

PAWEŁ FORYŚ*

ZASTOSOWANIE METODY ROJU CZĄSTEK
W OPTYMALNYM PROJEKTOWANIU
ELEMENTÓW KONSTRUKCJI

A PARTICLE SWARM OPTIMIZATION APPLIED
TO OPTIMAL DESIGN OF STRUCTURAL ELEMENTS

Streszczenie

Zmodyfikowana metoda optymalizacji rojem cząstek może być z powodzeniem stosowana jako uniwersalne narzędzie wspomagające optymalizację elementów konstrukcji. W artykule przedstawiono pewne aspekty przystosowania metody do rozwiązywania zadań programowania nieliniowego w przestrzeni zmiennych mieszanych. Jako przykłady numeryczne przedstawiono zadania: projektowania sprężyny, przestrzennej kratownicy oraz optymalizacji modelu ramy Koitera ze względu na stan pokrytyczny, tj. odtwarzanie i kształtowanie ścieżki pokrytycznej.

Słowa kluczowe: metoda roju cząstek, programowanie nieliniowe, optymalizacja konstrukcji

Abstract

A modified particle swarm optimization method can be successfully applied as a universal tool to aid optimal design of structural elements. Some improvements of the method to cope with nonlinear optimization tasks in mixed variable space are presented in the paper. Design of the helical spring, spatial truss and Koiter's frame model against postcritical state i.e. reconstruction and modeling of postcritical path are presented as the numerical examples.

Keywords: particle swarm optimization, nonlinear programming, structural optimization

*Dr inż. Paweł Foryś, Instytut Mechaniki Stosowanej, Wydział Mechaniczny, Politechnika Krakowska.

1. Wstęp

Inspiracją dla wielu nowoczesnych technik obliczeniowych jest obserwacja funkcjonowania organizmów żywych. Na przykład sztuczna sieć neuronowa jest uproszczonym modelem ludzkiego mózgu. Z kolei algorytmy genetyczne wzorowane są na teorii ewolucji istot żywych. Omawiana w artykule metoda roju cząstek naśladuje zachowania społeczne osobników tworzących zorganizowane populacje. Osobniki żyjące w rojach oddziałują na siebie przy równoczesnym wpływie środowiska bytowania. Cząstki, mając zdolność zapamiętywania swoich położeń, przystosowują się do środowiska, w którym żyją i powracają do obszarów o sprzyjających właściwościach. Mogą nimi być miejsca żerowania, nocowania czy wylegu. Ta kognitywna cecha cząstek umożliwia wyszukiwanie nowych obszarów o jeszcze lepszych właściwościach. Przykładami rojów w świecie przyrody są: kolonie owadów, stada ptaków, ławice ryb itd.

2. Metoda optymalizacji rojem cząstek

Algorytm numeryczny optymalizacji rojem cząstek (PSO – *Particle Swarm Optimization*) został zaproponowany w 1995 r. przez Kennedy’ego i Eberharta [1]. PSO ma wiele elementów wspólnych z technikami ewolucyjnymi, takimi jak algorytmy genetyczne (*Genetic Algorithms – GA*) czy programowanie ewolucyjne (*Evolutionary Programming – EP*).

Model numeryczny społecznego zachowania grupy obiektów traktuje populację jako rój, a każdego osobnika jako cząstkę. W trakcie kolejnych kroków zdyskretyzowanego czasu cząstki przemieszczają się do nowych położeń, symulując adaptację roju do środowiska, czyli poszukują optimum. „Liderem” roju zostaje cząstka o najlepszym dotychczas znanym położeniu. Każda cząstka ma przyporządkowanych „sąsiadów”, którymi są wybrane cząstki z roju. Przyporządkowanie jest zazwyczaj statyczne, co oznacza, że sąsiedzi są określani raz, na początku obliczeń. „Liderem sąsiadów” dla każdej cząstki jest sąsiad o najlepszym dotąd znalezionym położeniu. Położenia cząstek w obszarze poszukiwania stanowią potencjalne rozwiązania. W przeciwieństwie do algorytmów genetycznych PSO zazwyczaj nie korzysta z operatorów ewolucyjnych, takich jak krzyżowanie czy mutacja.

Zastosowanie metody roju cząstek do rozwiązywania zadań nieliniowej optymalizacji z ograniczeniami wymaga znacznego usprawnienia wydajności algorytmu w stosunku do jego wersji oryginalnej. Poprawę uzyskano poprzez modyfikację równań ruchu cząstek (połączenie wersji globalnej i lokalnej), wprowadzenie koncepcji *algorytmu dwustanowego* oraz idei *elastycznych ograniczeń* prędkości cząstek [2].

