

KATEDRA MECHANIKI I INŻYNIERII OBLICZENIOWEJ
Wydział Mechaniczny Technologiczny
Politechnika Śląska

ROZPRAWA DOKTORSKA

Optymalizacja kształtu oraz warunków brzegowych podzespołu
układu oczyszczania spalin

mgr inż. Joachim Jarosz

Promotor: dr hab. inż. Adam, Długosz prof. PŚ

GLIWICE 2023

Spis treści

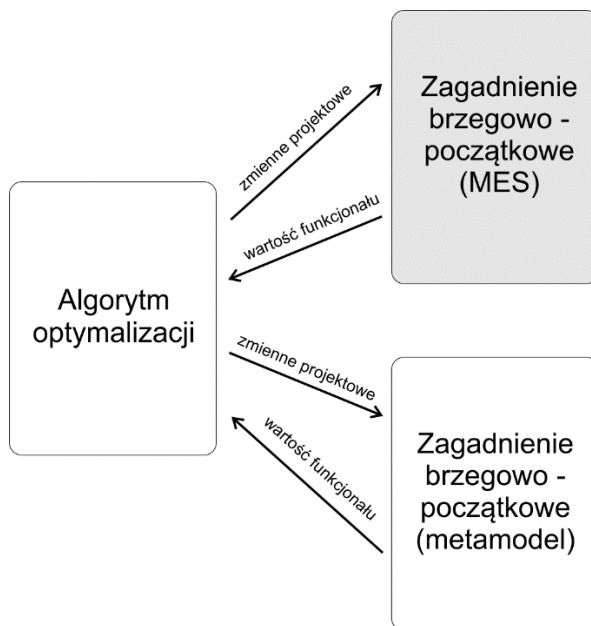
1. Wprowadzenie.....	4
1.1. Przegląd zawartości rozprawy.....	6
1.2. Cel i teza rozprawy.....	6
2. Osłony termiczne układów wydechowych.....	8
2.1. Geneza tematu.....	8
2.2. Rola i rodzaje osłon układów wydechowych.....	27
2.3. Wymagania stawiane osłonom układów wydechowych.....	30
3. Metody analizy optymalnego projektowania rozpatrywanego zagadnienia.....	32
3.1. Wstęp.....	32
3.2. Optymalizacja.....	36
3.3. Optymalizacja wielokryterialna.....	42
3.4. Metody gradientowe.....	45
3.5. Metody bezgradientowe.....	48
3.6. Metody ewolucyjne.....	49
3.7. Podstawy analizy MES układów sprężystych, termosprężystych oraz analizy modalnej.....	57
3.8. Zastosowanie sztucznych sieci neuronowych w metamodelowaniu przy użyciu MES.....	65
3.8.1. Sztuczne sieci neuronowe (SSN).....	65
3.8.2. Neuron – opis działania.....	65
3.8.3. Metamodel z wykorzystaniem MES.....	68
3.9. Zastosowane algorytmy optymalizacji wbudowane w oprogramowanie MES.....	70
3.9.1. Jednokryterialny algorytm NLPQL.....	70
3.9.2. Jednokryterialny algorytm MISQP.....	71
3.9.3. Jednokryterialny algorytm ASO.....	72
3.9.4. Wielokryterialny algorytm NSGA-II.....	72
3.9.5. Wielokryterialny algorytm MOGA.....	74
3.10. Zewnętrzne algorytmy optymalizacji (niewbudowane w oprogramowanie MES).....	76
3.10.1. Jednokryterialny algorytm Nelder-Mead.....	76
3.10.2. Jednokryterialny algorytm ewolucyjny EA.....	79
3.10.3. Jednokryterialny algorytm ewolucji różnicowej L-SHADE.....	80
3.10.4. Wielokryterialny algorytm MOOPTIM.....	82
3.11. Kryteria optymalizacji.....	84
3.12. Opracowany schemat optymalizacji jedno i wielokryterialnej.....	86
4. Model geometryczny oraz numeryczny osłony – budowa oraz weryfikacja numeryczna.....	89
4.1. Model geometryczny.....	89

4.2. Model MES osłony układu wydechowego.....	92
4.3. Wariant podstawowy – referencyjny.....	94
4.4. Dyskretyzacja i dobór siatki.....	96
4.5. Model uproszczony.....	102
4.6. Metamodel odpowiedzi numerycznej z zastosowaniem MES.....	105
5. Rezultaty optymalizacji.....	114
5.1. Optymalizacja jednokryterialna.....	114
5.2. Optymalizacja wielokryterialna.....	127
5.3. Optymalizacja z zastosowaniem metamodelowania.....	142
6. Podsumowanie, wnioski oraz kierunki dalszych badań.....	145
Bibliografia.....	147
Streszczenie.....	155
Summary.....	156

1. Wprowadzenie

Współczesne formy projektowania wymagają od konstruktora uwzględnienia wielu czynników. Należą do nich założenia projektowe, wymagania techniczne stawiane konstrukcji, a także normy, dyrektywy, wymogi bezpieczeństwa czy przepisy prawne. Projekt musi być konkurencyjny w relacji do istniejących rozwiązań na rynku. Aby sprostać wymaganiom rynkowym produkt musi spełniać coraz to wyższe wymagania i standardy, a konstruktor coraz częściej sięga do zagadnień optymalizacji konstrukcji, zazwyczaj z uwzględnieniem wielu kryteriów oraz ograniczeń. Optymalizacja konstrukcji, modelu geometrycznego, jest pojęciem bardzo szerokim. Najczęściej w praktyce, w złożonych projektach konstrukcyjnych czy skomplikowanych procesach przemysłowych mamy do czynienia z kryteriami które pozwalają sformułować cele w postaci funkcyjnej, które z kolei bardzo często są nieciągłe lub wielomodalne. Jest to spowodowane wielością parametrów opisujących rozpatrywany problem, opisem fizycznym modelowanego zjawiska, złożonością związaną z warunkami brzegowymi, modelami materiałowym, itp. Istnieje wiele metod optymalizacji, dlatego wybór odpowiednie metody w procesie projektowym staje się kluczowy. Proces optymalizacji jest z reguły procesem czasochłonnym. Spowodowane to jest koniecznością wyznaczania wartości funkcji celu z wykorzystaniem metod numerycznych. Ciągłe rosnąca moc obliczeniowa komputerów stwarza ku temu coraz większe możliwości, jednak nawet obecnie ciągle poszukuje się metod umożliwiających zwiększenie wydajności tego procesu. Do symulacji numerycznych w praktyce inżynierskiej najczęściej wykorzystywana jest metoda elementów skończonych (MES). Jej popularność i szeroka dostępność w postaci komercyjnego oprogramowania staje się podstawowym narzędziem dla konstruktora. Obliczenia i symulacje przeprowadzane za pomocą tej metody pozwalają przewidywać zachowanie się modelu konstrukcji uwzględniając specyfikę danego zjawiska fizycznego. Często w rozważaniach należy uwzględnić jednoczesne występowanie większej ilości zjawisk, jak również interakcję pomiędzy nimi. W szeroko rozumianej inżynierii mechanicznej numeryczne modele MES umożliwiają otrzymanie z wymaganą dokładnością rezultatów dotyczących wartości, przemieszczeń, odkształceń, naprężeń, temperatur, strumieni ciepła, itp. W kontekście optymalizacji wykorzystanie istniejącego komercyjnego oprogramowania z jednej strony umożliwia budowę i analizę bardzo złożonych modeli, jednak z drugiej wymaga zadbania o automatyzację procesu związanego z tworzeniem parametrycznych modeli geometrycznych, definicji warunków brzegowych oraz wymaganych parametrów dla danego typu analizy. Zadania te z reguły są bardzo kosztowne obliczeniowo, co kontekście zadań optymalizacji, których często należy wykonać tysiąca pojedynczych symulacji staje się problematyczne. W takich przypadkach uzasadnione jest poszukiwanie adekwatnych modeli zastępczych. Metamodeli modeli numerycznych tworzone są z wykorzystaniem różnych technik i algorytmów. W przypadku prostszych zadań często zwykłe modele wielomianowe mogą być wystarczające, jednakże dla większej liczby parametrów decyzyjnych oraz przy uwzględnieniu większej liczby kryteriów i ograniczeń już tak nie jest. Wykorzystanie sztucznych sieci neuronowych (SSN) w metamodelowaniu potrafi w takich przypadkach być uzasadnionym podejściem, pod warunkiem zadbania o prawidłową topologię sieci oraz odpowiednie jej wytrenowanie. Rys. 1.1 przedstawia ogólną koncepcję sprzężenia pomiędzy odpowiednim algorytmem optymalizacji a modulem wyznaczającym wartości funkcji kryterialnych za pomocą symulacji MES oraz metamodelu MES. Prawidłowo zaprojektowana i wytrenowana SSN umożliwia na skrócenie czasu rozwiązywania zadań optymalizacji nawet o kilka rzędów wielkości. Należy jednak zaznaczyć, że prawidłowe przygotowanie adekwatnego modelu zastępczego nie jest zadaniem łatwym oraz może być również czasochłonne. Obecnie używane rozbudowane systemy klasy CAE (ang. Computer Aided Engineering) coraz częściej posiadają również odpowiednie moduły umożliwiające rozwiązywanie zadań optymalizacji zarówno opartej bezpośrednio na modelu MES, jak również z użyciem modeli zastępczych. Jednakże zastosowanie podejścia wykorzystującego zewnętrzne

algorytmy optymalizacji oraz zewnętrzne metamodele umożliwia poszukiwanie alternatywnych rozwiązań konstrukcyjnych, nierzadko dających lepsze rezultaty.



Rys. 1.1 Schemat komunikacji pomiędzy modułami optymalizacji a modułami wyznaczającymi wartości funkcjonalów.

Jak wcześniej wspomniano kryteria oraz wymagania projektowe w problemach inżynierii mechanicznej, dotyczące m.in. przemysłu samochodowego, kolejowego czy lotniczego najczęściej związane są z własnościami wytrzymałościowymi, termofizycznymi lub kosztowymi. Niniejsza praca dotyczy właśnie takiej problematyki i związana jest z optymalnym projektowaniem osłon układów wydechowych silników samochodowych. Osłony takie, wykonane z różnych materiałów oraz posiadające specyficzne kształty projektowane są z uwzględnieniem powyższych wymagań. Optymalizacja tego typu układów, oprócz optymalnego doboru kształtu dotyczyć również może doboru materiałów oraz warunków brzegowych związanych z montażem. Prowadzi to do konieczności sformułowania zadań optymalizacji, które dotyczą jednocześnie optymalnego doboru kształtu osłony jak i warunków brzegowych. Wymaga to wprowadzenia znacznie większej liczby zmiennych parametrów projektowych w porównaniu do zadań w których te aspekty optymalizacji rozpatrywane byłyby osobo, a w konsekwencji utrudnia oraz komplikuje sam proces optymalizacji. Z kolei definicja większej liczby kryteriów i konieczności ich jednoczesnego rozpatrywania prowadzi naturalnie do formułowania zadań optymalizacji wielokryterialnej.

Autor niniejszej rozprawy zawodowo zajmuje się m.in. projektowaniem elementów układu wydechowego samochodów spalinowych, w tym osłon termicznych, od wielu lat. Pomimo faktu, że przyszłość przemysłu samochodowego należy raczej do elektromobilności, to tematyka rozpatrywana w ramach doktoratu w dalszym ciągu wydaje się zasadna i uzasadniona, ponieważ zaproponowana metoda optymalizacji kształtu oraz warunków brzegowych może zostać zastosowana z powodzeniem do różnego typu osłon termicznych, stosowanych nie tylko w silnikach spalinowych oraz nie tylko w przemyśle samochodowym.

1.1.Przegląd zawartości rozprawy

Niniejsza rozprawa składa się sześciorozdziałów. Pierwszy rozdział zawiera wprowadzenie ogólne, informacje na temat układów wydechowych oraz sformułowanie tezy rozprawy.

W drugim rozdziale przedstawiono charakterystykę osłon termicznych stosowanych najczęściej w układach wydechowych. Opisano różne typy osłon z uwzględnieniem ich zalet i wad oraz ich przeznaczenie. Rozdział ten zawiera również opis wymagań jakie są stawiane wspólnie stosowanym osłonom termicznym.

Z kolei w trzecim rozdziale przedstawiono zagadnienia dotyczące zastosowanych metod i algorytmów optymalnego doboru wybranych cech konstrukcyjnych osłon, opisano podstawy teoretyczne rozwiązywanego zagadnienia termosprężystego oraz zagadnienia własnego zarówno w kontekście teoretycznym jak i w ujęciu metody elementów skończonych (MES). Ponadto przedstawiono teoretyczne podstawy zagadnienia związanego ze sztucznymi sieciami neuronowymi, które zostały wykorzystane w pracy do utworzenia metamodelu osłony termicznej. Opisano algorytmy wykorzystane w optymalizacji jedno i wielokryterialnej. Sformułowano i zawarto również kryteria oraz ograniczenia niezbędne do zdefiniowania zadań optymalizacji. Przedstawiono schemat połączenia wszystkich etapów całego procesu optymalizacji wraz z narzędziami wykorzystywanymi na danym etapie. Pokazano połączenie komercyjnego oprogramowania MES z algorytmami optymalizacji w połączeniu ze skryptami stresującymi procesem.

W rozdziale czwartym opisano model geometryczny i numeryczny osłony. Model geometryczny został przedstawiony jako model w pełni sparametryzowany stanowiący integralną część proponowanej metody optymalizacji kształtu oraz warunków brzegowych wraz z utworzonymi procedurami, które nią sterują. Opisane zostały wszystkie parametry modelu jako dane do algorytmów optymalizacyjnych. Zostały omówione również zagadnienia wpływu jakości siatki elementów skończonych na dokładność wyznaczenia wybranych cech zastosowanego modelu. Ponadto przedstawiono również uproszczony model numeryczny, wykorzystany w zadaniach optymalizacji, nie uwzględniający kontaktu pomiędzy otworami montażowymi i miejscem zamocowania, wraz z porównaniem dokładności rozwiązania dla takiego modelu. Rozdział zamyka przedstawienie koncepcji wykorzystania sztucznych sieci neuronowych do utworzenia metamodelu.

Rozdział piąty zawiera z kolei rezultaty przeprowadzonych analiz osłony termicznej dla optymalizacji jedno- i wielokryterialnej. Ponadto opisano w nim wyniki optymalizacji jednokryterialnej z wykorzystaniem metamodelu. Przedstawione zostały przykłady zadań optymalizacyjnych osłon termicznych dla różnych kryteriów optymalizacyjnych.

Rozdział szósty zawiera podsumowanie całej rozprawy oraz wnioski z przeprowadzonych analiz. Przedstawiono także kierunki dalszych badań, rozbudowy opracowanego sposobu optymalizacji kształtu osłon.

Na końcu zamieszczono bibliografię oraz streszczenie niniejszej rozprawy.

1.2.Cel i teza rozprawy

Głównym celem pracy jest opracowanie i opis metody optymalizacji kształtu i warunków brzegowych osłon termicznych stosowanych w układach wydechowych przy wykorzystaniu metody elementów skończonych, różnych metod optymalizacji oraz metamodelowania w oparciu o sztuczne sieci neuronowe.

Sformułowano kryteria oraz ograniczenia dla poszczególnych wariantów zadań optymalizacji. Przedstawiono zarówno optymalizację jedno-, jak i wielokryterialną.

Podjęte badania mają na celu również opracowanie elastycznej metody nowoczesnego projektowania umożliwiającej nie tylko skrócenie czasu projektowania i weryfikacji numerycznej ale również elastycznego definiowania i uwzględniania innych kryteriów jakości i ograniczeń.

Sformułowano następującą tezę rozprawy:

Zaproponowana metoda optymalizacji wybranych cech konstrukcyjnych osłon termicznych układów wydechowych przy wykorzystaniu metody elementów skończonych, różnych algorytmów optymalizacji oraz metamodelowania z użyciem sztucznych sieci neuronowych jest skutecznym narzędziem wspomagającym i poprawiającym tradycyjne metody projektowania.

2. Osłony termiczne układów wydechowych

2.1. Geneza tematu

Powszechnie stosowane silniki spalinowe wymagają odpowiednich układów oczyszczania spalin. Konstrukcja dzisiejszych układów spalinowych różni się znacząco od tych pierwszych modeli. Bardzo proste modele budowane na początku XX wieku składały się zazwyczaj z jednej prostej lub lekko uformowanej rury. Jej zadaniem było odprowadzenie spalin poza zakres karoserii samochodu, tak aby odprowadzić spaliny w takie miejsce, w którym były najmniej uciążliwe dla kierowcy i pasażerów. Wraz z rozwojem przemysłu samochodowego, a wraz z nim kolejnych wersji silników spalinowych, układy wydechowe nabierały większego znaczenia. W pewnym momencie samo odprowadzenie spalin już nie wystarczało, gdy silniki stawały się coraz głośniejsze wraz ze wzrostem ich mocy, układ zaczął pełnić dodatkowo rolę tłumika hałasu. Dzisiaj do najważniejszych zadań układu oczyszczania należy bezpieczne odprowadzenie spalin, ograniczenie hałasu spowodowanego pracą silnika spalinowego oraz ograniczenie emisji szkodliwych i toksycznych składników spalin przez zastosowanie różnego rodzaju katalizatorów oraz filtrów cząstek stałych jak to pokazuje rys.2.1.



Rys. 2.1 Zadania układu wydechowego

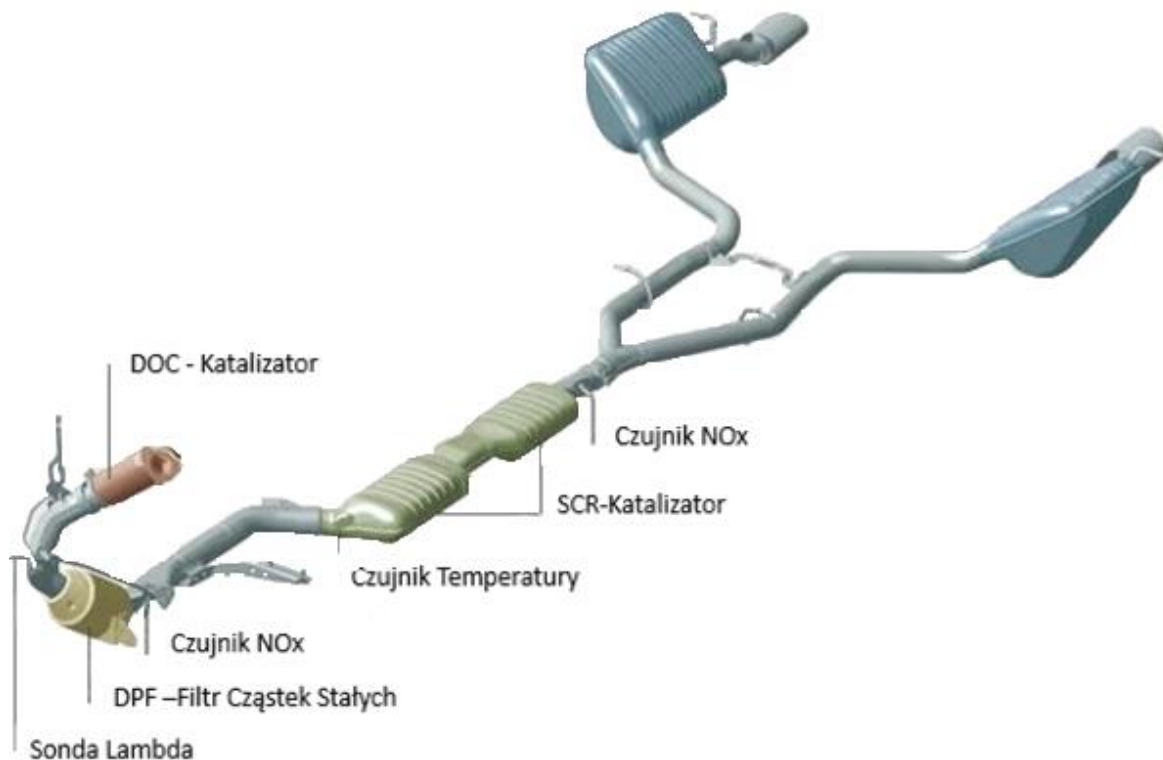
Układ wydechowy ma duży wpływ na prawidłową pracę silnika i jego parametry pracy. Udowodniono, że można wpłynąć na wartość mocy silnika poprzez odpowiednią konstrukcję układu. Przeciwnościem jakie generują kolejne podzespoły układu stojące na drodze przepływu spalin jest wartością dzisiaj regulowaną i opisaną precyzyjnie w każdym wymaganiach stawianych przez producentów samochodów.

Generalnie układy oczyszczania spalin zwane potocznie układami wydechowymi lub wylotowymi spalin zbudowane są głównie z dwóch podzespołów:

1. hot end – tzw. część gorąca układu, która obejmuje wszelkiego rodzaju katalizatory, DPF, SCR montowane w różnych konfiguracjach.
2. cold end – tzw. część zimna, do której zaliczamy najczęściej wszystkie tłumiki, rury, czasem katalizator LNT, który jest montowany za ostatnim katalizatorem SCR.

Oczywiście tak przedstawiona budowa układu wydechowego jest bardzo prostym opisem tego co spotykamy w praktyce na co dzień. Tak naprawdę wymagania stawiane układom wydechowych zawarte w tzw. ‘książce wymagań’ (*Book of Requirements*) definiują oraz narzucają pewne rozwiązania ostateczne, poprzez dokładne zdefiniowanie parametrów jakie układ ma spełniać. Generalnie część gorąca, to te wszystkie podzespoły, które są umieszczone blisko silnika, odpowiadają za oczyszczanie spalin. Druga część układu odpowiada głównie za jego akustykę. Poprzez zaprojektowanie odpowiednich tłumików i dobór odpowiedniej pojemności układu możemy uzyskać odpowiednie

brzmienie silnika. Rys. 2.2 przedstawia jeden z nowoczesnych układów wydechowych stosowanych w samochodach osobowych. Przy założeniu, że mamy do czynienia z idealnym procesem spalania, dostarczone paliwo składające się z węglowodorów spala się całkowicie wraz z tlenem dostarczonym z powietrza. W wyniku spalania powstają dwutlenek węgla CO_2 oraz para wodna H_2O . Natomiast azot oraz gazy obojętne nie biorą udziału w procesie spalania. Jednak w rzeczywistości proces spalania jest daleki od idealnego, a azot i siarka tworzą szkodliwe składniki spalin takie jak tlenek węgla CO , tlenki azotu NO_x , niespalone węglowodory HC , cząstki stałe (sadza) oraz siarkowodór. [102,103,148].



Rys. 2.2 Podstawowe elementy układu oczyszczania spalin [145]

W celu ograniczenia toksyczności spalin producenci silników spalinowych skupiają się głównie na:

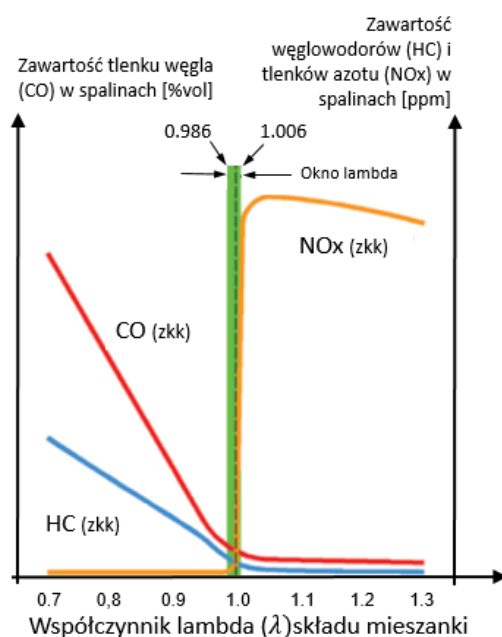
- regulacji i sterowaniu podzespołami silnika, co ma wpływ na jego cykl pracy (cykl roboczy)
- odpowiednich rozwiązaniach konstrukcyjnych silników spalinowych
- stosowaniu odpowiednich rodzajów paliw i wszelkich dodatków
- wyposażeniu silników w katalizatory i filtry spalin

Aktualnie stosujemy różne metody oczyszczania spalin, które są powiązane bezpośrednio z konstrukcją samych silników, jak i układów wylotowych spalin. Do tych pierwszych zaliczamy m.in. szeroko stosowane EGR (*Exhaust Gas Recirculation*) LP (*Low Power*) i HP (*High Power*), które odpowiadają za recyrkulację spalin, niosąc ze sobą odpowiednie korzyści, m.in. takie jak:

- a) obniżenie temperatury w komorze spalania
- b) zmniejszenie ilości tlenków azotu NO_x wygenerowanych w komorze spalania silnika
- c) poprawianie skuteczności spalania mieszanki,

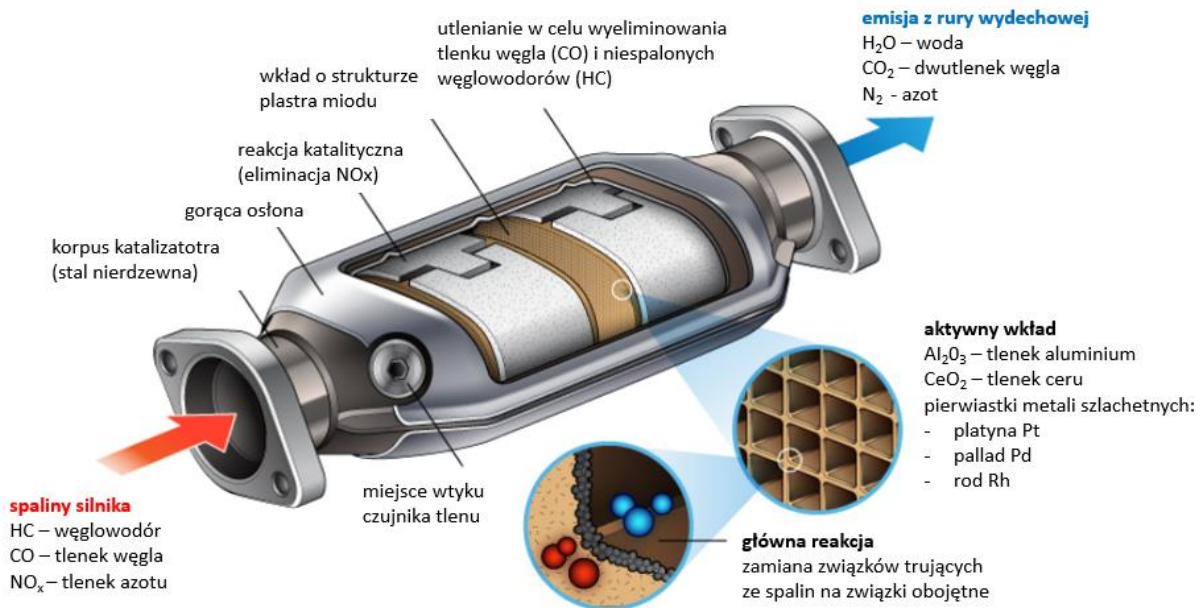
Pojawia się niestety również skutek uboczny w postaci wzrostu ilości cząstek stałych w spalinach. Do metod pozasilnikowych możemy głównie zaliczyć wszystkie najnowsze technologie oczyszczania spalin ze wszelkich szkodliwych emisji stosując:

- katalizatory trójfunkcyjne TWC (*Three Way Catalysts*),
- filtry cząstek stałych DPF (*Diesel Particulate Filter*)
- katalizatory SCR (*Selective Catalytic Reduction*) do redukcji głównie NO_x
- LNT (*Lean NOx Traps*)



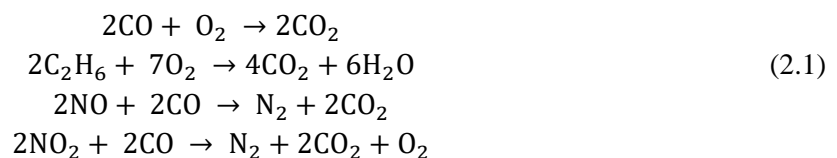
Rys. 2.3 Współczynnik Lambda (λ) składu mieszanki [151]

Wraz z rozwojem motoryzacji szybko zdano sobie sprawę, że samochodów przybywa zbyt szybko, i pojazdy emitują do atmosfery znaczne ilości zanieczyszczeń, szkodliwych substancji, które szkodzą środowisku jak i ludziom. Na Rys. 2.3 możemy zobaczyć jak kształtuje się ilość wydzielanych substancji szkodliwych w zależności od wartości współczynnika Lambd. Zaczęto rozważać i w końcu wprowadzono pierwsze rozporządzenia, dyrektywy, przepisy i normy emisji spalin, które narzuciły pewne ograniczenia i zmusiły producentów samochodów i wszelkich innych pojazdów do prac nad redukcją szkodliwych substancji emitowanych w spalinach. Pierwsze regulacje w tym zakresie pojawiły się dopiero pod koniec lat 80-tych, a pierwsze normy dedykowane dla poszczególnych kontynentów natomiast w latach 90-tych.

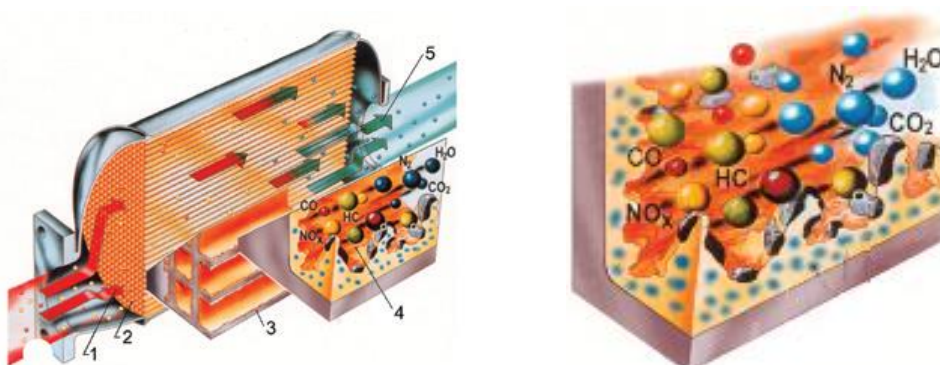


Rys. 2.4 Budowa katalizatora trójdrożnego [152]

Poprzez wprowadzenie normy Euro-3 na podstawie unijnej Dyrektywy 98/69/EC dla samochodów osobowych i 93/59/EEC dotyczącej wszystkich pojazdów wymuszono na producentach samochodów montowanie w układach wydechowych katalizatorów DOC (*Diesel Oxydation Catalyst*). Ich doskonalszą wersją są tzw. katalizatory trójdrożne (*Three Way Catalysts*) (Rys. 2.4), które utleniają w głównej mierze tlenki węgla CO i węglowodory CH oraz redukują tlenki azotu NO_x wg następujących reakcji chemicznych:



Produktami końcowymi przemiany są para wodna H₂O, dwutlenek węgla CO₂ oraz azot N₂ (Rys. 2.5).

