

n – liczba zmiennych niezależnych.

Algorytm obliczeń

- (1) Oblicz wartość funkcji celu w punktach wierzchołkowych sympleksu:

$$q_i = f(P_i) + \Delta f, \Delta f \in \left(0, T_0(1 - k/K)^\alpha\right) \quad (7.31)$$

- (2) Wykonaj algorytm *Sympleksu Neldera i Meada* - N z tym, że wartości funkcji celu w nowych wierzchołkach sympleksu oblicz z następującej zależności:

$$q_i = f(P_i) - \Delta f, \Delta f \in (0, T_0(1 - k/K)^\alpha) \quad (7.32)$$

- (3) Jeżeli nie zostało spełnione kryterium na minimum, powtórz procedurę począwszy od kroku (1).

7.2.9. Porównanie metod w oparciu o przykłady literaturowe

Jednymi z istotniejszych zagadnień podczas stosowania bezgradientowych metod optymalizacji jest dobór punktu startowego oraz czułość metody i efektywna szybkość zbieżności metody (interpretowana jako liczba obliczeń funkcji celu).

Jak wynika z przeprowadzonych badań opisanych w literaturze (Findeisen, Szymanowski, Wierzbicki, 1977; Jastriebow, Wciślík, 2004; Press i in., 1992; Mathews 1992; Weise, 2007), najmniej czułą na zmianę punktów startowych okazała się metoda *Rosenbrocka* R - Rosenbrocka. Chociaż ogólnie rzecz biorąc jest ona wolniejsza od metody P1 czy HJ, to jednak w prawie każdej sytuacji daje zadowalające wyniki. Natomiast rozpatrując kryterium szybkości zbieżności najefektywniejszymi są metody gradientowe, w szczególności metody zmiennej metryki DFP.

Metody P1 oraz DFP charakteryzują się dużą czułością na dobór punktu startowego. W przypadku punktów mało korzystnych krzywa zbieżności tej metody zbliżona jest do krzywej kwadratowej, a więc w początkowym okresie jest ona dosyć wolna, natomiast w otoczeniu minimum zaczyna szybko wzrastać. Tego rodzaju przebieg można wyjaśnić tym, że początkowo przeszukiwanie odbywa się wzdłuż ortogonalnej bazy kierunków, w wyniku czego przesunięcie jest znikomo małe, dopiero wyznaczenie sprzężonych kierunków określa właściwy kierunek poszukiwania minimum i od tego momentu efektywność minimalizacji wzrasta. W przypadku korzystnego doboru punktów startowych krzywa zbieżności metody P1 ma charakter wykładniczy typu $1 - e^y$, a więc znacznie lepszy od liniowego, którym charakteryzują się metody R i N.

Metoda sympleksu N posiada korzystną cechę - dobór punktów startowych położonych w obszarach funkcji reprezentujących sobą strome zbocze „dolin” nie powoduje zmniejszenia szybkości metody N. Fakt ten można wyjaśnić tym, że w trakcie posuwania się w kierunku minimum metoda ta automatycznie eliminuje silne nieliniowości. Wpływa to ze sposobu wyliczania środka symetrii sympleksu \bar{P} , który jest obliczany z pominięciem wierzchołka f_{max} . Metoda ta charakteryzuje się prawie liniową zbieżnością a tym samym kolejne dokładności osiągnane są z tą samą szybkością. Wadą jej natomiast jest dosyć duża czułość na ukształtowanie obszaru. Dla punktów startowych leżących na stosunkowo płaskich zboczach niewiele odległych od minimum metoda N staje się wolniejsza w stosunku do innych metod. Wykonuje wtedy wiele niepotrzebnych obliczeń wartości funkcji celu w wierzchołkach sympleksu w rezultacie nieznacznych różnic występujących pomiędzy nimi. Inną wadą

jest problem wyznaczenia początkowej odległości pomiędzy wierzchołkami sympleksu. Odległość ta musi być arbitralnie dobrana przez prowadzącego optymalizację.

W pewnych przypadkach metoda M może dać bardzo dobre rezultaty, znajdując minimum globalne. Jednakże nie jest trywialnym zagadnienie doboru odpowiedniej temperatury stopu oraz harmonogramu zmian tej temperatury. Aby więc przeprowadzić tę metodę, należy mieć dostateczną wiedzę ekspercką na temat optymalizowanego zagadnienia. Metoda ta, ze względu na swój stochastyczny charakter, jest niewątpliwie najwolniejszą z prezentowanych metod (wymaga największej liczby obliczeń funkcji celu).