Algorytm PSO funkcjonuje wg następującego schematu:

1. Nadanie cząstkom roju losowych położeń i prędkości początkowych.
2. Dokonanie oceny położenia cząstek za pomocą funkcji dopasowania (*fitness*).
3. Zmiana zapisu w pamięci cząstek dotycząca najlepszych własnych położeń oraz najlepszych położeń liderów sąsiadów. Wyłonienie lidera roju.
4. Aktualizacja wektora prędkości każdej cząstki

$$\mathbf{v}_{k+1}^i = w_{1k}^i \mathbf{v}_k^i + w_{2k}^i [c_1 r_{1k}^i (\mathbf{p}_k^{ii} - \mathbf{x}_k^i) + c_2 r_{2k}^i (\mathbf{p}_k^g - \mathbf{x}_k^i) + c_3 r_{3k}^i (\mathbf{p}_k^{ni} - \mathbf{x}_k^i)] \quad (1)$$

5. Aktualizacja położenia każdej cząstki

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \quad (2)$$

gdzie:

- \mathbf{x}_k^i – wektor położenia i -tej cząstki,
- \mathbf{v}_k^i – odpowiedni wektor prędkości,
- \mathbf{p}_k^{ii} – najlepsze dotąd znalezione położenie i -tej cząstki,
- \mathbf{p}_k^{ni} – najlepsze dotąd znalezione położenie lidera sąsiadów i -tej cząstki,
- \mathbf{p}_k^g – najlepsze dotąd znalezione położenie lidera roju,
- w_{1k}^i, w_{2k}^i – współczynniki wagowe, określane na poziomie i -tej cząstki.

W powyższych oznaczeniach dolny indeks k określa kolejny krok iteracji. Współczynniki c_1, c_2, c_3 są ustalonymi mnożnikami wagowymi określanymi w literaturze – w zależności od interpretacji – jako współczynniki przyspieszenia (*acceleration constants*) lub współczynniki uczenia się (*learning factors*). Współczynniki $r_{1k}^i, r_{2k}^i, r_{3k}^i$ są liczbami losowymi o rozkładzie równomiernym w przedziale $[0, 1]$. Generowane są w kolejnych krokach iteracji dla każdej cząstki.

Wielkość w_{1k}^i jest mnożnikiem interpretowanym jako współczynnik bezwładności ruchu cząstki (*inertia weight*). W algorytmie dwustanowym wartości współczynników wagowych są dobierane w zależności od zachowania się cząstek roju. Jeżeli cząstka przemieściła się do lepszego położenia, czyli nastąpiła poprawa wartości funkcji celu, to w kolejnym kroku wartość w_{2k}^i przyjmuje się jako równą 0, natomiast wartość $w_{1k}^i \geq 1$, zezwalając na swobodny ruch w tym kierunku i dalsze przyspieszanie cząstki (*stan 1*). Gdy cząstka nie poprawiła swego dopasowania, wartości obu współczynników przyjmuje się równe 1 i nowe położenie jest określane z wykorzystaniem historii ruchu roju (*stan 2*), czyli tak jak odbywa się to w wersji niemodyfikowanej.

Punkty 2–5 należy wykonywać w pętli aż do spełnienia przez algorytm kryterium zatrzymania. Dla zadań przedstawionych w artykule jako kryterium zatrzymania obliczeń przyjęto ustaloną liczbę kroków iteracji. Jeden przebieg pętli odpowiada kolejnemu krokowi czasowemu dla poruszającego się roju. Wartość kroku czasowego przyjmuje się jako jednostkową. Równania (1), (2), według których następuje aktualizacja prędkości i położenia cząstek roju na kolejnych krokach iteracji, noszą nazwę *równań aktualizacji* lub *reguły aktualizacji*.

Cząstki poruszają się ruchem odcinkowo prostoliniowym. Postać *równań aktualizacji* za współczynnikiem w_{2k}^i wynika z drugiej zasady dynamiki Newtona, gdzie wypadkowa siła wywołująca przyspieszenie każdej cząstki powstaje wskutek działania sił od „naciągniętych sprężyn” pomiędzy położeniem aktualnym a najlepszymi położeniami: własnym, lidera sąsiada oraz lidera roju. Kolejny wektor prędkości cząstki jest wynikiem jej przyspieszenia w kierunku nowego, potencjalnie lepszego położenia, bazując na stale zmieniających się najlepszych wcześniej znanych położeniach.

