

Optymalizacja nieliniowych modeli MES na przykładzie tłoczniaka

Nonlinear FEM models optimization on example of a stamping die

IRENEUSZ WRÓBEL
JACEK STADNICKI*

DOI: <https://doi.org/10.17814/mechanik.2018.7.61>

Rozwiązywanie zadań optymalizacji, w których model MES jest nieliniowy, stwarza problemy związane z dużym nakładem obliczeniowym. W artykule przedstawiono i porównano algorytm powierzchni odpowiedzi RSM z hybrydowym algorytmem przeszukiwania SHERPA w zastosowaniu do rozwiązywania zadań optymalizacji progów ciągowych tłoczniaka do blach karoseryjnych.

SŁOWA KLUCZOWE: optymalizacja, nieliniowe modele MES, tłoczniak, progi ciągowe

Solving of optimization tasks using nonlinear FEM models creates problems because of large expenditure of calculation time. The response surface methodology RSM and hybrid search SHERPA algorithms in application of optimization of stamping die drawbeads are presented and compared in the paper.

KEYWORDS: optimization, nonlinear FEM models, stamping die, drawbeads

Współczesne programy do komputerowego wspomaganie projektowania, wykorzystujące metodę elementów skończonych, coraz częściej zawierają moduły do rozwiązywania zadań optymalizacji. Użytkownik może formułować i rozwiązywać takie problemy, jak:

- **optymalizacja parametryczna** (*geometry optimization*), gdzie zmiennymi decyzyjnymi są wymiary lub parametry przekrojowe modelu MES, a funkcją celu może być naprężenie, przemieszczenie węzłowe lub częstość własna;
- **optymalizacja topologiczna** (*topologic optimization*), gdzie poszukuje się takiego kształtu projektowanej części, który zapewni jej najlepsze rozłożenie materiału w przestrzeni z uwagi na kryterium maksymalnej sztywności, minimalnej częstości własnej itp.;
- **analiza wrażliwości** (*sensitivity analysis*), w której bada się wpływ zmiany wybranych parametrów modelu MES na wskazaną wielkość (kryterium), aby ustalić parametry istotne i wyeliminować mało istotne z uwagi na poprawę danego kryterium.

W wymienionych problemach rozwiązuje się zadanie optymalizacji:

$$Q(\bar{x}) \rightarrow \min \quad (1)$$

przy ograniczeniach:

$$\begin{aligned} g_j(\bar{x}) &\leq 0 \quad j = 1, 2 \dots n_g \\ h_k(\bar{x}) &\leq 0 \quad j = 1, 2 \dots n_h \\ x_i^d &\leq x_i \leq x_i^g \quad i = 1, 2 \dots n \end{aligned}$$

gdzie: $\bar{x} = [x_1, x_2 \dots x_n]^T$ – wektor zmiennych decyzyjnych, $Q(\bar{x})$ – funkcja celu.

Rozwiązania poprawionego poszukuje się najczęściej według schematu:

$$\bar{x}_{k+1} = \bar{x}_k + \hat{\lambda}_k \bar{d}_k \quad (2)$$

gdzie: \bar{d}_k – kierunek poszukiwań; $\hat{\lambda}_k$ – długość kroku, po którym w iteracji k uzyskuje się rozwiązanie poprawione [4].

Za kierunek poszukiwań przyjmuje się np. kierunek przeciwny do gradientu funkcji celu:

$$\bar{d}_k = -\nabla^T Q(\bar{x}_k) \quad (3)$$

Ponieważ punkt \bar{x}_k ma pozostawać w zbiorze dopuszczalnym (spełniać ograniczenia), kierunek poszukiwań koryguje się tak, aby był dopuszczalnym kierunkiem poprawy [4], czyli aby spełniał warunki:

$$\begin{aligned} \bar{d}_k \nabla^T Q(\bar{x}_k) &< 0 \\ \bar{d}_k \nabla^T g_j(\bar{x}_k) &< 0 \quad j = 1, 2, \dots, n_g \end{aligned} \quad (4)$$

Ze względu na wspomniane problemy optymalizacji modeli MES w każdej iteracji dotyczącej rozwiązywania zadania optymalizacji trzeba rozwiązać podproblem analizy strukturalnej lub modalnej modelu MES. Jeśli przy dostępnych zasobach pamięci komputera i mocy obliczeniowej procesora czas rozwiązywania podproblemu nie jest zbyt długi, rozwiązanie zadania optymalizacji można uzyskać w czasie akceptowalnym z inżynierskiego punktu widzenia. Ewentualne skrócenie czasu obliczeń można osiągnąć dzięki poprawie efektywności numerycznej modelu MES (np. przez zredukowanie liczby stopni swobody) lub zdefiniowaniu początkowych wartości zmiennych decyzyjnych zadania optymalizacji w taki sposób, aby punkt startowy leżał bliżej rozwiązania optymalnego (to położenie punktu można ustalić np. na podstawie zgrubnego przeszukania zbioru dopuszczalnego).