7.3. Metodyka optymalizacji modelu procesu

7.3.1. Optymalizacja modelu procesu

W trakcie optymalizacji modelu w sposób automatyczny poszukiwano następujących wybranych współczynników występujących w równaniach opisujących model procesu biologiczno-mechanicznej oczyszczalni ścieków:

- współczynniki A i B w równaniach (2.11), (2.12) modelu osadnika wstępnego,
- współczynniki μ_H , μ_A , b_H , b_A , b_a , b_h w równaniach (2.18)–(2.51) modelu komory z osadem czynnym,
- współczynniki v_0 , B w równaniu (2.55) modelu osadnika wtórnego.

Biorąc pod uwagę strukturę modelu (rozdz. 3.8):

- 4 komory osadnika wstępnego,
- 4 strefy komory z osadem czynnym,
- 8 komór osadnika wstępnego,

uzyskujemy:

- 8 nieznanych współczynników w modelu osadnika wstępnego,
- 24 współczynniki w modelu komory z osadem czynnym,
- 16 współczynników w modelu osadnika wtórnego,

co daje razem 48 nieznanych współczynników, które powinny zostać poddane procesowi optymalizacji.

Jedna symulacja modelu dla całej szarży pomiarowej (czyli jedna iteracja procesu optymalizacji) trwa (korzystając z osobistego komputera o częstotliwości taktowania procesora 1,7 GHz, oraz pamięci RAM 512 MB) około 50 sek. Biorąc pod

uwagę 48 wymiarową przestrzeń optymalizacji, jest to czas zbyt długi, aby przeprowadzić serię badań w sensownym czasie.

Korzystając z modułowej konstrukcji modelu dokonano dekompozycji modelu procesu na dwa mniejsze modele:

- osadnika wstępnego,
- komory z osadem czynnym wraz z osadnikiem wtórnym.

W modelu osadnika wstępnego optymalizowano współczynniki B w równaniach (2.11), (2.12).

Następnie, po znalezieniu optymalnych współczynników B , zapisano dane generowane przez model osadnika wstępnego i użyto ich jako danych wejściowych do modelu komory z osadem czynnym wraz z osadnikiem wtórnym. Dzięki temu uzyskano:

- skrócenie czasu obliczeń (zmniejszenie ilości iteracji) ze względu na wydzielenie podział zadania optymalizacji na dwa mniejsze,
- skrócenie czasu obliczeń (czasu symulacji) ze względu na pominięcie w modelu komory z osadem czynnym obliczeń wykonywanych w osadniku wstępnym.

W komorze z osadem czynnym wraz z osadnikiem wtórnym optymalizowano następujące współczynniki:

- współczynniki $\mu_H, \mu_A, b_H, b_A, b_{hw}, b_a$ w równaniach (2.18)–(2.51) modelu komory z osadem czynnym,
- współczynniki v_0, B w równaniu (2.55) modelu osadnika wtórnego.

7.3.2. Optymalizacja struktury modelu

W pierwszym podejściu zadanie optymalizacji prowadzono dla struktury wyjściowej modelu (patrz rozdział 3.8) zaproponowanej w (Studziński, 2004). W niniejszej pracy postanowiono, oprócz pojęcia *optymalizacji modelu* rozumianego jako zadanie dopasowania współczynników w równaniach opisujących proces, wprowadzić pojęcie *optymalizacji struktury modelu*.

Optymalizacja struktury modelu prowadzi do liczby modelowanych komór zbiorników technologicznych. Modyfikacja ta sprowadza się przede wszystkim do minimalizacji ich liczby. W czasie optymalizacji modelu algorytmy poszukiwania nieznanymi parametrów mogą zwrócić optymalną wartość parametru równą zeru (dla zależności liniowych) lub mniejszą od zera (dla parametrów występujących w potęgach). Przy wyznaczeniu zerowych wartości parametrów dla równań opisujących szybkości reakcji, np. (2.18), można wnioskować, że w takim przypadku ko-

mora modelu jest zbędna do poprawnego modelu i można strukturę zredukować o tę komorę.

Działanie polegające na redukcji ilości komór, prowadzi do zmniejszenia komplikacji modelu oraz przyśpieszeniu obliczeń. W przypadku wykorzystania modelu jako narzędzia działającego przy rzeczywistym obiekcie, może być to bardzo przydatna cecha.

