

SAMOADAPTACYJNA METODA ALGORYTMÓW GENETYCZNYCH W OPTYMALIZACJI PRZESTRZENNYCH KRATOWNIC

Krzysztof Grygierek^{1a}

¹Katedra Mechaniki i Mostów, Politechnika Śląska

^a*krzysztof.grygierek@polsl.pl*

Streszczenie

W artykule przedstawiono optymalizację masy przestrzennych kratownic stalowych metodą algorytmów genetycznych w połączeniu z metodą elementów skończonych. W metodzie algorytmów genetycznych zastosowano kodowanie całkowitoliczbowe dyskretnego zbioru zmiennych projektowych oraz samoadaptacyjną metodę opartą na arytmetyce logiki rozmytej mającą poprawić jakość i szybkość optymalizacji. W przykładach zaprezentowano efektywność zaproponowanej metody do optymalizacji masy kratownic z uwzględnieniem dopuszczalnych przemieszczeń i naprężeń.

Słowa kluczowe: algorytmy genetyczne, metoda elementów skończonych, kratownice

SELF-ADAPTIVE METHOD OF GENETIC ALGORITHM IN OPTIMIZATION OF SPATIAL TRUSS STRUCTURES

Summary

In the article genetic algorithm method in combination with finite element method was used for the optimization of the weight of spatial steel truss structures. Integer encoding of a discrete set of design variables and self-adaptive method based on the fuzzy logic mechanism for improving the quality and speed of optimization were applied in the method of genetic algorithm. The examples present the efficiency of proposed method for weight optimization of spatial trusses taking into account allowable displacements and stresses.

Keywords: genetic algorithm, finite element method, truss

1. WSTĘP

Koszt materiału jest jednym z głównych kryteriów doboru rodzaju konstrukcji nowych obiektów. Oczywiście, aby obniżyć ten wskaźnik, powinno się dążyć do redukcji masy lub objętości materiału. Wiele metod oraz algorytmów zostało rozwiniętych w ostatnich kilkudziesięciu latach w celu optymalnego projektowania zagadnień technicznych [3]. Większość z nich stosuje metody programowania matematycznego oraz kryteria optymalności dla ciąglego zakresu zmiennych projektowych. W praktyce w większości przypadków jest to związane z dyskretnym zbiorem zmiennych projektowych, a tym samym ze standaryzacją rozmiarów elementów wytwarzanych przez producentów.

Algorytmy genetyczne są to algorytmy poszukiwania oparte na mechanizmach doboru naturalnego oraz dziedziczności [10]. Łącząc w sobie ewolucyjną zasadę przeżycia najlepiej przystosowanych z systematyczną, choć losową wymianą informacji, tworzą metodę poszukiwań. Wykorzystują przy tym doświadczenia poprzednich pokoleń do określenia nowych obszarów poszukiwań o spodziewanej wyższej wydajności. W ostatnich dwóch dekadach algorytmy genetyczne, mimo że nie gwarantują otrzymania globalnego optimum, stały się zaawansowanym narzędziem optymalizacji dla szerokiego spektrum zagadnień inżynierskich. W budownictwie prace naukowe dotyczą głównie optymalizacji konstrukcji kratowych i ramowych. Oto niektóre z nich. W pracy [6]

przedstawiono optymalizację kratownic płaskich i przestrzennych, a w [9] kratownic i ram. Optymalizację z uwzględnieniem nieliniowości w stalowych ramach opisano w artykułach [8, 12]. W pracy [1] poddano optymalizacji stalowe ramy o węzłach podatnych, zaś w artykule [5] stalowe mostowe łuki kratownicowe. Monolityczną ramową konstrukcję żelbetową zoptymalizowano w artykule [14], a w artykule [4] m.in. schody. Powyższe artykuły zajmują się optymalizacją wagi lub objętości konstrukcji. W literaturze problemu znajdziemy również pozycje, które optymalizują kształt kratownic [11, 15]. Algorytmy genetyczne, ze względu na ich bardzo wolną zbieżność, są bardzo czasochłonne zwłaszcza w przypadku złożonych problemów. Stąd poprawa efektywności jest tematyką wielu artykułów naukowych. Wprowadzono kodowanie całkowitoliczbowe dla dyskretnych zbiorów zmiennych projektowych [2, 7]. Przeanalizowano wpływ różnych operatorów krzyżowania i mutacji na osiągnięte wyniki, zaś w artykule [17] operatory te są automatycznie zamieniane podczas kolejnych kroków algorytmu, w zależności od aktualnego rozproszenia osobników. Wprowadzono zmienne, w czasie trwania symulacji, prawdopodobieństwa krzyżowania i mutacji, współczynniki stosowane w funkcjach kary, liczbę osobników. Tutaj najbardziej popularna jest metoda samoadaptacyjna (self-adaptive method). W metodzie tej wyniki są obliczane z wyrażenń zależnych od wartości funkcji przystosowania [16] lub za pomocą arytmetyki rozmytej logiki [13].