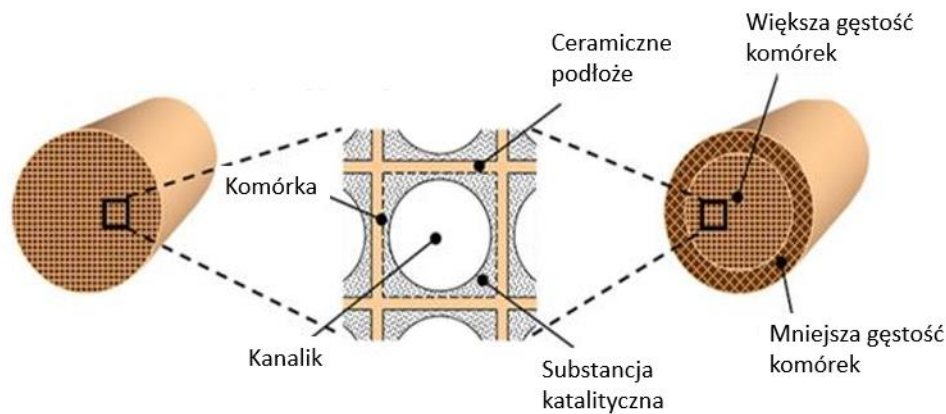


Rys. 2.5 Katalizator TWC – proces redukcji i utleniania substancji szkodliwych. [151]

1 - Gazy spalinowe nieoczyszczone, 2 - Kanaliki katalizatora pokryte materiałami szlachetnymi, 3 - Ceramiczna konstrukcja katalizatora, 4 - Warstwa materiałów utleniających, 5 - Oczyszczone gazy spalinowe

W katalizatorach trójfunkcyjnych nie ma możliwości redukcji wszystkich szkodliwych związków sterując tylko składem mieszanki paliwowej. Przy wzroście współczynnika lambda (λ) rośnie również stopień przemiany węglowodorów HC i tlenku węgla CO. Przy $\lambda=1$ udział tych dwóch toksycznych składników jest nieznaczący i pozostaje na tym samym poziomie dla jej wyższych wartości. Jeżeli chodzi o redukcję tlenków azotu NO_x, które powstają przy stosowaniu mieszanki bogatej $\lambda < 1$, to ich stężenie jest najmniejsze dla mieszanki stechiometrycznej gdy $\lambda = 1$. Jednak niewielkie zwiększenie tlenu w spalinach $\lambda > 1$ powoduje nagły wzrost stężenia NO_x, co można zauważyć na Rys. 2.3. Z tego powodu katalizatory trójfunkcyjne wymagają bardzo restrykcyjnej kontroli współczynnika λ i pracują efektywnie, biorąc pod uwagę wartości wszystkich trzech szkodliwych składników, tylko w bardzo wąskim przedziale wartości współczynnika λ .

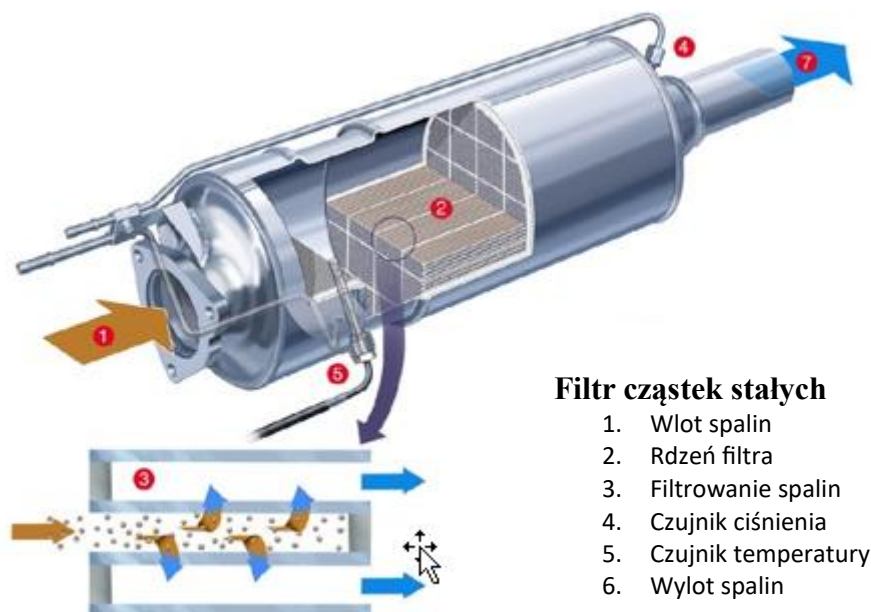
Wewnętrzna struktura katalizatora DOC (Rys. 2.6) składa się z tysięcy kanalików o przekroju przypominającym plaster miodu. Może on być wykonany z metalu, ale bardziej powszechne w użyciu są katalizatory ceramiczne z krzemianu aluminowo magnezowego. Cała powierzchnia pośrodkowa kanalików jest porowata i zapewnia dobrą przyczepność do powłoki aktywnej.



Rys. 2.6 Przekrój monolitu ceramicznego [153]

Powłoka aktywna katalitycznie wykonana jest z pierwiastków metali szlachetnych takich jak platyna, rod lub pallad. Pierwiastki te wchodzi w reakcje chemiczne ze składnikami spalin i zubożniają je do substancji nieszkodliwych dla środowiska. Są stosowane zarówno w silnikach benzynowych, jak również wysokoprężnych (Diesla). Oksydacja, czyli utlenianie w głównej mierze jest możliwe dzięki platynie, która powoduje, że proces utleniania rozpoczyna się już w temperaturze 200°C by w temperaturze około 550 - 600°C uzyskać skuteczność niemal 90% w przypadku CO i niektórych węglowodorów HC. Oczywiście katalizatory silników benzynowych różnią się od katalizatorów silników wysokoprężnych budową powłoki aktywnej ponieważ skład spalin dla obu rodzajów silników jest także różny. Przed i za katalizatorem wbudowane są liczne czujniki, gdzie na podstawie ich odczytów jednostka sterująca dobiera optymalną mieszankę paliwowo powietrzną. Punkt pracy katalizatora tzn. temperatura jaką muszą osiągnąć spaliny by w katalizatorze zachodziły reakcje katalityczne, wynosi minimum 300 stopni Celsjusza, jednak optymalne działanie katalizatora następuje, gdy wkład ceramiczny osiągnie temperaturę między 400 a 800 stopni Celsjusza. Temperatura punktu pracy katalizatora rośnie wraz ze wzrostem czasu użytkowania samochodu.

Coraz bardziej restrykcyjne normy emisji sprawiły, że same katalizatory przestały wystarczać do oczyszczania spalin, szczególnie emitowanych przez silniki wysokoprężne. W 1995 roku konieczne stało się dodatkowo stosowanie filtrów cząstek stałych (Rys. 2.7).



Filtr cząstek stałych

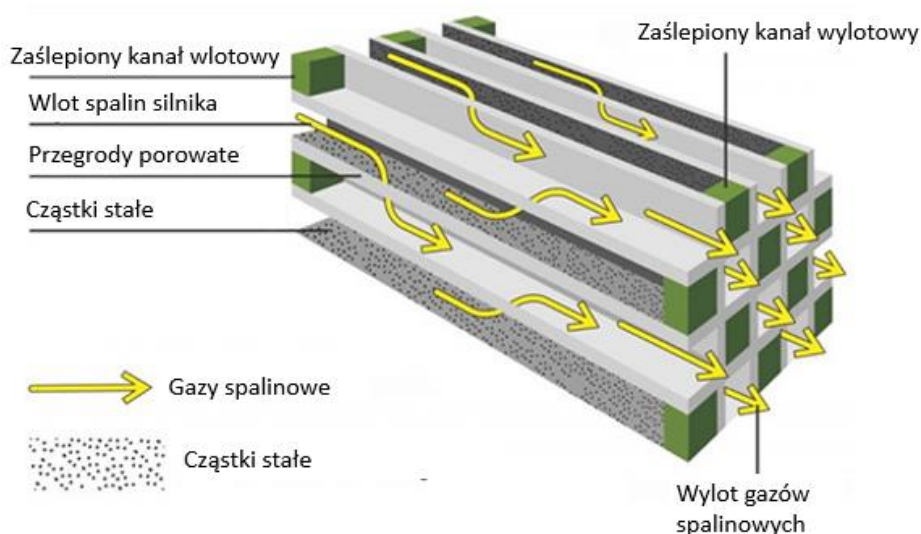
1. Wlot spalin
2. Rdzeń filtra
3. Filtrowanie spalin
4. Czujnik ciśnienia
5. Czujnik temperatury
6. Wylot spalin

Rys. 2.7 DPF - Struktura Filtra Cząstek Stałych [154]

Wymusiła to norma Euro-4. Jej głównym zadaniem jest po dzień dzisiejszy wyłapywanie ze spalin cząstek stałych, głównie sadzy. W fachowej literaturze możemy znaleźć kilka skrótów określających filtr cząstek stałych – najbardziej popularne to FAP (fr. *Filtre à Particules*) i DPF (ang. *Diesel Particulate Filter*) oraz GPF (ang. *Gasoline Particulate Filter*), od szeregu lat stosowane w samochodach z silnikiem benzynowym. Jak sama nazwa wskazuje zadaniem filtrów cząstek stałych jest zatrzymanie cząstek stałych znajdujących się w spalinach wytworzonych przez silnik. Są one niezwykle trujące. Niewielkie rozmiary ($2,5\mu\text{m}$) tych cząstek powodują wysokie zagrożenie dla zdrowia ludzi. Bez trudu dostają się one do układu oddechowego, a nawet krwionośnego. Filtr cząstek stałych, który pracuje w optymalnych dla niego warunkach, jest w stanie nawet kilkukrotnie obniżyć ilość sadzy wylatującej z układu wydechowego. Generalnie jego rola sprowadza się do dwóch funkcji, po pierwsze zatrzymuje cząstki stałe, po drugie je wypala. Proces wypalania jest potocznie określany jako regeneracja DPF.

Brak procesu wypalania cząstek stałych skraca żywotność filtra bardzo drastycznie, a można by zaryzykować stwierdzenie, że byłby to substrat jednorazowy. Filtr cząstek stałych swoją budową jest bardzo podobny do zwykłego katalizatora. Ceramiczny wkład filtra posiada mocno porowatą strukturę składającą się z bardzo dużo drobnych równoległe ułożonych kanałów, które z jednego swego końca są zaślepione (Rys. 2.8). Spaliny, chcąc się wydostać, muszą przenikać przez porowate ścianki pomiędzy kanalikami pozostawiając na ściankach cząsteczki sadzy, których wielkość jest większa od wielkości porów ceramiki. Ścianki kanalików wykonane są z węgla krzemowego albo kordierytu (tańsza wersja), pokrytego tlenkiem glinu oraz tlenkiem ceru. Na nich osadzone są cząstki metali szlachetnych jak np. platyna. Oprócz wkładów ceramicznych można także spotkać DPF-y z wkładem metalowym, które są trwalsze.

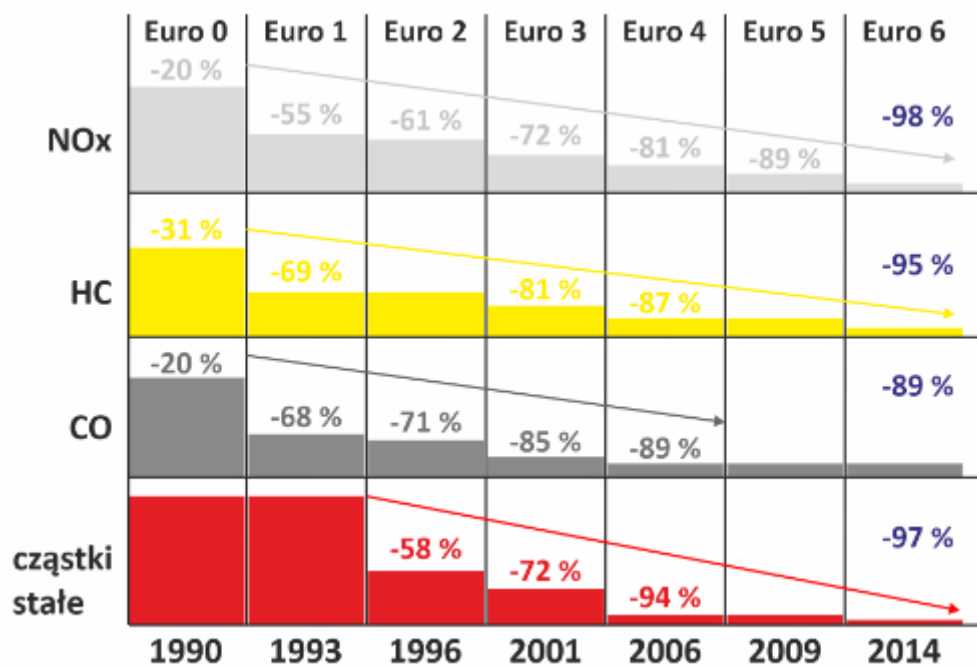
Filtr cząstek stałych



Rys. 2.8 Proces osadzania cząstek stałych w filtrze [155]

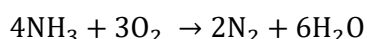
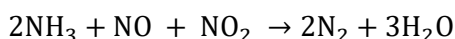
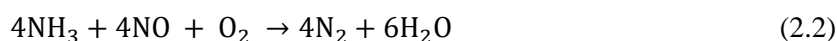
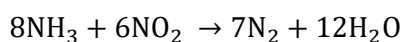
Z czasem nagromadzone cząstki stałe na ściankach kanalików blokują przepływ spalin przez filtr. Wtedy potrzebny jest proces wypalania tych cząstek w wyniku którego sadza zmienia się w dwutlenek węgla. Proces ten nazywany jest regeneracją filtra lub bardziej potocznie dopaleniem filtra. Wypalanie następuje w dwojaki sposób, aktywny oraz pasywny. Z pasywnym sposobem mamy do czynienia podczas przepływu gorących spalin w zakresie 350 do 500°C z odpowiednią prędkością poprzez filtr, który osiągnął już swoją temperaturę pracy. Wtedy następuje samoistne wypalanie pewnej części cząstek sadzy, ale jest to niestety tylko niewielka ich część. By oczyścić filtr z cząstek skutecznie, potrzebne jest wypalanie aktywne uruchamiane w miarę potrzeby poprzez sterownik silnika. O tym kiedy powinna rozpocząć się aktywna regeneracja informują sterownik odpowiednio zamontowane przed i za filtrem czujniki (czujnik temperatury, ciśnienia). W trakcie aktywnej regeneracji filtra minimalnie zwiększa się zużycie paliwa i podnosi się temperatura spalin nawet do 600°C.

W 2014 roku Parlament Europejski dyrektywą nr. 715/EC wraz z poprawkami z 2012 roku ujętymi w rozporządzeniu nr 459/2012 [80] wprowadził na terenie Unii Europejskiej normę emisji spalin Euro 6. Skupia się głównie na ograniczeniach emisji spalin dla samochodów z silnikami wysokoprężnymi. Dzisiaj mamy już kolejną wersję tejże normy, która jeszcze bardziej doprecyzowuje graniczne wartości emisji szkodliwych substancji w spalinach - Euro 6d. Jak to wyglądało na przestrzeni ostatnich 30 lat, jakie wartości emisji spalin dla poszczególnych norm zostały wprowadzane przedstawia Rys. 2.9.

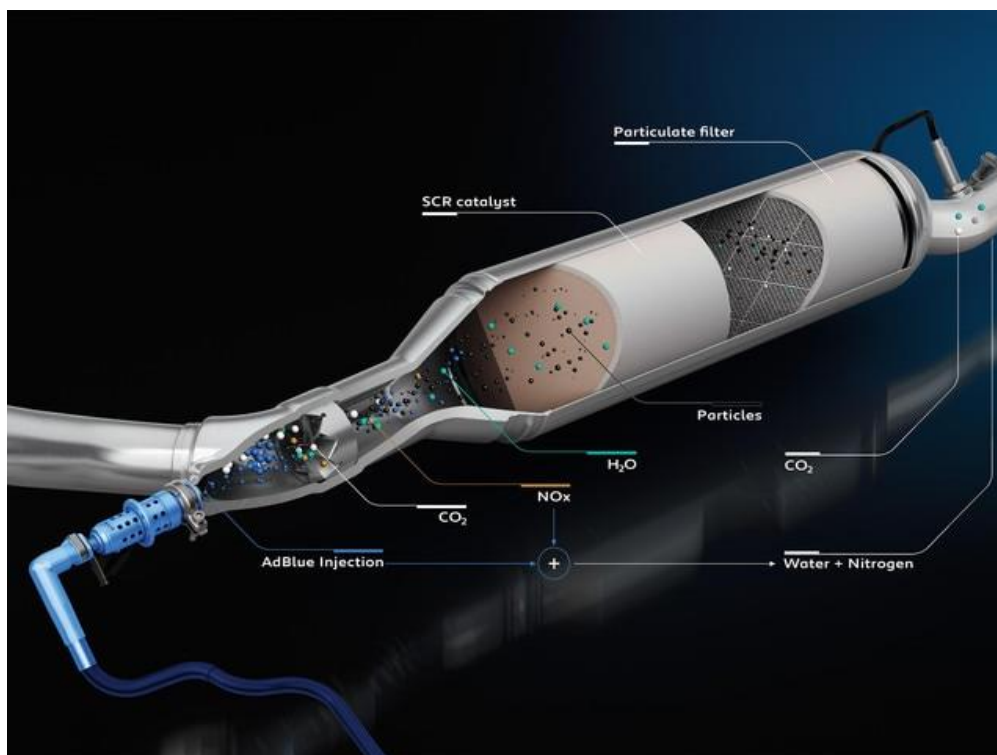


Rys. 2.9 Normy emisji spalin Euro na przestrzeni ostatnich 30 lat [156]

Norma Euro 6 wymusiła stosowanie dodatkowych technologii oczyszczania spalin. Jednym z powszechnie stosowanych rozwiązań jest katalizator SCR (*Selective Catalytic Reduction*), którego zadaniem jest znacząca redukcja szkodliwych tlenków azotu. Katalizator SCR pod względem budowy jest podobny do wcześniej opisanego DPF, ale montowany jest z dodatkowym wtryskiwaczem specjalnego płynu DEF (Diesel Exhaust Fluid)/Adblue. Jest to nietoksyczny, bezwonny 32.5% roztwór mocznika w dejonizowanej wodzie. Rozpylenie roztworu mocznika, ma na celu wymieszanie go ze gazami spalinowymi w wyniku czego następuje reakcja chemiczna podczas której redukowane są tlenki azotu. Substancję tę wtryskuje się do gorących spalin, gdzie po zetknięciu z nimi wydziela się amoniak, który w katalizatorze wiąże zawarte w spalinach tlenki azotu, zamieniając je w parę wodną i azot. W wyniku działania SCR następuje redukcja tlenków azotu o około 85%. Reakcje jakie zachodzą podczas tego procesu pokazano poniżej:



W starszych rozwiązaniach jest montowany za filtrem cząstek stałych. W najnowszych rozwiązaniach jednak można spotkać konstrukcje, gdzie w jednej obudowie zabudowany jest DPF i SCR (Rys. 2.10). W niektórych układach dla wyłapania resztek amoniaku ze spalin montowany jest dodatkowo katalizator LNT (*Lean NOx Traps*).



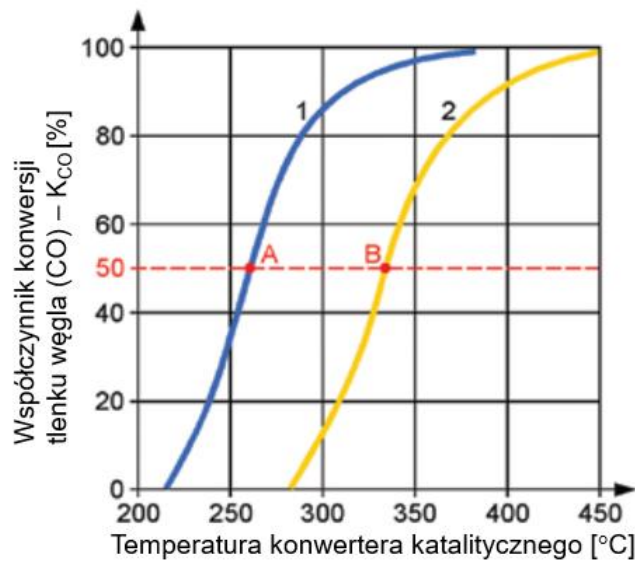
Rys. 2.10 SCR i DPF – wspólna zabudowa [157]

Informacje z czujnika temperatury spalin oraz z sond tlenku azotu, usytuowanych przed i za katalizatorem SCR, wykorzystywane są do prawidłowego sterowania pracą SCR.

Opisane powyżej podzespoły układu wydechowego wymagają specjalnych warunków pracy, by ich skuteczność w oczyszczaniu (utlenianiu, redukcji) zanieczyszczeń była wysoka. Jednym z bardzo ważnych parametrów pracy DPF-u, SCR-u czy DOC jest jego temperatura pracy – tzw. punkt pracy. Istotnym jest by osiągnąć go jak najszybciej. W tym celu albo przesuwa się te podzespoły jak najbliżej silnika, gdzie temperatura spalin jest znacznie wyższa niż w drugiej jego połowie – tzw. *'cold end'*, albo stosuje się dodatkowo przeróżne osłony termiczne wszystkich elementów poprzedzających DOC, DPF czy SCR w układzie oczyszczania spalin. Jak można zauważyć na Rys.2.11 temperatura np. katalizatora trójdrożnego, w której osiąga on swój optymalny punkt pracy wynosi 380°C.

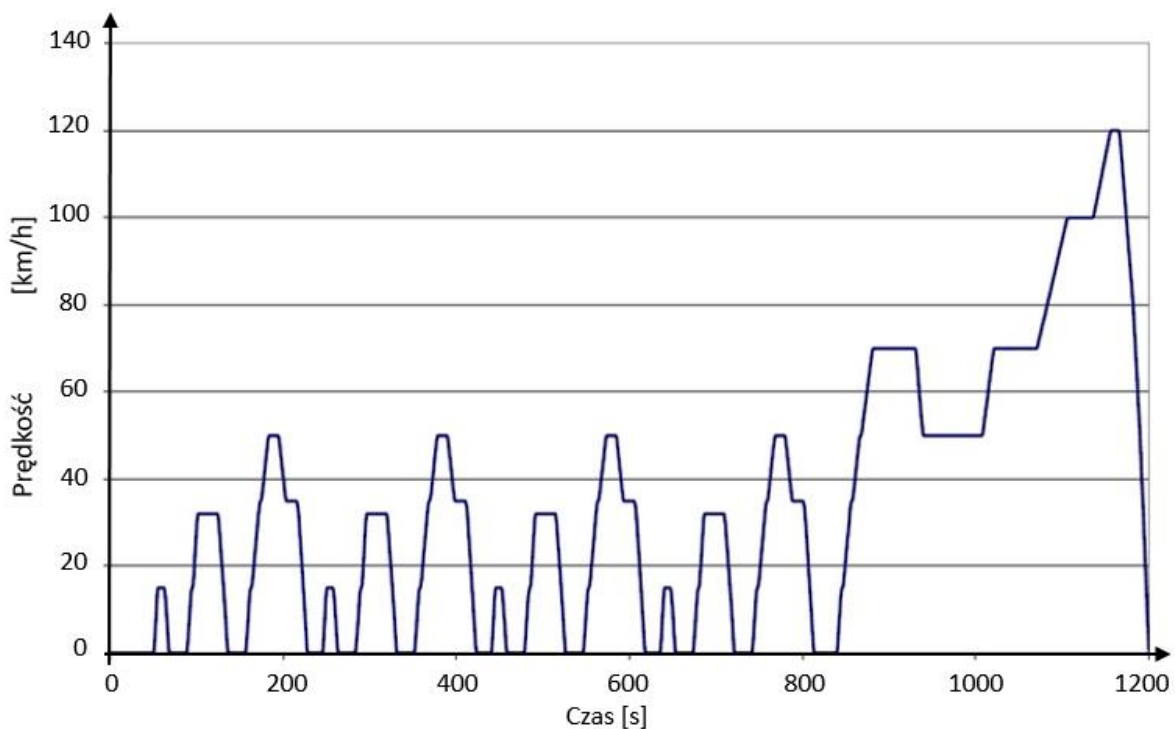
Jednakże wraz ze wzrostem czasu użytkowania, niestety ta temperatura rośnie i może wynosić nawet 450°C. Tak wysokie temperatury wymuszają stosowanie osłon termicznych podzespołów układu oczyszczania spalin i generują dodatkowe wymagania dla układu wydechowego.

Porównując układy wydechowe z lat 90-tych ubiegłego wieku z aktualnie produkowanymi, można dostrzec przepaść jaka je dzieli jeżeli chodzi o rodzaj zaawansowanej technologii użytej do ich produkcji. Wzrost stopnia zaawansowania technologicznego układów wydechowych wymuszały kolejno coraz bardziej restrykcyjne wprowadzane przez Komisję Europejską normy emisji spalin. To one wymuszały montowanie w układach dodatkowych podzespołów, które z kolei wymagają specyficznych warunków pracy, by mogły pracować efektywnie. Jednym z nich jest – jak już wspomniano - odpowiednia temperatura pracy. Utrzymanie jej na wyższym poziomie wymusza stosowanie osłon termicznych, które albo chronią przed nią sąsiadujące elementy płyty podłogowej samochodu, albo chronią spaliny przed nadmiernym i zbyt wczesnym ich wychłodzeniem.



Rys. 2.11 Temperatura pracy katalizatora [151]

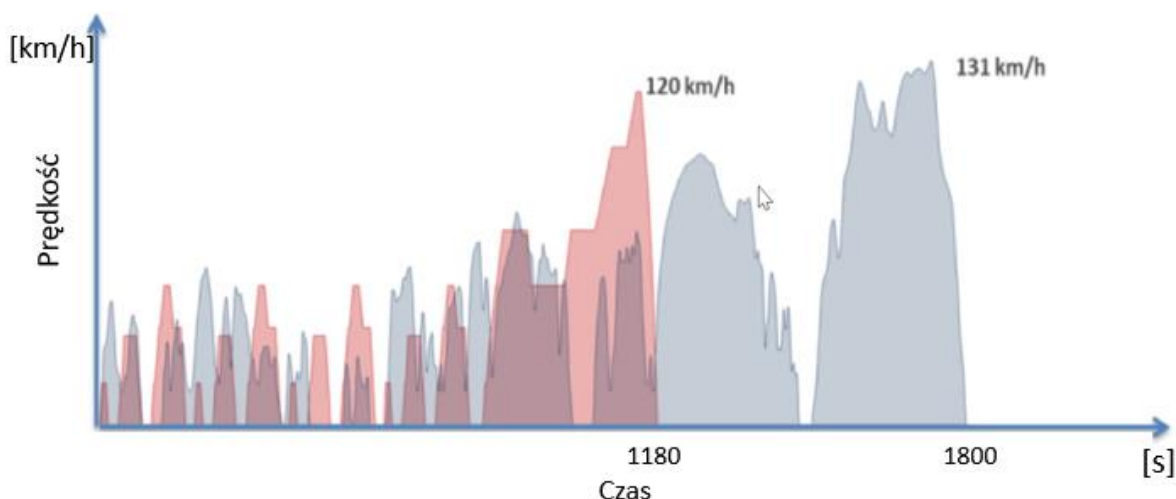
Wszyscy producenci samochodów są zobligowani do przestrzegania obowiązujących norm. To czy dana norma jest przestrzegana pokazuje nam emisja CO_2 i test zużycia paliwa NEDC, zdefiniowany w rozporządzeniu Komisji (WE) nr 692/2008 zastąpiony wg wymagań nowej dyrektywy Komisji (UE) 2017/1151, testem WLTP. Pomiary te dokonuje się przy pomocy zdefiniowanych procedur testowych, które w sposób precyzyjny określają przebieg testu wraz z definicją warunków brzegowych przeprowadzenia takiego testu. Na Rys. 2.12 możemy zobaczyć przykładowy jeden z kilku zdefiniowanych testów pomiaru zanieczyszczeń spalin NEDC (*New European Driving Cycle*) obowiązujących w Europie do roku 2017.



Rys. 2.12 Test NEDC [158]

Wykres przedstawia zależność prędkości samochodu od czasu. Precyzyjnie określa zależność wartości przebytego dystansu, prędkości oraz czasu jego realizacji oraz wymaganego czasu postoju. Odbywa się to wg ściśle opisanego schematu. O ile same wyniki przeprowadzone wg tej samej procedury można było porównać między sobą, to już wyniki z tego rodzaju testu w porównaniu z pomiarami podczas rzeczywistej eksploatacji samochodu różniły się między sobą. Dlatego opracowano parę lat temu nową procedurę - WLTP, która opisuje testowanie samochodów i określa ilości emitowanych zanieczyszczeń. Od dłuższego czasu Komisja Europejska dążyła do zastąpienia testu NEDC testem WLTP (*Worldwide harmonized Light vehicles Test Procedures*), przeprowadzonym w warunkach bliższym rzeczywistym, podobnie jak w USA. Nowy test WLTP jest bardziej dynamiczny, trwa trochę dłużej, a do pokonania jest dystans ponad dwukrotnie większy. Jednocześnie średnia prędkość rośnie o 40 procent, a czas postoju zmalał o połowę. Rys. 2.13 pokazuje ich porównanie.

Komisja Europejska postanowiła jednak sprawdzać poprawność wyników testu WLTP poprzez dodatkowy test skoncentrowany na weryfikacji emisji szkodliwych substancji w spalinach w warunkach drogowych. Test RDE '*Real Driving Emissions*' nie zastępuje lecz jedynie uzupełnia test WLTP. Badanie emisji w rzeczywistych warunkach jazdy ma na celu ilościowe określenie emisji w samochodach za pomocą przenośnych systemów pomiarowych tzw. '*Portable Emissions Measurement Systems*' (PEMS). Zebrane dane poddane są analizie by zweryfikować, czy warunki graniczne systemu RDE zostały spełnione oraz czy normy emisji mieszczą się w określonych limitach. Limity graniczne, wynikają z testu laboratoryjnego (WLTP).



Rys. 2.13 Porównanie testów NEDC i WLTP [159]

Test RDE, wprowadzony w 2017 r. na etapie Euro 6d-TEMP, okazał się skutecznym narzędziem, które wymusiło stosowanie bardziej niezawodnych systemów i strategii oczyszczania spalin z NO_x, przyczyniając się do znacznie zwiększonego poziomu zgodności emisji NO_x w pojazdach z silnikiem Diesla, w większości warunków eksploatacji. Test RDE odegrał również istotną rolę w wymuszeniu powszechnego stosowania filtrów cząstek stałych (GPF) '*Gasoline Particulate Filter*' w samochodach benzynowych.

Kolejnym wyzwaniem dla konstruktorów układów wydechowych jest minimalizacja drgań układu i jego poszczególnych podzespołów. Układ wydechowy podczas pracy silnika mimo elastycznego połączenia z silnikiem wpada w wibracje i przemieszcza się wraz z wychyleniami silnika. Aby układ wydechowy mógł się przemieszczać w dopuszczalnym zakresie i nie przenosił drgań na

nadwozie musi być zawieszony w kilku miejscach na elastycznych gumowych (elastomery) wieszakach (Rys. 2.14).



Rys. 2.14 Elastyczny element zawieszenia układu wydechowego [145]

Głównym źródłem drgań jest silnik spalinowy, który generuje dwa rodzaje drgań:

- mechaniczne – wywołane drganiem samego korpusu silnika podczas pracy
- wynikające z pulsacji gazów spalinowych, które poprzez zawory w silniku i kolektor tłoczone są do układu wydechowego

Najbardziej podatne na drgania są drobne, lekkie elementy, do których zaliczamy m.in. osłony termiczne.

Jednym z ważniejszych aspektów dotyczących projektowania osłon, jest ich optymalny kształt niwelujący prawdopodobieństwo powstania zjawiska rezonansu. Odpowiednio zaprojektowany i zoptymalizowany, pozwala uniknąć problemów w fazie testowania i późniejszej eksploatacji układu. Powierzchnia osłony, jej grubość, przetłoczenia usztywniające oraz liczba i położenie punktów mocowania, ma istotny wpływ na powstawanie lub przenoszenie drgań.

Równania opisujące teoretyczne podstawy drgań własnych w układach mechanicznych, wyprowadza się z ogólnego równania ruchu dla drgań wymuszonych tłumionych [53,58]. Dla układu o jednym stopniu x swobody ma ono postać:

$$m\ddot{x} + c\dot{x} + kx = P * \sin(\omega t + \varphi) \quad (2.3)$$

gdzie: m jest masą układu, k sztywnością, c współczynnikiem tłumienia, P siłą wymuszającą. Po obustronnym podzieleniu przez masę, równanie przyjmuje postać:

$$\ddot{x} + 2h\dot{x} + \omega_o^2 x = \frac{P}{m} * \sin(\omega t + \varphi) \quad (2.4)$$

gdzie:

$$2h = \frac{c}{m} \quad \omega_o^2 = \frac{k}{m} \quad (2.5)$$

stała $h = \frac{c}{2m}$ jest współczynnikiem tłumienia względnego, natomiast

$$\omega_o = \sqrt{\frac{k}{m}} \quad (2.6)$$

jest częstością drgań własnych.