Poniżej przedstawiono zaproponowany algorytm optymalizacji struktury modelu:

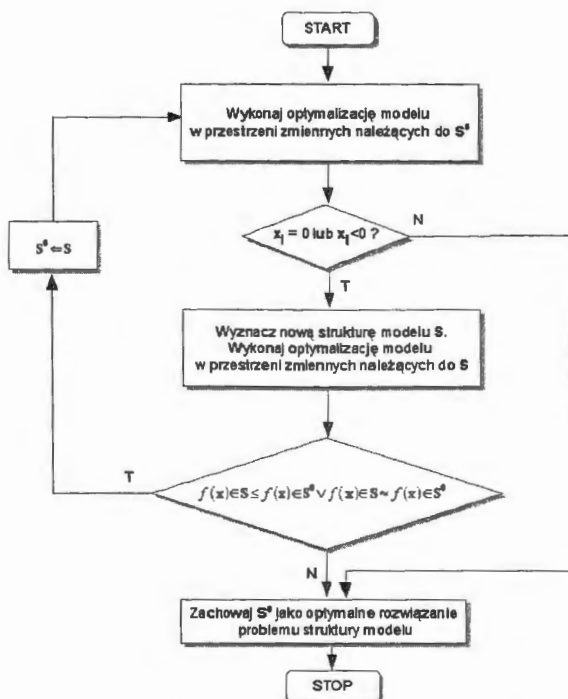
Informacje wejściowe:

- \mathbf{x}^0 – punkt startowy uzyskany w wyniku kalibracji modelu,
- S^0 – początkowa struktura modelu.

Algorytm obliczeń:

- (1) Wykonaj optymalizację modelu w przestrzeni zmiennych należącej do S^0 .
- (2) Przeanalizuj wyniki. Jeżeli parametry $x_i = 0$ lub $x_i < 0$, to (3), inaczej zakończ działanie procedury.
- (3) Wyznacz nową strukturę modelu S .
- (4) Wykonaj optymalizację modelu w przestrzeni S . Jeżeli: $f(\mathbf{x}) \in S^k \leq f(\mathbf{x}) \in S^0 \vee f(\mathbf{x}) \in S^k \equiv f(\mathbf{x}) \in S^0$, to podstaw $S^0 \leftarrow S^k$ i wróć do (1). Inaczej pozostaw S^0 jako optymalne i zakończ działanie procedury.

Sieć działań przytoczonego algorytmu przedstawiono na rysunku 7.11.



Rysunek 7.11. Sieć działań algorytmu optymalizacji struktury modelu

7.4. Przyjęte funkcje celu w zadaniach optymalizacji modelu

7.4.1. Funkcja celu dla zadania optymalizacji modelu osadnika wstępnego

Jak wspomniano w poprzednich rozdziałach, w zadaniu optymalizacji (oraz wcześniejszej kalibracji) modelu osadnika wstępnego jako funkcję celu przyjęto odchyłkę wartości zawiesiny mineralnej x_{min} , generowanych przez model, od rzeczywistych wartości zmierzonych na oczyszczalni, definiowaną jako:

$$f(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^K (x_{\min M}(\mathbf{x}, k) - x_{\min R}(k))^2 \quad (7.33)$$

gdzie:

$f(\mathbf{x})$ – wartość funkcji celu,

\mathbf{x} – wektor parametrów optymalizowanych,

k – krok symulacji,

K – ilość wszystkich kroków symulacji i pomiarów,

$x_{\min M}$ – wartość stężenia zawiesiny mineralnej generowana przez model,

$x_{\min R}$ – rzeczywista (zmierzona) wartość stężenia zawiesiny mineralnej.

7.4.2. Funkcja celu dla zadania optymalizacji modelu komory z osadem czynnym

W zadaniu optymalizacji komory z osadem czynnym jako funkcję celu przyjęto następujące wyrażenie:

$$f(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^K \left[\sum_{i=1}^n \sqrt{\left(\frac{\bar{z}_{M,i} - \bar{z}_{R,i}}{\bar{z}_{R,i}} \right)^2} + \sum_{i=1}^n \exp \alpha \left(\frac{\bar{z}_{M,i} - \bar{z}_{R,i}}{\bar{z}_{R,i}} - 0,5 \right) \right] \quad (7.34)$$

gdzie:

$z_{M,i}$ – stężenie i -tej, substancji, generowane przez model,

$z_{R,i}$ – stężenie i -tej substancji rzeczywiste (zmierzone),

α – współczynnik funkcji wykładniczej, $\alpha=30$.

Dzięki przyjętej postaci funkcji celu poszukiwany jest model możliwie najmniejszym względny błędzie modelu, a dodana funkcja eksponentialna zabezpiecza przed ucieczką wartości procesowych generowanych przez model poniżej połowy wartości procesowych rzeczywistych.

IBS PAN *Sená*

45905

Bibl. podręczna

ISBN 978-83-89475-15-2
ISSN 0208-8029
