

Maciej RABIJASZ^{*}
Politechnika Śląska

OCENA WPŁYWU WYBRANYCH PARAMETRÓW SYSTEMU KLASYFIKATORÓW NA PRZEBIEG OPTIMALIZACJI KRATOWNICY

Streszczenie. Przedstawiono zasadę działania systemu klasyfikatorów użytego do optymalizacji kształtu płaskiej kratownicy. Funkcją celu jest masa konstrukcji. Zmiennymi decyzyjnymi są przekroje elementów oraz współrzędne węzłów. Autor przedstawia dwa eksperymenty. W pierwszym analizuje wpływ długości procesu optymalizacyjnego, natomiast w drugim - liczby nowych osobników generowanych w każdej iteracji.

INFLUENCE ASSESSMENT OF SELECTED PARAMETERS OF A CLASSIFIER SYSTEM ON OPTIMISATION OF A TRUSS

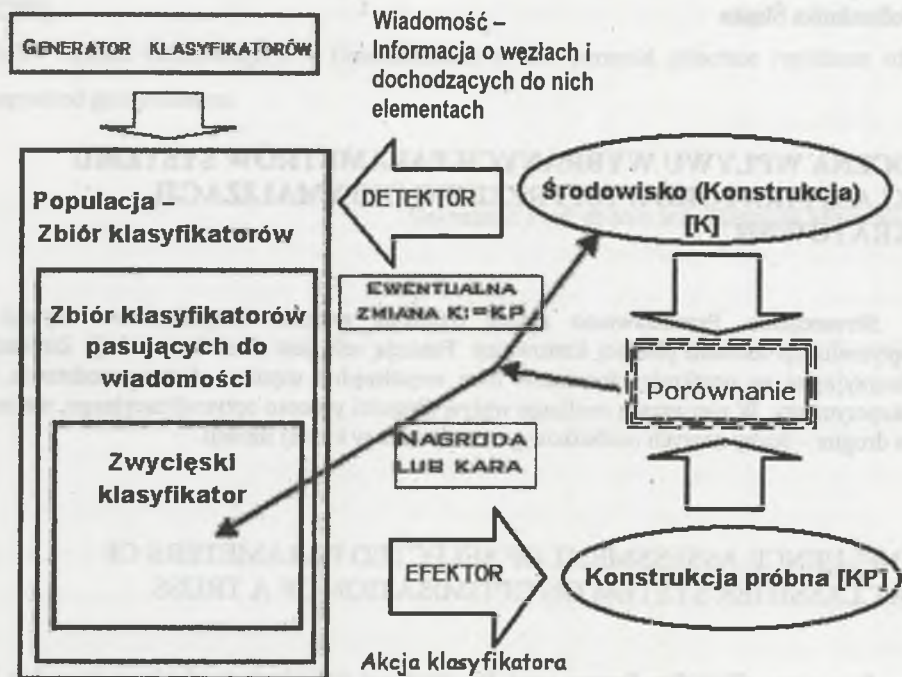
Summary. Classifier System used for structural optimization of planar truss has been presented. Weight of a truss is assumed to be an objective function. Decisive variables are: elements' sections and nodes' coordinates. Author presents two numerical experiments. The first analyses the influence of the total process time and the second- the influence of numbers of new classifiers generated in each iteration.

1. Wprowadzenie

System klasyfikatorów jest rodzajem genetycznego systemu uczącego się. Podstawy teoretyczne zostały opracowane przez Hollanda (1962), a pierwsza praktyczna implementacja miała miejsce w 1978 r. i polegała na treningu systemu w pokonywaniu labiryntu. Najbliższą tematycznie publikacją znaną autorowi jest ogólny algorytm optymalizacji kształtu konstrukcji za pomocą systemu klasyfikatorów- Richards (1995).

^{*} Opiekun naukowy: Dr hab. inż. Jerzy Skrzypczyk, prof. Pol. Śląskiej

2. Zasada działania



Rys. 1. Schemat działania algorytmu w jednej iteracji

Fig. 1. The scheme of the algorithm operation in a single iteration

W celu przebadania SK autor napisał obiektowo zorientowany program w języku C++, wykorzystując mechanizmy biblioteki STL (Standard Template Library) oraz język VRML (Virtual Reality Modelling Language) do prezentacji danych. Algorytm w jednej iteracji działa w sposób przedstawiony na rys. 1. Najpierw następuje wygenerowanie wiadomości ze środowiska, zawierającej zakodowane dane dotyczące optymalizowanego węzła. Są w niej zawarte informacje na temat liczby dochodzących do węzła elementów, ich wzajemnego ułożenia, przekroju, działających sił i wyężenia. Następnie z bazy danych wybierana jest lista klasyfikatorów pasujących do tej wiadomości. Każdy z nich składa się z części przedziałowej - „warunek” i liczbowej - „akcja”, przez co każdy klasyfikator jest pewną regułą.