W przypadku nietłumionych drgań swobodnych o jednym stopniu swobody, równanie (2.3) przybiera postać:

$$m\ddot{x} + kx = 0 \quad (2.7)$$

Rozwiązaniem równania drgań własnych jest całka ogólna równania, dla której x przyjmuje postać

$$x = C_1 \cos(\omega t) + C_2 \sin(\omega t) \quad (2.8)$$

W rozwiązaniu należy uwzględnić również warunek początkowy w postaci:

$$x_0 = A_0 \sin(\omega t) \quad (2.9)$$

gdzie: ω jest częstotliwością drgań własnych, A_0 jest amplitudą początkową drgań własnych

Zgodnie ze wzorem (2.6) wyższa sztywność przenosi częstotliwość drgań w wyższe zakresy. Typowy silnik spalinowy 4 cylindrowy o zapłonie iskrowym pracuje w zakresie obrotów pomiędzy 600 – 6900obr/min. Korzystając ze wzoru (2.10):

$$f = \frac{\text{obr}}{60} \cdot \frac{\text{il.cylindrów}}{\text{rząd silnika}} \quad [\text{Hz}] \quad (2.10)$$

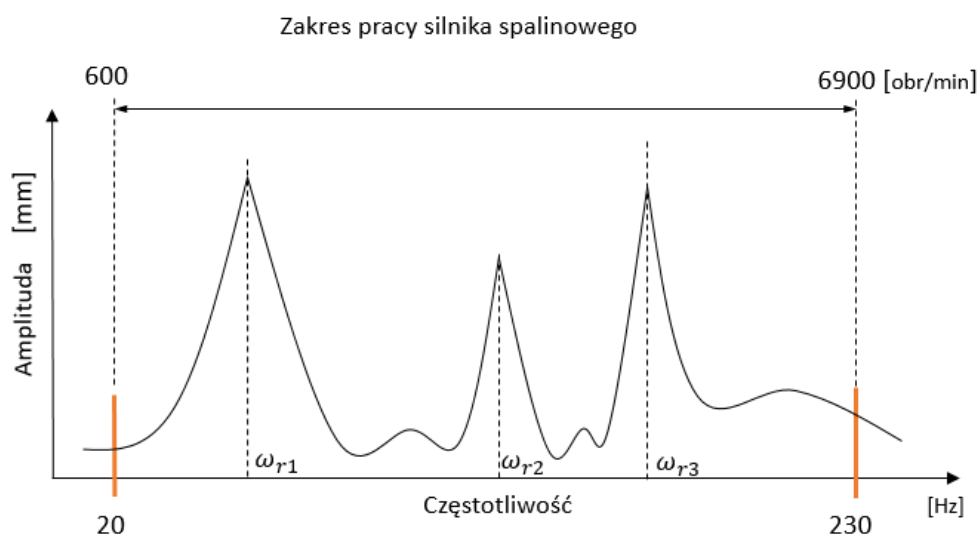
gdzie:

obr – obroty silnika spalinowego [obr/min]

ilość cylindrów - silnik spalinowy 4-ro cylindrowy

rząd silnika – dla wybranego silnika 4-cylindrowego jest równy 2

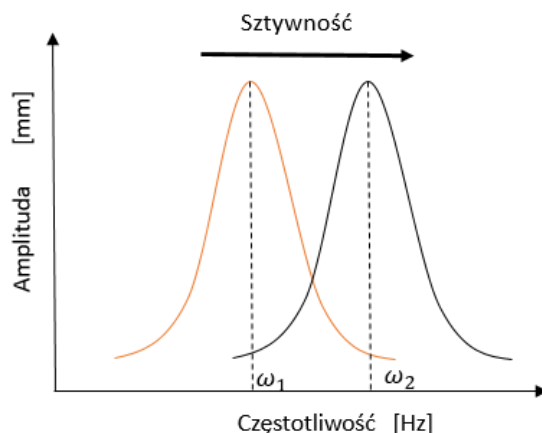
można przedstawić obroty silnika w widmie częstotliwościowym. Podany zakres obrotów odpowiada zakresowi częstotliwości pracy silnika pomiędzy 20 – 230Hz (Rys. 2.15). Z tego względu jednym z celów jest tak zaprojektować m.in. osłony, by ich pierwsza częstotliwość drgań własnych była wyższa niż 230Hz, co pozwala na uniknięcie zjawiska rezonansu.



Rys. 2.15 Zakres pracy silnika spalinowego benzynowego, 4 cylindrowego

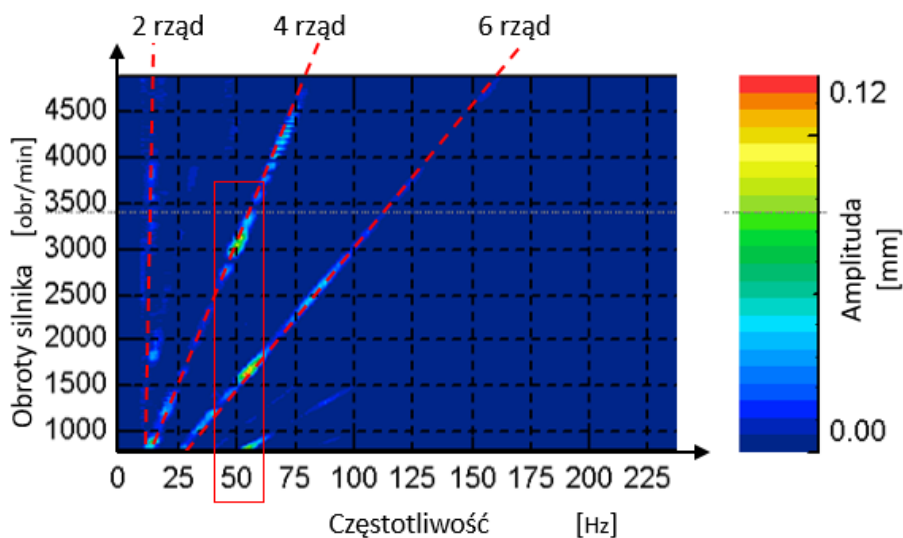
Dla analizy modalnej układu wydechowego, najważniejszymi z punktu widzenia występowania rezonansu jest pierwszych pięć częstotliwości drgań własnych będące zwykle drganiami całego układu. Drgania te prowadzą głównie do przenoszenia dźwięków poprzez zawieszenie układu do kabiny samochodu. Tłumiki umieszczone w tylnej części układu drgają jak ciała sztywne. Powyżej 180Hz tworzą się pierwsze częstotliwości drgań własnych osłon. W celu zmniejszenia amplitud drgań tłumika czy zamontowanej na nim osłony, a tym samym emitowanego hałasu, odpowiednio sterujemy parametrami sztywności i tłumienia. Zwiększenie sztywności układu skutkuje przesunięciem rezonansu w wyższe zakresy częstotliwości, jak pokazano na Rys. 2.16. Celem jest, przesunięcie rezonansu poza

zakres pracy silnika tj. 230Hz. Usztywnienie konstrukcji nie eliminuje jednak drgań tylko przenosi je w inny zakres częstotliwości. Jak możemy zauważyć na Rys. 2.15, gdzie w całym zakresie pracy silnika spalinowego przedstawionego w widmie częstotliwościowym widoczne są trzy częstotliwości rezonansowe, które w dalszym etapie projektowania układu należy wyeliminować.



Rys. 2.16 Wpływ sztywności na zjawisko rezonansu

Te same częstotliwości możemy zobaczyć na innym wykresie często wykorzystywanym przez akustyków, tj. wykresie (Rys. 2.17). Ten wykres jasno pokazuje poszczególne częstotliwości rezonansowe oraz niesie dodatkowe informacje o tzw. rzędzie silnika, w którym rezonans występuje, co pozwala na szybsze znalezienie przyczyny problemu.



Rys. 2.17 Wykres Chambell'a z wyznaczonymi 'rzędami' silnika i zaznaczonym rezonansem ok. 50Hz [145]

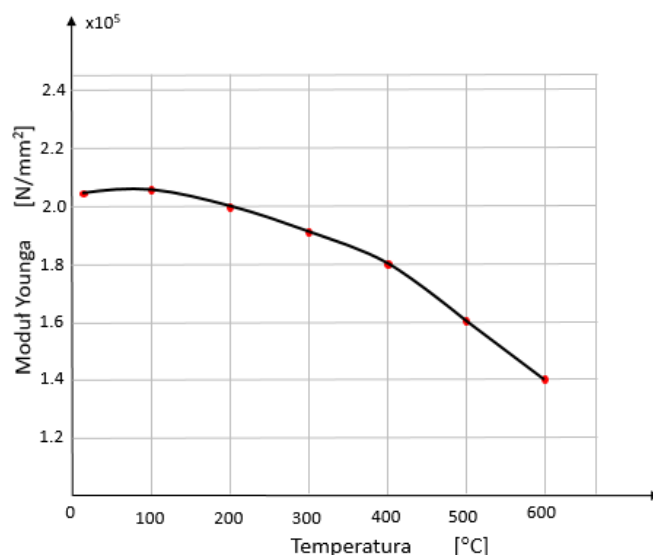
Nie zawsze jednak jest możliwe tak przekonstruować układ by jego wysoka częstotliwość rezonansowa została przesunięta poza zakres pracy silnika. Dla częstotliwości rezonansowych w dolnym zakresie pracy silnika ta metoda się nie sprawdzi. Stąd w takich przypadkach, zgodnie ze wzorem (2.6), pomocne jest zwiększenie masy elementu będącego w rezonansie. Zwiększenie masy przesuwa częstotliwości rezonansowe w niższe zakresy pracy silnika. Zdarzają się czasem przypadki, że samo pogrubienie

ścianki rury, czy zwiększenie grubości osłony problemu nie rozwiązuje. Wtedy używane są zw. masy wyrównowazające (Rys. 2.18), które skutecznie dzielą częstotliwość rezonansową na dwie mniejsze. Są montowane bezpośrednio na danym podzespole układu.



Rys. 2.18 Masa wyrównowazająca [160]

Obniżeniu częstotliwości rezonansowych pomagają również wysoka temperatura komponentów układu. Moduł sprężystości podłużnej Younga i gęstość materiału zależą od temperatury. Rys. 2.19 pokazuje wartości modułu sprężystości dla stali 1.4510, w zależności od temperatury. Wraz ze wzrostem temperatury sztywność konstrukcji maleje, co skutkuje niższymi częstotliwościami drgań własnych. Pomaga ona przesunąć częstotliwości z dolnego zakresu pracy silnika poza zakres jego pracy, ale stanowi problem, gdy częstotliwości rezonansowe zidentyfikowano na granicy górnego zakresu pracy silnika, lub niewiele powyżej tego zakresu.



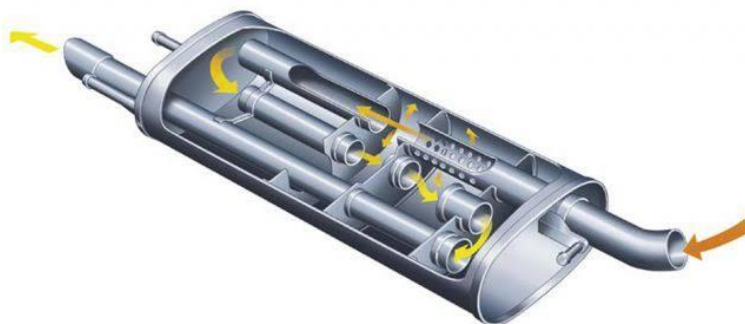
Rys. 2.19 Zależność modułu Younga od temperatury dla stali 1.5410

Temperatura powoduje także dodatkowe naprężenia w komponentach. W przypadku tłumików na przykład ścianki działowe rozszerzają się i tym samym usztywniają konstrukcje całego tłumika. Płaszczki natomiast rozszerzają się i zmniejszają jego sztywność.

Tłumiki projektuje się tak, aby ich pierwsza częstotliwość drgań własnych znajdowała się powyżej zakresu częstotliwości wymuszenia. Problemem stają się rozległe, płaskie powierzchnie które posiadają małą sztywność i mają silną tendencję do wibrowania. Aby zminimalizować tę niedogodność można te elementy usztywnić poprzez następujące zmiany konstrukcyjne:

- wytłoczenia kuliste, stosowane w denkach tłumików
- stosowanie przetłoczeń powierzchni płaszcza tłumika
- stosowanie płaszczy jednowarstwowych
- stosowanie spawanych tłumików w miejsce związanych
- zwiększenie grubości płaszcza tłumika
- stosownie ponadgabarytowych wymiarów przegród wewnętrznych tłumika

Głównym zadaniem tłumika jest jednak redukcja hałasu wywołanego falą ciśnienia spalin przemieszczających się w całym układzie wydechowym (Rys.2.20). Tłumiki są dalszymi częściami składowymi układu wydechowego połączone odpowiednio wyprofilowanymi rurami. Niepożądane dźwięki w tłumikach można ograniczać poprzez zastosowanie różnych ich konstrukcji.



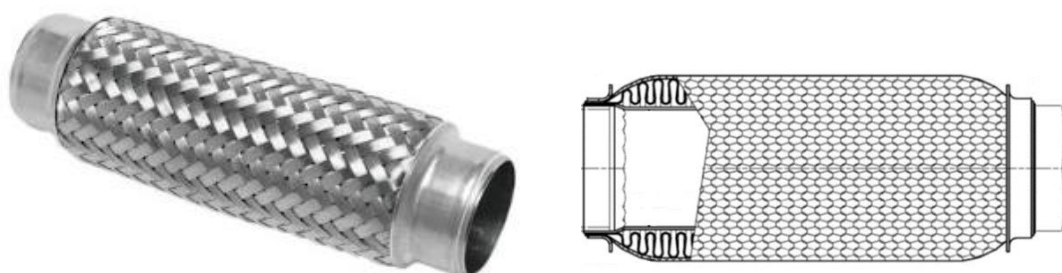
Rys. 2.20 Przebieg spalin w tłumiku refleksyjnym [161]

Rozróżniamy następujące konstrukcje tłumików:

- Tłumik absorpcyjny – wykorzystuje perforowaną rurę, która otoczona jest watą tłumiącą lub całe sektory nią wypełnione. Przepływające przez rurę/przegrodę z otworkami spaliny mają kontakt z watą tłumiącą. Wata tłumiąca w czasie kontaktu ze spalinami tłumi ich drgania na skutek tarcia mechanicznego. Tego rodzaju tłumik tłumi hałas o dużej częstotliwości powyżej 500Hz.
- Tłumik refleksyjny – w tego rodzaju tłumiku fale dźwiękowe poruszające się ze spalinami odbijają się wielokrotnie, gdy zmienia się przekrój rury prowadzącej spaliny lub gdy jakiś element stoi na drodze spalin. Aby uzyskać pożądany efekt stosuje się kilka komór i rur o zmiennym przekroju rozmieszczonych szeregowo lub równolegle względem kierunku przepływu spalin. Wielokrotnie odbite spaliny tracą energię, a odbite fale dźwiękowe wpadają w rezonans. Zjawisko rezonansu powstaje, gdy częstotliwość drgań pokrywa się z częstotliwością drgań własnych spalin w komorze, w której się przemieszczają.
- Tłumik interferencyjny – w tego typu konstrukcji wykorzystuje się zjawisko wzajemnego wygaszania się fal dźwiękowych o tej samej częstotliwości, jeśli nakładają się na siebie i są przesunięte w fazie o połowę. Zjawisko to można osiągnąć przez rozwidlenie głównego przewodu na dwa lub trzy przewody o różnych długościach, a następnie ich ponowne połączenie. W rezultacie rozdzielony strumień płynąc drogami o różnych długościach wpadając do wspólnej komory ma różne fazy pulsacji. Niestety nie istnieje konstrukcja doskonała, która by tłumiła wszystkie dźwięki o wszystkich częstotliwościach. Stąd też, w praktyce stosuje się

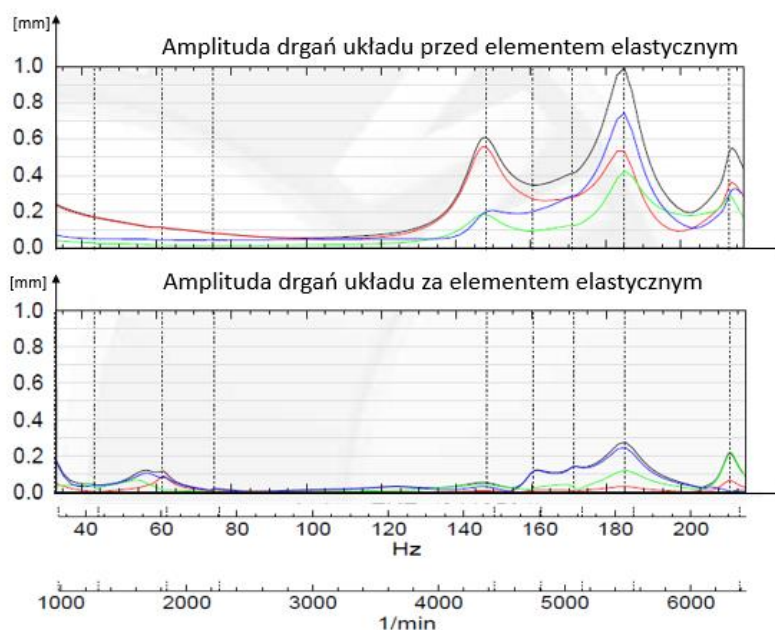
tłumiki, w których stosuje się jednocześnie różne rozwiązania konstrukcyjne a tym samym różne metody tłumienia drgań.

Silnik spalinowy zawieszony jest na elastycznych poduszkach. Silnik podczas swojej pracy wibruje i wychyla się wraz ze zmianą obciążenia. Połączenie silnika z układem wydechowym musi być elastyczne. Odpowiednio dobrane, zmniejsza przemieszczanie się układu wydechowego podczas pracy silnika, podczas zmiany jego obciążenia oraz pochłania drgania i wibracje. Jest to kolejna metoda – poza już dwoma opisanymi - ograniczenia propagacji drgań silnika. Elastyczne połączenie najczęściej umiejscowione jest pomiędzy częścią gorącą i zimną układu wydechowego. Możemy spotkać różne konstrukcje tego elementu. Najczęściej spotykanym rozwiązaniem jest elastyczny łącznik wykonany z tzw. plecionki (Rys. 2.21).



Rys. 2.21 Element elastyczny układu wydechowego [145]

Tego typu połączenie dobrze znosi moment gnący i w mniejszym stopniu siły skręcające. Sztywność tego elementu jest regulowana m.in. ilością zwojów drutu i własnościami elementu wewnętrznego ukształtowanego w postaci falującej blachy, co widać na poniższym przekroju. Prawidłowy dobór parametrów elementu elastycznego pomaga w eliminowaniu niepożądanych częstotliwości rezonansowych. (Rys. 2.22).



Rys. 2.22 Tłumienie drgań - element elastyczny [145]

Zjawisko rezonansu mechanicznego można zaobserwować, gdy na układ, który może wykonywać drgania harmoniczne z własną częstością kołową ω działa zewnętrzna siła wymuszająca F_{zewn} , zmieniająca się periodycznie w czasie:

$$F_{zewn} = F_m \sin(\omega_w t) \quad (2.11)$$

Sumę wszystkich sił działających na układ z kolei można zapisać w postaci równania różniczkowego:

$$m \frac{\partial^2 x}{\partial t^2} = -b \frac{\partial x}{\partial t} - kx + F_m \sin(\omega_w t) \quad (2.12)$$

gdzie:

x – wychylenie ciała z położenia równowagi,

m – masa ciała;

b – współczynnik oporu ośrodka,

k – współczynnik sprężystości,

F_m – amplituda siły wymuszającej;

ω_w – częstość siły wymuszającej.

Rozwiązanie równania (2.12) zależy od warunków początkowych i warunków brzegowych. Drgania harmoniczne powstają, gdy układ jest wyprowadzony z położenia równowagi i pojawia się siła wprost proporcjonalna do wychylenia, która próbuje ten stan równowagi przywrócić. Powszechnie rozważanym przykładem w literaturze [94] jest ciało zamocowane do sprężyny, która zostanie napięta i puszczona swobodnie. Ciało to wpada w drgania z charakterystyczną częstotliwością dla tego układu. Częstością drgań własnych określamy częstość drgań swobodnych, kiedy na układ nie działają siły zewnętrzne. Oczywiście w praktyce zawsze występują jakieś siły oporu, które ostatecznie powodują wytłumienie drgań.

Jeżeli siły oporu ośrodka są bardzo małe, czyli gdy b jest bliskie zero, to rozwiązaniem równania (2.12) jest funkcja:

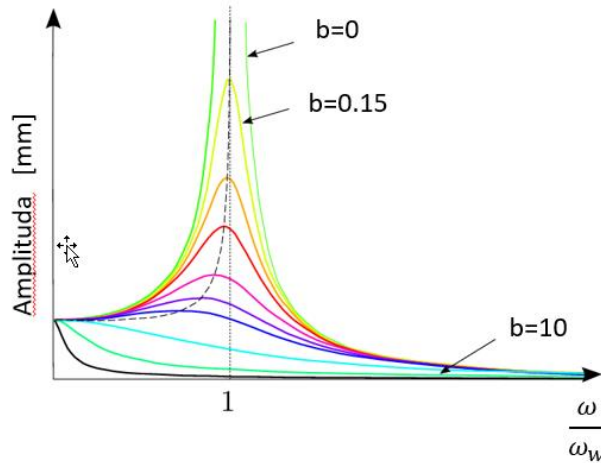
$$x = A \sin(\omega t + \alpha) + A^* \sin(\omega_w t - \varphi) \quad (2.13)$$

gdzie A i α zależą od warunków początkowych a ω jest częstością drgań swobodnych

$$\omega^2 = \frac{k}{m} \quad (2.14)$$

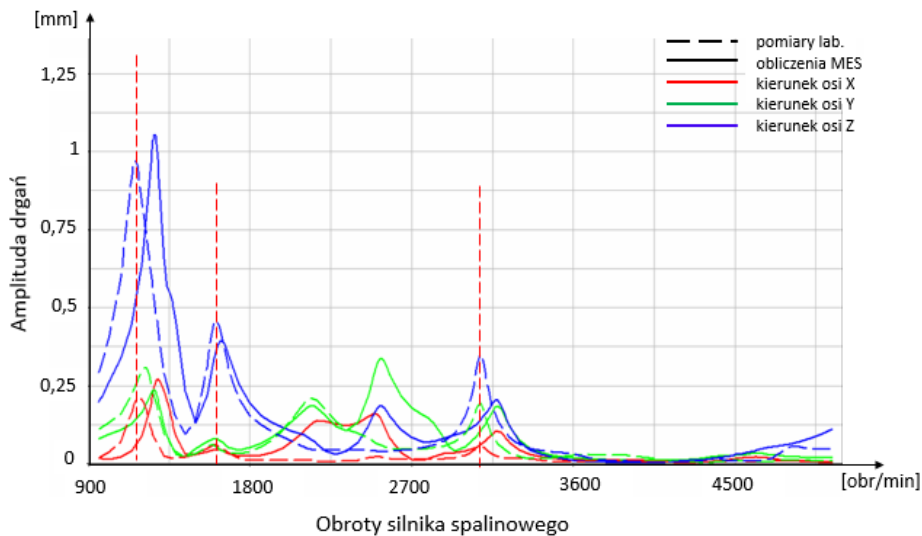
Gdy wyrażenie $(\omega_w - \omega) \rightarrow 0$, to dla $b = 0$, amplituda drgań A^* dąży do nieskończoności. Zachodzi wtedy zjawisko rezonansu, amplituda drgań wymuszonych zwiększa się w czasie do nieskończoności i układ zostaje zniszczony (Rys. 2.23).

Jest to efekt nakładających się drgań własnych z częstością ω i drgań wymuszanych przez siłę zewnętrzną z częstością ω_w . Jeżeli różnica częstości $\omega - \omega_w$ będzie nieduża w porównaniu z ω , to zaobserwujemy tzw. dudnienia (naprzemienne narastanie i zmniejszanie się amplitudy w czasie).



Rys. 2.23 Zjawisko rezonansu - wykres amplitudy względem częstotliwości

Na Rys. 2.24 pokazano zależność amplitudy drgań w widmie częstotliwości dla przykładowego tłumika środkowego układu wydechowego, który jest przymocowany do silnika pracującego pod pełnym obciążeniem. Przedstawione są krzywe dla trzech kierunków X,Y,Z. Linia czerwona odpowiada kierunkowi X, zielona kierunkowi Y, a niebieska kierunkowi Z. Krzywe przerywane są danymi pomiarowymi, a krzywe ciągłe są wynikiem symulacji i obliczeń z wykorzystaniem oprogramowania MES. Jak można zauważyć, występuje kilka częstotliwości rezonansowych (pionowa czerwona linia przerywana) w całym zakresie pracy silnika (silnik z zapłonem samoczynnym), które zostały potwierdzone w pomiarach laboratoryjnych w firmie Tenneco.



Rys. 2.24 Porównanie amplitudy drgań tłumika środkowego dla pomiarów i obliczeń [145]

Natomiast na tabeli 2.1 pokazano tzw. macierzę MAC, która pokazuje podobieństwo postaci drgań własnych układu otrzymanych na drodze pomiarów laboratoryjnych oraz obliczeń MES. Porównuje wartości częstotliwości drgań własnych, ale przede wszystkim ich postaci drgań. Wektor całkowity przemieszczenia układu wyrażony jest jako wartość od 0 do 1. Przy czym wartość równa 1, odpowiada pełnej zgodności wektorów przemieszczenia otrzymanego z pomiarów laboratoryjnych i symulacji

MES. W praktyce wartości powyżej 0.85 potwierdzają zgodność wyników. Metoda ta pozwala na wyłapanie istotnych, ważnych dla dalszego procesu walidacji układu, częstotliwości rezonansowych. Częstotliwości poboczne – zwane szumem (zakres 32-56Hz, tabela 2.1) – dla których wartości współczynnika są mniejsze od 0.85, są eliminowane z dalszej analizy inżynierskiej i nie są brane pod uwagę w procesie dalszego udoskonalania układu. Sytuację tę radykalnie zmienia istotna zmiana konstrukcyjna układu wydechowego. Wtedy proces dostrajania wyników należy przeprowadzić ponownie.

Tabela 2.1 Matryca MAC - częstotliwości drgań własnych układu - porównanie wyników symulacji MES z wynikami pomiarowymi z laboratorium [145]

		Wartości częstotliwości - obliczenia MES [Hz]							
		32	37,04	43,02	56,26	96,55	115,5	146	168
Wartości częstotliwości - pomiar laboratoryjne [Hz]	32	0,41	0,73	0,51	0,25	0,06	0,16	0,01	0,09
	36,95	0,5	0,92	0,48	0,41	0,02	0,1	0,02	0,05
	43,66	0,23	0,44	0,8	0,57	0,03	0,1	0,01	0
	53,04	0,02	0,33	0,33	0,97	0,04	0,03	0,03	0
	93,45	0,05	0,01	0,01	0,06	0,97	0,4	0,01	0,02
	115,6	0,11	0,1	0,05	0,02	0,2	0,86	0,04	0,06
	147,7	0,03	0	0,02	0,05	0,01	0,07	0,94	0,31
	168	0,06	0,04	0	0,01	0,02	0,02	0,51	0,96

Należy podkreślić, że cały zakres pracy silnika powinien być wolny od częstotliwości rezonansowych. Częstotliwości własne osłon termicznych lub podobnych do nich lekkich podzespołów układu wydechowego występują z reguły w górnym zakresie pracy silnika.

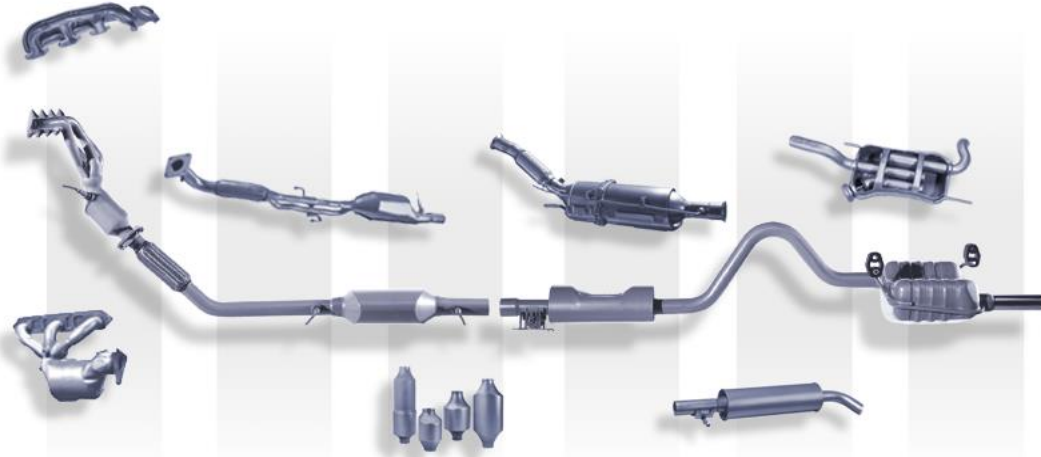
2.2.Rola i rodzaje osłon układów wydechowych

Oslony cieplne są powszechnie stosowane nie tylko w przemyśle motoryzacyjnym ale również i lotniczym. Oslony termiczne zwane także osłonami cieplnymi są w samochodzie niezbędne. Powinny chronić zwłaszcza podwozie, skrzynię biegów, części nadwozia oraz zbiornik paliwa, ale także części wrażliwe termicznie, jak sonda lambda, przewody paliwowe i hydrauliczne, przewody elektryczne itp. Podczas pracy samochodu wymienione podzespoły są narażone na wysokie temperatury, które są wytwarzane przez gorące spaliny, które mogą bez osłony cieplnej prowadzić do uszkodzenia podzespołów samochodu. Poprawną osłonę termiczną musi cechować wysoka odporność termiczna i emisyjność cieplna, która powinna skutecznie odbijać lub pochłaniać ciepło. Poprzez odbijanie lub pochłanianie ciepła osłony chronią najbliższe jej otoczenie przed niepożądaną temperaturą. Pochłanianie ciepła jest emitowane jednocześnie na zewnątrz przez promieniowanie cieplne. W samochodzie napędzanym silnikiem spalinowym układ wydechowy od kolektora silnika do rury wydechowej jest największym źródłem ciepła poza samym silnikiem. Powierzchnie części, odprowadzające spaliny, mogą osiągać temperatury 900°C, a nawet 1000°C. Ponieważ układ wydechowy często przechodzi w pobliżu wielu wrażliwych na temperaturę podzespołów, ważnym jest zabezpieczenie tych części i modułów przed przegrzaniem, oraz zapobieganie emisji emitowanego ciepła do wnętrza pojazdu. Główne stosujemy osłony do ochrony następujących podzespołów układu wydechowego:

- Osłony termiczne kolektora wydechowego
- Osłony termiczne katalizatora DOC, DPF, SCR, LNT
- Osłony termiczne turbosprężarek

- Osłony termiczne tłumików, rur

Przegląd wybranych elementów układu wydechowego, na których montuje się osłony, a zarazem będące źródłem emisji ciepła, przedstawia Rys. 2.25.