W artykule przedstawiono nowe połączenie kodowania całkowitoliczbowego z metodą samoadaptacyjną opartą na arytmetyce rozmytej. Umożliwia ono, w połączeniu z MES, efektywne optymalizowanie konstrukcji o dyskretnym zbiorze optymalizowanych wartości. W zaproponowanym algorytmie wszystkie podstawowe parametry sterujące przebiegiem procesu optymalizacji są dobierane automatycznie. Zmodyfikowano również operację mutowania, która stała się równoważną z krzyżowaniem. Zaproponowany algorytm przetestowano na przykładach przestrzennych kratownic.

2. ALGORYTMY GENETYCZNE

W algorytmach genetycznych, ze względu na ich stochastyczną naturę, głównym problemem jest powolna zbieżność, a co za tym idzie, czasochłonność. W celu poprawy zbieżności w artykule wprowadzono pewne mechanizmy mające usprawnić ten proces. Są one opisane w następnych punktach.

2.1 KODOWANIE

W metodzie algorytmów genetycznych można zastosować różne metody kodowania. W optymalizacji konstrukcji stalowych w artykułach przewodzi kodowanie binarne, gdzie ciąg znaków zero-jedynkowych opisuje cechy charakterystyczne optymalizowanej

konstrukcji. Ten klasyczny rodzaj kodowania ma niestety pewne wady, które uwidaczniają się szczególnie w przypadku dyskretnego zbioru zmiennych projektowych. W przypadku kratownic takim zbiorem może być typoszereg wybranych kształtowników stalowych, w którym każdemu kolejnemu numerowi przyporządkowane są charakterystyki geometryczno-wytrzymałościowe odpowiedniego profilu. Można wtedy zastosować kodowanie całkowitoliczbowe [2, 7]. Do każdej liczby przyporządkowany jest jeden kształtownik, a co za tym idzie – odpowiednie właściwości jego przekroju. W stosunku do kodowania binarnego w kodowaniu tym chromosom jest opisany za pomocą krótszego łańcucha znaków oraz odpada konieczność dekodowania, ponieważ genotyp jest identyczny z fenotypem. Oszczędza się w ten sposób pamięć komputera oraz czas symulacji. Nie występuje problem zgodności rozmiarów zakresu zmiennych projektowych i opisującego je łańcucha binarnego. Unika się klifu Hamminga (w niektórych przypadkach potrzebna jest znaczna zmiana łańcucha binarnego dla niewielkiej zmiany parametru), który spowalnia zbieżność algorytmu genetycznego. Ze względu na powyższe zalety, w artykule do optymalizacji kratownic o dyskretnym zbiorze zmiennych projektowych zastosowano kodowanie całkowitoliczbowe.

2.2 KRZYŻOWANIE I MUTACJA

W algorytmie zastosowano krzyżowanie jednorodne, które w kodowaniu całkowitoliczbowym przebiega analogicznie do krzyżowania w kodowaniu binarnym. W krzyżowaniu binarnym wymieniane fragmenty chromosomów przez rodziców mogą prowadzić do rozrywania „dobrych” genów i przez to do zmniejszania funkcji przystosowania ich dzieci. Taki proces nie zachodzi w krzyżowaniu całkowitoliczbowym. Wymianie ulegają zawsze całe geny. Z drugiej strony, taki przebieg operacji krzyżowania powoduje, że operuje się wyłącznie na genach wylosowanych w pierwszej populacji, co znacznie spowalnia proces optymalizacji. Za wprowadzenie nowego materiału do procesu optymalizacji całkowicie odpowiada mutacja, która staje się operacją równorzędną do krzyżowania. Zaproponowana w artykułach [2, 7] operacja mutowania polegająca na dodaniu lub odjęciu wartości jeden do mutowanego genu wydaje się zbyt zachowawcza (potrzeba wielu takich operacji, żeby znacznie zmienić wartość genu). Większe i zmienne w czasie ewolucji wartości dodawane do mutowanego genu powinny przynieść lepszy efekt na proces zbieżności. Oczywiście, innym rozwiązaniem jest zwiększenie rozmiaru populacji chromosomów, co jednak znacznie wydłuży czas działania algorytmu. W artykule zastosowano samoadaptacyjną operację mutowania. Szerzej zostanie ona opisana w kolejnym rozdziale.