Algorytmy optymalizacji nieliniowych modeli MES

W optymalizacji nieliniowych modeli MES warto zwrócić uwagę na dwa problemy, z którymi należy się uporać, aby przy dostępnych zasobach uzyskać rozwiązanie zadania w akceptowalnym czasie:

- czas rozwiązywania zadania nieliniowej analizy MES jest na tyle długi, że wykonanie niezbędnej do uzyskania rozwiązania optymalnego liczby powtórzeń pojedynczej iteracji algorytmu optymalizacji skutkuje zbyt długim łącznym czasem poszukiwania rozwiązania optymalnego – w tym przypadku mamy do czynienia z niedostateczną **efektywnością numeryczną algorytmu** optymalizacji;
- natura problemu analizy nieliniowej MES powoduje, że wyznaczone kierunki poszukiwań rozwiązań poprawionych nie prowadzą do uzyskania rozwiązania optymalnego po akceptowalnej liczbie powtórzeń pojedynczej iteracji algorytmu optymalizacji (punkt startowy został niewłaściwie określony lub wrażliwość funkcji celu na zmiany wektora zmiennych decyzyjnych zadania jest niejednoznaczna) – w tym przypadku mamy do czynienia z niedostateczną **odpornością algorytmu** optymalizacji.

Wciąż poszukuje się algorytmów optymalizacji, które byłyby dostatecznie efektywne numerycznie i odporne w zastosowaniu do problemów optymalizacji modeli nieliniowych. Dalej omówiono dwa algorytmy zaimplementowane m.in. w programie DynaForm.

* Dr hab. inż. Ireneusz Wróbel, prof. ATH (iwrobel@ath.bielsko.pl); prof. dr hab. inż. Jacek Stadnicki (jstadnicki@ath.bielsko.pl) – Akademia Techniczno-Humanistyczna w Bielsku-Białej

Algorytm SHERPA [7]

SHERPA (*simultaneous hybrid exploration that is robust, progressive and adaptive*) to w wolnym tłumaczeniu algorytm jednoczesnego hybrydowego przeszukiwania, który jest odporny, progresywny i adaptacyjny. W istocie nie jest on jednym algorytmem, lecz kombinacją od dwóch do dziesięciu różnych algorytmów, które działają równocześnie, przy czym wybór algorytmu i jego parametrów decydujących o zbieżności i efektywności odbywa się automatycznie. Dzięki temu SHERPA rozpoznaje charakter optymalizowanej funkcji celu i zbiór dopuszczalny, a następnie adaptuje się do rozwiązywanego zadania optymalizacji, wybierając najlepszy w danym momencie algorytm z najlepszymi parametrami. Podobnie jak w przypadku każdego innego uniwersalnego algorytmu optymalizacji nie ma gwarancji, że algorytm zawsze niezawodnie i efektywnie znajdzie rozwiązanie optymalne. Praktyka numeryczna potwierdza jednak, że statystycznie SHERPA jest algorytmem sprawdzającym się w wielu zastosowaniach.

Najważniejsze z algorytmów optymalizacji, które wykorzystuje SHERPA to:

- **algorytmy optymalizacji lokalnej**, takie jak:
 - algorytm Levenberga-Marquardta [2], który jest szybko zbieżny (zwłaszcza dla niezbyt złożonych modeli), a jednocześnie wrażliwy na wybór punktu startowego;
 - algorytm Neldera-Meada (pełzającego sympleksu) [3], który jest odporny i umiarkowanie zbieżny (zwłaszcza dla złożonych modeli);
- **algorytmy optymalizacji globalnej**, takie jak:
 - algorytm ewolucji różnicowej (*differential evolution – moncar*) [6], który najszybciej próbuje liczne zbiory dopuszczalne;
 - algorytm systematycznego przeszukiwania (*gridsearch*) [4], który zgrubnie i szybko rozpoznaje charakter funkcji celu i zbiór dopuszczalny.