Rys. 2.25 Podzespoły typowego układu wydechowego [145]

Rola osłon na przestrzeni ostatnich lat znacznie wzrosła. Oprócz roli chroniącej elementy otaczające układ, dodatkowo utrzymują odpowiednią temperaturę spalin, by pewne procesy chemiczne mogły zaistnieć. Związane jest to przede wszystkim z eliminacją zanieczyszczeń takich jak tlenki azotu, wodorotlenki itp. Prawidłowo zaprojektowane osłony muszą spełniać wiele kryteriów związanych z ich charakterystykami wytrzymałościowymi, cieplnymi, akustycznymi, itp. Osłony te, aby spełnić swoją rolę muszą odznaczać się:

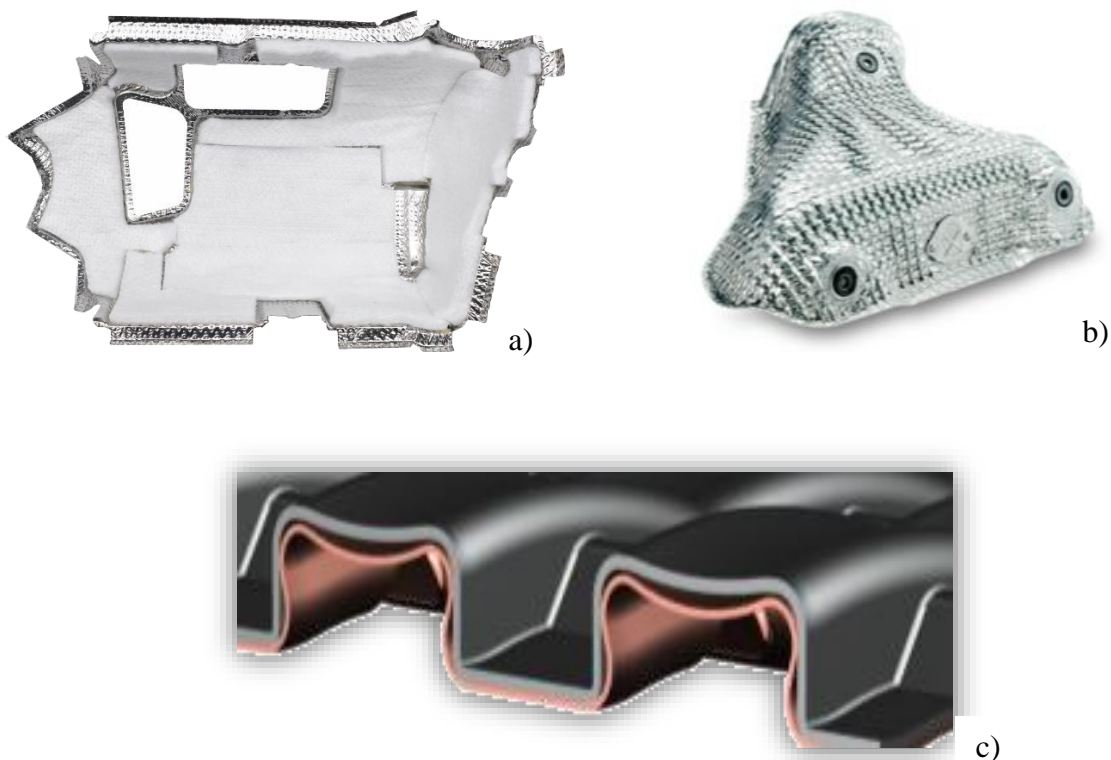
- niską przewodnością cieplną (wysoka odporność termiczna),
- wysoką emisyjnością
- dobrą stabilnością termiczną (ogniotrwałość).

Występują one w wielu konfiguracjach i mogą przyjmować różne skomplikowane kształty. Rozróżniamy m.in.:

- osłony jednowarstwowe nieizolowane lub czasem wyklejane matami odpornymi na wysokie temperatury od wewnątrz, stosowane dla ochrony elementów trudno dostępnych i o nieregularnych kształtach (Rys. 2.26)
- osłony wielowarstwowe przykręcane do podzespołu osłanianego (Rys. 2.26b)
- osłony dwuwarstwowe, składające się z dwóch arkuszy blachy o różnych grubościach każdy, ze szczeliną powietrzną lub materiałem izolacyjnym pomiędzy nimi. (Rys. 2.26c)
- osłony powłokowe, jednowarstwowe o grubości 0.25 -2.5mm (Rys. 2.27)

Podstawową zaletą osłon jednowarstwowych powłokowych jest ich łatwość formowania oraz niski koszt produkcji. Wadą jest to, że nie redukują skutecznie hałasu oraz mają tendencję do wibracji. Są także najmniej skuteczne jeżeli chodzi o izolację cieplną. Dlatego też, niektóre odmiany stosowane dzisiaj, są wyklejane odpowiednimi matami odpornymi na wysokie temperatury od wewnątrz, polepszając tym samym znacznie ich skuteczność cieplną i zdolność do tłumienia drgań porównywalną z osłonami dwuwarstwowymi. Osłony dwuwarstwowe natomiast, składają się z dwóch arkuszy blachy

o różnych grubościach każda, ze szczeliną powietrzną lub materiałem izolacyjnym pomiędzy nimi (Rys. 2.26c).

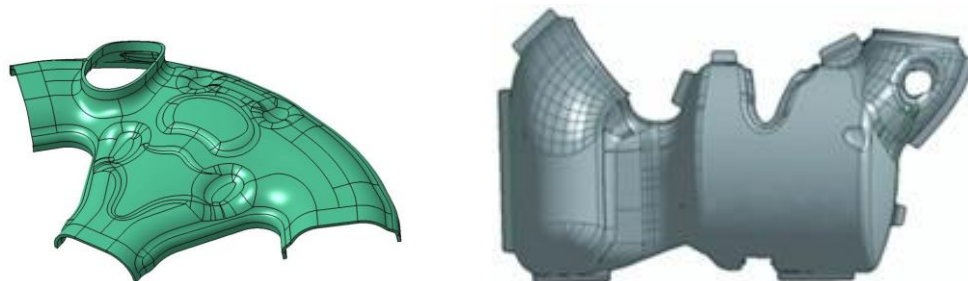


Rys. 2.26 Rodzaje osłon układu wydechowego [145]

W praktyce osłony tego typu posiadają przetłoczenia w celu zwiększenia sztywności samej osłony oraz redukcji maksymalnych wartości przemieszczeń. Ponadto ich struktura poprawia własności termiczne, akustyczne i wibracyjne. Większa grubość blachy natomiast skutkuje lepszym tłumieniem i pochłanianiem energii akustycznej.

Osobną grupę osłon, stanowią osłony dla samochodów terenowych, które są wyposażone w tzw. „osłony przeciwpożarowe”, które mocuje się pod układem wydechowym. Te osłony termiczne pomagają uniknąć zapalenia się suchej trawy i zarośli podczas jazdy po otwartych terenach leśnych czy łące.

Do osłon jednowarstwowych o w miarę prostym kształcie stosuje się blachy żaroodporne oraz z podwyższoną odpornością na korozję. Są to zwykle arkusze blachy, której grubość waha się od 0.25mm do nawet 2.5mm. Arkusze mogą również zawierać wytłoczenia dla zwiększenia sztywności (Rys. 2.27).

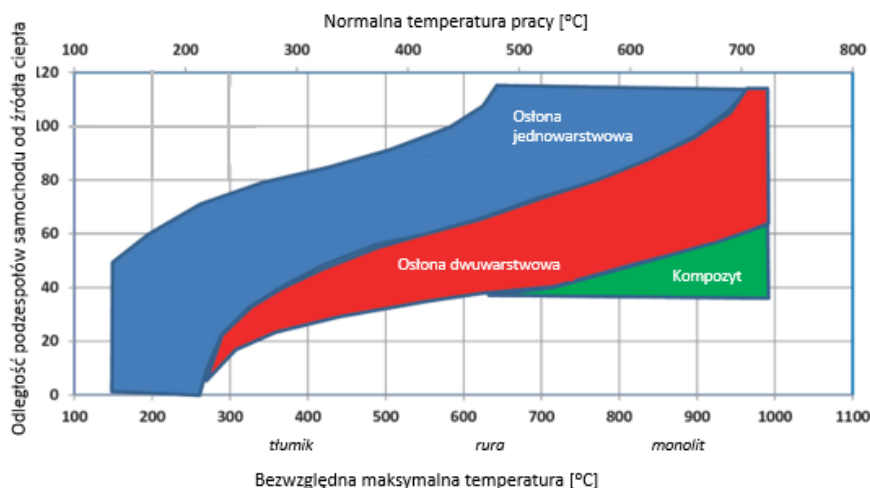


Rys. 2.27 Rodzaje osłon jednowarstwowych układu wydechowego [145]

Często materiałem stosowanym do ich budowy było aluminium. Aktualnie wykonywane są z blachy aluminiowej lub austenitycznej, rzadziej z materiałów kompozytowych. Dzięki wysokiej odporności cieplnej nie ulegają one uszkodzeniu nawet w wysokich temperaturach.

2.3. Wymagania stawiane osłonom układów wydechowych

Obecnie podstawowym materiałem stosowanym na osłony termiczne są stopy aluminium lub cienkie blachy austenityczne. Aluminium posiada wysoki współczynnik refleksji i niską emisyjność powierzchni, nawet po pokryciu naturalną warstwą tlenku glinu. Wysoka przewodność cieplna aluminium zapewnia szybkie odprowadzanie ciepła z potencjalnych gorących punktów w osłonie termicznej. Aluminium ma również wysoką właściwą pojemność cieplną. Co oznacza, że wzrost temperatury po pochłonięciu określonej ilości energii cieplnej jest mniejszy niż w przypadku wielu innych materiałów. Wykazuje się doskonałą odpornością na korozję, dobrą plastycznością na zimno, odpowiednią wytrzymałością na rozciąganie oraz doskonałe pochłanianie energii. Wszystko to sprawia, że jest to materiał preferowany do produkcji osłon termicznych zarówno pod względem łatwości wykonania, jak i przeznaczenia. Oprócz podstawowego EN AW-1050A stosuje się również stopy o podwyższonej wytrzymałości, EN AW-3003, EN AW-5052 i EN AW-5182. Stopy o wyższej wytrzymałości pozwalają na konstruowanie osłon termicznych, które mimo mniejszej grubości są bardzo odporne na uszkodzenia. Stopy te jednak zmniejszają plastyczność osłon na zimno, co ogranicza ich zastosowanie w wielu przypadkach, ze względu na złożoną geometrię osłon. Alternatywą dla stopów aluminium o podwyższonej wytrzymałości jest cienka blacha austenityczna. Jej cienka grubość pozwala zazwyczaj znacznie łatwiej dopasować się do kształtu osłanianego elementu. Stosuje się je do ochrony przed najwyższymi temperaturami oraz w przypadku ograniczonego miejsca podczas ich montażu. Najważniejszym atutem folii austenitycznej jest jej dobra formowalność, aby sprostać ekstremalnym wymaganiom. Osłony termiczne muszą jak najszczelniej zakrywać gorący element. Niewielka przestrzeń, w szczególności w komorze silnika i pod podwoziem sprawia, że osłony termiczne mają zwykle dość skomplikowane kształty, by ściśle przylegać do powierzchni elementów chronionych. Zarządzanie ciepłem i odpowiednia izolacja są niezwykle ważne i bardzo wymagające dla funkcjonowania pojazdów. Parametry konstrukcyjne osłony cieplnej, takie jak rodzaj materiału izolacyjnego, skład materiału izolacyjnego, grubość izolacji, gęstość izolacji, grubość szczeliny powietrznej i materiał warstwy zewnętrznej mają wpływ na temperaturę powierzchni osłony. Rys. 2.28 przedstawia typy osłon w odniesieniu do temperatury i odległości podzespołu osłanianego od źródła ciepła.



Rys. 2.28 Rodzaj osłony vs temperatura [162]

Dodatkowo możemy zauważyć, na jakie temperatury są narażone najczęściej wybrane komponenty układu.

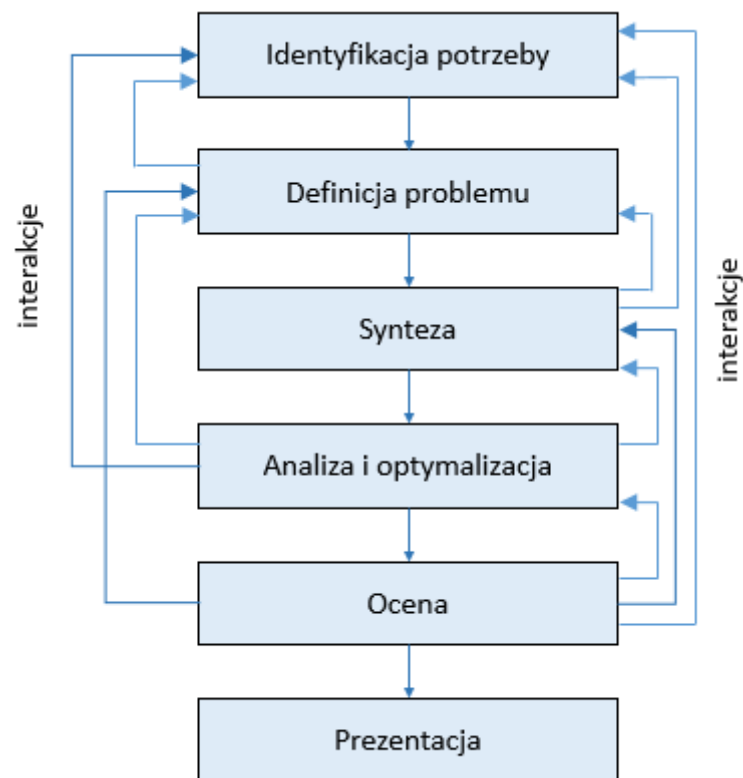
Kolejnym ważnym wymogiem stawianym osłonom jest wysoka odporność na korozję. Wilgotne środowisko pracy, podwyższone temperatury oraz dodatkowe czynniki środowiskowe, silnie sprzyjają efektom korozji. Duża część osłon termicznych jest bezpośrednio narażona na zanieczyszczenia z podłoża. Zanieczyszczenia takie jak sól, błoto czy zwykła woda przyspieszają w całym okresie użytkowania samochodu zjawisko korozji osłony termicznej. Bardzo często aluminiowe arkusze osłony są przymocowane bezpośrednio do elementów stalowych, co skutkuje powstawaniem warunków do rozwoju innego rodzaju korozji - korozji galwanicznej. Wymaga ona szczególnej uwagi na etapie dobierania materiału na osłony. Ponadto materiał osłony termicznej musi zapewniać odpowiednią odporność na pękanie. Uderzenia kamieni lub mocniejszy kontakt z podłożem mogą spowodować uszkodzenie osłon termicznych. Stąd odpowiedni materiał powinien zapobiegać powstawaniu pęknięć.

Jednym z ważniejszych aspektów dotyczących projektowania osłon, jest ich optymalny kształt niwelujący prawdopodobieństwo powstania zjawiska rezonansu. Odpowiednio zaprojektowany i zoptymalizowany, pozwala uniknąć problemów w fazie testowania i późniejszej eksploatacji układu. Powierzchnia elementu, jego grubość, a także grubość materiału izolacyjnego oraz liczba i położenie połączeń śrubowych, ma istotny wpływ na powstawanie lub przenoszenie drgań.

3. Metody analizy optymalnego projektowania rozpatrywanego zagadnienia

3.1. Wstęp

Celem projektowania jest tworzenie nowych obiektów technicznych motywowanych konkretnymi potrzebami i ograniczonymi dostępnymi środkami ich osiągnięcia [53]. Proces projektowania rozpoczyna się od potrzeby, wymagania lub pomysłu, a kończy informacją o sposobie wytworzenia i użytkowania produktu, na przykład w postaci zestawu rysunków lub reprezentacji komputerowej [43]. Projektowanie to innowacyjny, wysoce iteracyjny proces składający się z wielu interaktywnych faz [35]. Proces podejmowania decyzji jest ważną fazą projektowania. Etapy związane z procesem projektowania są często ilustrowane w sposób pokazany na Rys. 3.1.



Rys. 3.1 Schemat ogólnego procesu projektowania

Strzałki na Rys. 3.1 oznaczają iterację pomiędzy poszczególnymi fazami procesu. Proces rozpoczyna się od identyfikacji potrzeby. Po identyfikacji następuje definicja problemu, podczas której określone są wielkości wejściowe i wyjściowe, cechy i ograniczenia pożądanego produktu. Synteza natomiast, czasami nazywana także wynalezieniem koncepcji lub projektem koncepcyjnym, w którym są proponowane, badane, oceniane i ulepszone pewne rozwiązania danego problemu. Na etapie analizy i optymalizacji konieczne jest zbudowanie abstrakcyjnych modeli układu, do których można zastosować narzędzia matematyczne. Model matematyczny lub numeryczny ma w zadowalający sposób symulować układ mechaniczny. Model taki służy do ilościowego określenia, czy zidentyfikowane wcześniej wymagania zostały spełnione do dalszego ich doskonalenia w procesie optymalizacji. Najogólniej model matematyczny to zbiór zmiennych opisujących rozpatrywany obiekt. W modelu optymalizacyjnym ten zbiór zmiennych składa się ze stałych parametrów i zestawu zmiennych projektowych. Parametry stałe podczas procesu optymalizacji mogą obejmować na przykład własności fizyczne (np. moduł Younga lub gęstość) systemu, a zmienne projektowe (np. współrzędne czy średnica

otworu) ulegają zmianie podczas procesu optymalizacji w celu ulepszenia projektu w kontekście konkretnej potrzeby. Ogólnie rzecz biorąc, na zmienne projektowe można nałożyć ograniczenia, aby odzwierciedlić rzeczywiste ograniczenia systemu. Aby ocenić jakość rozwiązania, należy ustalić funkcję celu (zwaną także funkcją przystosowania) jako funkcję zmiennych projektowych. Funkcja celu ma reprezentować konkretne, rzeczywiste potrzeby stawiane produktowi, takie jak na przykład niski koszt, mała masa czy wysoka sztywność. Proces ewaluacji powinien odpowiedzieć na wiele pytań dotyczących realizacji potrzeb, stanu produktu, jego wytwarzania, opłacalności i konkurencyjności na rynku. Ostatnim etapem procesu jest prezentacja, podczas której nowe rozwiązanie zostaje przedstawione szerszej publiczności. Należy je odpowiednio przedstawić, opisać, biorąc pod uwagę zróżnicowany poziom wiedzy odbiorców na dany temat.

Większość inżynierów rozwiązujących różne problemy projektowe dysponuje szeroką gamą narzędzi usprawniających ten proces. Popularne oprogramowanie wspomagania komputerowego CAx (CAD/CAE/CAM) zapewnia narzędzia ułatwiające projektowanie, analizę i symulację systemów mechanicznych. Obecne oprogramowanie CAE często zawiera moduły optymalizacyjne, a w wielu przypadkach oferuje narzędzia do optymalizacji jednokryterialnej i wielokryterialnej.

Odpowiednią konstrukcję osłony termicznej można otrzymać, przeprowadzając analizę optymalizacyjną w różnych obszarach, takich jak: optymalizacja wymiarów (ang. *sizing optimization*), optymalizacja kształtu (ang. *shape optimization*), optymalizacja warunków brzegowych (ang. *boundary condition optimization*) [38,47,99], optymalizacja materiału konstrukcyjnego (ang. *material optimization*), a nawet w pewnych przypadkach możliwa jest również optymalizacja topologiczna (ang. *topology optimization*). W ciągu ostatnich 20 lat nastąpił intensywny rozwój narzędzi komputerowych umożliwiających przeprowadzenie tego rodzaju rodzajów zadań. Nowoczesne oprogramowanie CAx bardzo często zawiera zintegrowane moduły umożliwiające rozwiązywanie tego typu problemów. Projektowanie przy wykorzystaniu metod optymalizacji topologicznej nazywane jest również projektowaniem generatywnym (ang. *Generative Design*). Optymalizacja topologii kontinuum materialnego dokonywana jest wewnątrz ściśle określonego obszaru projektowego, gdzie podczas procesu optymalizacji tworzą się podobszary wypełnione materiałem i podobszary pozbawione materiału. Zwykle celem optymalizacji topologicznej jest maksymalizacja sztywności, a zatem minimalizacji podatności układu, która jest wyrażona jako praca sił i obciążenia zewnętrznego

Optymalizację topologiczną rozważano m.in. w pracach nad wybranymi elementami zawieszenia [68, 91] czy elementami zespołu napędowego [21, 108]. Oprócz optymalizacji jednokryterialnej [89] stosowano również metody optymalizacji wielokryterialnej [59]. Stosunkowo niewiele prac poświęcono zagadnieniu optymalnego zaprojektowania kształtu osłon tłumika układu wydechowego [2, 81, 98], co sprawia, że podjęcie tematu wydaje się zasadne, zwłaszcza jeśli dotyczy zwiększenia możliwości rozwiązywania tego typu problemów z wykorzystaniem nie tylko algorytmów wbudowanych w systemy CAE.

W przypadku osłon termicznych montowanych w układach wydechowych najbardziej uzasadniony wydaje się dobór wybranych cech geometrycznych oraz sposób montażu tego typu układów. Sposób zamocowania osłon, realizowany zwykle za pomocą odpowiednich otworów montażowych, związany jest z optymalnym projektowaniem sposobu podparcia lub doboru wartości siły wywieranej przez połączenie śrubowe. Taka optymalizacja jest wtedy optymalizacją warunków brzegowych. Dla innych układów mechanicznych optymalizacja warunków brzegowych może być rozumiana w szerszym kontekście i dotyczyć może np. lokalizacji różnych obciążeń mechanicznych lub termicznych, ich wartości lub sposobu rozłożenia w przestrzeni. Przykłady tego typu zadań znaleźć można m.in. w [37].

Natomiast optymalizacja kształtu osłony termicznej dotyczyć może w tym przypadku poszukiwania optymalnego kształtu geometrii, ewentualnie dodatkowych przetłoczeń lub ich trajektorii. Dla modelu osłony, który jest przedmiotem dalszej analizy, trajektoria przetłoczeń oparta jest na krzywych typu *B-spline*, które to są wynikiem składania krzywych Béziera. Najczęściej kształt krzywej jest

reprezentowany przez wielomiany trzeciego stopnia, ponieważ wielomiany niższego stopnia są już zbyt mało elastyczne. Krzywa określona jest przez dwa punkty końcowe oraz dwa punkty pośrednie nie należące do krzywej.

Typowa krzywa Beziera opisywana jest za pomocą równania :

$$C(u) = \sum_{i=0}^n p_i B_i^n(u) \quad \text{dla} \quad u \in [0,1] \quad (3.1)$$

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-1} \quad (3.2)$$

gdzie:

n – liczba punktów kontrolnych

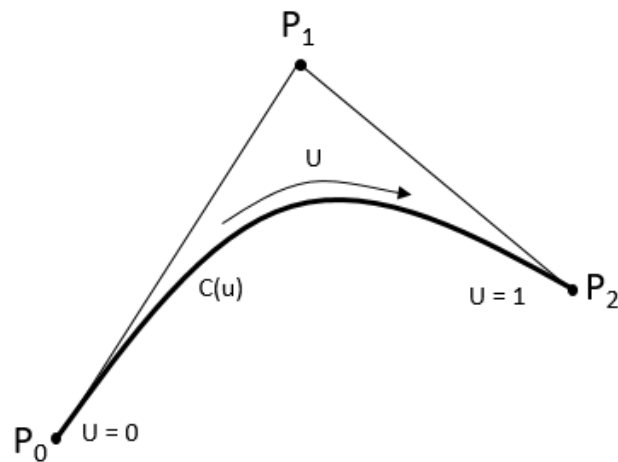
$B_i^n(u)$ – wielomiany bazowe Bernsteina

p – punkt kontrolny krzywej Beziera

W praktyce wykorzystuje się krzywe wielomianowe niskich stopni, opisywane niewielką liczbą punktów kontrolnych. Najczęściej stosowane są krzywe drugiego stopnia (Rys. 3.2) lub trzeciego (odpowiednio trzy i cztery punkty kontrolne). Znacznie rzadziej stosuje się krzywe wyższych stopni.

Krzywe drugiego stopnia definiujemy za pomocą 3 wierzchołków tzw. punktów kontrolnych P_0 , P_1 , P_2 , zgodnie ze wzorem:

$$C(u) = P_0(1-u)^2 + P_1 2u(1-u) + P_2 u^2 \quad (3.3)$$



Rys. 3.2 Krzywa wielomianowa w reprezentacji Beziera

Niezależnie od rodzaju krzywej, jej przebieg definiuje zawsze łamana, określona za pomocą punktów kontrolnych, których liczba jest zwykle niewielka. Kształt krzywej opisują wielomiany, dla których przyjęto dziedzinę znormalizowaną $[0,1]$. Stopień wielomianu krzywej zależy wprost od liczby punktów kontrolnych i wynosi $n-1$. Zalety reprezentacji Béziera widoczne są w momencie przesuwania wierzchołków kontrolnych. Można wówczas w elastyczny sposób modyfikować krzywą zachowując kontrolę nad jej kształtem.

Krzywe wielomianowe drugiego stopnia nie mogą jednak służyć do reprezentowania dowolnych krzywych stożkowych, ponieważ zawsze są parabolami. Uniwersalność krzywych Beziera pozwala rozwiązać ten problem stosując krzywe opisywane za pomocą punktów kontrolnych i związanymi z

nimi współczynnikami wagowymi. Krzywe typu NURBS (*Non Uniform Rational B-Spline*), dla których położenie dowolnego punktu na krzywej oblicza się zgodnie ze wzorem:

$$C(u) = \frac{\sum_{i=0}^n w_i p_i N_i^n(u)}{\sum_{i=0}^n w_i N_i^n(u)} \quad u \in [0,1], \quad (3.4)$$

gdzie:

n - liczba punktów kontrolnych minus 1 (punkty kontrolne liczone są od zera: p_0, \dots, p_n)

p_i - i -ty punkt kontrolny,

w_i - waga i -ego punktu kontrolnego (dowolna liczba rzeczywista),

jeśli $w = 0$ punkt kontrolny nie jest brany pod uwagę

$N_i^n(u)$ - bazowe funkcje typu B-spline (wzór 3.6, 3.7)

Krzywe typu NURBS dają nam bardzo subtelne narzędzie do modyfikacji kształtu krzywej bez zmiany położenia jej punktów kontrolnych. Zmiana ta odbywa się na podstawie modyfikacji wag poszczególnych punktów kontrolnych. Zwiększając wagę wybranego punktu, powodujemy, że krzywa zbliża się do tego punktu.

Jednakże krzywe wielomianowe, posiadają jedną wspólną wadę – bardzo trudno za pomocą tylko jednej krzywej opisać skomplikowane kształty. Można co prawda edytować krzywą, dodając nowy punkt kontrolny, ale skutkuje to tym, że kontrola kształtu krzywej jest trudniejsza. Spowodowane to jest tym, że przemieszczenie tylko jednego punktu wpływa na kształt całej krzywej. Ponadto im wyższy stopień, tym zmiana położenia punktów kontrolnych jest mniej widoczna. Fakt ten przeważał, że bardziej powszechne w użytku stały się krzywe B-sklejane (*B-spline*). Są to takie krzywe, które składają się z fragmentów krzywych bądź to wielomianowych, bądź wymiernych (względnie niskiego stopnia), natomiast matematyczne równania opisujące taką krzywą gwarantują, iż w punktach połączenia różnych fragmentów krzywa będzie gładka. Krzywą B-sklejaną można modelować i edytować za pomocą punktów kontrolnych na podobnych zasadach, jak krzywe Béziera. W wyniku przemieszczenia wybranego punktu kontrolnego zmianie ulega tylko jego najbliższe otoczenie. Ich budowa pozwala na bardziej precyzyjną lokalną kontrolę kształtu krzywej.

Dla krzywej B-sklejanej charakterystyczne są dwa parametry:

n - stopień sklejaných krzywych wielomianowych

m - liczba podprzedziałów, na których definiowane są kolejne części krzywej

Dla krzywej B-spline dowolny punkt opisujący krzywą obliczany jest wg wzoru, wynikającego z algorytmu de Boor'a:

$$C(t) = \sum_{i=0}^{m-n-1} p_i N_i^n(t) \quad \text{dla} \quad t \in [u_n, u_{m-n}], \quad (3.5)$$

gdzie:

$m+1$ - liczba węzłów,

n - stopień krzywej,

p_i - punkty kontrolne,

$N_i^n(t)$ - unormowana funkcja B-spline stopnia n

Unormowana funkcja B-spline jest przedstawiana za pomocą rekurencyjnego wzoru Mansfielda-de-Boora-Coxa, będącego podstawą algorytmu de Boora:

$$N_i^0(t) = \begin{cases} 1 & \text{dla } t \in [u_i, u_{i+1}) \\ 0 & \text{w przeciwnym razie} \end{cases} \quad (3.6)$$

$$N_i^n(t) = \frac{t-u_i}{u_{i+n}-u_i} N_i^{n-1}(t) + \frac{u_{i+n+1}-t}{u_{i+n+1}-u_{i+1}} N_{i+1}^{n-1}(t) \quad (3.7)$$

dla $n > 0$

3.2. Optymalizacja

Celem optymalizacji jest wybór najlepszego (minimalnego lub maksymalnego) rozwiązania danego problemu, które spełnia wszystkie ograniczenia. Na co dzień rozwiązania optymalne zadań praktycznych są wynikiem zdobytego doświadczenia, posiadanej wiedzy lub są skutkiem metody prób i błędów. Proces ten jest niestety długotrwały. Metody optymalizacyjne pozwalają na znalezienie bardziej efektywnego sposobu poszukiwanego ‘optimum’ w sposób algorytmiczny. Pozwalają na poszukiwanie rozwiązań optymalnych na drodze analitycznej lub numerycznej, bez konieczności analizowania wszystkich możliwych wariantów.

Zagadnienie poszukiwania rozwiązań optymalnych możemy zaliczyć do dziedziny matematyki stosowanej, której zadaniem jest w sposób matematyczny sformułowanie analizowanego problemu optymalizacji, zbioru poszukiwań oraz funkcji celu optymalizacji (kryterium jakości) [95]. Przy tak zdefiniowanym problemie optymalizacji, zadanie polega na znalezieniu optymalnego rozwiązania, które będzie spełniać założone kryterium optymalizacji. Proces znalezienia rozwiązań optymalnych można przedstawić w sposób ogólny i sprowadzić do następujących zadań:

1. Opracowanie modelu matematycznego analizowanego procesu
2. Zdefiniowanie funkcji celu optymalizacji (funkcjonału/kryterium jakości)
3. Poszukiwanie optymalnego rozwiązania przy pomocy jednej z metod optymalizacji

Założmy, że są znane:

funkcja celu $f: \mathbb{R}^n \rightarrow \mathbb{R}$,
 zbiór $X_d \subset \mathbb{R}^n$

gdzie: \mathbb{R}^n n -wymiarowa przestrzeń wektorów rzeczywistych

Zadaniem optymalizacji jest poszukiwanie takiego elementu $x^{opt} \in X_d$, który spełnia wszystkie zdefiniowane ograniczenia, i tak w przypadku poszukiwania:

- minimum ($x^{min} = x^{opt}$)

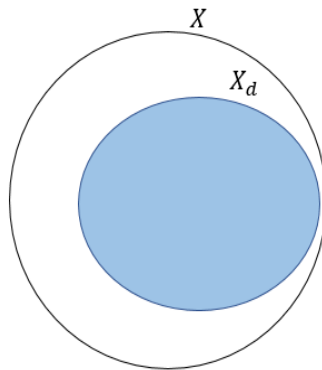
$$f(x^{opt}) \leq f(x), \quad \forall x \in X_d \quad (3.8)$$

- maksimum ($x^{max} = x^{opt}$)

$$f(x^{opt}) \geq f(x), \quad \forall x \in X_d \quad (3.9)$$

W praktyce algorytmy optymalizacyjne znajdują pewne rozwiązanie x^* , które powinno stanowić przybliżenie minimum/maksimum globalnego optymalizowanej funkcji celu. Innymi słowy wartość funkcji celu $f(x^*)$ powinna być dobrym przybliżeniem liczby $f(x^{opt})$. Znalezione minimum lub maksimum jest globalne jednak tylko wewnątrz zdefiniowanego obszaru poszukiwań zmiennych optymalizacji X_d , w którym znajdują się dopuszczalne rozwiązania. Jak pokazano na Rys. 3.3 jeżeli w przestrzeni zmiennych optymalizacji X zdefiniujemy zbiór rozwiązań dopuszczalnych X_d , a jednocześnie przyjmemy, że $X = \mathbb{R}^n$, gdzie \mathbb{R}^n oznacza n -wymiarową przestrzeń wektorów

rzeczywistych ($n \in \mathbb{N}$), to problem optymalizacji określamy jako statyczny lub skończenie wymiarowy [95]. Taki też model został przyjęty w tej pracy.



Rys. 3.3 Zbiór rozwiązań dopuszczalnych X_d w przestrzeni wszystkich rozwiązań X .