2.3 STEROWANIE ALGORYTMEM

Wiele parametrów wpływa na jakość i szybkość procesu optymalizacyjnego w algorytmach genetycznych np.: wybór funkcji przystosowania, definicja operacji krzyżowania i mutacji, rozmiar populacji, liczba pokoleń, prawdopodobieństwo operacji krzyżowania i mutacji. W klasycznym algorytmie genetycznym wszystkie te parametry są stałe i zdefiniowane przed rozpoczęciem algorytmu. Wiele prac naukowych [13,16] pokazuje jednak, że lepsze rezultaty można otrzymać, stosując zmienne w trakcie trwania symulacji parametry. Niestety duża liczba parametrów komplikuje proces sterowania. Zależą one od wielu czynników, a relacje pomiędzy nimi i ich wpływ na właściwości algorytmu genetycznego są bardzo skomplikowane. Dlatego też w większości metod wyizolowane są pojedyncze parametry, które są modyfikowane w kolejnych pokoleniach. Przykładowo w artykule [7] prawdopodobieństwo krzyżowania i mutacji zależy od funkcji przystosowania, a w pracy [17] operacje krzyżowania i mutacji zależą od rozproszenia chromosomów. Prawdopodobieństwo krzyżowania i mutacji oraz liczba bitów biorących udział w operacji mutacji są sterowane w [13]. Parametry te uzależniono od funkcji przystosowania danego chromosomu, średniej funkcji przystosowania całej populacji, kroku symulacji oraz liczby kroków, w których nie uzyskano poprawy wyników. Jak widać, badacze stosują różne parametry wejściowe do obliczania wartości parametrów odpowiedzialnych za przebieg procesu optymalizacyjnego. Kolejnym problemem jest brak ścisłych zależności pomiędzy parametrami wejściowymi i wyjściowymi. W artykule [16] zastosowano metodę samoadaptacyjną, w której zaproponowano ścisłe zależności. Jako dane wejściowe przyjęto wartości funkcji przystosowania, jako parametry wyjściowe prawdopodobieństwo mutacji i krzyżowania. Zaproponowane zależności matematyczne bazują wyłącznie na obserwacji przebiegu algorytmu genetycznego i wynikach testów. Sformułowanie ścisłych związków, potrzebnych do uwzględnienia więcej danych wejściowych, jest praktycznie niemożliwe. Interesującą alternatywą dla takiego przypadku jest zastosowanie logiki rozmytej [18]. Algebra rozmyta jest stosunkowo prostą i efektywną metodą służącą do opisu złożonych relacji szczególnie wówczas, kiedy relacje te bazują na obserwacji i wynikach eksperymentu.

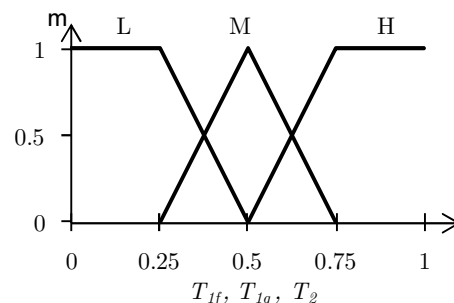
Za rozproszenie chromosomów odpowiedzialne są operacje krzyżowania i mutacji. Prawdopodobieństwa krzyżowania P_C i mutacji P_M decydują o tym, jak często chromosomy poddane są tym operacjom w procesie ewolucji. Z jednej strony duże P_C i P_M umożliwiają lepsze przeszukanie przestrzeni zadania i stąd zwiększa się prawdopodobieństwo znalezienia lepszego rozwiązania, a nawet globalnego optimum. Z drugiej

strony, duże P_C i P_M zwiększają prawdopodobieństwo zniszczenia i rozbitcia dobrych chromosomów, spowalniając proces optymalizacji i zbieżności. Podobny wpływ na proces ewolucyjny będzie miała wartość S , która określa maksymalną wartość, jaką można dodać do mutowanego genu. Badania naukowe pokazują, że stosowanie zmiennych wartości, dostosowanych do aktualnej sytuacji procesu, znacznie zwiększają efektywność i jakość rozwiązania [13, 17]. Problem jednak polega na tym, że nie ma ścisłych zależności matematycznych potrzebnych do obliczenia powyższych wartości.