Pasujące klasyfikatory walczą ze sobą poprzez aukcję o możliwość aktywowania modyfikacji konstrukcji zakodowanej w części „akcja”. Jeśli dana modyfikacja przyniosła poprawę funkcji celu, to system zwiększa „żywołność” reguły, która do niej doprowadziła dając tym samym szanse jej zwycięstwa w kolejnych aukcjach. W innej chwili, gdy system będzie szukał optymalnego ustawienia innego węzła o podobnej konfiguracji dochodzących do niego elementów i sił, spróbuje zastosować już sprawdzoną regułę. Pula reguł jest poddawana operacjom rynkowym – dobre reguły „zarabiają”, a złe „bankrutują” i są usuwane. Poszukiwanie nowych potencjalnie lepszych reguł odbywa się za pomocą mechanizmów zaczerpniętych z algorytmu genetycznego. Warto zauważyć, że to nie sama konstrukcja, jak w dotychczasowych aplikacjach algorytmu ewolucyjnego, lecz reguły jej zmiany są „osobnikami” populacji. Ostatecznie, sprawdzoną grupę reguł można zapisać w pliku tekstowym i wykorzystać do optymalizacji innej kratownicy.

W trakcie pracy programu system klasyfikatorów ma możliwość zmiany przekroju dowolnego elementu, jak również współrzędnych wszystkich węzłów, z wyjątkiem obciążonych i podporowych.

W trakcie działania programu przedstawiony schemat pojedynczej iteracji jest realizowany wielokrotnie (zwykle kilka tysięcy razy). Na bieżąco gromadzone są informacje o postępie optymalizacji, stanie populacji i konstrukcji. Przebieg algorytmu od pierwszej iteracji, gdzie wprowadza się konstrukcję początkową, aż do ostatniej, gdzie otrzymuje się wynik w postaci konstrukcji końcowej (zoptymalizowanej), nazwano „Zadaniem”.

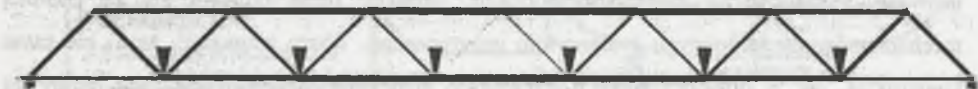
Ze względu na fakt, iż genetyczne systemy uczące się należą do grupy metod optymalizacji niedeterministycznej, wielokrotne przeprowadzenie „zadania”, nawet dla takich samych warunków, zawsze doprowadzi do różnych wyników. W związku z tym wprowadzono pojęcie „Eksperyment”. Polega on na wielokrotnym przeprowadzeniu „zadania” (10-50 razy) i uśrednieniu wszelkich wartości wyników i przebiegów algorytmu. Prezentowane w artykule wykresy przedstawiają właśnie takie uśrednione wartości.

3. Przeprowadzone eksperymenty

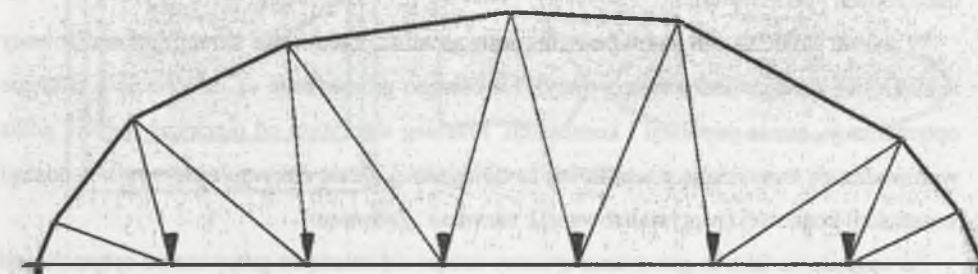
Na potrzeby niniejszej publikacji przeprowadzono dwa „eksperymenty”. W obu przypadkach zoptymalizowano wiązarkę kratową swobodnie podpartą o rozpiętości 14 m i wysokości 1m, obciążony siłami skupionymi po 70 kN przyłożonymi do węzłów pasa

dolnego. Przekroje początkowe dobrano z warunku naprężeń dopuszczalnych, tak aby masa układu była jak najmniejsza. Przy dostępnej bazie stu profili od 1 cm^2 do 100 cm^2 ze zmianą co 1 cm^2 – masa minimalna dla konfiguracji początkowej wynosi 1507 kg.