Metoda powierzchni odpowiedzi (RSM) [1]

Metoda RSM (*response surface methodology*) polega na aproksymowaniu odpowiedzi modelu MES na zmiany zmiennych decyzyjnych zadania optymalizacji tzw. powierzchnią odpowiedzi, która staje się surogatem (metamodelem) pierwotnego zadania optymalizacji. Jeśli powierzchnię odpowiedzi $y(\bar{x})$ przybliży się szeregiem funkcji bazowych:

$$y(\bar{x}) \cong \sum_{i=1}^L a_i \phi_i(\bar{x}) \quad (5)$$

to stałe a_i wyznacza się, minimalizując sumę kwadratów odchyłeń powierzchni odpowiedzi w P punktach kontrolnych odpowiedzi modelu MES:

$$\sum_{p=1}^P [y(\bar{x}_p) - \sum_{i=1}^L a_i \phi_i(\bar{x}_p)]^2 \rightarrow \min \quad (6)$$

Rozwiązaniem zadania (6) jest wektor stałych:

$$\bar{a} = ([X]^T [X])^{-1} [X]^T \bar{y}, \quad (7)$$

gdzie: $[X] = [X_{pi}] = [\phi_i(X_{pi})]$.

Losując położenie punktów kontrolnych, np. metodą LHS (*latin hypercube sampling*), otrzymuje się metamodel pierwotnego zadania optymalizacji, który z racji przybliżenia według wzoru (5) jest o wiele bardziej efektywny numerycznie.

Dalej opisano przykład praktycznego wykorzystania algorytmów optymalizacji nieliniowego modelu MES do poprawy konstrukcji tłoczniaka do blach karoseryjnych.

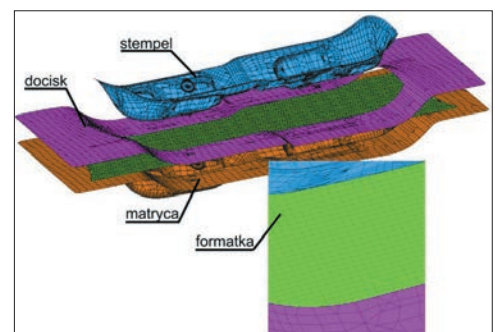
Optymalizacja progów ciągowych tłoczniaka

Aby wyprodukować wytłoczkę bez wad, tłoczona blacha (formatka) musi być w odpowiednich miejscach przytrzymywana z właściwą siłą między stemplem a matrycą, aby wywołać odkształcanie plastyczne potrzebne do uzyskania projektowanego kształtu wytłoczki. Tę funkcję realizują tzw. progi ciągowe, czyli rowki w matrycy lub docisku, współpracujące z progami w stemple, które, stawiając opór przeciąganej przez nie blasze podczas tłoczenia, przytrzymują ją z odpowiednią siłą. Zadanie optymalizacji progów ciągowych polegało na wyznaczeniu takiego profilu i rozmieszczenia progów, które eliminuje wady wytłoczki w postaci pęknięć i zmarszczek. Wytłoczkę z wadami przed optymalizacją progów przedstawiono na rys. 1.



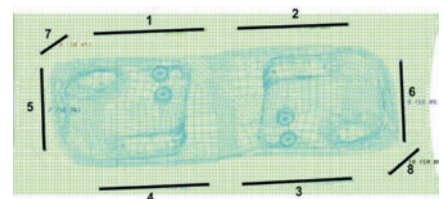
Rys. 1. Wytłoczka z wadami w postaci pęknięć i zmarszczek

Na potrzeby zadania zbudowano w programie DynaForm modele MES formatki oraz powierzchni roboczych części tłoczniaka, tj. matrycy, stempla i docisku. Modele powierzchni roboczych tłoczniaka utworzono z czterowęzłowych elementów powierzchniowych typu *shell*, które były nieodkształcalne, a formatkę starannie zdyskretyzowano za pomocą ośmiowęzłowych elementów powierzchniowych typu *shell16* (tzw. *fully integrated shell*). Formatce przypisano materiał HSLA250, tj. stal do tłoczenia na zimno części karoserii o granicy plastyczności 250 MPa oraz granicy wytrzymałości 345 MPa. Wzajemne oddziaływanie powierzchni roboczych tłoczniaka i formatki potraktowano jako jednostronne więzy z tarcieniem i zamodelowano elementami kontaktowymi typu powierzchnia–powierzchnia (*one-way-surface-to-surface*). Przyjęto współczynnik tarcia $\mu = 0,125$. W ten sposób powstał kompletny model MES tłoczniaka i formatki, pokazany na rys. 2, który wykorzystano do symulacji tłoczenia i rozwiązania zadania optymalizacji progów ciągowych.