W artykule przyjęto, że danymi wejściowymi do obliczenia P_C , P_M oraz S będą aktualne rozproszenie populacji, jakość chromosomu oraz krok algorytmu. W większości artykułów miarą rozproszenia populacji jest rozrzut funkcji przystosowania. Można jednak pokazać przykłady, w których chromosomy mające bardzo zbliżony getotyp różnią się stosunkowo mocno wartościami funkcji przystosowania oraz odwrotnie. Dlatego w pracy przyjęto dwie miary rozproszenia populacji. Pierwsza miara opiera się na funkcji przystosowania [13]:

$$T_{1f} = (f_{max} - f_{avg}) / (f_{max} - f_{min}), \quad (1)$$

gdzie, przez f_{max} , f_{avg} , f_{min} oznaczono odpowiednio maksymalną, średnią i minimalną wartość funkcji przystosowania w populacji. Drugą miarą jest rozproszenie genotypu T_{1g} . Zakres zmiennych został podzielony na przedziały i sprawdza się, czy w całej populacji przynajmniej raz występuje wartość z przedziału w każdym genie. Obydwie miary przyjmują wartości z zakresu [0,1]. Duże wartości oznaczają znaczne rozproszenie chromosomów. Duże P_C , powinno pomóc w uzyskaniu lepszej zbieżności. W przeciwnym wypadku chromosomy mogą zbiegać do lokalnego optimum, a duże P_M zwiększy ich rozproszenie. Z drugiej strony małe S umożliwi znalezienie tego optimum (może ono się okazać globalnym optimum).



Rys. 1. Funkcje przynależności dla danych wejściowych

Jeśli przez f oznaczy się funkcję przystosowania pewnego chromosomu, to współczynnik:

$$T_2 = (f_{max} - f) / (f_{max} - f_{min}), \quad (2)$$

określa jakość tego chromosomu w całej populacji. Małe wartości oznaczają, że jest to bardzo dobry chromosom

i dlatego stosuje się małe P_C , P_M i S , aby ochronić go przed zniszczeniem. W przeciwnym wypadku zastosuje się duże wartości, żeby go znacząco zmienić.

W opisie wpływu powyższych parametrów na wartości sterowane pojawiły się nieprecyzyjne wartości typu: małe, duże. O ile sformułowania takie są zrozumiałe w potocznej mowie, o tyle zamiana ich na konkretne wartości jest już mocno kłopotliwa. Z pomocą przychodzi w takich nieprecyzyjnych sformułowaniach system logiki rozmytej. Zbiory rozmyte mają za zadanie matematyczną reprezentację niejednoznaczności i niepewności. Podstawowym celem logiki opartej na zbiorach rozmytych jest naśladowanie ludzkiego sposobu wnioskowania.

Tab. 1. Baza reguł rozmytych

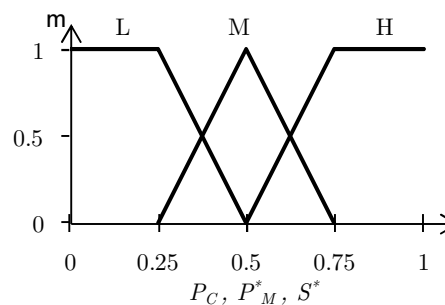
	T_{I_g}	T_{I_f}	T_2	P_C	P_M	S
1	L	L	L	L	L	L
2	M	L	L	L	M	L
3	H	L	L	L	M	L
4	L	M	L	L	L	L
5	M	M	L	L	L	L
6	H	M	L	L	L	L
7	L	H	L	L	L	M
8	M	H	L	M	L	M
9	H	H	L	M	L	M
10	L	L	M	L	H	L
11	M	L	M	L	H	L
12	H	L	M	M	H	L
13	L	M	M	M	M	M
14	M	M	M	M	M	M
15	H	M	M	M	M	M
16	L	H	M	M	L	H
17	M	H	M	H	L	H
18	H	H	M	H	M	H
19	L	L	H	M	H	M
20	M	L	H	M	H	M
21	H	L	H	H	H	M
22	L	M	H	H	H	H
23	M	M	H	H	H	H
24	H	M	H	H	H	H
25	L	H	H	H	M	H
26	M	H	H	H	M	H
27	H	H	H	H	H	H