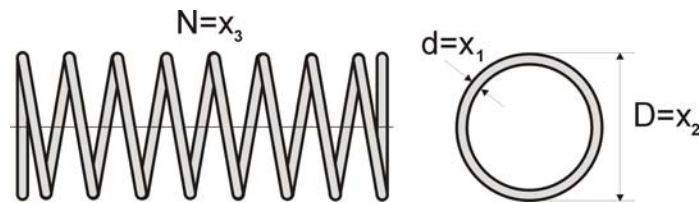
Przystosowanie algorytmu do obsługi ograniczeń nierównościowych i równościowych zostało szerzej opisane w pracy [3]. Z kolei modyfikacje umożliwiające obsługę całkowitych oraz mieszanych zmiennych projektowych zostały przedstawione w pracach [4 i 5]. Dobór parametrów sterujących pracą algorytmu omówiono w [4].

W przedstawionych w artykule zadaniach liczba cząstek wynosiła początkowo 3. Stopniowo była ona zwiększana do poziomu, powyżej którego nie obserwowano zmiany wyniku końcowego rozwiązywanego zadania. Liczba sąsiadów zawiera się w przedziale 0,3–0,6 liczby wszystkich cząstek.

3. Przykłady obliczeniowe

3.1. Optymalizacja sprężyny ściskanej – zmienne ciągłe

W zadaniu minimalizowana jest objętość rozciąganej/ściskanej sprężyny przedstawionej na rysunku 1 (np. [6]).



Rys. 1. Projektowanie walcowej sprężyny ściskanej – zmienne decyzyjne
Fig. 1. Helical spring design – the decision variables

Matematycznie zadanie jest wyrażone następująco (np. [7]).

Minimalizuj
$$f = (x_3 + 2)x_2x_1^2 \quad (3)$$

Przy ograniczeniach
$$g_1 = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \quad (4)$$

$$g_2 = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$$

$$g_3 = 1 - \frac{140,45x_1}{x_2^2x_3} \leq 0$$

$$g_4 = \frac{x_2 + x_1}{1,5} - 1 \leq 0$$

gdzie:

- x_1 – średnica drutu,
- x_2 – średnia średnica zwoju,
- x_3 – liczba czynnych zwojów.

W równaniu (3) pominięto stały współczynnik $\frac{\pi^2}{4}$. Cztery ograniczenia są związane z: minimalnym ugięciem g_1 , maksymalnym naprężeniem stycznym g_2 , częstotliwością drgań własnych g_3 i średnicą zewnętrzną g_4 . Wartości zmiennych decyzyjnych wyrażone w calach [in] są przyjmowane z zakresów

$$0,05 \leq x_1 \leq 2; \quad 0,25 \leq x_2 \leq 1,3; \quad 2 \leq x_3 \leq 15 \quad (5)$$

W tabeli 1 przedstawiono rozwiązanie optymalne uzyskane dla projektowania sprężyny. Do obliczeń zastosowano rój utworzony z 30 cząstek. Każda cząstka posiadała 18 sąsiadów. Liczba kroków iteracji do zakończenia obliczeń wynosiła 100. Stałe akceleracji przyjęto jako: $c_1 = c_2 = c_3 = 0,5$. Wyniki są porównane z uzyskanymi wcześniej przez innych autorów [7–9] i [6].