Zbiór rozwiązań dopuszczalnych $X_d \subset X$ często definiujemy również poprzez podanie pewnych warunków (ograniczeń), które musi spełniać wektor $\mathbf{x} \in X$, aby należeć do zbioru X_d . Przykładowo:

$$g_i(\mathbf{x}) < 0 \text{ dla } i = 1, \dots, k \quad (3.10)$$

$$h_j(\mathbf{x}) = 0 \text{ dla } j = 1, \dots, m$$

W przypadku kiedy nie zdefiniowano żadnych ograniczeń, to przyjmujemy, że $X_d = X$. W przypadku gdy $X_d = X = \mathbb{R}^n$, to mówimy o optymalizacji bez ograniczeń. W przeciwnym razie, tzn. gdy $X \subseteq \mathbb{R}^n$ mamy do czynienia z optymalizacją z ograniczeniami.

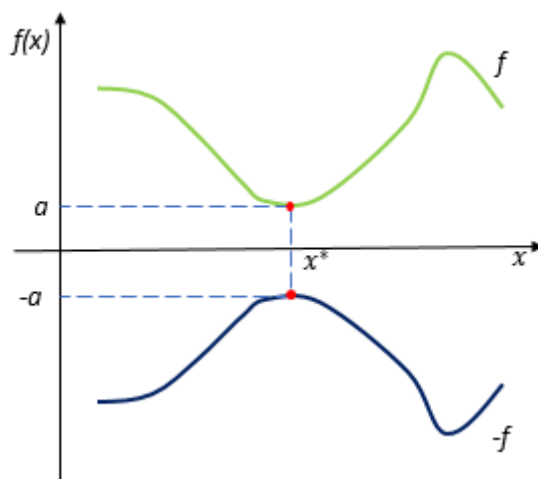
W zagadnieniach optymalizacji wprowadza się pojęcie funkcji celu (ang. *fitness function*), która często nazywana jest także w literaturze kryterium optymalizacji, kryterium jakości lub funkcjonalem jakości. Z definicji wynika, że funkcja celu

$$f: X \rightarrow \mathbb{R}$$

przyporządkowuje każdemu rozwiązaniu $\mathbf{x} \in X$ pewną wartość liczbową $f(\mathbf{x}) \in \mathbb{R}$.

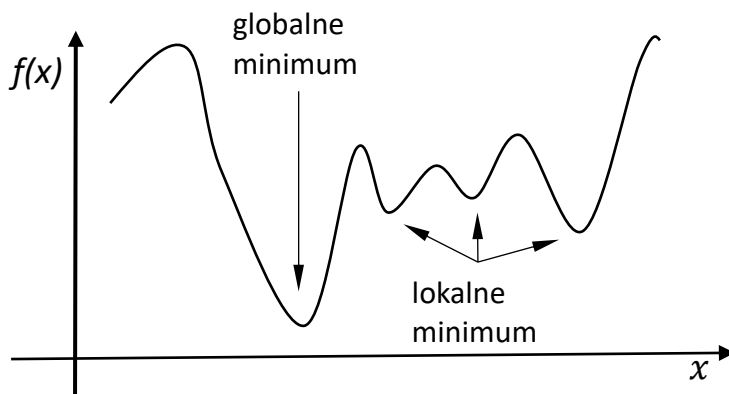
Pozwala to na jej przyporządkowanie poszczególnym rozwiązaniom \mathbf{x} ze zbioru rozwiązań dopuszczalnych $X_d \subset X$, co umożliwia porównanie rozwiązań, a co za tym idzie, daje możliwość wyznaczenia takiego rozwiązania dopuszczalnego, które daje minimalną bądź maksymalną wartość funkcji celu.

Na uwagę zasługuje fakt, że pojęcie *rozwiązanie minimalne* i *rozwiązanie maksymalne* są do pewnego stopnia wymienne. Poszukiwanie minimum funkcji celu może być albowiem zastąpione zadaniem poszukiwania maksimum. Jeżeli wyznaczymy np. rozwiązanie minimalne x^* funkcji f , to będzie ono jednocześnie rozwiązaniem maksymalnym dla funkcji $-f$ i odwrotnie (Rys. 3.4). Innymi słowy, kiedy znamy algorytm wyznaczający minimum dowolnej funkcji celu, tym samym znamy również od razu sposób na wyznaczenie maksimum tej funkcji.



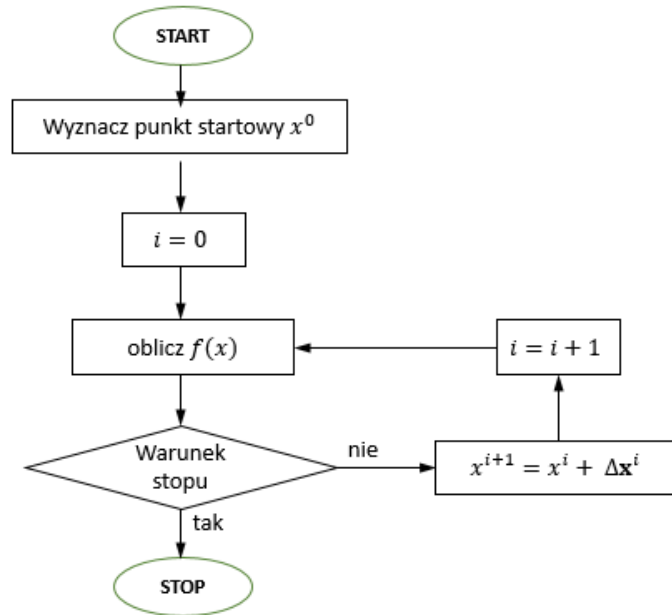
Rys. 3.4 Minimum i maksimum funkcji celu jako pojęcia wymienne

Często zdarza się, że w analizowanym obszarze poszukiwań, występują zarówno ekstrema globalne jak i lokalne. Dla ekstremum lokalnego (minimum/maksimum) wartość optymalizowanej funkcji przybiera gorszą wartość niż dla rozwiązania globalnego. Ekstrema lokalne skutecznie zakłócają proces poszukiwania tego jednego, właściwego minimum/maksimum globalnego. Wynika to z faktu, że metody optymalizacji mają tendencję do ‘grzeźnięcia’ w minimum/maksimum lokalnym i mają problem by dotrzeć do rozwiązania globalnego. Dlatego też, zdolność do docierania do globalnego minimum lub maksimum, jest jednym z kryteriów oceny jakości metod optymalizacji. Większość metod optymalizacji, znajduje przeważnie tylko minima lokalne albo nie znajdują niczego. Algorytm kończy swoje działania w punktach, które nawet nie są minimami lokalnymi. Na Rys. 3.5 pokazano przykładowy przebieg funkcji jednej zmiennej, która posiada wiele minimów lokalnych, z których tylko jedno jest globalne.



Rys. 3.5 Wykres funkcji jednej zmiennej z wieloma minimami lokalnymi.

Większość metod optymalizacji poszukuje rozwiązania optymalnego opierając się na ogólnej, iteracyjnej strategii poszukiwania. A mianowicie dla dowolnie wybranego punktu startowego x^0 należącego do zbioru rozwiązań dopuszczalnych, obliczana jest wartość początkowa funkcji celu. Jeżeli dla tej wartości jest spełniony przyjęty warunek stopu optymalizacji, to procedura optymalizacyjna kończy działanie. Jeżeli się jednak okaże, że warunek stopu nie został spełniony, wtedy obliczana jest wartość funkcji w nowym punkcie $x^{i+1} = x^i + \Delta x^i$ i cała procedura jest powtarzana. Ogólny schemat strategii poszukiwania rozwiązania optymalnego pokazano na Rys. 3.6.



Rys. 3.6 Ogólny schemat blokowy strategii optymalizacji funkcji celu.

Istnieje wiele metod optymalizacyjnych, ale każda indywidualnie definiuje algorytm określania wartości wektora $\Delta \mathbf{x}$, jednakże w większości metod jest on uzależniony od wyznaczenia kierunku poszukiwań d oraz długości kroku h , jaki należy wykonać wzdłuż kierunku poszukiwań d . Można to zapisać w następujący sposób $\Delta \mathbf{x} = h\mathbf{d}$. W ostateczności, w każdej kolejnej iteracji i nowy punkt poszukiwań rozwiązania optymalnego x^{i+1} można przestawić w ogólnej postaci:

$$x^{i+1} = x^i + \Delta \mathbf{x}^i = x^i + h^i \mathbf{d}^i \quad (3.11)$$

Znajomość kierunku poszukiwań np. minimum funkcji celu sprowadza problem znalezienia minimum, do wyznaczenia minimum jednej zmiennej, wzdłuż prostej zgodnej z kierunkiem poszukiwań. Do wyznaczenia tego minimum wykorzystuje się jedną z metod optymalizacji jednowymiarowej (metod poszukiwania minimum wzdłuż kierunku) opisanych w rozdziale 3.4. Cała procedura poszukiwania minimum funkcji celu kończy się po spełnieniu warunku stopu. W praktyce bardzo trudno jest uzyskać rzeczywiste x^{min} funkcji celu. Stosując różne metody optymalizacyjne i wykorzystując różne algorytmy z reguły jesteśmy w stanie wyznaczyć jedynie jakąś wartość x^* bardzo bliską wartości optymalnej, ale nie jest to rzeczywiste minimum. Jak duże odchyłki od wartości rzeczywistego minimum dopuszczamy, możemy zdefiniować za pomocą warunku stopu. Stosowane są różne, a do najczęściej wykorzystywanych możemy zaliczyć tzw. testy zbieżności.

Jednym z nich jest warunek wyrażony za pomocą normy euklidesowej:

$$\|x^{i+1} - x^i\|_2 < \delta \quad (3.12)$$

gdzie: δ jest małą liczbą bliską zeru, ustaloną na początku procesu optymalizacji

który śledzi różnicę pomiędzy kolejnymi generowanymi punktami w kolejnych iteracjach i przy jej niewielkiej wartości można przypuszczać, że następne, kolejne iteracje już nie spowodują znacznej poprawy funkcji celu, nawet w przypadku dużej liczby iteracji.

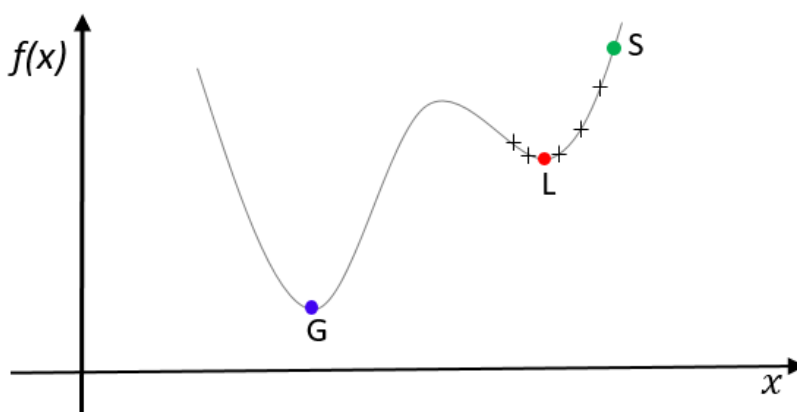
Kolejnym warunkiem stopu, który może być wykorzystywany w procesach optymalizacyjnych jest warunek oceniający wartość funkcji celu. Opisany równaniem (3.13) oblicza bezwzględną wartość

różnicy funkcji celu w dwóch kolejnych iteracjach. Jeżeli otrzymana różnica wartości jest mniejsza od ε algorytm kończy działanie. Zakłada się, że dalsze iteracje nie przyniosą już znacznej poprawy funkcji celu.

$$|f(x^{i+1}) - f(x^i)| < \varepsilon \quad (3.13)$$

gdzie: $\varepsilon > 0$ jest stałą dodatnią bliską zeru

Stałe δ i ε należy interpretować jako założoną dokładność obliczeń. W zależności od algorytmu mogą zostać wykorzystane oby dwa warunki stopu jednocześnie lub żaden z nich, albo tylko jeden, gdyż niektóre algorytmy wykorzystują warunek stopu wynikający ze specyfiki użyte metody. Często stosowanym warunkiem stopu jest liczba iteracji. Ustalona na początku procesu liczba iteracji wymusza zakończenie działania algorytmu bez względu na wartości δ i ε . Dla tego warunku stopu – co jest zaletą – jesteśmy pewni, że algorytm na pewno skończy swoje działanie, czego nie możemy zagwarantować dla innych warunków stopu. Ponadto należy zaznaczyć, że wybór warunku stopu ma ogromny wpływ na jakość uzyskanych wyników. Może się zdarzyć, że niewłaściwie dobrany warunek stopu spowoduje zatrzymanie działania algorytmu w najmniej korzystnym momencie dla całego procesu optymalizacji, znajdując minimum lokalne funkcji celu zamiast globalnego. Rys. 3.7 pokazuje w uproszczeniu wspomniany przypadek. Jak można zauważyć dla wybranego punktu startowego (S), istnieje duże prawdopodobieństwo, że algorytm ‘ugrzęźnie’ w lokalnym minimum (L) bez dużej szansy na dotarcie do minimum globalnego (G).

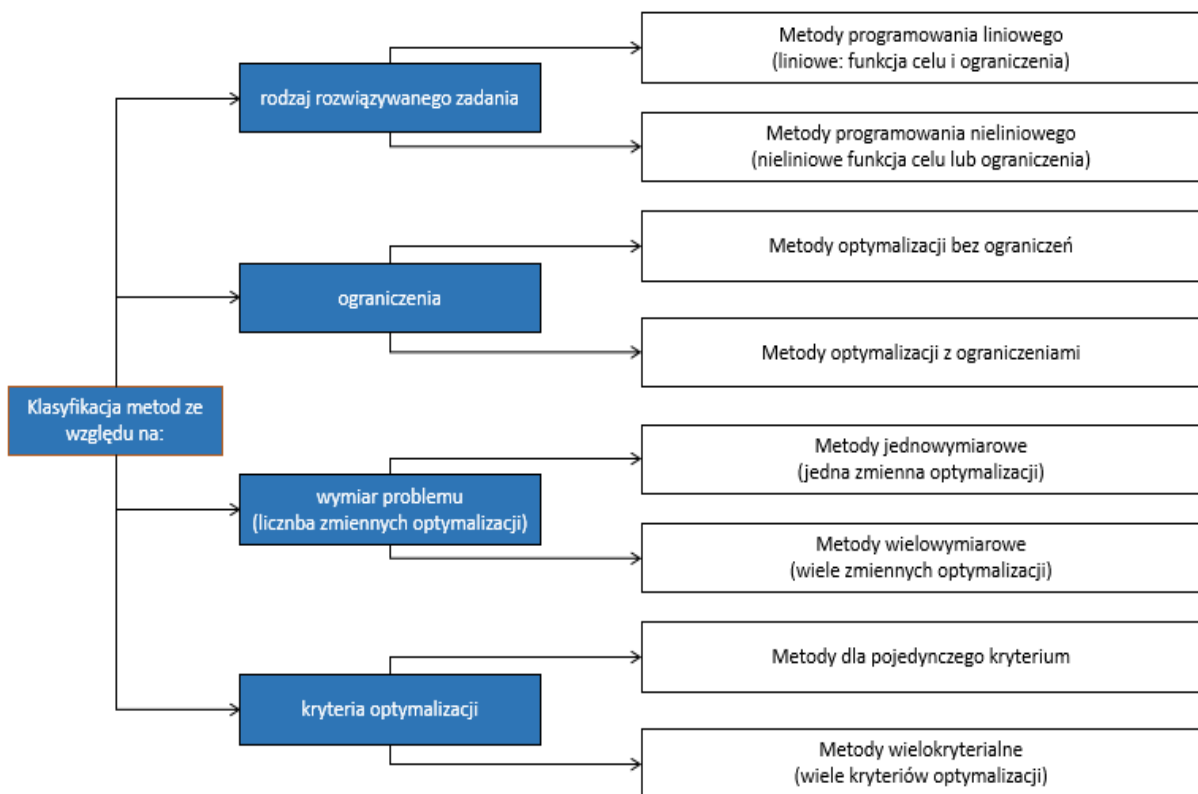


Rys. 3.7 Poszukiwanie funkcji celu zakończone w minimum lokalnym (L), kolejne iteracje oznaczono czarnymi krzyżykami, minimum globalne literą G.

Powyższy rysunek dotyka jeszcze jednego bardzo ważnego aspektu związanego z poszukiwaniem rozwiązania optymalnego. Jest nim odpowiedni dobór punktu startowego, punktu, od którego algorytm rozpoczyna swoje działanie, a który ma ogromne znaczenie na ostateczny wynik obliczeń i jakość procesu optymalizacji. Nieodpowiedni wybór punktu może prowadzić do sytuacji, kiedy algorytm, jak już wspomniano wcześniej – ‘ugrzęźnie’ w lokalnym minimum. Nie ma jednej dedykowanej metody dedykowanej do sposobu doboru punktu startowego. Stąd w praktyce algorytmy wymagające jego definicji uruchamia się kilka razy z różnymi punktami startowymi. Następnie metodą porównania wyników wybierany jest wariant najlepszy. Problem ten został opisany szerzej w rozdziale 5.1 przy okazji analizy wyników optymalizacji jednokryterialnej.

Metody optymalizacji, podobnie jak analizowane zadania optymalizacji, sklasyfikować można na wiele różnych sposobów. Główny z nich dotyczy podziału metod ze względu na rodzaj analizowanego

zadania, gdyż w praktyce, to właśnie rodzaj rozwiązywanego zadania optymalizacji jest głównym czynnikiem implikującym klasę metod, które mogą być użyte w procesie poszukiwania rozwiązania. Bazując na tym założeniu metody optymalizacji można sklasyfikować jak na pokazano na Rys. 3.8 [95] W ramach pracy wykorzystano metody optymalizacji jednokryterialnej i optymalizacji wielokryterialnej. Wykorzystano algorytmy metod programowania nieliniowego. W procesie optymalizacji zdefiniowano kilka kryteriów optymalizacyjnych zwanych funkcjonalami opisanymi szczegółowo w rozdziale 3.11. Z rodziny algorytmów programowania nieliniowego został wykorzystany algorytm NLPQL (Nonlinear Programming by Quadratic Lagrangian) [125]. Jest to algorytm wbudowany w środowisko obliczeniowe MES, podobnie jak dwa pozostałe MISQP (Mixed-Integer Sequential Quadratic Programming) [104] i ASO (Adaptive Single-Objective) [104], które zostały użyte w optymalizacji jednokryterialnej. Poza tymi algorytmami wykorzystano w pracy jeszcze algorytmy zewnętrzne, które również zastosowano w procesie optymalizacji. Pierwszym z nich jest algorytm ewolucji różnicowej L-SHADE (Success-History based Adaptive DE with Linear Population Size Reduction) [166], drugim algorytm Nelder-Meada a trzecim klasyczny algorytm ewolucyjny EA. W kolejnych podrozdziałach opisano bardziej szczegółowo ich wady i zalety wraz dokładniejszym opisem sposobu ich działania.



Rys. 3.8 Klasyfikacja metod optymalizacji.

Przeglądając literaturę najczęściej napotykamy rozwiązania optymalizacji konstrukcji oparte o metody gradientowe, bezgradientowe czy metaheurystyczne, których krótką charakterystykę przedstawiono w poniższych rozdziałach.

3.3. Optymalizacja wielokryterialna

Część problemów na które napotykamy w życiu to problemy, do rozwiązania których potrzebujemy brać pod uwagę wiele kryteriów jednocześnie. Optymalizację wielokryterialną możemy zastosować w wielu różnych dziedzinach przemysłu, w projektowaniu produktu i procesu produkcji, finansów, w projektowaniu samochodów oraz wszędzie tam, gdzie decyzje muszą być podjęte na podstawie kompromisów pomiędzy dwoma lub więcej sprzecznymi kryteriami niejednokrotnie poddanymi dodatkowym ograniczeniom. Przykładem optymalizacji wielokryterialnej może być maksymalizacja zysków przy minimalizacji kosztów produkcji, minimalizacja masy elementu przy maksymalizacji jego wytrzymałości mechanicznej oraz maksymalizacja sztywności osłony przy minimalizacji jej maksymalnych naprężeń i przemieszczeń. Często zdarza się, że obrane kryteria są sprzeczne, tzn. jedno kryterium jest zaprzeczeniem drugiego. Może się zdarzyć także, że niektóre z nich są bardziej istotne dla rozważanego problemu niż inne. Przypisuje się im wtedy różne wagi. W takiej sytuacji jest bardzo trudno o jednoznaczne rozwiązanie. W przypadku gdy mamy do czynienia ze skończonym zbiorem, w ramach którego poszukujemy optymalne rozwiązanie sprawa jeszcze nie jest najtrudniejsza. Problem komplikuje się znacznie w sytuacji, gdy zbiór rozwiązań jest nieograniczony. Wtedy musimy się posłużyć metodami bardziej zaawansowanymi. W takich przypadkach wykorzystuje się metody optymalizacji wielokryterialnej, czyli poszukujemy optymalnego rozwiązania, które będzie akceptowalne z punktu widzenia każdego kryterium. Znalezienie rozwiązania kompromisowego jest nadrzędnym celem optymalizacji wielokryterialnej [19,44,85,106,144].

Rozwiązanie problemu optymalizacji wielokryterialnej polega na znalezieniu wektora zmiennych projektowych, reprezentujących np. geometrie i warunki brzegowe itp., dla którego zdefiniowane funkcje celu osiągają ekstremum. Funkcja celu jest wektorem i jeżeli którykolwiek z jej składników będzie konkurował z innym to niemożliwe będzie znalezienie jednoznacznego rozwiązania. Zazwyczaj każdy z tych składników wektora funkcji celu osiąga ekstremum przy innym wektorze zmiennych decyzyjnych. Optymalizacja wielokryterialna polega więc na minimalizacji lub maksymalizacji wektora funkcji celu, który podlega ograniczeniom. Jeżeli rozważymy przypadek minimalizacji funkcji celu, to poszukujemy wektora zmiennych projektowych $\mathbf{x} = [x_1, x_2, \dots, x_n]$,

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n \quad (3.14)$$

na który można narzucić l ograniczeń równościowych,

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, l \quad (3.15)$$

oraz m ograniczeń nierównościowych

$$g_i(\mathbf{x}) \geq 0 \quad i = 1, 2, \dots, m \quad (3.16)$$

który będzie minimalizował zestaw p funkcji celu,

$$f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})] \quad (3.17)$$

Wynika z tego, że dla optymalizacji wielokryterialnej przestrzeń decyzyjna jest n -wymiarowa, a przestrzeń celów p -wymiarowa.

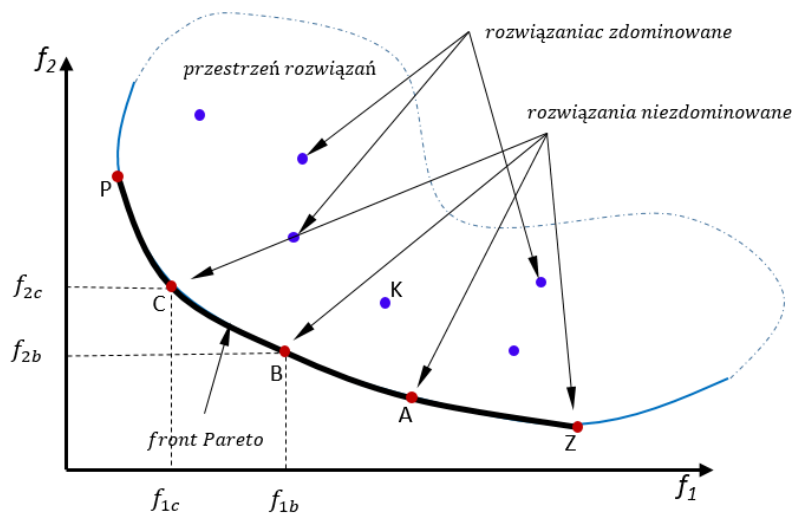
W optymalizacji wielokryterialnej, w celu porównania rozwiązań wprowadza się pojęcie rozwiązań niezdominowanych w sensie Pareto. Z rozwiązaniami niezdominowanymi mamy do czynienia, gdy poprawa wartości jednej z funkcji celu powoduje pogorszenie innej. Innymi słowy, w czasie procesu optymalizacji, rozwiązanie nie jest w stanie poprawić swojej wartości, z punktu widzenia pewnego

kryterium, bez jednoczesnego pogorszenia z punktu widzenia innego kryterium. Takie rozwiązanie nazywamy optymalnym w sensie Pareto (Pareto optymalne) [170].

Z teorii wynika, że punkt x_1 stanowi niezdominowane rozwiązanie (dominuje nad innym punktem x_2), jeżeli spełnione są oba następujące warunki [49]:

1. punkt x_1 zapewnia nie gorsze rozwiązanie niż punkt x_2 dla wszystkich funkcji celu
2. punkt x_1 zapewnia lepsze rozwiązanie niż punkt x_2 dla przynajmniej jednej funkcji celu

Na Rys. 3.9 przedstawiono graficzną interpretację frontu Pareto.



Rys. 3.9 Ilustracja niezdominowanego frontu Pareto

Przykładowe punkty A, B i C są punktami niezdominowanymi. Wynika to z faktu, że poprawa pierwszej funkcji celu f_1 powoduje pogorszenia drugiej funkcji f_2 . Punkt K jest punktem zdominowanym, albowiem nie spełnia pierwszego z powyższych dwóch warunków dla pozostałych punktów. Punkty Z, A, B, C i P dominują nad punktem K. Niezdominowane rozwiązania tworzą tzw. zbiór Pareto. Zazwyczaj tworzą one (te punkty) pewną krzywą, dlatego też stosuje się również nazwę niezdominowany front Pareto. Na Rys. 3.9 pokazano front Pareto dla dwuwymiarowej przestrzeni poszukiwań rozwiązań optymalnych. Przedstawienie wyników – frontu Pareto – przy zdefiniowanych dwóch funkcjach optymalizacji jest stosunkowo łatwe. Operujemy wtedy na płaszczyźnie 2D. W przypadku, gdy rozpatrywane są trzy kryteria to rozwiązania należące do frontu Pareto tworzą powierzchnię. Opracowanie wyników dla większej liczby kryteriów staje się trochę bardziej skomplikowane. W takich przypadkach uzyskane rozwiązania ogranicza się do poszczególnych par kryteriów w celu ich wizualizacji i przedstawia się je w postaci macierzy tzw. (*scatter-plot matrix*). [46,61]. Uogólniając, dla k -kryteriów, liczba niezbędnych par do wizualizacji wynosi $k(k - 1)/2$. Dla dużej liczby kryteriów ten sposób wizualizacji staje się mało efektywnym. Bardziej dogodnym - w takich przypadkach - sposobem wizualizacji wyników optymalizacji jest zastosowanie samoorganizujących się map (SOM – *self-organizing maps*). Koncepcja ta jest oparta na sieciach neuronowych Kohonena [90].

Przez skuteczność działania algorytmów optymalizacji wielokryterialnej można rozumieć jakość otrzymanego zbioru rozwiązań Pareto optymalnych i/lub koszt obliczeniowy. Koszt obliczeniowy może być wyrażony np. jako liczba wywołań funkcji celu lub czas, przy założeniu że obliczenia są przeprowadzane na tak samo wydajnych komputerach. W przypadku optymalizacji jednokryterialnej jakość uzyskanego rozwiązania można wyrazić za pomocą jednego parametru, którym może być błąd

otrzymanego rozwiązania, jeśli znamy faktyczną wartość minimum globalnego, lub np. różnica wartości funkcji celu otrzymana w porównywanych algorytmach.

W zadaniach optymalizacji wielokryterialnej, aby porównać jakość uzyskanego rozwiązania różnymi algorytmami, należy wziąć pod uwagę więcej aspektów a mianowicie:

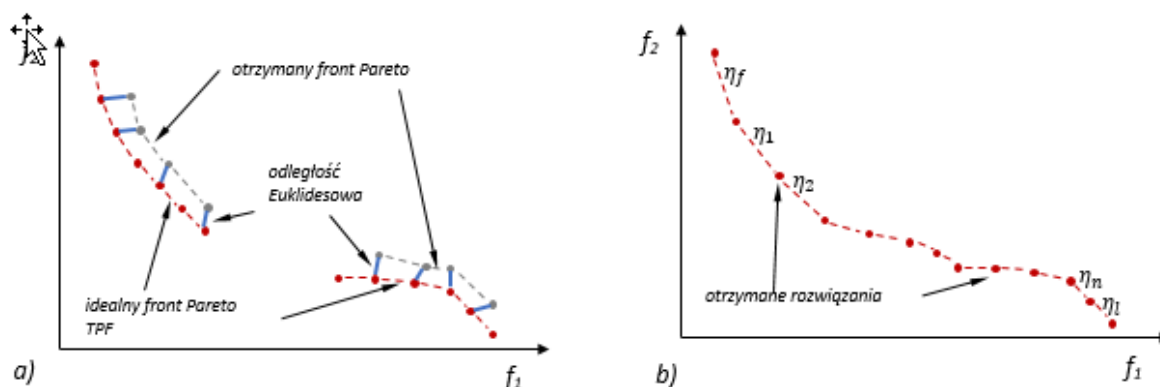
- Odległość znalezionych rozwiązań Pareto optymalnych od idealnego frontu Pareto (TPF – *true Pareto front*),
- Sposób rozłożenia rozwiązań na froncie – im bardziej równomiernie, tym lepiej
- Rozpiętość frontu Pareto – im większa odległość pomiędzy rozwiązaniami skrajnymi tym lepiej

Do porównania jakości otrzymanych zbiorów rozwiązań stosuje się odpowiednie metryki. Najbardziej popularne to:

- Metryka CM (*convergence metric*) – określająca zbieżność do TPF [36,141]. Metryka ta jest obliczana za pomocą wzoru (3.18), natomiast Rys. 3.10a przedstawia schemat wyznaczania odległości dla dwóch kryteriów.

$$CM = \frac{\sqrt{\sum_{i=1}^N \eta_i^2}}{N} \quad (3.18)$$

gdzie: η_i jest odległością Euklidesową pomiędzy danym rozwiązaniem na otrzymanym froncie a najbliższym rozwiązaniem na TPF, N jest liczebnością zbioru rozwiązań Pareto optymalnych



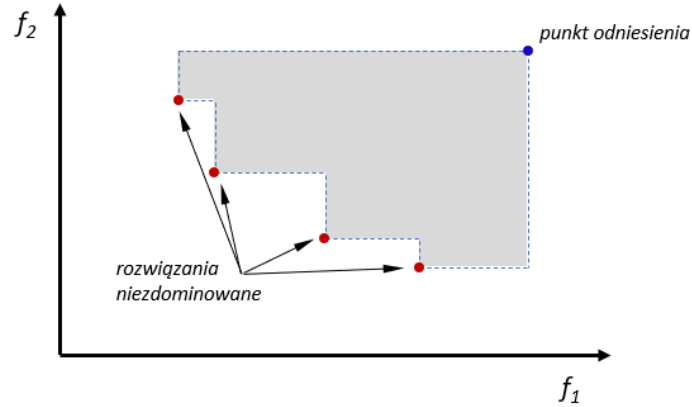
Rys. 3.10 Graficzna prezentacja sposobu wyznaczania metryk CM i DM na przykładzie optymalizacji dwukryterialnej.

- Metryka DM (*diversity metric*) – określająca równomierne rozłożenie rozwiązań w otrzymanym froncie [36,141]. Metryka ta jest obliczana za pomocą wzoru (3.19), natomiast Rys. 3.10b przedstawia obliczane odległości dla dwóch kryteriów.

$$DM = \frac{\eta_f + \eta_l + \sum_{i=1}^{N-1} |\eta_i - \bar{\eta}|}{\eta_f + \eta_l + (N-1)\bar{\eta}} \quad (3.19)$$

gdzie: η_i jest odległością pomiędzy rozwiązaniami sąsiednimi, $\bar{\eta}$ jest średnią wszystkich obliczonych odległości, η_f i η_l są odległościami pomiędzy rozwiązaniami skrajnymi a najbliższymi z otrzymanych w zbiorze rozwiązań pareto optymalnych,

- Metryka hiperobjętości HV (hipervolume metric) – będąca miarą obszaru zdominowanego przez wszystkie rozwiązania Pareto optymalne [36,86,169]. Metryka ta wymaga określenia punktu odniesienia, względem, którego jest wyznaczany obszar (hiperobjętość)(Rys. 3.11).