Rys. 2. Model dyskretny elementów tłoczniaka i formatki

W zadaniu optymalizacji zdefiniowano osiem progów ciągowych, których rozmieszczenie początkowe, pokazane na rys. 3, wynika z doświadczenia konstruktorów. Zadanie rozwiązano w programie DynaForm, w którym próg ciągowy zastępuje się siłą w płaszczyźnie formatki,

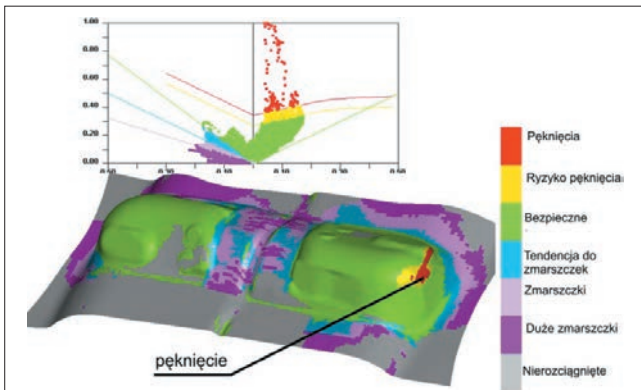


Rys. 3. Rozmieszczenie progów ciągowych na formatce

rozłożoną równomiernie wzdłuż zdefiniowanej linii. Wartość siły zależy od profilu progów ciągowych oraz grubości i rodzaju materiału formatki, a oblicza się ją według algorytmu Stoughtona [5]. Program operuje względną wartością siły wzdłuż progów, będącą procentem siły maksymalnej, która uniemożliwia przesuwanie blachy przez próg. Domyślną wartością początkową jest 50% tej siły. Po rozwiązaniu zadania wyznaczone względne wartości sił wzdłuż progów można przeliczyć na odpowiednie profile progów według wspomnianego algorytmu Stoughtona.

Aby przeprowadzić symulację tłoczenia, zadano kinematykę procesu, zgodnie z którą najpierw docisk zamyka formatkę w matrycy, a następnie dosuwający się stempel kształtuje formatkę aż do zamknięcia stempla i matrycy.

Po przeprowadzeniu symulacji jakość wytłoczki należy sprawdzić za pomocą diagramu krzywych granicznych tłoczności (tzw. diagramu FLD), na którym podawane jest ryzyko pęknięć i powstania zmarszczek w wytłoczce. Na rys. 4 przedstawiono wynik symulacji dla wytłoczki przed optymalizacją progów ciągowych, który potwierdza ryzyko pęknięć zmarszczek w miejscach, w których te wady rzeczywiście wystąpiły (rys. 1). To świadczy o poprawności przyjętego modelu MES. Aby otrzymać wytłoczkę bez wad, należy tak dobrać siły wzdłuż progów ciągowych, aby pęknięcia i zmarszczki nie powstawały.



Rys. 4. Diagram FLD dla wytłoczki przed optymalizacją progów ciągowych

Rozwiązanie zadania optymalizacji

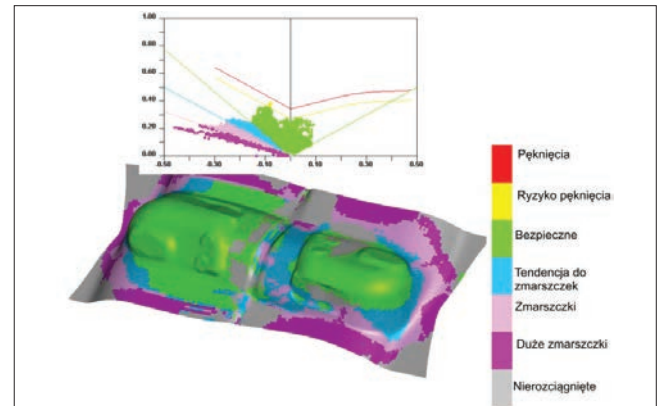
W zadaniu optymalizacji progów ciągowych, rozwiązaniem w programie DynaForm, składowymi wektora zmiennych decyzyjnych \bar{F} były wartości sił wzdłuż progów ciągowych, a funkcją celu $Q(\bar{F})$ – wielkość obszaru wytłoczki, w którym występuje ryzyko pęknięcia lub powstawania zmarszczek. Ograniczeniem był procentowy zakres siły wzdłuż progów ciągowych: przyjęto $5\%F_{max}^i \leq F_i \leq 80\%F_{max}^i$. Założono ponadto, że wartości sił wzdłuż progów 1 i 4 oraz 2 i 3 (rys. 3) są takie same, co pozwoliło zredukować liczbę zmiennych decyzyjnych zadania do sześciu.