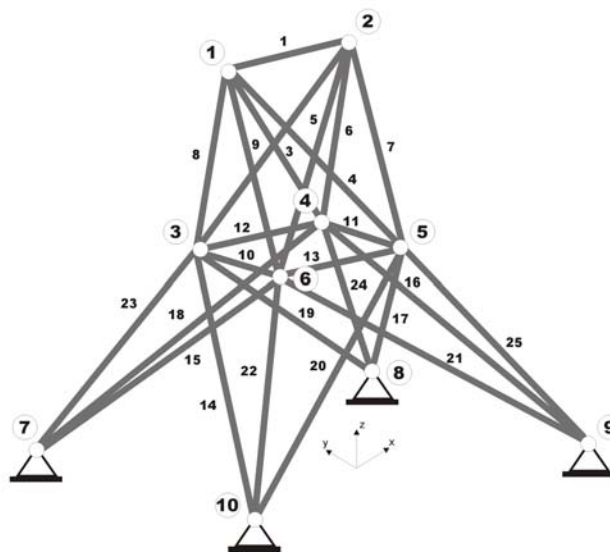
Tabela 1

Rozwiązanie optymalne projektowania sprężyny

	Wynik	[7]	[8]	[9]	[6]	SI	Wynik
x_1 [in]	0,051683	0,051690	0,052160	0,051480	0,053396	x_1 [cm]	0,131
x_2 [in]	0,356572	0,356750	0,368159	0,351661	0,399180	x_2 [cm]	0,906
x_3	11,297521	11,287126	10,648442	11,632201	9,185400	x_3	11,297
$f(x)$ [in ³]	0,0126652	0,0126653	0,0126692	0,0127048	0,0127303	$f(x)$ [cm ³]	0,208

3.2. Optymalizacja kratownicy przestrzennej – zmienne dyskretne

Projektowana jest kratownica 25-prętowa, będąca prostym modelem wieży transmisyjnej (rys. 2). Zadanie analizy rozwiązano metodą elementów skończonych MES. Pełne dane dotyczące geometrii, materiału, grup prętowych oraz przypadków obciążenia wieży przedstawiono w pracy dotyczącej optymalizacji kratownic [3].



Rys. 2. Kratownica 25-prętowa
Fig. 2. The 25-bar truss

Dyskretnych zmiennych decyzyjnych jest 8 ze względu na zaszeregowanie prętów w grupy o jednakowych przekrojach poprzecznych. Dopuszczalne wartości zmiennych decyzyjnych są przyjmowane wg następującego wyrażenia

$$x_k = 0,1n \quad n = 1, 2, \dots, 50 \quad (6)$$

co oznacza, że są to liczby całkowite z przedziału $[1, 50]$, mnożone przez wartość 0,1.

Krata pracuje pod dwoma przypadkami obciążenia. Minimalizowana jest objętość prętów kraty, przy ograniczeniach naprężeniowych i wyboczeniowych. Ograniczenia naprężeniowe są symetryczne. Łącznie zadanie ma $(25 + 25) \cdot 2$, czyli 100 ograniczeń.

Zadanie optymalizacji przedstawia się następująco:

$$\text{Minimalizuj} \quad f(\mathbf{x}) = \sum_{i=1}^{25} A_i l_i \quad (7)$$

$$\text{Przy ograniczeniach} \quad |\sigma_{ij}(\mathbf{x})| \leq \sigma_a \quad i = 1..25 \quad j = 1, 2 \quad (8)$$

$$\sigma_{ij}(\mathbf{x}) \geq \sigma_i^{\text{kr}}(\mathbf{x}) \quad i = 1..25 \quad j = 1, 2 \quad (9)$$

$$x_k^{\min} \leq x_k \leq x_k^{\max} \quad k = 1..8 \quad (10)$$

gdzie:

A_i – pole powierzchni przekroju i -tego pręta,

l_i – długość i -tego pręta,

x_k – pole powierzchni przekroju prętów k -tej grupy,

x_k^{\min} – dolne ograniczenie przekroju wynoszące $0,1[in^2]$ ($64,516[mm^2]$),

x_k^{\max} – górne ograniczenie przekroju wynoszące $5[in^2]$ ($3226[mm^2]$),

σ_{ij} – naprężenie w i -tym pręcie dla j -tego przypadku obciążenia,

σ_a – naprężenie dopuszczalne,

σ_i^{kr} – naprężenie wyboczeniowe w i -tym pręcie, określone następująco

$$\sigma_i^{\text{kr}} = \frac{-100,01\pi EA_i}{8l_i^2} \quad i = 1..25 \quad (11)$$

Wyniki optymalne przedstawiono w tab. 2 i porównano z uzyskanymi przez [10]. Żadne z ograniczeń nie jest dokładnie równe wartości dopuszczalnej, jednak te, które są najbliższe, przedstawiono w tab. 3. Oznacza to, że kolejny krok algorytmu optymalizacji wygenerowałby rozwiązanie poprawiające wartość funkcji celu, jednak naruszające jedno z tych ograniczeń.

Do obliczeń zastosowano rój utworzony z 15 cząstek. Każda cząstka miała 5 sąsiadów. Liczba kroków iteracji do zakończenia obliczeń wynosiła 100. Stałe akceleracji przyjęto jako: $c_1 = 0$, natomiast $c_2 = c_3 = 1,2$. Projektowanie zostało przeprowadzone z wykorzystaniem dwóch wersji algorytmu roju cząstek: dla zmiennych mieszanych oraz dla zmiennych całkowitych. Wersja dedykowana dla zadań programowania całkowito-liczbowego wymagała jedynie 40% czasu obliczeniowego koniecznego do uzyskania wyniku przy zastosowaniu wersji dla zmiennych mieszanych.