Rys. 3.11 Graficzna prezentacja metryki HV na przykładzie optymalizacji dwukryterialnej

Mniejsze wartości CM i DM oraz większe metryki HV dla zadań minimalizacji świadczą o otrzymaniu lepszego rozwiązania. W pracy do oceny skuteczności algorytmów spośród kilku różnych mierników oceny jakości otrzymanych frontów wybrano metrykę hiperobjętościową (HV), ze względu na to, że dla rozwiązania zadania nie jest znana lokalizacja TPF (ang. *True Pareto Front*). Ponadto metryka HV uwzględnia wszystkie trzy wymienione cechy: odległość znalezionych rozwiązań optymalnych Pareto od TPF, równomierność rozkładu rozwiązań na froncie Pareto oraz rozpiętość Frontu Pareto [149].

3.4. Metody gradientowe

Gradientowe metody optymalizacji, w odróżnieniu od metod optymalizacji bezgradientowej, wykorzystują oprócz wartości funkcji celu, również wartość jej gradientu lub wielkości z nim związanych. Z tego względu, funkcja celu musi być określona i różniczkowalna w całej przestrzeni rozwiązań problemu optymalizacyjnego \mathbb{R}^n . Ponieważ zdefiniowany problem optymalizacji w pracy jest zagadnieniem wielu zmiennych stąd też zostaną opisane tylko wybrane metody dotyczące optymalizacji wieloparametrowej.

I tak, w przypadku metod gradientowych, gradient w punkcie $x^{(t)}$ możemy obliczyć za pomocą następującej formuły:

$$\nabla f(x^{(t)}) = \begin{bmatrix} \frac{\partial f(x^{(t)})}{\partial x_1} \\ \frac{\partial f(x^{(t)})}{\partial x_2} \\ \dots \\ \frac{\partial f(x^{(t)})}{\partial x_N} \end{bmatrix}, \quad \text{gdzie:} \quad \frac{\partial f(x^{(t)})}{\partial x_i} = \frac{f(x_i^{(t)} + \Delta x_i^{(t)}) - f(x_i^{(t)} - \Delta x_i^{(t)})}{2\Delta x_i^{(t)}} \quad (3.20)$$

Hesjan w punkcie $x^{(t)}$ natomiast obliczymy następująco:

$$\nabla^2 f(x^{(t)}) = \begin{bmatrix} \frac{\partial^2 f(x^{(t)})}{\partial x_1^2} & \frac{\partial^2 f(x^{(t)})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x^{(t)})}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f(x^{(t)})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x^{(t)})}{\partial x_2^2} & \dots & \frac{\partial^2 f(x^{(t)})}{\partial x_2 \partial x_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(x^{(t)})}{\partial x_N \partial x_1} & \frac{\partial^2 f(x^{(t)})}{\partial x_N \partial x_2} & \dots & \frac{\partial^2 f(x^{(t)})}{\partial x_N^2} \end{bmatrix} \quad (3.21)$$

gdzie:

$$\frac{\partial^2 f(x^{(t)})}{\partial x_i^2} = \frac{f(x_i^{(t)} + \Delta x_i^{(t)}) - 2f(x^{(t)}) + f(x_i^{(t)} - \Delta x_i^{(t)})}{(\Delta x_i^{(t)})^2} \quad (3.22)$$

$$\frac{\partial^2 f(x^{(t)})}{\partial x_i \partial x_j} = \frac{\frac{f(x_i^{(t)} + \Delta x_i^{(t)})}{\partial x_j} - \frac{f(x_i^{(t)} - \Delta x_i^{(t)})}{\partial x_j}}{2\Delta x_i^{(t)}} \quad (3.23)$$

Pochodne cząstkowe w ostatnim wyrażeniu powyżej są z kolei obliczone tak, jak składowe gradientu, tylko że w innym punkcie. Wyrażenie $x_i^{(t)} + \Delta x_i^{(t)}$ reprezentuje wektor

$$(x_1^{(t)}, \dots, x_i^{(t)} + \Delta x_i^{(t)}, \dots, x_N^{(t)})^T \quad (3.24)$$

Żeby obliczyć gradient potrzebnych jest $2N$ różnych wartości funkcji. A żeby wyznaczyć Hessian potrzebnych jest $3N + 4 \binom{N}{2} = 2N^2 + N$.

Ponieważ gradient jest kierunkiem najszybszego wzrostu funkcji celu, ujemna wartość gradientu jest kierunkiem, w którym wartość funkcji celu zmniejsza się i nazywany jest kierunkiem najszybszego spadku funkcji. Ta własność jest bardzo pomocna przy poszukiwaniu minimum/maksimum funkcji celu i jest wykorzystywana we wszystkich metodach gradientowych.

Większość znanych metod gradientowych wykorzystuje ten sam schemat działania. Ogólnie strategia ta sprowadza się do dwóch etapów:

- a) określenie kierunku poszukiwania minimum d . Do tego celu wykorzystywany jest gradient funkcji celu i wielkości z nim powiązanych
- b) dobór kroku h o odpowiedniej długości, który należy wykonać zgodnie z wcześniej wytyczonym kierunkiem poszukiwań.

Kierunek poszukiwań (ang. *search direction*) $d^{(t)}$ jest kierunkiem spadku w punkcie $x^{(t)}$, jeśli w otoczeniu tego punktu spełniony jest następujący warunek:

$$\nabla f(x^{(t)}) \cdot d^{(t)} \leq 0 \quad (3.25)$$

Im niższa ujemna wartość $\nabla f(x^{(t)}) \cdot d^{(t)}$ tym większy spadek funkcji w kierunku $d^{(t)}$. W metodach gradientowych często w ramach pojedynczej iteracji dokonuje się poszukiwań w wybranym kierunku. Mamy do czynienia wtedy z optymalizacją jednowymiarową.

Poszczególne metody bazujące na gradientach różnią się między sobą sposobem wyznaczania kierunku poszukiwań i metodami określania długości kroku.

W zależności od sposobu wyznaczania kierunku poprawy rozwiązania d , metody gradientowe posiadają wiele odmian, wśród których można wyróżnić [33]:

1. Metodę (Cauchy'ego) – która jest uważana za jedną z najprostszych metod optymalizacyjnych wykorzystujących informację o różniczkowości funkcji celu.

Kierunek poszukiwań w metodzie Cauchy'ego jest kierunkiem najszybszego spadku:

$$s^{(k)} = -\nabla f(x^{(k)}) \quad (3.26)$$

Algorytm ten gwarantuje poprawę w każdej iteracji, tzn. gwarantuje spadek wartości funkcji. Metoda najszybszego spadku działa dobrze, gdy $x^{(0)}$ jest daleko od minimum x^* , albowiem wtedy zmiany gradientu są znaczące. Natomiast, gdy bieżący punkt jest w pobliżu minimum, zmiana gradientu jest mała, wobec czego następny punkt powstały w wyniku poszukiwania w jednym kierunku jest blisko punktu bieżącego. Poniżej przedstawiono pseudo kod metody najszybszego spadku:

- a) Zdefiniuj maksymalną liczbę iteracji N , dwa parametry zakończenia ϵ_1 i ϵ_2 oraz punkt początkowy i ustal $k = 0$.
- b) Oblicz $\nabla f(x^{(k)})$
- c) Jeśli $\|\nabla f(x^{(k)})\| \leq \epsilon_1$, zakończ zadanie
Jeśli $k \geq N$; zakończ zadanie
W przeciwnym razie przejdź do kroku 4).
- d) Rozpocznij wyszukiwanie wzdłuż kierunku, żeby znaleźć α^k tak, aby wyrażenie

$$f(x^{(k+1)}) = f(x^{(k)}) + \alpha^k s^{(k)} \quad (3.27)$$

było minimalne.

Jednym z kryteriów zakończenia procesu jest

$$\left| \nabla f(x^{(k+1)}) \cdot \nabla f(x^{(k)}) \right| \leq \epsilon_2 \quad (3.28)$$

- e) Jeśli

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{x^{(k)}} \leq \epsilon_1 \quad (3.29)$$

zakończ zadanie

W przeciwnym przypadku ustal $k = k + 1$ i przejdź do kroku 2).

2. Metodę Newtona – w której kierunek poprawy rozwiązania definiowany jest wzorem (3.31). Równanie to otrzymujemy rozwijając funkcję f wokół punktu $x^{(k)}$ w szereg Taylora drugiego rzędu, która przyjmuje następującą postać:

$$f(x^{(k+1)}) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2} (x - x^{(k)})^T \nabla^2 f(x^{(k)}) (x - x^{(k)}) \quad (3.30)$$

Kierunek poszukiwań w metodzie Newtona jest zdefiniowany wyrażeniem:

$$s^{(k)} = - [\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)}) \quad (3.31)$$

Jeśli macierz $[\nabla^2 f(x^{(k)})]^{-1}$ jest półdefinitnie określona, to dowodzi, że jest określona w otoczeniu minimum, a kierunek $s^{(k)}$ jest kierunkiem spadku. Stąd metodę Newtona wykorzystujemy, kiedy punkt początkowy jest blisko minimum.

Algorytm jest bardzo podobny do metody najszybszego spadku, lecz poszukiwania prowadzone są w innym kierunku. Możliwy warunek zakończenia optymalizacji wzdłuż kierunku wygląda następująco:

$$\left| \nabla f(x^{(k+1)}) \cdot [\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)}) \right| \leq \epsilon_2 \quad (3.32)$$

3. Metodę Levenberga-Marquardta jest metodą, która łączy obydwie poprzednie. Wykorzystana została na potrzeby uczenia SSN podczas tworzenia metamodelu w niniejszej pracy. Metoda Cauchy'ego działa poprawnie, w przypadku, gdy punkt początkowy jest daleko od minimum, podczas, gdy metoda Newtona sprawdza się lepiej, gdy punkt początkowy jest blisko minimum. Dlatego w metodzie Marquardta, stosujemy metodę Cauchy'ego na początku by następnie w końcowej fazie procesu optymalizacji użyć metodę Newtona.[97]

Algorytm tej metody można opisać następująco:

- a) Wybierz maksymalną liczbę iteracji N , parametr zakończenia ϵ i punkt początkowy.
Ustaw $k = 0$ oraz $\lambda^{(0)} = 10^4$
- b) Oblicz $\nabla f(x^{(k)})$
- c) Jeśli $\|\nabla f(x^{(k)})\| \leq \epsilon$, zakończ zadanie
Jeśli $k \geq N$, zakończ zadanie
W przeciwnym razie przejdź do kroku 4).
- d) Oblicz wartość wyrażenia:

$$x^{(k+1)} = x^{(k)} - [\nabla^2 f(x^{(k)}) + \lambda^{(k)} I]^{-1} \nabla f(x^{(k)}) \quad (3.33)$$

- e) Jeśli zachodzi nierówność

$$f(x^{(k+1)}) < f(x^{(k)}) \quad (3.34)$$

przejdź do punktu 6).

W przeciwnym przypadku przejdź do punktu 7).

- f) Ustaw $\lambda^{(k+1)} = \frac{1}{2} \lambda^{(k)}$, $k = k + 1$ i przejdź do punktu 2).
- g) Ustaw $\lambda^{(k+1)} = 2 \lambda^{(k)}$, $k = k + 1$ i przejdź do punktu 4).

Oczywiście opisane powyżej metody optymalizacji nie wyczerpują tematu w tej dziedzinie, gdyż stanowią tylko wycinek z całej puli stosowanych metod. W tym miejscu należałoby wspomnieć o metodzie gradientu prostego, gradientów sprzężonych Hestenesa-Stiefela, o metodach Quasi-Newtonowskich (DPF – Davidon, Powell, Fletcher; BFGS - Broyden, Fletcher, Goldfarb, Shanno) oraz o metodzie Fletchera-Reevesa czy metodzie zaufanych regionów. [95,97].

3.5. Metody bezgradientowe

Poszukiwanie rozwiązania optymalnego z wykorzystaniem metod bezgradientowych oparte jest jedynie na znajomości wartości funkcji celu. Dlatego też metody te nazywane są często metodami bezpośredniego szukania. Charakteryzują się względną prostotą, gdyż nie wymagają dodatkowych obliczeń pochodnych funkcji celu. Często jednak okupione jest to ich niską efektywnością i długimi czasami obliczeń związanych z poszukiwaniem rozwiązań optymalnych. Poszczególne metody precyzują sposób określania zarówno kierunku poszukiwań, jak i długości kroku w kolejnych iteracjach na podstawie znajomości wartości funkcji celu w wybranych punktach zbioru rozwiązań dopuszczalnych. Dzięki temu, że nie wymagają obliczania pochodnych funkcji celu, mogą być wykorzystane we wszystkich zadaniach optymalizacji, w których funkcja celu opisana jest skomplikowanymi zależnościami, dla których trudno jest wyznaczyć pochodne w sposób analityczny. Metody te wykorzystywane są również w zadaniach, gdy nie ma możliwości dokładnego opisanie

funkcji celu wzorami matematycznymi. Z przypadkami takimi spotykamy się często w analizie optymalizacyjnej złożonych procesów przemysłowych, w których do obliczenia wartości funkcji celu wykorzystane są skomplikowane modele symulacji numerycznej nie pozwalające na jednoznaczne zapisanie funkcji celu wzorami matematycznymi. Metody optymalizacji bezgradientowych obejmują optymalizację funkcji jednej jak i wielu zmiennych. Ze względu na tematykę tej pracy w dalszej części skupiono się tylko na metodach optymalizacji wielu zmiennych. Jedną z prostszych metod bezgradientowej optymalizacji wielowymiarowej funkcji celu jest metoda Hooke'a-Jeevesa [95], w której kierunki poszukiwania rozwiązania optymalnego są równoległe do osi układu współrzędnych i niezmiennie w całym procesie poszukiwań. Długość kroku określany jest prostymi zależnościami. Metoda ta składa się z dwóch etapów, próbnego oraz roboczego [95]. Metodą podobną do metody Hooke'a-Jeevesa jest metoda Rosenbrocka [95]. Różnica polega na tym, że kierunki poszukiwania rozwiązania optymalnego są definiowane wstępnie wzdłuż przyjętych na starcie ortogonalnych kierunków $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$, które zazwyczaj są wersorami układu współrzędnych. Kierunki te jednak nie są stałe, lecz w trakcie pracy algorytmu mogą ulec zmianie poprzez obrót układu współrzędnych. Metodę Powella wyróżnia natomiast fakt, że poszukiwania rozwiązań optymalnych odbywa się wzdłuż tzw. kierunków sprzężonych. Zadanie optymalizacji wielowymiarowej sprowadza się wtedy do zadań optymalizacji jednowymiarowej wzdłuż każdego kierunku. Tak więc do znalezienia minimum można użyć dowolnej bezgradientowej metody funkcji jednej zmiennej np. metody złotego podziału czy metoda Fibonacciego. Działanie tych metod może się okazać w praktyce bardzo czasochłonne, szczególnie gdy funkcja celu posiada dużą liczbę zmiennych projektowych [95]. W rzeczywistych problemach optymalizacji z reguły funkcja celu ma postać dosyć skomplikowaną, niejednokrotnie z wieloma lokalnymi ekstremami, dla której znalezienie optymalnego rozwiązania nie jest proste. Z pomocą przychodzą algorytmy niedeterministyczne opisane w rozdziale 3.10. Metodę sympleksu Nelder-Meada [70] dokładnie opisano w rozdziale 3.10.1, ponieważ algorytm ten został użyty do rozwiązania jednego z wariantów zadań optymalizacji.

3.6. Metody ewolucyjne

W rzeczywistych problemach optymalizacyjnych, wymóg ciągłości i różniczkowalności funkcji celu oraz odporności na utykanie w minimum lokalnym znacząco ogranicza możliwość stosowania metod gradientowych. W związku z tym opracowano i wdrożono szereg metod obliczeniowych, które wykorzystują czynnik losowy, zebrano pod wspólną nazwą metod metaheurystycznych. Pierwsze informacje dotyczące symulowanej ewolucji pojawiły się w latach 1950-1960. Wówczas to A.S. Fraser, G.E.P. Box, G.J. Friedman, W.Bledsoe, H.J. Bremermann podjęli badania dotyczące zastosowania systemów ewolucyjnych. W tym też czasie pojawiły się pierwsze artykuły [29,31,32,67], dotyczące prezentowanego tematu. W dużej mierze celem prac prowadzonych przez prekursorów symulowanej ewolucji było naśladowanie, komputerowe symulowanie zachodzącego w naturze procesu ewolucji. W kolejnych latach kształtowały się główne kierunki rozwoju symulowanej ewolucji jak programowanie ewolucyjne [66], strategie ewolucyjne [127], algorytmy genetyczne [78]. Przedstawiona przez Hollanda w 1975 roku teoria schematów dla algorytmów genetycznych [78] poddawano częstym weryfikacjom [1,4,6,8,23,74,79,96,126]. Pod koniec lat osiemdziesiątych ubiegłego wieku (1989) J.D. Schaffer opisał skuteczność operatora mutacji w algorytmach genetycznych [131], natomiast D.B. Fogel zajął się analizą operatora krzyżowania [64]. Tematykę wpływu operatora krzyżowania na wydajność algorytmu podjęli również w swoich pracach De Jong, Spears, Vose [48,147]. W dalszych badaniach Schaffer i Eshelmana nad obydwoma parametrami, doszli do wniosku, że operator mutacji nie jest wystarczający w każdym przypadku [132].

Obecnie, metody ewolucyjne zaliczamy do szerszej grupy metod metaheurystycznych. Nowo powstałe algorytmy mają różną inspirację biologiczną. Główne kierunki badań dotyczących algorytmów

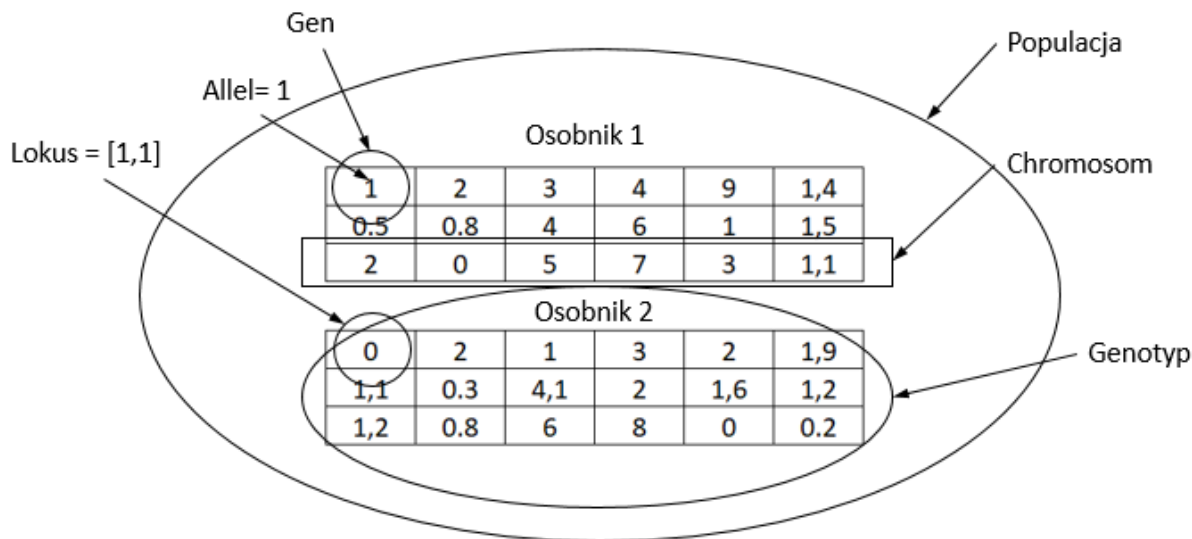
ewolucyjnych (ang. *evolutionary algorithms*) można znaleźć w pracach [10,12,13,14,15,16,25,111,122]. Najbardziej popularne metody inspirowane biologicznie obejmują również inne algorytmy takie jak: algorytmy rojowe (ang. *Particle swarm optimization, PSO*), algorytmy mrówkowe (ang. *Ant Colony Optimization*), programowanie ewolucyjne (ang. *evolutionary programming*) [8,25,65], które wykorzystuje tylko operator mutacji, strategie ewolucyjne (ang. *evolution strategies*) [7,8,26,27,75], algorytmy genetyczne (ang. *genetic algorithms*) [8,11,17,18,24,96], oraz programowanie genetyczne (ang. *genetic programming*) [88,92,93], które z kolei opierają się na wykorzystaniu zarówno operatora krzyżowania jak i mutacji dla poprawy jakości rozwiązania. Przebieg procesu ewolucji w dużej mierze zależy od wyników, jakie uzyskują poszczególne osobniki w „środowisku” zdefiniowanym w postaci funkcji przystosowania (ang. *fitness function*). Na jej podstawie przypisuje się osobnikom wartości prawdopodobieństwa przejścia do nowej populacji. Rozwiązania gorsze jakościowo zostają odrzucone. Osobniki, które przetrwały selekcję poddane zostają działaniu operatorów genetycznych krzyżowania i mutacji.

Algorytm Ewolucyjny (AE), programowanie ewolucyjne, strategie ewolucyjne, algorytmy genetyczne oraz programowanie genetyczne tworzą grupę metaheurystycznych metod iterowanego poszukiwania skończonej przestrzeni stanów, posiadającej odwzorowanie w postaci funkcji celu. Wspólną cechą algorytmów z tej grupy jest podobieństwo do ewolucji organizmów, występujących w naturze, wyrażające się głównie we wzorowaniu się na mechanizmach doboru naturalnego i dziedziczności. Wprawdzie teoretyczne opisy AE jako całości często stosują bardziej matematyczny, precyzyjny język, ale ciągle używa się terminologii inspirowanej naturą [5], która jest bardzo użyteczna, jeśli chodzi o zrozumienie podstaw tej dziedziny. Stąd pojęcia używane w ramach genetyki komputerowej są w większości odpowiednikami nazw odnajdywanych w przyrodzie. Zaliczamy do nich chromosom składający się z uporządkowanego łańcucha genów (ang. *gen*). Każdy gen ma swoje miejsce w chromosomie (*locus*) i swoją wartość (*allele*). Strukturę stworzoną przez chromosomy dla danego osobnika nazywamy genotypem (ang. *genotype*), jego zespół wartości reprezentowany jest przez fenotyp. Osobnik (ang. *individual*) z kolei to element populacji, punkt w przestrzeni poszukiwań, reprezentowany przez genotyp¹ wraz z odpowiadającym mu wskaźnikiem przystosowania. Osobniki tworzą populację (ang. *population*), których liczebność jest ściśle określona. W większości modeli AE wyróżnia się populację rodzicielską, składającą się z osobników z przeliczonymi wskaźnikami przystosowania (μ) i populację potomną, na której operatory modyfikują osobniki (λ). Dwa różne symbole stosowane są dlatego, że nie zawsze te dwa rozmiary populacji są sobie równe. Populacje w kroku czasowym (t) jest przeważnie oznaczana P^t , natomiast populacja potomna O^t . Należy jeszcze wspomnieć o operatorze reprodukcji (ang. *reproduction*), który wybiera z aktualnej populacji rodzicielskiej pewne osobniki i kopiuje je do populacji potomnej w celu dalszego przetwarzania. Zazwyczaj kieruje się wartościami wskaźników przystosowania, ale czasami jest realizowany losowo. Wartość wskaźnika przystosowania decyduje o jego prawdopodobieństwie wyboru. Sukcesja (ang. *succession*) natomiast zastępuje populację rodzicielską przez osobniki z populacji potomnej. W zależności od algorytmu stosujemy sukcesję prostą (cała populacja rodzicielska jest zastępowana poprzez populację potomną), losową czy opartą o wskaźniki przystosowania. Częścią operatora sukcesji także może być selekcja (ang. *selection*). Jest to wspólna nazwa na procedurę wybierającą osobniki w zależności od wartości wskaźnika przystosowania. Selekcja jest z reguły najistotniejszą częścią operatora reprodukcji. Działanie wszelkich operatorów, w których występuje selekcja prowadzi do powstania zjawiska zwanego naciskiem selektywnym lub presją selektywną (ang. *selection pressure*).

¹ Przy małej liczbie zmiennych genotyp składa się z jednego chromosomu. Dopuszczalnym jest wówczas stwierdzenie, że osobnik w algorytmie reprezentowany jest w postaci chromosomu.

$$Osobnik = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,2} & a_{3,3} \end{bmatrix}$$

$$Fenotyp = \begin{bmatrix} 1 & 2 & 3 & 4 & 9 & 1.4 \\ 0.5 & 0.8 & 4 & 6 & 1 & 1.5 \\ 2 & 0 & 5 & 7 & 3 & 1.1 \end{bmatrix}$$



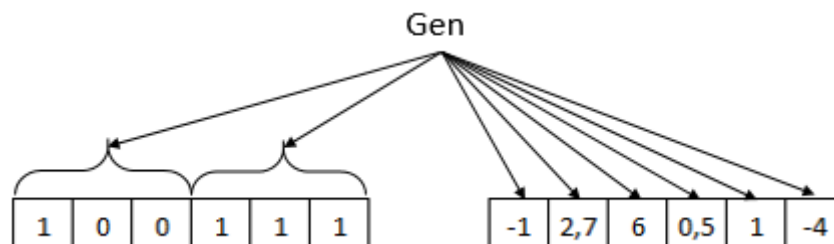
Rys. 3.12 Graficzna reprezentacja osobników w algorytmie ewolucyjnym

Mianem tym określa się stopniowy wzrost średniego wskaźnika przystosowania w populacji. Jest to główny mechanizm działania algorytmu ewolucyjnego. W algorytmach z reguły wykorzystywany jest jeszcze operator mutacji (ang. *mutation*), który wprowadza do genotypu losowe modyfikacje oraz operator krzyżowania (ang. *crossing-over*) polegający na wymianie części chromosomu pomiędzy wybranymi osobnikami, z reguły dwoma, ale możliwa jest również wymiana pomiędzy większą ilością osobników. Wtedy jest mowa o krzyżowaniu wieloosobniczym. Dodatkowo rozróżniamy krzyżowanie jedno i wielopunktowe, arytmetyczne i heurystyczne.

Opisane powyżej pojęcia pokazano graficznie na Rys. 3.12. Powyższa lista pojęć związanych z genetyką zapewne nie jest kompletna, albowiem niektóre algorytmy AE korzystają z własnego, bardziej szczegółowego nazewnictwa. Do tej grupy pojęć należy dodać, nie spotykaną w genetyce naturalnej, funkcję przystosowania (ang. *fitness function*). Funkcja ta wyznacza wartość wskaźnika przystosowania, który stanowi o wyborze osobników najbardziej wartościowych. Pozwala ocenić stopień adaptacji poszczególnych osobników. Przyjmuje się, że osobniki najbardziej wartościowe generują największe wartości funkcji przystosowania. Wybór osobników najbardziej wartościowych polega na wyznaczeniu osobników, którym odpowiadają największe wartości funkcji celu.

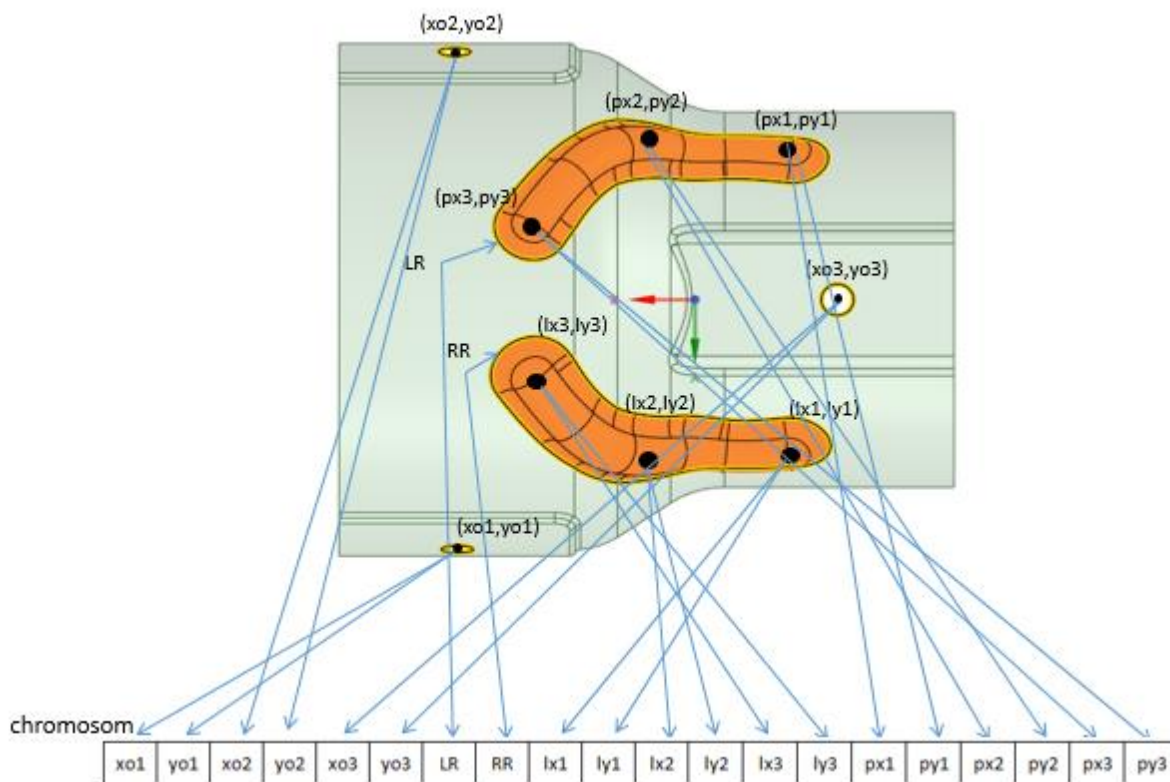
Jak można zauważyć na Rys. 3.12 do populacji należą dwa osobniki. Każdy z nich zawiera trzy chromosomy. Chromosomy te podlegają ewolucji realizowanej za pomocą selekcji i działaniu operatorów genetycznych. Każdy z nich składa się z sześciu genów. Wartość poszczególnych genów tzw. allele przedstawiają zakodowane, poszukiwane parametry zadania. Funkcja przystosowania określa stopień konkurencyjności osobnika w populacji. Jej argumentami są należące do ocenianego fenotypu allele. Wyznaczana jest osobno dla każdego pojedynczego chromosomu lub dla jednego fenotypu. Wartości funkcji przystosowania automatycznie klasyfikują osobniki od najlepiej do najgorzej przystosowanego. Wartości najwyższe funkcji przystosowania dają wysokie szanse na przetrwanie selekcji.

Zasada działania algorytmów genetycznych wymusza kodowanie poszukiwanych parametrów zadania. Algorytmy heurystyczne pozwalają na określenie wartości genów w chromosomie w postaci rzeczywistej. Kodowanie to przyjmuje postać chromosomów. Dobór odpowiedniej metody kodowania parametrów nie jest zadaniem prostym. Wybrana metoda kodowania powinna być neutralna. Tzn. nie powinna faworyzować żadnego z możliwych do otrzymania rozwiązań. Typ algorytmu ewolucyjnego również może mieć wpływ na sposób kodowania. Przykładem może być klasyczny algorytm genetyczny, w którym stosuje się kodowanie binarne, gdy w algorytmach ewolucyjnych, strategiach ewolucyjnych lub ewolucji różnicowej dozwolone są wartości ze zbioru liczb rzeczywistych. Umożliwia to różną reprezentację genu w algorytmach genetycznych i np. w strategiach ewolucyjnych (Rys. 3.13). Najczęstsze metody kodowania parametrów zadania, które możemy spotkać to kodowanie binarne, logarytmiczne [130] czy kodowanie rzeczywiste.