Dla porównania jakości rozwiązań i czasu obliczeń to samo zadanie optymalizacji rozwiązano za pomocą algorytmów SHERPA i RSM. W tabelicy zestawiono optymalne wartości zmiennych decyzyjnych. Jak widać, w przypadku obu algorytmów rozwiązania są podobne, co potwierdza poprawność wyników. Orientacyjne czasy rozwiązania zadania optymalizacji na komputerze z procesorem Intel i7/3,2GHz/16GB RAM były różne i wyniosły: dla algorytmu SHERPA – ok. 12 h, dla algorytmu RSM – ok. 192 h.

Ponieważ celem optymalizacji było poprawienie jakości wytłoczki, warto dla optymalnych progów ciągowych ponownie przeprowadzić symulację tłoczenia i sprawdzić diagram FLD. Na rys. 5 przedstawiono taki diagram, z któ-

TABLICA. Optymalne siły wzdłuż progów ciągowych \hat{F}_i jako procent siły F_{max}^i uniemożliwiającej przesuw blachy przez próg

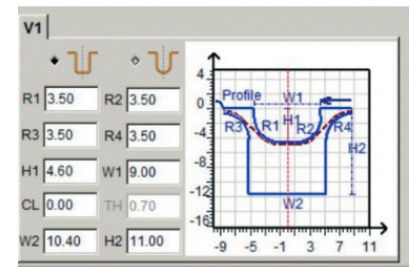
| Numer progów i | 1, 4 | 2, 3 | 5 | 6 | 7 | 8 |
|---------------------------|------|------|---|---|----|----|
| \hat{F}_i w % wg SHERPA | 21 | 20 | 3 | 3 | 11 | 9 |
| \hat{F}_i w % wg RSM | 20 | 20 | 2 | 2 | 10 | 11 |



Rys. 5. Diagram FLD dla wytłoczki po optymalizacji progów ciągowych

rego wyniku, że optymalizacja progów ciągowych przyniosła zamierzony cel (por. rys. 4) i wytłoczka nie ma wad.

W praktyce optymalne wartości sił wzdłuż progów ciągowych automatycznie przelicza się na wymiary progów. Na rys. 6 pokazano okno dialogowe programu DynaForm, które podaje wymiary przykładowego progów nr 6 o profilach półokrągłym i prostokątnym.



Rys. 6. Wymiary półokrągłego i prostokątnego progów ciągowych nr 6 dla optymalnej siły

Wnioski

- Dostępne w oprogramowaniu inżynierskim algorytmy rozwiązywania zadań optymalizacji, w których w pojedynczej iteracji należy wyznaczyć odpowiedź nieliniowego modelu MES na zmiany zmiennych decyzyjnych zadania, są wystarczająco skuteczne w przypadku rozwiązywania rzeczywistych zadań projektowych.
- Jakość rozwiązania uzyskanego algorytmami RMS i SHERPA jest porównywalna, lecz algorytm SHERPA jest efektywniejszy numerycznie – czas obliczeń był ok. 16 razy krótszy.

LITERATURA

1. Daberkov D.D., Mavris D.N. "An investigation of metamodeling techniques for complex system design". *Symposium on Multidisciplinary Analysis and Design*, Atlanta, October 2002.
2. Marquardt D.W. "An algorithm for least-squares estimation of nonlinear parameters". *Journal of the Society for Industrial and Applied Mathematics*. 11, 2 (1963): s. 431–441.
3. Nelder J.A., Mead R. "A simplex method for function minimization". *Computer Journal*. 7 (1965): s. 308–313.
4. Stadnicki J. „Teoria i praktyka rozwiązywania zadań optymalizacji”. Warszawa: Wydawnictwa Naukowo-Techniczne, 2006.
5. Stoughton T.B. "Model of drawbead forces in sheet metal forming". *15th Biennial Congress of IDDRG*, Dearborn, MI, 1988: s. 205–215.
6. Storn R., Price K. "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces". *Journal of Global Optimization*. 11 (1997): s. 341–359.
7. www.redcedartech.com/pdfs/SHERPA.pdf.