W pierwszym „eksperymentie” analizowano wpływ liczby iteracji w „zadaniu” na przebieg i efekt końcowy. W eksperymencie przeprowadzono 10 zadań kolejno przy 5000, 7500, 10000, 20000 i 50000 iteracji.



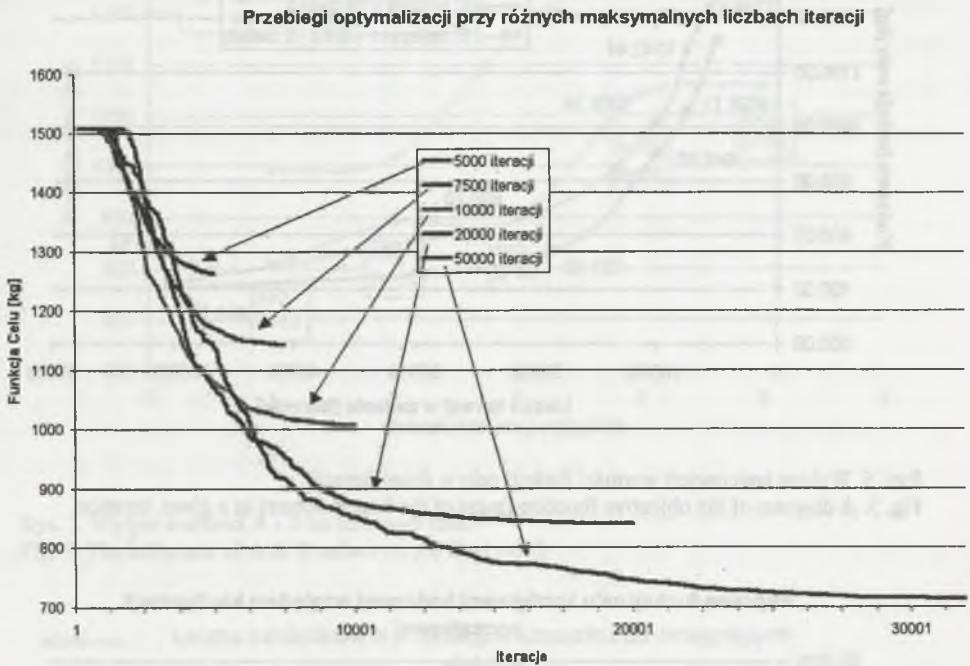
Rys. 2. Konfiguracja początkowa konstrukcji
Fig. 2. Initial configuration of the structure



Rys. 3. Najlepsza uzyskana w eksperymencie konfiguracja końcowa o masie 699 kg
Fig. 3. The best final configuration achieved in the experiment, of mass 699 kg

Na rys. 4 przedstawiono postęp optymalizacji kratownicy. Im większa ilość iteracji, tym lepszy jest końcowy efekt. Na rys. 5 porównano końcowe wartości funkcji celu. Widać z niego, że opłacalna liczba iteracji to 15000-25000. Przy 50000 iteracji wartości średnie i najlepsze prawie się pokrywają. Wynika z tego, że osiągnięto granicę możliwości programu i dalsze zwiększanie liczby iteracji nie ma sensu. Rysunek 6 przedstawia uzyskane oszczędności w masie konstrukcji. Przy 50000 iteracji algorytm był w stanie zredukować wagę o ponad połowę.