Tabela 2

Rozwiązanie optymalne projektowania kratownicy 25-prętowej

Wynik autora			[10]
Zmienna decyzyjna	[in ²]	[cm ²]	[in ²]
x_1	0,1	0,645	0,1
x_2	0,8	5,161	0,9
x_3	0,8	5,161	0,9
x_4	0,1	0,645	0,1
x_5	0,2	1,29	0,2
x_6	0,6	3,871	0,6
x_7	1,0	6,452	1,0
x_8	0,8	5,161	0,8
$f(x)$	239,8266 [in ³]	3930,05 [cm ³]	249,3187 [in ³]

Tabela 3

**Wartości naprężeń bliskich dopuszczalnych
w prętach kratownicy 25-prętowej**

Przypadek obciążenia	Numer pręta	Naprężenie	Naprężenie krytyczne
1	2	-1,8149035e + 004	-1,8447893e + 004
1	5	-1,8149035e + 004	-1,8447893e + 004
1	19	-1,1465515e + 004	-1,1969169e + 004
1	20	-1,1465515e + 004	-1,1969169e + 004
2	24	-1,7366470e + 004	-1,7638775e + 004

3.3. Optymalizacja modelu ramy Koitera – zachowanie pokrytyczne

Optymalizowany jest model ramy Koitera ze względu na zachowanie pokrytyczne, tj. po utracie stateczności. Zadanie sformułowano na dwa sposoby. W pierwszym odtwarzana jest założona ścieżka pokrytyczna. W drugim kształtowany jest przebieg ścieżki pokrytycznej przy ograniczeniach nałożonych na jej nachylenie. Sformułowania te obszerniej rozważano w pracy [11].

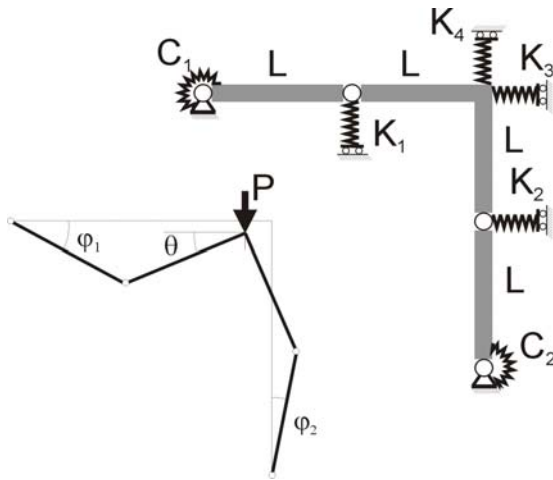
Model ramy Koitera jest sztywno-sprężystym układem o skończonej liczbie stopni swobody (rys. 3). Składa się on z dwóch sztywnych belek i sztywnej kątowej ramy. Połączenia prętów, ramy oraz połączenia z podporami są zrealizowane przez sprężyste przeguby.

Układ jest wyposażony w sprężyny skrętne o sztywnościach C_1 , C_2 oraz wzdłużne o sztywnościach K_1 , K_2 , K_3 , K_4 , gdzie sprężyny skrętne reprezentują sztywność podparcia, K_1 , K_2 – sprężystość ramy, a K_3 , K_4 – sztywność dodatkowych podparć. Zakłada się, że sprężyny wzdłużne zachowują początkową orientację w trakcie deformacji układu.

Nieliniową funkcję przedstawiającą ścieżkę pokrytyczną uzyskano z warunku stacjonarności energii potencjalnej układu. Kąt θ przyjęto jako przemieszczenie uogólnione do kontroli zachowania układu po utracie stateczności.