Sterownik rozmyty składa się z trzech podstawowych części (bloków) [18]: fuzyfikacji (rozmywanie), wnioskowania (interferencja) i defuzyfikacji (ostrzenie). W bloku fuzyfikacja przeprowadzana jest operacja rozmywania, czyli obliczania stopnia przynależności do poszczególnych zbiorów rozmytych, otrzymanych na wejściu wartości liczbowych T_{I_f} , T_{I_g} , T_2 . Aby operację tę przeprowadzić, blok ten musi posiadać dokładnie zdefiniowane funkcje przynależności do zbiorów rozmytych poszczególnych wejść. Przyjęto, że dane wejściowe T_{I_f} , T_{I_g} , T_2 są określone przez zbiór

zmiennych lingwistycznych: małe (L), średnie (M), duże (H). Założono, że wszystkie dane wejściowe są określone za pomocą takich samych zbiorów rozmytych (rys. 1). Dzięki funkcji przynależności w bloku fuzyfikacji zamieniono wartości liczbowe na zmienne lingwistyczne stosowane w kolejnych blokach. Etap wnioskowania, na podstawie wejściowych stopni przynależności, wyznacza wynikową funkcję przynależności. Funkcja ta ma często złożony kształt, a jej obliczanie odbywa się w drodze tzw. inferencji. Moduł wnioskowania działa na podstawie bazy reguł rozmytych. Składa się na nią zbiór instrukcji warunkowych, które powstają na podstawie wiedzy eksperta i określają zależności przyczynowo-skutkowe istniejące w systemie pomiędzy zbiorami rozmytymi wejść i wyjść. W tym przypadku reguły te przedstawiono w tabeli 1, na podstawie przesłanek opisanych powyżej. Reguły są skonstruowane na zasadzie IF... THEN..., np.: pierwsza reguła to:

IF ($T_{I_g}=L$) AND ($T_{I_f}=L$) AND ($T_2=L$)
THEN $P_C=L$, $P_M=L$, $S=L$

Przyjęte funkcje przynależności dla zmiennych wyjściowych pokazano na rys. 2 ($P_M=P_M^* \cdot 0,4/L_{CH}$, $S=S^* \cdot 0,6P_z$, L_{CH} - liczba genów w chromosomie, P_z - liczba dopuszczalnych przekrojów). Na podstawie stopni aktywizacji konkluzji poszczególnych reguł, stosując metodę MIN-MAX [18], określono wynikowe postacie funkcji przynależności wyjść. W etapie ostrzenia, z obliczonych rozmytych zbiorów wyjść, wyznaczono dokładną wartość wyjściową (liczba) stosowaną w dalszych obliczeniach. W defuzyfikacji zastosowano metodę środka ciężkości.



Rys. 2. Funkcje przynależności dla funkcji wyjścia

Wartość, która jest dodawana do mutowanego genu, jest losowo wybierana z przedziału $max([-1,1], [-S, S] \cdot k)$, gdzie współczynnik k zależy od kroku algorytmu i zmienia się liniowo od 1 na początku symulacji do 0 w $\frac{3}{4}$ kroków. Parametr ten mógłby zostać wprowadzony bezpośrednio do sterownika rozmytego. Jednak, jeśli przyjęto by dla niego analogiczne krzywe przynależności jak na rys. 2, to trzykrotnie wzrosłaby liczba reguł w tab. 1. Dlatego w celu uproszczenia mechanizmu sterowania zdecydowano się tutaj na takie uproszczenie.

3. PROCES OPTYMALIZACJI

Podstawowym kryterium przy projektowaniu konstrukcji jest osiągnięcie jak najmniejszego ciężaru. W pracy tej optymalizacja przestrzennych kratownic przebiega tak, by osiągnąć minimalny ciężar konstrukcji. Z tego powodu funkcja celu W została zdefiniowana jako:

$$\min W = \sum_{i=1}^{lp} \rho_i L_i A_i, \quad (3)$$

gdzie ρ_i - to ciężar właściwy materiału, L_i - długość pręta, A_i - pole przekroju poprzecznego pręta, lp - liczba prętów. Optymalizowana konstrukcja musi spełniać również ograniczenia związane z dopuszczalnymi przemieszczeniami i naprężeniami:

$$\delta_i \leq \delta_u, \quad i = 1, 2, \dots, p \quad (4)$$

$$\sigma_j \leq \sigma_u, \quad j = 1, 2, \dots, lp \quad (5)$$

gdzie δ_i, δ_u - to odpowiednio obliczone w MES przemieszczenia oraz przemieszczenia dopuszczalne, p - liczba węzłów, w których są ograniczone przemieszczenia, σ_i, σ_u - obliczone i dopuszczalne naprężenia.