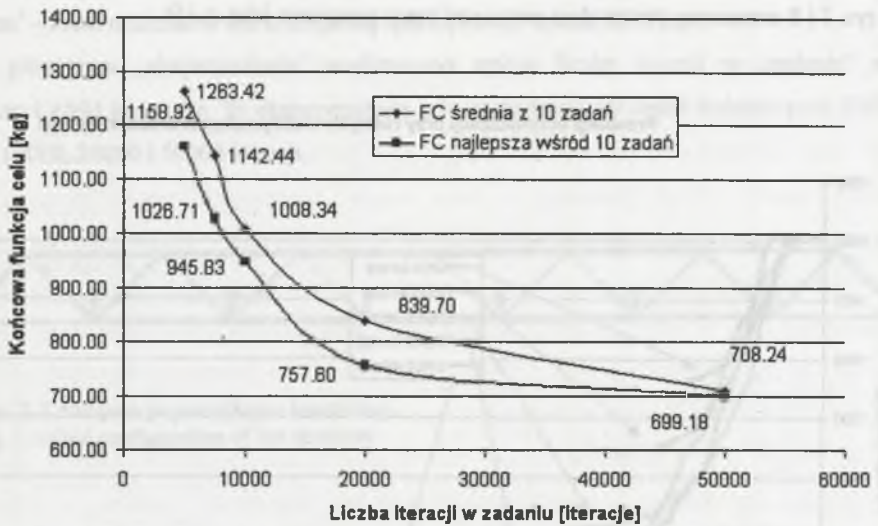
W drugim „eksperymentcie” analizowano wpływ liczby generowanych nowych osobników (reguł) na końcową wartość funkcji celu. Liczba iteracji była tym razem zawsze równa 10000. Na rys. 7 i 8 oznaczono pewne dwa parametry pracy programu jako A i B.



Rys. 4. Wykres funkcji celu (masa najlepszej konfiguracji kratownicy) w danej iteracji
Fig. 4. A diagram of the objective function (mass of the best structure) in a given iteration

Algorytm działa w ten sposób, że jeśli dla danej konfiguracji konstrukcji liczba pasujących reguł (klasyfikatorów) jest mniejsza od B, to generator klasyfikatorów (patrz rys. 1) tworzy nowe, pasujące w liczbie A. Widać z wykresów, że im mniejsze są A i B, tym lepsze końcowe wartości funkcji celu. Dla ustawień eksperymentu 4 i 5 (rys. 8) widać znacząco większą ilość klasyfikatorów w populacji końcowej i równoczesny spadek efektywności. Wynika z tego wniosek, iż generator dodatkowych reguł jest jedynie dodatkiem w mechanizmie optymalizacji. Główną przyczyną polepszania się konstrukcji są mechanizmy rynkowe charakterystyczne dla systemu klasyfikatorów, którym poddawane są reguły.

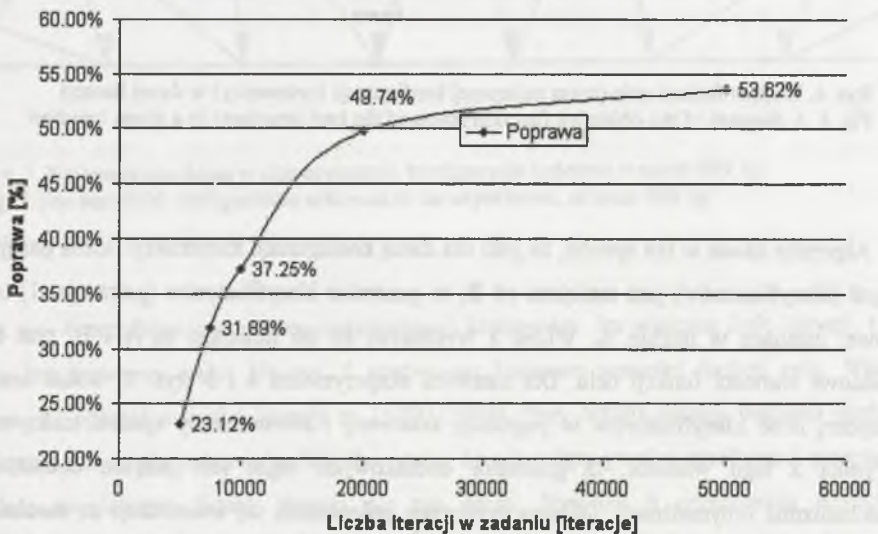
Wpływ liczby iteracji na końcowy efekt



Rys. 5. Wykres końcowych wartości funkcji celu w danej iteracji

Fig. 5. A diagram of the objective function (mass of the best structure) in a given iteration