Energię potencjalną układu przedstawia następujące równanie

$$\begin{aligned} \Pi = & \frac{1}{2} C_1 \varphi_1^2 + \frac{1}{2} C_2 \varphi_2^2 + \frac{1}{2} K_1 (L \sin \varphi_1)^2 + \frac{1}{2} K_2 (L \sin \varphi_2)^2 + \\ & + \frac{1}{2} K_3 (2L - L \cos \theta \cos \varphi_2)^2 + \frac{1}{2} K_4 (2L - L \cos \theta \cos \varphi_1)^2 - \\ & - P(2L - L \cos \theta - L \cos \varphi_2) \end{aligned} \quad (12)$$



Rys. 3. Model ramy
Fig. 3. The model of the frame

Uwzględniając dodatkowe związki geometryczne pomiędzy kątami (prace [11, 2]) i przyrównując do zera pierwszą pochodną energii po niezależnym kącie θ , otrzymuje się funkcję opisującą nieliniową ścieżkę równowagi

$$\begin{aligned} p(\theta) = & \frac{c_1 \varphi_1 \varphi_1' + c_2 \varphi_2 \varphi_2' + (1/2) k_1 \varphi_1' \sin 2\varphi_1 + (1/2) k_2 \varphi_2' \sin 2\varphi_2}{\varphi_2' \sin \varphi_2 + \sin \theta} \\ & + \frac{k_3 (2 - \cos \varphi_1 - \cos \theta) (\varphi_1' \sin \varphi_1 + \sin \theta) + k_4 (2 - \cos \varphi_2 - \cos \theta) (\varphi_2' \sin \varphi_2 + \sin \theta)}{\varphi_2' \sin \varphi_2 + \sin \theta} \end{aligned} \quad (13)$$

gdzie

$$\varphi_1' = \frac{\sin(\varphi_2 + \theta) + \cos(\varphi_2 + \theta)}{\cos(\theta - \varphi_2)}; \quad \varphi_2' = \frac{-\sin(\varphi_1 + \theta) + \cos(\varphi_1 + \theta)}{\cos(\theta - \varphi_2)} \quad (14)$$

W powyższym wyrażeniu wprowadzono następujące wielkości bezwymiarowe (C_0 oznacza sztywność odniesienia)

$$c_i = \frac{C_i}{C_0}; \quad k_j = \frac{K_j L^2}{C_0}; \quad p = \frac{PL^2}{C_0}; \quad i = 1, 2; \quad j = 1, 2, 3, 4 \quad (15)$$

Wartość siły krytycznej dla modelu jest wyrażona przez

$$p_{kr} = \frac{1}{2} (c_1 + c_2 + k_1 + k_2) \quad (16)$$

Widać, że nie jest ona funkcją sztywności k_3 i k_4 . Sztywności te jednak wpływają na zachowanie się układu po utracie stateczności.

3.3.1. Odtwarzanie założonej ścieżki pokrytycznej

Założono określony przebieg statecznej ścieżki pokrytycznej: $p^{(0)}(\theta) = 0,00338\theta^2 + 1$. Ścieżka jest odtwarzana poprzez minimalizację błędu średniokwadratowego pomiędzy punktami leżącymi na aktualnej i założonej ścieżce pokrytycznej. Dodatkowo dodano ograniczenie na ustaloną wartość siły krytycznej.

$$\text{Minimalizuj} \quad \sum_i [P(\theta_i; k_1, k_2, k_3, k_4) - p^{(0)}(\theta_i)]^2 \quad i = 1..N \quad (17)$$

$$\text{Przy ograniczeniu} \quad p_{kr}(k_1, k_2, k_3, k_4) = 1 \quad (18)$$

gdzie:

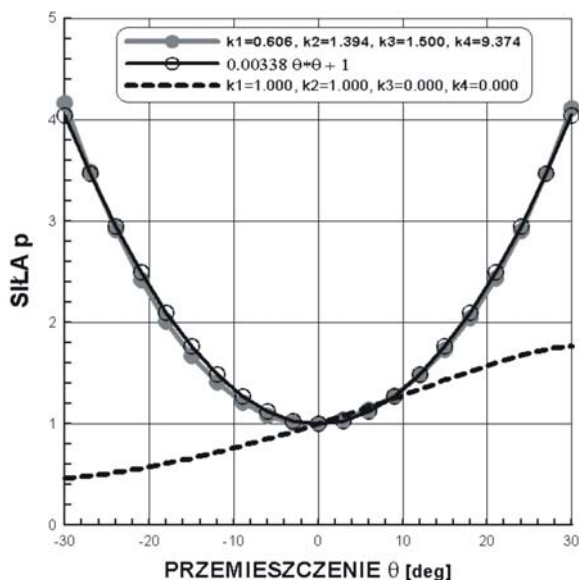
$p(\theta_i; k_1, k_2, k_3, k_4)$ – punkty na aktualnej ścieżce równowagi,

$p^{(0)}(\theta_i)$ – punkty na ścieżce założonej,

θ_i – wartości przemieszczenia uogólnionego w przyjętych punktach,

k_j – zmienne decyzyjne (sztywności sprężyn).