Rys. 3.13 Reprezentacja genu w algorytmach genetycznych i ewolucyjnych

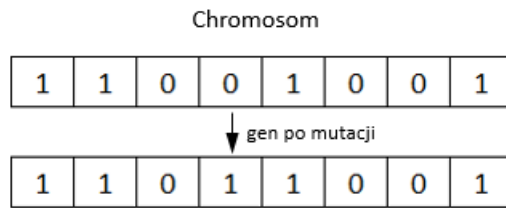
Geny jako zmienne decyzyjne w zależności od rodzaju zadania optymalizacji są używane w niniejszej pracy m.in. do modelowania kształtu przetłoczeń osłony oraz warunków brzegowych. Prezentują one charakterystyczne cechy modelu, tj. długości krawędzi, głębokość przetłoczenia, promień otworu, współrzędne ich położenia itp. Na Rys. 3.14 przedstawiono przykład parametryzacji osłony termicznej, w której kodowane zmienne projektowe reprezentują: współrzędne położenia otworów montażowych ($x_{o1}, y_{o1}; x_{o2}, y_{o2}; x_{o3}, y_{o3}$), promienie przetłoczeń (LR, RR) oraz współrzędne punktów kontrolnych krzywej parametrycznej typu 'spline' ($lx_1, ly_1; lx_2, ly_2; lx_3, ly_3; rx_1, ry_1; rx_2, ry_2; rx_3, ry_3$).



Rys. 3.14 Przykład kodowania zmiennych projektowych w chromosomie

W przypadkach optymalizacji wybranych cech geometrycznych, przy modelowaniu kształtu, zasadne jest stosowanie krzywych parametrycznych. Najczęściej wykorzystuje się krzywe Beziera, krzywe B-spline oraz krzywe NURBS (Non Uniform Rational B-Spline) [118]. Zastosowanie krzywych parametrycznych pozwala na redukcję liczby zmiennych projektowych. Bardziej szczegółowo krzywe te zostały opisane w rozdziale poświęconym optymalizacji kształtu (rozdział 3.1).

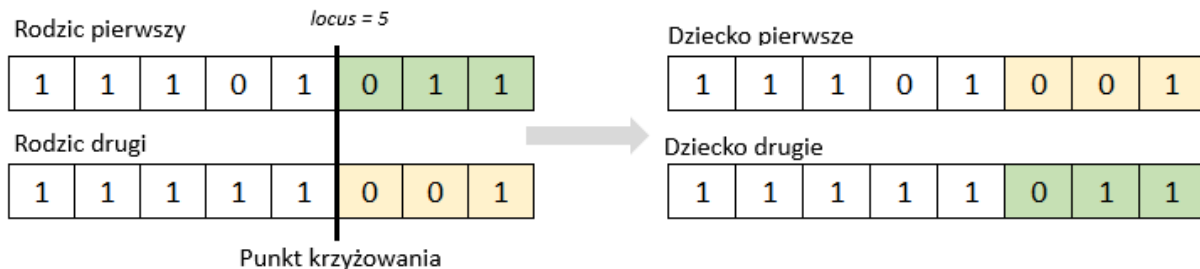
Operatory ewolucyjne możemy podzielić na operatory jednoargumentowe, do których zaliczamy mutację [9,93,132], która zmienia losowo wybrane geny pojedynczego osobnika, oraz operatory dwuargumentowe, do których zaliczamy krzyżowanie [30,48,60], które tworzy z kolei osobniki potomne bazując na dwóch losowo wybranych osobnikach z populacji rodzicielskiej. W klasycznym algorytmie genetycznym operator krzyżowania ma priorytet wyższy od operatora mutacji. Tzn. operator mutacji odgrywa rolę drugoplanową. Jeżeli chodzi o mutację, to wprowadza ona do genotypu losowe zmiany. Ma to na celu podtrzymanie różnorodności w populacji, innymi słowy ma przynajmniej częściowo zapobiegać przedwczesnej zbieżności algorytmu. Nie służy temu zbyt silna mutacja. Daje wtedy efekt odwrotny do zamierzonego. Niszczy dobre rozwiązania – zamiast je subtelnie różnicować. Stąd w procesie ewolucji mutacja ma znaczenie drugorzędne. Jej rangę określa wartość przyjmowanego z reguły bardzo małego prawdopodobieństwa jej wystąpienia ($0 \leq p_m \leq 0.1$). Podstawowym zadaniem mutacji jest umożliwienie odtworzenia zniszczonego podczas krzyżowania chromosomu. Następuje to z reguły, gdy wartość funkcji przystosowania potomka jest mniejsza od wartości funkcji przystosowania rodzica. Dla podstawowego algorytmu genetycznego, z binarną reprezentacją chromosomu wykonanie mutacji z prawdopodobieństwem p_m sprowadza się w praktyce do wylosowania dla każdego *locus* liczby z przedziału np.: (0,1). Następnie wybiera się jednego lub większą ilość genów (decyduje o tym rodzaj zastosowanej mutacji), dla których wylosowana liczba jest mniejsza bądź równa p_m . Wybrane geny w GA zmieniają swoją wartość na przeciwną (Rys. 3.15), bądź wartość losową.



Rys. 3.15 Operacja mutacji.

W strategiach ewolucyjnych natomiast mutacja jest operatorem głównym. W zależności od implementacji, polega na zamianie jednego lub większej liczby alleli na inne. Dla kodowanych binarnie chromosomów, losuje się przeważnie dwa geny i zamienia się je miejscami. W alternatywnym przypadku odpowiednio wylosowany gen zostaje zanegowany. Permutacje z kolei stosuje się w przypadku genotypów zakodowanych liczbami całkowitymi. W sytuacji, gdy genotypy zakodowane są liczbami rzeczywistymi, wprowadza się losowe zmiany o danym rozkładzie (najczęściej normalnym), do przypadkowych genów. Najczęściej wybierana jest mutacja równomierna. Wtedy, każdy gen w chromosomie ma równe szanse być poddany procesowi mutacji, zgodnie z prawdopodobieństwem jej wystąpienia. [76]

Drugim operatorem, który dla algorytmów ewolucyjnych jest operatorem dominującym jest operator krzyżowania. Prawdopodobieństwo jego występowania przyjmuje z reguły wartość z zakresu $0.5 < p_c < 1$. Efektem pracy operatora krzyżowania pary rodzicielskiej są dwa osobniki potomne. Głównym zadaniem krzyżowania jest wymieszanie materiału genetycznego. W wyniku czego otrzymuje się populację potomną (ang. *offspring*), której liczebność jest zwykle redukowana do jednego losowo wybranego potomka. Rozróżniamy wiele wariantów krzyżowania. Do najczęściej spotykanych należy krzyżowanie jedno i wielopunktowe.



Rys. 3.16 Krzyżowanie jednopunktowe

Krzyżowanie jednopunktowe (Rys. 3.16) polega na przecięciu chromosomów pary rodzicielskiej w punkcie krzyżowania oraz złączenia odciętych części chromosomów naprzemiennie. Niedoskonałością krzyżowania prostego jest to, że może prowadzić do sytuacji, w której niektóre ciągi kodowe nigdy nie zostaną osiągnięte. Było to powodem do opracowania innych bardziej efektywny sposobów rekombinacji genów. Możemy tutaj wymienić krzyżowanie dwupunktowe, wielopunktowe, równomierne [88], arytmetyczne, heurystyczne czy ukośne [60]. Modyfikacją operatora krzyżowania jest krzyżowanie równomierne [134]. Ten rodzaj krzyżowania wprowadza pojęcie losowego wzorca dziedziczenia. Wzorec ten wskazuje, które geny dziedziczone są od pierwszego rodzica, a które od drugiego.

Za pomocą operatorów, omówionych w poprzednim rozdziale, tworzy się populację potomną zwiększając wielkość populacji pośredniej. Metoda selekcji redukuje jej liczebność do rozmiaru populacji rodzicielskiej. Pożądane jest takie działanie metody selekcji, aby kolejne generacje populacji rodzicielskiej P charakteryzowały się wyższą średnią wartości funkcji przystosowania:

$$\bar{P}_t \leq \bar{P}_{t+1} \quad (3.35)$$

gdzie: \bar{P} – średnia wartość funkcji przystosowania dla P w chwili t
 \bar{P}_{t+1} – średnia wartość funkcji przystosowania dla P w chwili $t + 1$

Podczas gdy w klasycznych algorytmach genetycznych mechanizmem reprodukcji była tylko selekcja proporcjonalna, to w przypadku algorytmów ewolucyjnych można wymienić dodatkowo sekcję rangową czy turniejową, w przypadku których istnieje możliwość sterowania naporem selekcyjnym. Ponadto nie wolno nam zapominać o selekcji typu (μ, λ) czy $(\mu + \lambda)$.

Jedną z podstawowych i często stosowanych metod selekcji była metoda koła ruletki, która definiuje dla każdego osobnika w populacji \mathbf{P}^t prawdopodobieństwo jego reprodukcji według wzoru:

$$p_s(ch_t^j) = \frac{F(ch_t^j)}{\sum_{j=1}^N F(ch_t^j)} \quad (3.36)$$

gdzie: $F(ch_t^j)$ to wartość funkcji przystosowania osobnika $ch_t^j, j = 1, 2, \dots, N; t = 1, 2, \dots$ [poz4]

Prawdopodobieństwo wylosowania osobnika jest więc wprost proporcjonalne do jego wartości funkcji przystosowania (selekcja proporcjonalna). Graficznie można ją przedstawić w postaci koła, które jest podzielone na N części odpowiadającej liczbie osobników w populacji. Wielkość każdego wycinka koła zależna jest od prawdopodobieństwa wylosowania poszczególnych osobników do następnego pokolenia. Można ją opisać wzorem:

$$v(ch_t^j) = \frac{F(ch_t^j)}{\sum_{j=1}^N F(ch_t^j)} \cdot 100\% \quad (3.37)$$

Zgodnie z wzorem (3.37) osobniki o większym prawdopodobieństwie wylosowania do następnego pokolenia będą mieć większe pole na kole ruletki. Liczba losowań - obrotów kołem ruletki - uzależniona jest od liczebności populacji potomnej. Łatwo można zauważyć, że w tej metodzie, osobnik o największym stopniu przystosowania ma szansę na wielokrotny wybór do populacji rodzicielskiej. Cechę tę zwykle postrzega się za wadę tej metody. Dominacja jednego z osobników może powodować przedwczesną zbieżność do ekstremum lokalnego. Bezsporną zaletą tej metody jest możliwość wprowadzenia do populacji potomnej osobników o najgorszym stopniu przystosowania.

W metodzie selekcji rankingowej następuje sortowanie osobników z populacji rodzicielskiej według wartości funkcji przystosowania. Następnie przypisuje się im wartość rangi (prawdopodobieństwo przetrwania) na podstawie miejsca w populacji, według wzoru:

$$p_s(ch_t^j) = (1 - \alpha)^i \quad (3.38)$$

gdzie i – to jest numer osobnika po sortowaniu, α - współczynnik naporu selekcji.

Współczynnik naporu selekcji może zmieniać się w zakresie $[0, 1]$. Im wartość α jest wyższa, tym szansa na znalezienie się w populacji potomnej osobników o mniejszej wartości funkcji przystosowania maleje. W ten sposób posortowane osobniki tworzą listę rankingową. Najmniejsza wartość rangi jest równa jeden, a największa jest równa liczebności populacji rodzicielskiej. Ranga osobnika staje się argumentem funkcji R w selekcji rankingowej. Za pomocą tej funkcji wyznacza się współczynnik selekcji osobnika. Stanowi on podstawę do obliczenia prawdopodobieństwa jego wyboru do tworzonej populacji rodzicielskiej. Pozycja [120] opisuje w sposób dokładny wykorzystywane funkcje w selekcji rankingowej.

Dla każdego osobnika, który znajduje się na liście rankingowej obliczona jest wartość funkcji selekcji rankingowej R :

$$R'_i = \frac{R_i}{\sum_{j=1}^N R_j} \quad \text{gdzie: } i = \{1, 2, \dots, N\} \quad (3.39)$$

Wyznacza ona prawdopodobieństwo selekcji i -tego osobnika do tworzonej populacji. Osobniki o największych wartościach funkcji rangi mają największe prawdopodobieństwo przetrwania.

Selekcja rangowa, w porównaniu do innych metod, stwarza przestrzeń na przetrwanie osobników o małej wartości funkcji przystosowania. Jest to uznawane jako zaleta tej metody, ponieważ ważnym aspektem jest zachowanie różnorodności osobników w populacji. Dodatkową zaletą jest brak konieczności skalowania w związku z problemem przedwczesnej zbieżności, czy σ -obcinania (brak ujemnych wartości w rankingu), co może wystąpić przy zastosowaniu metody ruletki.

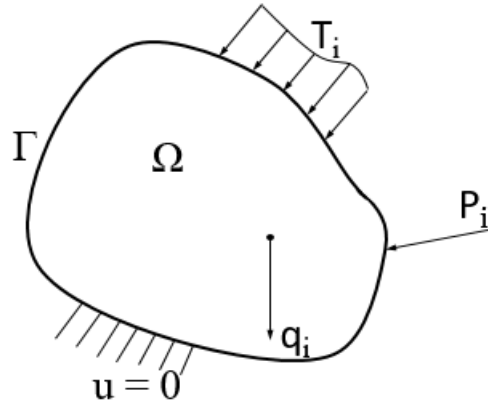
Kolejnym rodzajem selekcji jest selekcja turniejowa [74]. W tej metodzie nie obliczamy wprost prawdopodobieństwa wyboru, jak to ma miejsce dla innych metod selekcji. Proces wyłaniania najlepszych osobników odbywa się na zasadzie turnieju. Polega on na wylosowaniu z populacji (zakres od 1 do μ , możliwe powtórzenia) n -elementowy zbiór osobników, których porównywane są ich wskaźniki przystosowania. Do populacji potomnej przechodzi osobnik o najwyższym wskaźniku przystosowania. Najlepszy osobnik może być wybierany w sposób losowy lub deterministyczny. Proces turnieju oraz identyfikacji zwycięzcy odbywa się iteracyjnie, aż do momentu wypełnienia populacji potomnej. Ilość turniejów oraz ich rozmiar jest liczbą dowolną, jednak często liczba osobników w każdym z turniejów nie przekracza czterech. Parametr n jest nazywany rozmiarem turnieju (ang. *Tournament size*). Przy założeniu, że $n \ll \mu$, co zazwyczaj jest spełnione, koszt obliczeniowy jest niski (co jest za zaletą tej metody). Z uwagi na łatwość implementacji oraz brak konieczności dostępu do całej populacji rodzicielskiej w każdym kroku (ważne dla implementacji równoległych w systemach NUMA) reprodukcja turniejowa zdobyła dużą popularność. Wykazuje również całkowity brak wrażliwości na wartości ujemne, bądź też bardzo wysokie wartości wskaźników przystosowania. Dodatkową zaletą metody selekcji turniejowej, jest to, że może być stosowana zarówno w zadaniach minimalizacji jak i maksymalizacji funkcji celu. Ponadto może być także wykorzystana do optymalizacji wielokryterialnej.

Odmianą reprodukcji turniejowej, jest jej alternatywna odmiana, nazywana turniejem probabilistycznym. W tej metodzie wybór najlepszego osobnika spośród wylosowanych uczestników turnieju, odbywa się przy pomocy procedury reprodukcji proporcjonalnej, a nie deterministycznie jak poprzednio. Powyższa charakterystyka metod selekcji zawiera tylko najpopularniejsze techniki reprodukcji. W literaturze oczywiście można napotkać również inne metody lub modyfikacje powyżej opisanych. Przykładem może być reprodukcja losowa, która jest równoważna selekcji turniejowej, gdy parametr $n = 1$. Nie bierze pod uwagę wartości wskaźnika przystosowania. Osobniki są wybierane całkowicie losowo. Prawdopodobieństwo wyboru każdego osobnika jest takie samo i równe $p_j = \frac{1}{\mu}$. Z kolei selekcja prosta występuje w przypadku, kiedy cała populacja rodzicielska jest przepisywana do populacji potomnej. Jest to możliwe w przypadku, kiedy reprodukcja nie wymaga nacisku selektywnego, a $\mu = \lambda$. Należy wspomnieć jeszcze o selekcji typu (μ, λ) oraz $(\mu + \lambda)$. Należą do klasy metod, w których osobniki wybiera się na zasadzie deterministycznego wyboru – poprzez wybór μ osobników z N chromosomów o największych wartościach funkcji przystosowania. Różnią się podejściem do budowy populacji pośredniej.

Dla selekcji (μ, λ) wybiera się osobniki do populacji rodzicielskiej z populacji pośredniej, która jest identyczna populacji potomnej.

3.7. Podstawy analizy MES układów sprężystych, termosprężystych oraz analizy modalnej

W analizie układów sprężystych wykorzystuje się model ciała odkształcalnego, tzn. zmieniającego swoje wymiary i kształt pod wpływem przyłożonych do niego obciążeń. Zmiana wymiarów oraz kształtów powoduje powstanie pola przemieszczeń u_i . Rys. 3.17 przedstawia model układu sprężysty zajmujący pewien obszar Ω ograniczony brzegiem Γ , na który działają obciążenia w postaci: sił powierzchniowych T_i , sił skupionych P_i oraz sił objętościowych q_i zdefiniowane są przemieszczenia



Rys. 3.17 Model rozpatrywanego ciała

Dla analizowanego modelu ciała, w przestrzeni trójwymiarowej naprężenia wyrażane są za pomocą tensora w postaci macierzy symetrycznej:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \quad (3.40)$$

gdzie:

$\sigma_x, \sigma_y, \sigma_z$ – to są składowe normalne naprężenia

$\tau_{xy}, \tau_{xz}, \tau_{yx}, \tau_{yz}, \tau_{zx}, \tau_{zy}$ – składowe styczne naprężenia

Symetria tensora naprężeń $\boldsymbol{\sigma}$ wynika z równoważności odpowiadających sobie naprężeń stycznych:

$$\tau_{xy} = \tau_{yx}, \tau_{xz} = \tau_{zx}, \tau_{yz} = \tau_{zy} \quad (3.41)$$

Odpowiednio tensor odkształceń $\boldsymbol{\varepsilon}$ definiowany jest następująco:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \varepsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \varepsilon_z \end{bmatrix} \quad (3.42)$$

gdzie:

$\varepsilon_x, \varepsilon_y, \varepsilon_z$ – to są składowe normalne odkształcenia

$\gamma_{xy}, \gamma_{xz}, \gamma_{yx}, \gamma_{yz}, \gamma_{zx}, \gamma_{zy}$ – składowe odkształceń postaciowych

Podobnie jak tensor naprężeń, tensor odkształceń jest również symetryczny, stąd:

$$\gamma_{xy} = \gamma_{yx}, \gamma_{xz} = \gamma_{zx}, \gamma_{yz} = \gamma_{zy} \quad (3.43)$$

W ramach liniowej teorii sprężystości zależność pomiędzy naprężeniami oraz odkształceniami jest wprost proporcjonalna. Opisuje ją uogólnione prawo Hooke'a. [42,58,20], które może być zapisane w postaci sześciu poniższych zależności:

$$\varepsilon_x = \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E} - \nu \frac{\sigma_z}{E} \quad (3.44)$$

$$\varepsilon_y = -\nu \frac{\sigma_x}{E} + \frac{\sigma_y}{E} - \nu \frac{\sigma_z}{E} \quad (3.45)$$

$$\varepsilon_z = -\nu \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E} + \frac{\sigma_z}{E} \quad (3.46)$$

$$\gamma_{xy} = \frac{\tau_{xy}}{G} \quad (3.47)$$

$$\gamma_{yz} = \frac{\tau_{yz}}{G} \quad (3.48)$$

$$\gamma_{xz} = \frac{\tau_{xz}}{G} \quad (3.49)$$

gdzie:

ν – współczynnik Poissona

E – moduł sprężystości wzdłużnej tzw. moduł Younga,

G – moduł sprężystości poprzecznej tzw. moduł Kirchhoffa

Zależności te zapisuje się w następujący sposób:

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \quad (3.50)$$

gdzie: \mathbf{D} jest macierzą sprężystości materiału izotropowego, która w ogólnej postaci jest określona za pomocą dwóch parametrów, tj. modułu Younga oraz współczynnika Poissona:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5-\nu \end{bmatrix} \quad (3.51)$$

Pole temperatury występujące w odkształcalnym ciele stałym powoduje powstanie w nim pola odkształceń, które mogą prowadzić do powstania naprężeń [117]. Naprężenia te mogą wywierać znaczny wpływ na wytrzymałość i trwałość elementów konstrukcji. W wielu przypadkach udział naprężeń termicznych może mieć znacznie większy udział w porównaniu z naprężeniami wywołanymi obciążeniami mechanicznymi. Istotne znaczenie praktyczne naprężeń cieplnych występuje m.in. w takich układach jak: urządzenia energetyczne, silniki spalinowe, urządzenia hutnicze i np. w układach wydechowych. Dla stanu nieustalonego przepływu ciepła [73], w układach mechanicznych zwykle przyjmowany jest model Fouriera, w którym strumień ciepła \mathbf{q} definiowany jest następującym równaniem:

$$\mathbf{q} = -\mathbf{k}\nabla T \quad (3.52)$$

W równaniu tym \mathbf{k} jest tensorem przewodności cieplnej, natomiast ∇T jest gradientem pola temperatury, który w trójwymiarowej przestrzeni występuje w postaci:

$$\nabla T = l_x \frac{\partial T}{\partial x} + l_y \frac{\partial T}{\partial y} + l_z \frac{\partial T}{\partial z} \quad (3.53)$$

gdzie:

l_x, l_y, l_z – to składowe wektora przemieszczenia

Podstawiając równanie (3.52) do równania bilansu energii otrzymuje się równanie różniczkowe nieustalonego przewodzenia ciepła zwanego równaniem Fouriera-Kirchhoffa

$$\nabla \cdot \mathbf{k} \nabla T + Q = \rho c \frac{\partial T}{\partial t} \quad (3.54)$$

gdzie Q – oznacza wydajność wewnętrznych źródeł ciepła, ρ – gęstość, a c – ciepło właściwe.

Równanie (3.54) może przybierać różne formy dla różnych w zależności od typu rozpatrywanego zagadnienia. Wprowadzając współczynnik wyrównywania temperatury zwany także dyfuzyjnością termiczną

$$\alpha = \frac{k}{\rho c} \quad (3.55)$$

dla materiału izotropowego ze stałymi właściwościami termofizycznymi można uzyskać równanie Fouriera- Biota

$$\nabla^2 T + \frac{Q}{k} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (3.56)$$

Przy założeniu braku wewnętrznych źródeł ciepła oraz niezmienności współczynnika przewodności cieplnej uzyskujemy równanie różniczkowe Fouriera

$$\nabla^2 T = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (3.57)$$

Z kolei dla stanu ustalonego przewodzenia ciepła i stałym współczynnikiem przewodności cieplnej otrzymujemy równanie różniczkowe Poissona

$$\nabla^2 T + \frac{Q}{k} = 0 \quad (3.58)$$

a dla ustalonego stanu przewodzenia ciepła oraz braku wewnętrznych źródeł ciepła otrzymujemy równanie różniczkowe Laplace'a

$$\nabla^2 T = 0 \quad (3.59)$$

Równania 3.52.- 3.59 są szeroko opisane w pozycjach [73, 150] dotyczących wymiany ciepła.

Ośłona termiczna zimnej części układu wydechowego pracuje również w podwyższonej temperaturze, która na powierzchni zewnętrznej może się wahać pomiędzy 50-120°C. Sprzężenie pomiędzy polem mechanicznym i termicznym można rozważać jako problem silnie lub słabo sprzężony. W przypadku sprzężenia silnego pola przemieszczeń oraz temperatury są wzajemnie sprzężone. Rozwiązanie problemu uzyskuje się w wyniku rozwiązania sprzężonych równań termosprężystości. [114,115,116,40,56]. Dla sprzężenia słabego pole przemieszczeń zależy od pola temperatury, natomiast pole temperatury jest niezależne od pola przemieszczeń (niesprężona termosprężystość). Uzyskanie

rozwiązania w tego typu zagadnieniach jest możliwe przez rozwiązanie w kolejnych krokach problemu przepływu ciepła, a następnie zagadnień sprężystości. Dla modeli rozważanych w pracy w ramach liniowej teorii termosprężystości, niska temperatura osłony powoduje, iż wpływ rozkładu przemieszczeń na pole temperatury jest znikomy. Przyjęcie założeń niesprężonej termosprężystości jest w zupełności wystarczające i takie podejście jest wykorzystane w pracy.[54]

Ponieważ do symulacji numerycznej zagadnień brzegowych [168] wykorzystano MES, stąd po przeprowadzonej dyskretyzacji oraz wprowadzeniu warunków brzegowych dla ustalonego przepływu ciepła macierzowa forma układu równań algebraicznych przyjmuje postać [22]

$$\mathbf{K}_T \mathbf{T} = \mathbf{Q} \quad (3.60)$$

gdzie: \mathbf{K}_T jest macierzą przewodności cieplnej, \mathbf{T} i \mathbf{Q} odpowiednio są macierzami kolumnowymi węzłowych wartości temperatur oraz strumieni ciepła.

Po rozwiązaniu układu równań (3.60) obliczana jest macierz kolumnowa węzłowych wartości sił wynikających z odkształceń termicznych. Macierz ta dla pojedynczego elementu jest wyznaczana z zależności:

$$\mathbf{F}_T^e = \int_V (\mathbf{B} \mathbf{D} \boldsymbol{\varepsilon}^T) dV \quad (3.61)$$

gdzie \mathbf{D} - jest macierzą sprężystości, a \mathbf{B} macierzą geometryczną. Macierz geometryczna to macierz zawierająca pochodne funkcji kształtu i jest ona zależna od typu elementu skończonego oraz zastosowanych funkcji kształtu.

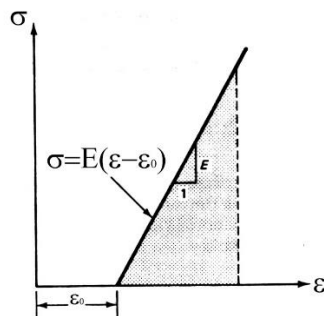
Następnie macierz (3.61) jest uwzględniana jako dodatkowy wektor obciążeń w rozwiązaniu zagadnienia liniowej sprężystości. Macierzowe równanie MES dla liniowej teorii sprężystości przyjmuje wtedy postać:

$$\mathbf{K} \mathbf{U} = \mathbf{F} + \mathbf{F}_T \quad (3.62)$$

gdzie: \mathbf{K} to macierz sztywności, \mathbf{U} , \mathbf{F} i \mathbf{F}_T to odpowiednio węzłowe wektory przemieszczeń, przyłożonych obciążeń mechanicznych oraz obciążeń wynikających z odkształceń termicznych. Globalna macierz sztywności \mathbf{K} powstaje w wyniku agregacji lokalnych macierzy sztywności dla poszczególnych elementów skończonych, która ma postać

$$\mathbf{K}_M^e = \int_V (\mathbf{B}^T \mathbf{D} \mathbf{B}) dV \quad (3.63)$$

Przy wyznaczaniu naprężeń dla problemów termosprężystości należy uwzględnić powstałe odkształcenia termiczne ε_0 , ponieważ modyfikują one zależność odkształcenie-naprężenie jak pokazano to na Rys. 3.18



Rys. 3.18 Wpływ odkształceń termicznych na zależność pomiędzy naprężeniem i odkształceniem.

W pracy kryterium wytrzymałościowe obliczane jest na podstawie naprężeń zastępczych, czyli zredukowanych. Naprężenia te wyznacza się przy użyciu odpowiednich hipotez wyężeniowych. Celem każdej hipotezy wyężeniowej jest zamiana przestrzennego układu naprężeń na jednoosiowy, w którym naprężenie będzie powodowało podobne wyężenie materiału, jak w przestrzennym układzie. Naprężenie zredukowane zwykle oznaczane jest jako σ_{red} . Wyężenie opisywane jest za pomocą funkcji, której argumentami jest tensor stanu naprężenia (3.40) oraz parametry materiałowe.

$$W = F(\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yx}, C) \quad (3.64)$$

gdzie: W – to wyężenie, a C to parametry materiałowe

Naprężenie zredukowane to taka wartość naprężenia, wyznaczona dla danego złożonego stanu naprężenia przy użyciu przyjętej hipotezy wyężeniowej, która przy jednoosiowym rozciąganiu tego samego materiału, wywołałaby identyczne wyężenia jakie ma miejsce w rozpatrywanym stanie naprężenia.

$$\sigma_{red} = f(\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yx}, C) \quad (3.65)$$

$$W = f(\sigma_{red}, C) \quad (3.66)$$

Naprężenia zredukowane często porównujemy z naprężeniami dopuszczalnymi dla danego materiału, wyznaczanymi metodami doświadczalnymi (np. próba rozciągania pręta) w celu określenia współczynnika bezpieczeństwa konstrukcji.

$$\sigma_{red} \leq k_r \quad (3.67)$$

gdzie: k_r – to dopuszczalne naprężenia wyznaczone z próby

W literaturze zostało opisanych wiele różnych hipotez [129,20,58]. Niektóre z nich mają obecnie znaczenie historyczne, jak np. hipoteza Galileusza [113], w której o wyężeniu decyduje wartość maksymalna naprężeń rozciągających. Obecnie bardziej popularne hipotezy użyteczne w analizie materiałów konstrukcyjnych, takich jak stal oraz niektóre stopy to hipotezy największych naprężeń stycznych (Coulomba-Tresca-Guesta) oraz hipoteza energii odkształcenia postaciowego (Hubera-Missesa-Hencky'ego). Wyrażona jest ona następującą zależnością [34]:

$$\sigma_{red} = \frac{1}{\sqrt{2}} \cdot \sqrt{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2)} \quad (3.68)$$

Jak można zauważyć, hipoteza ta nie uwzględnia danych materiałowych a jedynie bazuje na składowych tensora stanu naprężenia. Jest najczęściej implementowaną hipotezą w oprogramowaniu dedykowanym MES. Hipotezę tę wykorzystano w niniejszej pracy.

Częstotliwości drgań własnych modelu oraz jego postaci wyznacza się na podstawie rezultatów analizy modalnej MES. Równania ruchu opisuje się przy pomocy równania Lagrange'a drugiego rodzaju, pochodzące pośrednio od II zasady dynamiki Newton'a:

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{x}} \right) + \frac{\partial E_d}{\partial \dot{x}} + \frac{\partial E_p}{\partial x} = P(t) \quad (3.69)$$

gdzie: E_k – energia kinetyczna układu

E_d – energia tłumienia układu

E_p - energia potencjalna układu

Dla układu o jednym stopniu swobody wzór na energię kinetyczną, energię tłumienia oraz energię potencjalną opisują wzory 3.70 - 3.78

$$E_k = \frac{m\dot{x}^2}{2} \quad (3.70)$$

$$E_d = \frac{c\dot{x}^2}{2} \quad (3.71)$$

$$E_p = \frac{kx^2}{2} \quad (3.72)$$

$$E_k = E_{k1} + E_{k2} \quad (3.73)$$

$$E_d = E_{d1} + E_{d2} \quad (3.74)$$

$$E_p = E_{p1} + E_{p2} \quad (3.75)$$

$$E_k = \frac{m_1\dot{x}_1^2}{2} + \frac{m_2\dot{x}_2^2}{2} \quad (3.76)$$

$$E_d = \frac{c_1\dot{x}_1^2}{2} + \frac{c_2(\dot{x}_2 - \dot{x}_1)^2}{2} \quad (3.77)$$

$$E_p = \frac{k_1x_1^2}{2} + \frac{k_2(x_2 - x_1)^2}{2} \quad (3.78)$$

Na podstawie tych równań tworzony jest układ równań różniczkowych ruchu:

$$\begin{cases} m_1\ddot{x}_1 + [(c_1 + c_2)\dot{x}_1 - c_2\dot{x}_2] + [(k_1 + k_2)x_1 - k_2x_2] = P_1(t) \\ m_2\ddot{x}_2 + [(-c_2)\dot{x}_1 - c_2\dot{x}_2] + [(-k_2)x_1 - k_2x_2] = P_2(t) \end{cases} \quad (3.79)$$

Układ równań (3.76) można zapisać w postaci macierzowej:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} P_1(t) \\ P_2(t) \end{bmatrix} \quad (3.80)$$

lub jako,

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{P}(t) \quad (3.81)$$

W przypadku, kiedy nie występują siły wymuszające, równanie przyjmuje postać:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = 0 \quad (3.82)$$

W ujęciu MES wyrażenie to można zapisać jako:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = 0 \quad (3.83)$$

gdzie, przy założeniu braku tłumienia wzór przyjmuje następującą postać:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{K}\mathbf{U} = 0 \quad (3.84)$$

gdzie \mathbf{M} – globalna macierz bezwładności

- C** - globalna macierz tłumienia
- K** – globalna macierz sztywności
- P(t)** – wektor sił uogólnionych
- \ddot{U} – globalna macierz wektorów przyspieszeń
- \dot{U} – globalna macierz wektorów prędkości
- U** – globalna macierz wektorów przemieszczeń

Popularność oprogramowania wykorzystującego metodę elementów skończonych jest coraz większa. Z każdym rokiem jej użyteczność potwierdzają kolejne dziedziny życia, w których jest stosowana. Symulacje inżynierskie w oprogramowaniu CAE wykorzystujące MES z reguły obejmują trzy podstawowe etapy. Zgodnie ze schematem (Rys. 3.19) rozróżniamy etap przygotowania modelu zwany preprocesingiem, następnie wykonywane są obliczenia MES przy pomocy modułu obliczeniowego zwanego potocznie solverem, a na końcu przeprowadzany jest etap tzw. postprocesingu, w którym następuje opracowywanie uzyskanych wyników.