Ponieważ algorytmy genetyczne są metodami rozwiązującymi zadania optymalizacyjne bez ograniczeń,

dlatego przekształcono rozwiązywane zadanie z ograniczeniami na zadanie bez ograniczeń poprzez obciążenie funkcji celu funkcją kary. W zastosowanym algorytmie najlepszy osobnik przechodzi do następnej iteracji bez zmian (elitaryzm), co gwarantuje nie pogorszenie wyniku w kolejnych krokach.

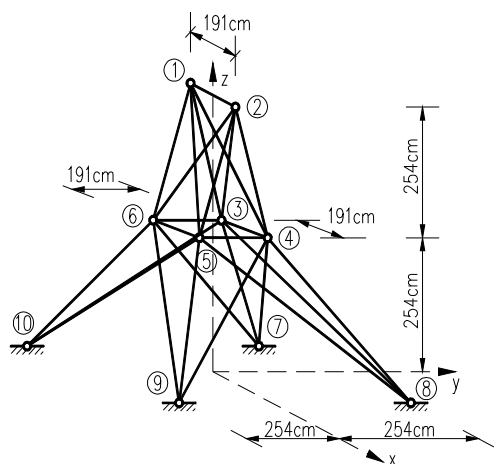
4. PRZYKŁADY NUMERYCZNE

Zaproponowany algorytm może być stosowany do szerokiej klasy zagadnień, w której zmienne projektowe należą do dyskretnego zbioru.

Efektywność algorytmu sprawdzono na przykładach optymalizacji dwóch przestrzennych kratownic: A - 25 prętów, B- 72 pręty. Przykłady te są optymalizowane w wielu artykułach, zarówno dla zmiennych ciągłych jak i dyskretnych. W większości artykułów zostały przyjęte jednostki anglosaskie, które tutaj zostały przeliczone na układ SI. Obliczenia przeprowadzono w autorskim programie. Przyjęto rozmiar populacji 40 oraz 200 kroków iteracji. Schemat kratownicy A przedstawiono na rys. 3. Składa się ona z 25 prętów, kolejno ograniczonymi węzłami: 1-2, 1-4, 2-3, 1-5, 2-6, 2-4, 2-5, 1-3, 1-6, 6-3, 4-5, 3-4, 6-5, 3-10, 6-7, 4-9, 5-8, 4-7, 3-8, 5-10, 6-9, 6-10, 3-7, 4-8, 5-9. Pręty podzielono na 8 grup. Wszystkie przyjęte założenia zestawiono w tabeli 2.

Tab. 2. Dane dla kratownicy A i B

	Kratownica A					Kratownica B							
	moduł sprężystości $E=68950$ MPa, ciężar właściwy $\rho=0,02714$, N/cm ³												
Dopuszczalne naprężenia, MPa	275,79					172,36							
Dopuszczalne przemieszczenia, cm	0,889					0,635 w węzłach 17, 18, 19, 20							
Dopuszczalne przekroje poprzeczne, cm ²	0,65 1,29 1,94 2,58 3,23 3,87 4,51 5,16 5,81 6,45 7,10 7,74 8,39 9,03 9,68 10,32 10,97 11,61 12,26 12,90 13,55 14,19 14,84 15,48 16,13 16,77 18,07 19,36 20,65 21,94					1,12 1,42 1,45 1,74 1,85 2,26 2,67 2,78 3,08 3,28 3,79 3,87 4,30 4,48 4,80 5,69 5,86 6,31 6,56 6,91 8,24 8,70 9,03 9,40 10,10 11,00 11,50 11,90 12,30 13,20 14,10 14,30							
Grupy prętów nr grupy - nr prętów	1	1				1	1, 2, 3, 4				9	37, 38, 39, 40	
	2	2, 3, 4, 5				2	5, 6, 7, 8, 9, 10, 11, 12				10	41, 42, 43, 44, 45, 46, 47, 48	
	3	6, 7, 8, 9				3	13, 14, 15, 16					49, 50, 51, 52	
	4	10, 11				4	17, 18				11	53,54	
	5	12, 13				5	19, 20, 21, 22				12	55, 56, 57, 58	
	6	14, 15, 16, 17				6	23, 24, 25, 26, 27, 28, 29, 30				13	59, 60, 61, 62, 63, 64, 65, 66	
	7	18, 19, 20, 21				7	31, 32, 33, 34				14	67, 68, 69, 70	
	8	22, 23, 24, 25				8	35, 36				15	71,72	
Obciążenia, kN	Gr	W	X	Y	Z	Gr	W	X	Y	Z			
grupa (Gr) - nr	1	1	4,448	-44,48	-44,48	1	17	22,24	22,24	-22,24			
węzła (W) - obciążenia w kierunku osi X, Y, Z		2	0	-44,48	-44,48	2	17-20	0	0	-22,24			
		3	2,224	0	0								
		6	2,67	0	0								