Wynik projektowania wraz z optymalnymi wartościami zmiennych decyzyjnych przedstawiono na rys. 4. Przerywaną linią pokazano ścieżkę równowagi ramy nieoptymalizowanej. Liniami ciągłymi przedstawiono ścieżki: czarną – ścieżkę założoną, szarą – ścieżkę odtworzoną.



Rys. 4. Odtwarzanie założonej ścieżki równowagi
Fig. 4. Reconstruction of the predicted equilibrium path

3.3.2. Kształtowanie przebiegu ścieżki pokrytycznej przy ograniczeniach nałożonych na jej nachylenie

Minimalizowana jest suma sztywności sprężyn wzdłużnych ramy. Ograniczane jest nachylenie ścieżki pokrytycznej przez zadanie dopuszczalnych wartości gradientów w wybranych punktach kontrolnych. Przepis określający wartości dopuszczalne podzielono na dwa podzbiory ze względu na wartości kąta θ . W sąsiedztwie punktu krytycznego, tj. dla $|\theta| \leq 6^\circ$, narzucono wartość gradientu $\hat{G}_i = 0$, natomiast na zewnątrz tego przedziału wartości \hat{G}_i reprezentują gradienty funkcji $0,00338 \cdot \theta^2 + 1$, użytej w przykładzie poprzednim do odtwarzania ścieżki.

$$\text{Minimalizuj} \quad k_1 + k_2 + k_3 + k_4 \quad (19)$$

Przy ograniczeniach

$$p(\theta_{j+1}; k_1, k_2, k_3, k_4) - p(\theta_j; k_1, k_2, k_3, k_4) \geq \hat{G}_{1j} \quad \text{dla} \quad \theta_j \geq 0 \quad j = 1, 2 \dots m \quad (20)$$

$$p(\theta_{k+1}; k_1, k_2, k_3, k_4) - p(\theta_k; k_1, k_2, k_3, k_4) \leq \hat{G}_{2k} \quad \text{dla} \quad \theta_k \leq 0 \quad k = 1, 2 \dots l \quad (21)$$

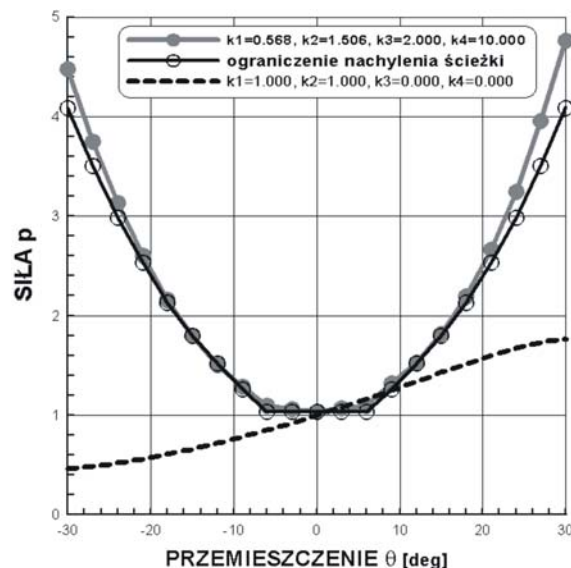
Ze względu na niesymetryczną postać ścieżki równowagi dla zadania, ograniczenia (20) i (21) postawiono osobno dla dodatnich i ujemnych wartości kąta θ , wprowadzając – odpowiednio – \hat{G}_{1j} oraz \hat{G}_{2k} . Sformułowanie powyższe wymusza taki kształt ścieżki pokrytycznej, który zapewni określone nachylenie funkcji uogólnionego kąta sterującego deformacją modelu ramy. Dodatkowo wymuszona została symetria ścieżki równowagi w punkcie krytycznym. Ukształtowana, optymalna ścieżka została przedstawiona na rys. 5.

Do rozwiązania zadania optymalizacji modelu ramy Koitera zastosowano rój 7 cząstek. Każda cząstka posiadała 3 sąsiadów. Liczba kroków iteracji do zakończenia obliczeń wynosiła 100. Przyjęto następujące wartości stałych akceleracji: $c_1 = c_2 = 1$, $c_3 = 0$.