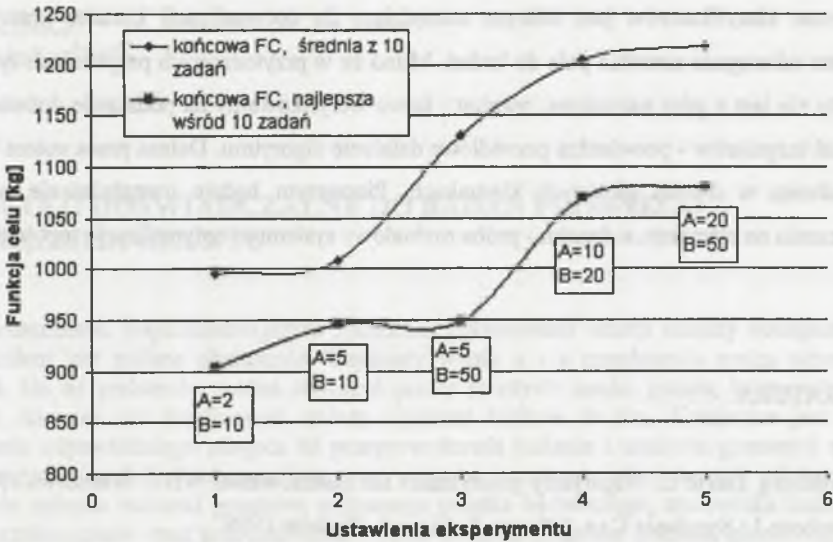
Poprawa funkcji celu konfiguracji końcowej względem konfiguracji początkowej



Rys. 6. Wykres funkcji celu (masa najlepszej konfiguracji kratownicy) w danej iteracji

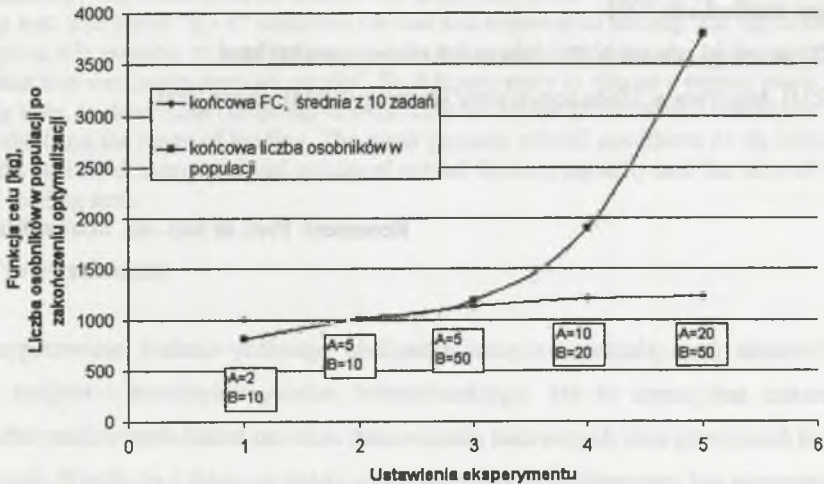
Fig. 6. A diagram of the objective function (mass of the best structure) in a given iteration

Wpływ ustawień wartości A i B na końcowy efekt



Rys. 7. Wpływ wartości A i B na końcowy efekt
 Fig. 7. The influence of A & B values on the final result

Liczba osobników w populacji w stosunku do osiągniętych efektów



Rys. 8. Liczba osobników w populacji końcowej w stosunku do osiągniętych efektów
 Fig. 8. The final population size versus final results

4. Wnioski i kierunki dalszej pracy

System klasyfikatorów jest dobrym narzędziem do optymalizacji kształtu kratownicy. Program udostępnia szerokie pole do badań. Mimo że w przytoczonych przykładach symetria kształtu nie jest z góry narzucona, rezultat - łatwo weryfikowalny na podstawie doświadczeń pokoleń inżynierów - potwierdza prawidłowe działanie algorytmu. Dalsza praca autora będzie prowadzona w dwóch głównych kierunkach. Pierwszym będzie uwzględnienie wpływu wyoboczenia na elementy, a drugim - próba rozbudowy systemu o optymalizację topologii.

LITERATURA

1. Goldberg David E.: Algorytmy genetyczne i ich zastosowania. WNT, Warszawa 1998.
2. Grębosz J.: Symfonia C++. Oficyna Kallimach, Kraków 1999.
3. Shrestha S.M., Ghaboussi J.: Evolution of optimum shapes using genetic algorithm. *Journal of Structural Engineering*, NOV 1998.
4. Taura T., Nagasaka I., Yamagishi A.: Application of evolutionary programming to shape design. Elsevier Science Ltd CAD VOL30, 1998.
5. Richards R.A.: Zeroth-ordered shape optimization utilizing a learning classifier system. www.stanford.edu 1995.
6. STL www.sgi.com/tech/stl/ www.cs.rpi.edu/~musser/stl.html.
7. VRML <http://www.3dsite.com/n/sites/3dsite/cgi/VRML-index.html>.

Recenzent: Prof. dr hab. inż. Marian Klasztorny