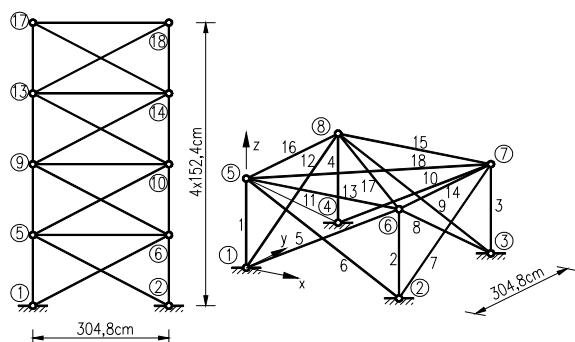
Rys. 3. Schemat kratownicy A

Porównanie otrzymywanych wyników dla tego przykładu można znaleźć w artykule [7]. Ciężar konstrukcji otrzymany programem autorskim jest identyczny z najlepszym rezultatem tam prezentowanym. Przekroje dla poszczególnych grup zestawiono w tabeli 3.

Tab. 3. Optymalne przekroje poprzeczne kratownicy A.

Pole, cm ²	Dede [7]	Autorskie
A ₁	0,65	0,65
A ₂	1,94	1,94
A ₃	21,4	21,4
A ₄	0,65	0,65
A ₅	13,55	13,55
A ₆	6,45	6,45
A ₇	3,23	3,23
A ₈	21,94	21,94
Ciężar, kN	2,157	2,157

Schemat kratownicy B przedstawiono na rys. 4. Zbudowana jest ona z 72 przętów podzielonych na 16 grup i jest obciążona dwoma grupami obciążeń. Dane przyjęte w obliczeniach zestawiono w tabeli 2.



Rys. 4. Schemat kratownicy B

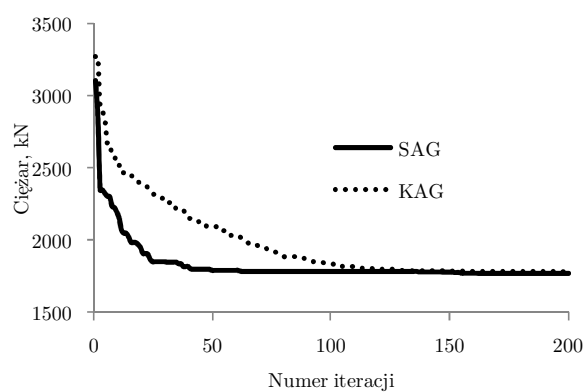
Porównanie wyników zestawiono w tabeli 4. Ciężar konstrukcji otrzymany z zastosowaniem przedstawionej powyżej metody jest około 2% mniejszy od najlepszego prezentowanego w literaturze. Otrzymano go przy

stosunkowo niewielkiej populacji 40 osobników (w literaturze stosowano od 80 do 100 osobników), co może świadczyć o wysokiej efektywności zaproponowanego algorytmu.

Tab. 4. Optymalne przekroje poprzeczne kratownicy B.

Pole, cm ²	Dede [7]	Autorskie
A ₁	13,20	12,30
A ₂	3,79	3,28
A ₃	1,12	1,12
A ₄	1,12	1,12
A ₅	6,31	8,24
A ₆	3,79	3,28
A ₇	1,12	1,12
A ₈	1,12	1,12
A ₉	3,28	3,28
A ₁₀	3,08	3,28
A ₁₁	1,12	1,12
A ₁₂	1,12	1,12
A ₁₃	1,12	1,12
A ₁₄	3,79	3,79
A ₁₅	2,26	2,26
A ₁₆	3,28	3,79
Ciężar, kN	1,806	1,774

Na rys. 5 przedstawiono porównanie zbieżności klasycznego algorytmu genetycznego (KAG) z zaproponowanym tutaj samoadaptacyjnym algorytmem genetycznym (SAG). W przypadku KAG przyjęto stałe $P_C=0.7$, $P_M=0.1$ oraz $S=1$ kodowanie całkowitoliczbowe oraz elitaryzm. Z rysunku tego jednoznacznie można odczytać znacznie szybszą zbieżność algorytmu samoadaptacyjnego. Najniższy ciężar, jaki uzyskano, stosując KAG, to 1780 kN i jest on większy od uzyskanego SAG.