4. Wnioski

Optymalizacja rojem cząstek (PSO) jest nowoczesną metodą przeszukiwania stochastycznego, opartą na idei zachowania społecznego w populacjach organizmów żywych. Metoda posiada pewne elementy wspólne z szeroko znanymi technikami ewolucyjnymi, takimi jak algorytmy genetyczne (*Genetic Algorithms* – GA) oraz programowanie ewolucyjne (*Evolutionary Programming* – EP). W porównaniu z nimi ma jednak istotną zaletę. Cząstki, pamiętając najlepsze znalezione wcześniej położenia, dysponują pseudogradientową informacją o przestrzeni poszukiwania. Nie jest to więc metoda rzędu zerowego w ścisłym znaczeniu tego słowa. Dodatkowo przez sprzężenie układu *równań aktualizacji* cząstki dzielą się tą wiedzą pomiędzy sobą. Prowadzi to w konsekwencji do znacznego ograniczenia liczby niezbędnych wywołań funkcji dopasowania (*fitness*) w trakcie procesu poszukiwania rozwiązania. Zaleta ta staje się szczególnie istotna w rozwiązywaniu zadań wymagających znacznych nakładów obliczeniowych na wyznaczenie wartości funkcji celu i/lub ograniczeń (np. zadania z analizą MES).

W minionych kilkunastu latach powstało wiele algorytmów optymalizacyjnych bazujących na metodzie roju cząstek. Jednym z nich jest algorytm przedstawiony w niniejszym



Rys. 5. Kształtowanie ścieżki równowagi przy ograniczeniach nałożonych na jej nachylenie

Fig. 5. Modeling of the equilibrium path subject to constraints imposed on its slope

artykule. Algorytm jest przystosowany do rozwiązywania zadań programowania nieliniowego. W tej klasie zadań optymalizacji mieści się wiele problemów optymalnego projektowania elementów konstrukcji. Dobór przykładów pokazuje szerokie spektrum możliwości obliczeniowych algorytmu. Zmienne projektowe w przedstawionych zadaniach są mieszane: ciągłe, dyskretne oraz całkowite. Nieliniowe funkcje celu i ograniczeń mają jawną lub niejawną postać. Optymalne rezultaty zadań projektowych pokazują, że algorytm znajduje najlepsze rozwiązania notowane w literaturze. Predysponuje to opracowany algorytm do szerokiego stosowania w zadaniach praktycznej optymalizacji inżynierskiej. W szczególności dotyczy to nowo formułowanych zadań optymalizacji o nieznanym dotąd rozwiązaniu.

Literatura

- [1] Kennedy J., Eberhart R.C., *Particle Swarm Optimization*, Proc. IEEE Int. Conf. On Neural Networks, Piscataway, NJ 1995, 1942-1948.
- [2] Foryś P., *Numerical optimization with particle swarms*, Proc. of 10th International Conference on Numerical Methods in Continuum Mechanics, Zilina, August 23–26, 2005, (CD-ROM), 13.
- [3] Foryś P., *A modified particle swarm optimizer applied to mixed variable design of truss structures*, Proc. of 17th International Conference on Computer Methods In Mechanics, CMM-2007, June 19–22, Łódź–Spała 2007, (CD-ROM), 8.
- [4] Foryś P., *Nowy algorytm optymalizacji rojem cząstek i jego zastosowanie w kształtowaniu elementów konstrukcji*, praca doktorska, Politechnika Krakowska, 2007, www.optalg.com.

- [5] Bochenek B., Foryś P., *Particle swarms in engineering design problems*, C.A. Mota Soares et.al. (eds.), III European Conference on Computational Mechanics, Solids, Structures and Coupled Problems in Engineering, Lisbon 5–8 June 2006.
- [6] Arora J.S., *Introduction to optimum design*, McGraw–Hill, New York 1989.
- [7] He S., Prempan E., Wu Q.H., *An improved particle swarm optimiser for mechanical design optimization problems*, Eng. Opt. 36, 5, 2004, 585-605.
- [8] Ray T., Liew K.M., *Society and civilization: An optimisation algorithm based on the simulation of social behaviour*, IEEE Transactions on Evolutionary Computation 7, 4, 2003, 386-396.
- [9] Coello C.A., *Use of a self-adaptive penalty approach for engineering optimization problems*, Computers in Industry 41, 4, 2000, 113-127.
- [10] Rao S.S., Xiong Y., *A hybrid genetic algorithm for mixed-discrete design optimization*, Journal of Mechanical Design, Trans. ASME, 127, 2005, 1100-1112.
- [11] Bochenek B., Foryś P., *Structural optimization for post-buckling behaviour using particle swarms*, Struct Multidisc Optim. 32, 6, 2006, 521-531.