Rys. 5. Porównanie zbieżności

5. PODSUMOWANIE

Efektywność i zbieżność optymalizacji w metodzie algorytmów genetycznych zależy od wielu parametrów powiązanych ze sobą trudnymi do opisanego, w sposób dokładny, zależnościami. Dlatego w artykule do podniesienia efektywności algorytmu zastosowano sterownik rozmyty, gdzie nieprecyzyjne przesłanki są

zamieniane na liczby umożliwiające sterowanie procesem. W połączeniu z kodowaniem całkowitoliczbowym, które rozwiązuje szereg problemów pojawiających się w klasycznym kodowaniu binarnym oraz MES, powstało efektywne narzędzie do optymalizacji konstrukcji inżynierskich. W kodowaniu całkowitoliczbowym skraca się czas dekodowania, długość genów w chromosomach, która w tym przypadku nie zależy od liczby zmiennych projektowych. Nie ma również problemu ze zgodnością rozmiarów liczby zmiennych projektowych z ciągiem znaków

binarnych oraz unika się klifu Hamminga. Zaprezentowane przykłady pokazują, że zastosowanie sterownika rozmytego znacznie przyspiesza zbieżność algorytmu, a otrzymane wyniki są takie same lub lepsze od prezentowanych w literaturze.

W artykule optymalizowano wyłącznie ciężar konstrukcji. Dalsze prace będą próbą rozbudowy zaprezentowanego tutaj algorytmu na równoczesną optymalizację topologii i ciężaru kratownic.

Literatura

1. Ali N. B. H., Sellami M., Cutting-Decelle A.F., Mangin J. C.: Multi-stage production cost optimization of semi-rigid steel frames using genetic algorithms. "Engineering Structures" 2009, 31, p. 2766 – 2778.
2. Bekiroglu S., Dede T., Ayvaz Y.: Implementation of different encoding types on structural optimization based on adaptive genetic algorithm. "Finite Elements in Analysis and Design" 2009, 45, p. 826 – 835.
3. Blachut J., Eschenauer H.A.: Emerging methods for multidisciplinary optimization. CISM No. 425. Wien, New York: Springer, 2001.
4. Burczyński T., Poteralski A.: Advanced evolutionary optimization of 3-D structures. European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004.
5. Cheng J.: Optimum design of steel truss arch bridges using a hybrid genetic algorithm. "Journal of Constructional Steel Research" 2010, 66, p. 1011 – 1017.
6. Coelle C. A., Christiansen A.D.: Multiobjective optimization of trusses using genetic algorithms. "Computers and Structures" 2000, 75, p. 647 – 660.
7. Dede T., Bekiroglu S., Ayvaz Y.: Weight minimization of trusses with genetic algorithm. "Applied Soft Computing", 2011, 11, p. 2565 – 2575.
8. Degertekin S. O., Saka M. P., Hayalioglu M.S.: Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm. "Engineering Structures" 2008, 30, p. 197-205.
9. Erbatur F., Hasancebi O., Tutuncu I., Kilic H.: Optimal design of planar and space structures with genetic algorithms. "Computers and Structures" 2000, 75, p. 209 – 224.
10. Goldberg D. E.: Algorytmy genetyczne i ich zastosowania. Warszawa: WNT, 1995.
11. Hajela P., Lee E.: Genetic algorithms in truss topological optimization. "Int. J. Solids Structures" 1995, Vol. 32, No. 22, p. 3341 – 3357.
12. Hayalioglu M. S.: Optimum design of geometrically non-linear elastic-plastic steel frames via genetic algorithm. "Computers and Structures" 2000, 77, p. 527 – 538.
13. Hu X.B., Wu S.F.: A self-adaptive genetic algorithm based on fuzzy mechanism. In: 2007 IEEE Congress on Evolutionary Computation, p. 4646 – 4652.
14. Kwak H-G., Kim J.: An integrated genetic algorithm complemented with direct search for optimum design of RC frames. "Computer-Aided Design" 2009, 41, p. 490 – 500.
15. Lingyun W., Mei Z., Guangming W., Guang M.: Truss optimization on shape and sizing with frequency constraints based on genetic algorithm. "Comput. Mech." 2005, 35, p. 361 – 368.
16. Srinivas M., Patnaik L. M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE "Transaction on Systems, Man and Cybernetics" 1994, Vol. 24, No. 4, s. 656 - 667.
17. Varnamkhasti M. J., Lee L. S.: A fuzzy genetic algorithm based on binary encoding for solving multidimensional knapsack problem. "Journal of Applied Mathematics" 2012, Article ID 703601, 2012.
18. Yager R. R.: Podstawy modelowania i sterowania rozmytego. Warszawa: WNT, 1995.