Preprocessing to pierwszy etap, który odpowiada za przygotowanie modelu do obliczeń. Przygotowania obejmują w pierwszej kolejności przygotowanie modelu geometrycznego, wraz z możliwymi uproszczeniami nie wpływającymi na jakość modelu. Model ten może być opracowany bezpośrednio w modelerze geometrycznym preprocesora, ale częściej jest importowany z popularnych i znacznie wszechstronniejszych pod tym względem systemów CAD (Catia v5, v6, NX Unigraphics, Creo, SolidWorks itp.). Jest to najbardziej pracochłonny etap całego procesu obliczeń. Z reguły preprocesor jest wyposażony w tzw. interfejs użytkownika (GUI), który w znacznym stopniu przyspiesza prace nad modelem. Szczególnie jest to widoczne podczas przygotowywania siatki elementów skończonych i definicji wektora obciążeń. W tym module użytkownik decyduje o wyborze rodzaju, typie elementu skończonego, który będzie wykorzystany w modelu. Jego rodzaj i właściwości mają ogromny wpływ na jakość uzyskanych później wyników. Po stworzeniu modelu, siatki elementów skończonych następuje definicja warunków brzegowych i materiałowych modelu. Przed uruchomieniem obliczeń sprawdzana jest jeszcze poprawność zadeklarowanych danych wejściowych. Ich poprawność kończy fazę preprocessingu.

Moduł obliczeniowy (solver) jest najważniejszym modułem każdego oprogramowania MES. Jest odpowiedzialny za zbudowanie koniecznych macierzy i wektorów bazując na modelu przygotowanym w pierwszej fazie. Rozwiązanie tak przygotowanego macierzowego układu równań pozwala na wyznaczenie poszukiwanych wielkości fizycznych. Oprogramowanie dedykowane do analizy MES może wykorzystywać różne moduły obliczeniowe w zależności od złożoności analizowanego problemu. Ostatnim etapem modelowania z użyciem MES jest postprocesing. Ten etap pozwala nam na szybkie opracowanie uzyskanych wyników. Niejednokrotnie służy temu to samo środowisko użytkownika co w fazie pierwszej, jednakże wykorzystywane jest już inny zakres funkcjonalności GUI. Tworzenie map przemieszczeń, naprężeń czy analiza częstotliwości rezonansowych należy już do użytkownika, który jest odpowiedzialny za ich prawidłową interpretację.

PREPROCESSOR

- Definicja modelu geometrycznego
- Definicja warunków brzegowych
- Wybór rodzaju elementu skończonego
- Tworzenie siatki elementów skończonych

SOLVER

- Budowa globalnej macierzy sztywności
- Budowa wektora obciążeń
- Rozwiązywanie układu równań algebraicznych
- Wyznaczenie symulowanych wielkości (przemieszczenia, naprężenia, częstotliwość drgań własnych itp.)

POSTPROCESSOR

- Analiza - opracowanie wyników obliczeń
- Tworzenie raportów
- Graficzna prezentacja wyników

Rys. 3.19 Ogólna koncepcja obliczeń MES

Dynamiczny rozwój oprogramowania CAD/CAE stwarza dzisiaj możliwość przeprowadzenia całego procesu modelowania i obliczeń w ramach jednego zwartego systemu CAD. Bardzo często można spotkać dosyć zaawansowane implementacje środowiska MES w systemach CAD. Przykładem jest tutaj NX Unigraphics lub Catia v5, gdzie środowisko czysto CAD-owskie jest zintegrowane ze środowiskiem MES. Funkcjonalność modułów MES w środowisku CAD nie jest jednak identyczna z możliwościami oprogramowania ściśle dedykowanych do analizy MES. Bardzo często wystarczają one tylko na poglądowe oszacowanie zaimplementowanych w modelu zmian, pod względem jego funkcjonalności czy przeznaczenia.

W niniejszej pracy do przygotowania modelu parametrycznego jak i późniejszych obliczeń MES zostało wykorzystane oprogramowanie ANSYS. Ponieważ możliwości modelera ANSYSa są ograniczone w stosunku do potrzeb tejże pracy, to model parametryczny opracowano i przygotowano za pomocą skryptu napisanego w języku IronPython w module SpaceClaim. Jest to moduł geometryczny wbudowany w środowisko ANSYS, kompatybilny z pozostałymi modułami przeznaczonymi do obliczeń i opracowywania wyników. Obliczenia wybranych wielkości wytrzymałościowych osłony przeprowadzono w module Mechanical Structure i Modal Analysis. Oprogramowanie ANSYS posiada bogatą bibliotekę elementów skończonych oraz szerokie możliwości definiowania warunków brzegowych. Współpracuje z prawie wszystkimi liczącymi się na rynku systemami CAD, co ułatwia wymianę danych pomiędzy tymi systemami..

3.8. Zastosowanie sztucznych sieci neuronowych w metamodelowaniu przy użyciu MES

3.8.1. Sztuczne sieci neuronowe (SSN)

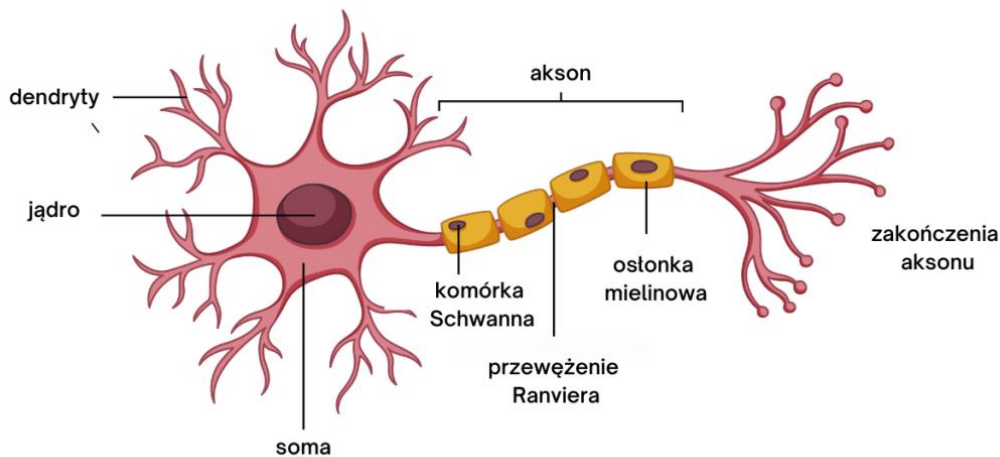
Pierwowzorem wszelkich sieci neuronowych jest mózg ludzki. Sieci neuronowe są nowymi, pracującymi współbieżnie systemami przetwarzania informacji. Są to układy przetwarzające informacje, mające w stosunku do typowych systemów obliczeniowych dwie zasadnicze zalety. Po pierwsze, obliczenia są w sieciach neuronowych wykonywane równolegle. Tysiące sztucznych neuronów składających się na sieć, wykonują przypadające im zadania obliczeniowe równocześnie. Po drugie, sieci nie trzeba programować. Zdefiniowaną strukturę sieci poddaje się procesowi uczenia. Liczne znane obecnie metody uczenia i samouczenia sieci pozwalają uzyskać ich celowe i skuteczne działanie nawet w sytuacji, kiedy twórca sieci nie zna algorytmu, według którego można rozwiązać postawione zadanie. Wymienione zalety spowodowały w ostatnich latach ogromny wzrost zainteresowania tą problematyką. [140].

Początki SNN sięgają połowy XX wieku, gdy w 1943 roku Warrena McCulloch i Waltera Pitts [101] wydali pracę, w której po raz pierwszy przedstawiono model matematyczny komórki nerwowej. Praca ta obejmowała również zasady przetwarzania informacji na podstawie tego modelu. Pierwszym bardzo popularnym przykładem działającej sieci neuronowej jest Perceptron opracowany w 1957 [128].

Interesującą siecią neuronową jest sieć Madaline, skonstruowana przez Bernarda Widrowa w latach sześćdziesiątych XX wieku. Sieć ta składała się z wielu pojedynczych elementów Adaline (ang. *Adaptive linear element*). W latach 70-tych zmalało zainteresowanie sieciami neuronowymi, by ponownie odżyć pod koniec lat 80-tych za sprawą publikacji Jamesa Andersona [3] i Johna Josepha Hopfielda opisującego sieci ze sprzężeniem zwrotnym. Zainteresowanie sieciami neuronowymi ożywiła również praca zredagowana przez McClellanda i Rumelharta na temat równoległego przetwarzania rozproszonego [100]. Od lat 90-tych XX wieku nieprzerwanie po dzień dzisiejszy zainteresowanie sieciami rośnie. Obecnie są one wykorzystywane w wielu dziedzinach. Popularnym stają się tzw. głębokie SSN. W przeciwieństwie do standardowego uczenia maszynowego, głębokie uczenie tego typu sieci wykorzystuje dane nieustrukturyzowane. Skutkuje to tym, że zostaje wyeliminowana duża część wstępnego przetwarzania danych, związana zazwyczaj z uczeniem maszynowym. Ta sieć naśladując ludzki mózg, umożliwia systemom grupowanie danych i dokonywanie dokładnych przewidywań. Możliwe jest wykonywanie zadań takich jak rozpoznawanie mowy, identyfikacja obrazu czy analizowania trendów rynkowych z wykorzystaniem sieci konwolucyjnych czy rekurencyjnych. Odgrywa coraz większą rolę we współczesnych technologiach, takich jak Alexa i Siri. Równie często wykorzystuje się je podczas rozwiązywania złożonych problemów inżynierskich, szczególnie w przypadkach, w których zdefiniowane zadanie trudno jednoznacznie opisać od strony matematycznej [123].

3.8.2. Neuron – opis działania

Podstawowym elementem SSN jest sztuczny neuron, który jest bardzo mocno uproszczoną wersją neuronu ludzkiego. Na Rys. 3.20 pokazano budowę neuronu biologicznego. Podstawowym elementem neuronu biologicznego jest ciało komórki. Do niego dołączone są dendryty i akson z rozgałęzieniami zakończony synapsami. Dendryty odpowiadają za sygnały wejściowe do neuronu, natomiast akson odpowiada za sygnał wyjściowy z neuronu. Sygnał ten jest przekazywany za pomocą synaps, które są połączone z innymi komórkami neuronowymi.



Rys. 3.20 Budowa komórki nerwowej [164]

Pojedynczy neuron nie jest skomplikowany. O sile w przewarżaniu informacji stanowi jego zdolność łączenia się w bardzo skomplikowane struktury połączeń, które licząc miliardy neuronów są zdolne do przetwarzania ogromnych ilości informacji w krótkim czasie. Odpowiednikiem biologicznej komórki nerwowej jest model matematyczny neuronu w SSN. Jak już wspomniano, model ten jest dużym uproszczeniem oryginału. Przetawiono go na Rys. 3.21. Model matematyczny sztucznego neuronu posiada warstwę wejściową, w której zdefiniowane jest n wejść sygnałów. Sygnały te są przemnażane poprzez odpowiednio dobrane wagi w_i . Następnie następuje zsumowanie wszystkich sygnałów i podawany jest dodatkowy sygnał (ang. *bias*), stanowiący skalarne przesunięcie. Uzyskana suma przekazywana jest do funkcji aktywacji, która wyznacza wartość wyjścia z neuronu. Matematycznie dla pojedynczego neuronu można to zapisać jako:

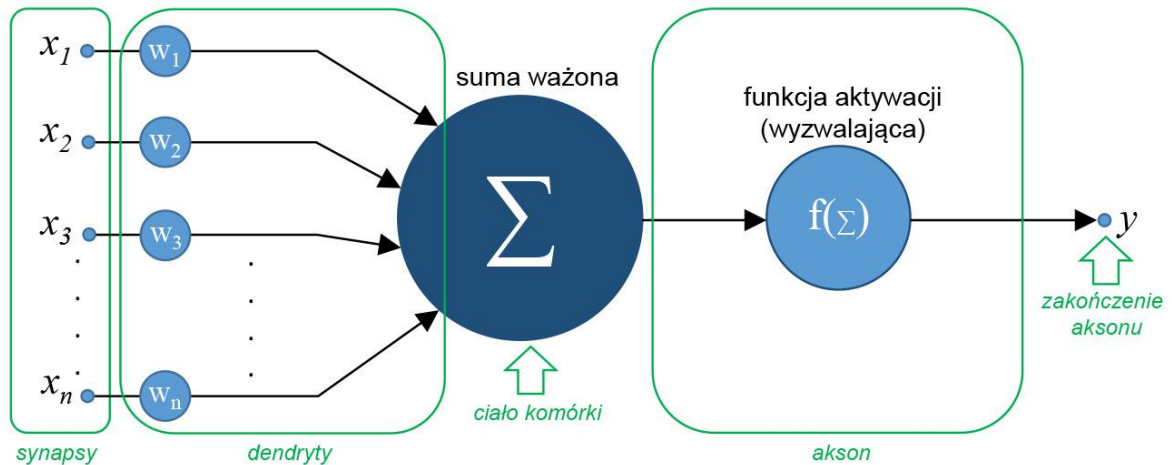
$$y = f_{\varphi}(s) \quad (3.85)$$

$$s = \sum_{i=1}^n x_i w_i + b \quad (3.86)$$

gdzie:

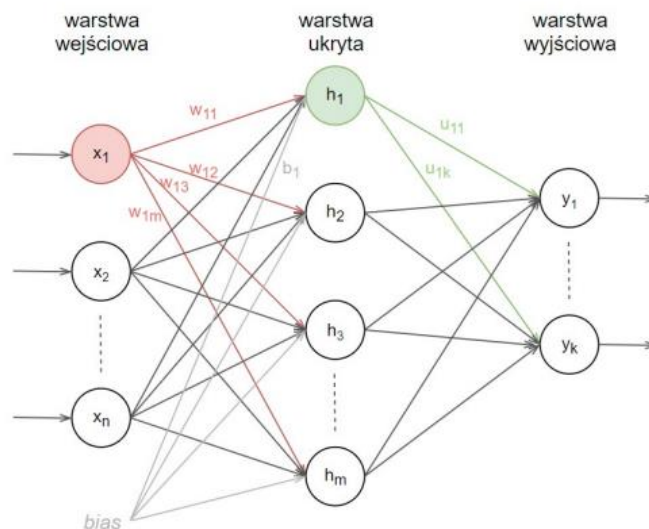
- y – wartość wyjściowa z neuronu
- f_{φ} – funkcja aktywacji
- s – suma ważona wejść
- x_i – wartość i -tego wejścia
- w_i – wartość i -tej wagi
- b – skalarne przesunięcie (*bias*)

Pojedynczy neuron podobnie jak pojedyncza komórka nerwowa ma ograniczone możliwości. Dlatego, do rozwiązywania złożonych problemów technicznych pojedyncze neurony składa się w dowolne struktury. W ten sposób od prostego pojedynczego perceptronu przechodzimy poprzez sieć jednowarstwową do jej rozbudowanej wersji w postaci sieci wielowarstwowej tzw. MLP (Multiperceptron).



Rys. 3.21 Model matematyczny neuronu [163]

Poniżej na Rys. 3.22 pokazano przykładową strukturę dwuwarstwowej sztucznej sieci neuronowej. Obejmuje ona warstwę sygnałów wejściowych (ang. *input layer*), jedną warstwę ukrytą (ang. *hidden layer*) oraz warstwę wyjściową (ang. *output layer*).

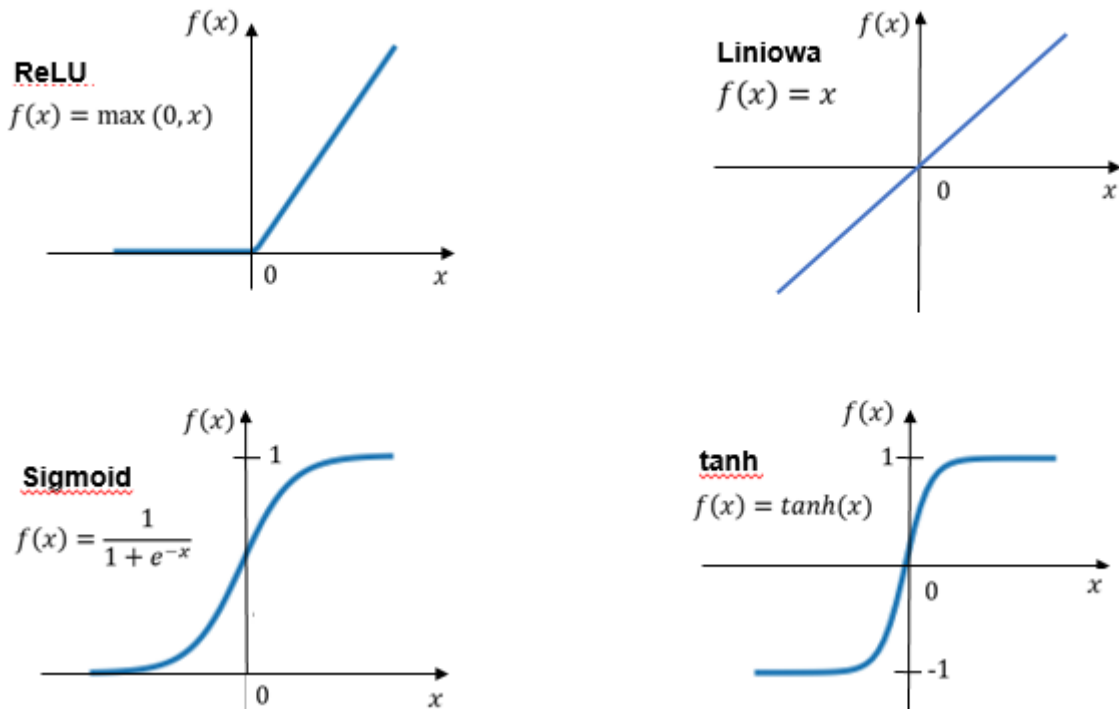


Rys. 3.22 Architektura 2-warstwowej jednokierunkowej sieci neuronowej

W takiej strukturze wyjścia jednej warstwy stanowią wejścia warstwy następnej. Warstwa wejściowa przeważnie nie jest numerowana, dlatego sieć z jedną warstwą ukrytą i warstwą wyjściową zwykle określona jest jako 2-warstwowa. Działanie takiej sieci polega na zadaniu na wejściu wektora wartości \mathbf{x} , który obejmuje wszystkie sygnały wejściowe. Każda wartość tego wektora przekazywana jest do każdego neuronu w warstwie ukrytej (ang. *hidden layer*). Następnie wektor \mathbf{x} jest przemnażany z wektorem wag \mathbf{w} i sumowany z wektorem skalarnych przesunięć \mathbf{b} (ang. *bias*). Otrzymany wynik podawany jest do funkcji aktywacji f (ang. *activation function*) i przekazywany do warstwy wyjściowej. W warstwie wyjściowej wektor wyjść z warstwy ukrytej jest przemnażany przez wektor wag \mathbf{u} i przetwarzany przez funkcję aktywacji g . Wyjście y' całej przykładowej sieci neuronowej z Rys. 3.22. opisuje wzór 3.87:

$$y' = g\left(\mathbf{U}(f(\mathbf{W}\mathbf{x} + \mathbf{b}))\right) \quad (3.87)$$

gdzie, \mathbf{x} jest wektorem sygnałów wejściowych, \mathbf{W} i \mathbf{U} to macierze wag dla odpowiednio warstwy ukrytej i warstwy wyjściowej, a \mathbf{b} to wektor przesunięć warstwy ukrytej. Celem funkcji aktywacji jest wprowadzenie do sieci nieliniowości i mapowanie wartości na wyjściu do ograniczanego przedziału, najczęściej od -1 lub 0 do 1. Rodzaj funkcji aktywacji użytkownik deklaruje podczas projektowania sieci. Do najpopularniejszych z nich zaliczamy funkcję sigmoidalną, funkcję liniowo rektyfikowaną (ang. ReLU - *Rectified Linear Unit*) oraz funkcję tangensa hiperbolicznego (tanh). Na Rys. 3.23 pokazano wykresy i ich wzory.



Rys. 3.23 Przykładowe funkcje aktywacji neuronu w SSN

3.8.3. Metamodel z wykorzystaniem MES

Wykorzystanie SNN w metamodelowaniu wymaga od użytkownika przestrzegania pewnych reguł. Algorytm zastosowania SSN w praktyce zwykle rozpoczyna się od przygotowania zbioru danych wejściowych. Zbiór ten z reguły zawiera pewnego rodzaju niepożądane dane, określane jako szum, który zakłóca etap uczenia sieci i należy go wyeliminować. Ponadto, zbiór danych uczących musi być zbiorem reprezentatywnym. Powinien zawierać dane odpowiadające statystycznej strukturze populacji, którą reprezentuje. Często, ze względu na dużą różnicę zakresów zmienności danych wejściowych, dane ze zbioru poddaje się normalizacji, tzn. sprowadza się je do zdefiniowanego zakresu, przeważnie jest to zakres $[0,1]$ lub $[-1, -1]$ [109]. Tak przygotowany zbiór dzielimy na trzy podzbiory, treningowy, testowy i walidacyjny w odpowiednich proporcjach np. 70%, 15%, 15%. Dane ze zbioru treningowego służą do uczenia sieci. Celem uczenia jest generalizacja. Dążymy do minimalizacji błędów dla danych, które nie zostały użyte w treningu. Zbiór danych walidacyjnych wykorzystywany jest do oceny postępu uczenia się sieci. Pozwala to na ocenę błędów generalizacji i jest przydatne przy określaniu punktu zatrzymania

treningu. Natomiast dane ze zbioru testowego, tzn. dane, które nie brały udziału w procesie uczenia sieci, służą do końcowej oceny jakości sieci. Tak nauczona sieć powinna przewidywać, z minimalnym błędem, prawidłową odpowiedź modelowanego procesu na dowolne sygnały wejściowe. W kolejnym kroku należy zaprojektować strukturę sieci. Dla wielowarstwowego perceptronu strukturę SSN definiujemy przy pomocy kilku parametrów sieci. Zaliczamy do nich ilość warstw ukrytych, ilość neuronów w każdej warstwie, rodzaj funkcji aktywacji dla każdego neuronu (warstwy), wielkość biasu czy sposób inicjalizacji wag.

Ze względu na kierunek przepływu sygnałów rozróżniamy dwa typy architektury sieci. Jednym z nich są sieci jednokierunkowe (ang. *feedforwarded*) to znaczy sieci o jednym kierunku przepływu sygnałów. Zdecydowanie najpopularniejszą topologią dla tej architektury sieci są sieci warstwowe. Drugim natomiast są sieci rekurencyjne (*feedback, bidirectional*) to znaczy sieci ze sprzężeniem zwrotnym. Przykładem tutaj może być np. sieć Hopfielda lub sieci uczenia się przez współzawodnictwo np. Kohonena.

Dla zapewnienia odpowiedniego sygnału wyjściowego sztuczny neuron powinien, biorąc wzór z jego biologicznego pierwowzoru, prawidłowo reagować na sygnały wejściowe. Umiejętność tę zdobywa w procesie uczenia, którego celem jest odpowiednie dopasowanie poszczególnych wag w_i przypisanych do każdego sygnału wejściowego, by nauczony neuron prawidłowo, zgodnie z oczekiwaniami, odpowiadał na zadany wektor sygnałów wejściowych.

Istnieją dwie, zupełnie odmienne strategie uczenia sieci. Pierwszą jest tzw. uczenie nadzorowane (z nauczycielem). Proces uczenia nadzorowanego polega na przyjęciu początkowych wartości wag w_i . W kolejnym kroku na wejście neuronu podawany jest wektor sygnałów wejściowych \mathbf{x} ze zbioru danych uczących U . Zbiór ten składa się z odpowiednio przygotowanych par wektorów. Daną parę tworzą wektor sygnałów wejściowych \mathbf{x} oraz odpowiadający im wektor sygnałów wyjściowych o oczekiwanych wartościach d , czyli

$$U = (\mathbf{x}^1, d^1), (\mathbf{x}^2, d^2), \dots, (\mathbf{x}^m, d^m) \quad (3.88)$$

W rezultacie na każdy wektor wejściowy neuronu \mathbf{x}^i otrzymujemy sygnał wyjściowy neuronu y^i . Sygnał ten jest porównywany z wartością oczekiwaną sygnału uczącego d^i . Przy odpowiednio dobranych wagach synaptycznych, błąd $\delta^i = d^i - y^i$ między tymi dwoma sygnałami jest bliski zeru. Minimalizacja błędu δ^i jest podstawowym celem procesu uczenia, a wagi synaptyczne neuronu, jako zmienne optymalizacyjne są modyfikowane w trakcie procesu uczenia.

Drugą strategią uczenia, zupełnie odmienną od poprzedniej, jest uczenie nienadzorowane zwane potocznie bez nauczyciela. Stosujemy ją w przypadkach, dla których nie dysponujemy zbiorem danych uczących. W strategii tej nie znamy wartości oczekiwanej d sygnału wyjściowego neuronu, dlatego nie jest możliwe określenie błędu uczenia δ . Jako podstawę tej strategii wykorzystuje się regułę Hebba [77]. Działanie reguły polega na wzmacnianiu połączeń silnych sygnałów i osłabianiu innych połączeń. Sygnał wyjściowy y neuronu jest sygnałem uczącym. Współczynnik wagowy jest korygowany proporcjonalnie do iloczynu sygnału wejściowego i wyjściowego, czyli $\Delta w_i = \eta x_i y$. Reguła ta nazywana jest często korelacyjną, ponieważ dodatnia wartość współczynnika korelacji pomiędzy sygnałem wejściowym a sygnałem wyjściowym skutkuje wzrostem wagi, a w rezultacie, mocniejszą reakcją neuronu w odpowiedzi na ten sam sygnał wejściowy x_i .

Proces uczenia sieci odbywa się zgodnie z wybranym algorytmem uczenia. Najczęściej stosowanym, w przypadku uczenia z nauczycielem, jest tzw. algorytm wstecznej propagacji błędu (*backpropagation*). Podobnie jak to było w przypadku pojedynczego neuronu wyznaczany jest błąd warstwy wyjściowej sieci. Na jego podstawie obliczane są wartości błędów warstw poprzedzających. To znaczy sygnał błędu przesuwany się wstecz od warstwy wyjściowej aż do warstwy wejściowej. Na podstawie tak określonych

błędów następuje korekcja wag wszystkich połączeń wewnątrz sieci. Sposób, w jaki algorytm koryguje poszczególne współczynniki wag opiera się na gradientowych metodach optymalizacji. Wybór ten ma wpływ na szybkość uczenia oraz wynik końcowy. Na potrzeby tej pracy wybrano algorytm LM (Levenberg-Marquardt) [69], którego pseudo kod opisany został w rozdziale 3.4. Algorytm ten został opracowany na początku lat 60 XX wieku w celu rozwiązywania nieliniowych problemów metodą najmniejszych kwadratów. Algorytm Levenberga-Marquardta łączy w sobie dwa algorytmy minimalizacji numerycznej: Cauchy'ego, wykorzystującą metodę najszybszego spadku gradientu i metodę Gaussa-Newtona. Adaptacyjnie przełącza się pomiędzy metodą najszybszego spadku gradientu a metodą Gaussa-Newtona. Metodę najszybszego spadku gradientu wykorzystuje, gdy parametry są dalekie od ich optymalnych wartości, a metodę Gaussa-Newtona, gdy parametry są bliskie ich optymalnym wartościom.

SSN uznajemy jako sieć nauczoną, kiedy wartość błędu MSE (Mean Squared Error) lub RMSE (Root Mean Squared Error) mieści się w określonym zakresie, zdefiniowanym na początku procesu uczenia. Tak przygotowaną sieć można użyć do budowy metamodelu. Szczegółowo proces metamodelowania dla rozpatrywanego w pracy zagadnienia został opisany w rozdziale 4.6.

3.9. Zastosowane algorytmy optymalizacji wbudowane w oprogramowanie MES

W ramach pracy wykorzystano zarówno algorytmy optymalizacji jednokryterialnej jak i algorytmy optymalizacji wielokryterialnej wbudowane w oprogramowanie MES. Zastosowano pięć algorytmów. Trzy z nich wykorzystane są w optymalizacji jednokryterialnej oraz dwa w optymalizacji wielokryterialnej. Do pierwszej grupy należy algorytm NLQPL (*Nonlinear Programming by Quadratic Lagrangian*), MISQP (*Mixed-Integer Sequential Quadratic Programming*) oraz ASO (*Adaptive Single Optimization*). Do drugiej natomiast algorytm MOGA (*Multi Objective Genetic Algorithm*), NSGA-II (*Non-dominated Sorting Genetic Algorithm*).

Ogólnie problem optymalizacji dla algorytmów jednokryterialnych, dla przypadku minimalizacji funkcji celu f przy nieliniowych ograniczeniach równościowych i nierównościowych został opisany w rozdziale 3.2 za pomocą wzorów 3.8 do 3.10. W przypadku optymalizacji wielokryterialnej problem minimalizacji funkcji celu f (kryteria 3.110, 3.111) oraz maksymalizacji (kryterium 3.112) został opisany ogólnie za pomocą wzorów 3.14 do 3.17.

Wszystkie użyte algorytmy w procesie optymalizacji - na cele tej rozprawy – zostały opisane w tym rozdziale.

3.9.1. Jednokryterialny algorytm NLPQL

NLPQL (Nonlinear Programming by Quadratic Lagrangian) to algorytm optymalizacji opracowany przez Klausa Schittkowskiego. Przeznaczony jest do rozwiązywania problemów programowania nieliniowego z ograniczeniami. Algorytm wymaga, aby funkcja celu i ograniczenia były różniczkowalne w sposób ciągły. NLPQL jest implementacją FORTRAN-owską metody sekwencyjnego programowania kwadratowego służącej do rozwiązywania nieliniowo ograniczonych problemów optymalizacyjnych z różniczkowalnymi funkcjami celu. [125]. W każdej iteracji kierunkiem poszukiwań jest rozwiązanie podproblemu programowania kwadratowego. Idea algorytmu polega na wygenerowaniu sekwencji podproblemów programowania kwadratowego uzyskanych przez kwadratowe przybliżenie funkcji Lagrange'a i linearyzację ograniczeń. Informacje drugiego rzędu są aktualizowane za pomocą formuły quasi-Newtona. Rozmiar zdefiniowanego problemu powinien być dobrze skalowalny i nie powinien być zbyt duży, a pochodne funkcji celu (gradienty) są obliczane numerycznie

Techniki programowania nieliniowego za pomocą metody NLPQL są szeroko stosowane do rozwiązywania rzeczywistych problemów optymalizacyjnych, również w mechanice konstrukcji. Mimo to, techniki NLPQL są podatne na błędy w parametrach i ocenach gradientów. Zazwyczaj otrzymanie rozwiązania przy większej liczbie wywołań symulacji zajmuje więcej czasu. W przypadku zaszumionych wartości funkcji znaczące zwiększenie wydajności można uzyskać poprzez adaptacyjne programowanie nieliniowe metodą kwadratowego lagrangianu w wersji A-NLPQL.

3.9.2. Jednokryterialny algorytm MISQP

MISQP – (Mixed-Integer Sequential Quadratic Programming) - to algorytm optymalizacji matematycznej opracowany przez Olivera Exlera, Thomasa Lehmana i Klause Schittkowskiego. MISQP rozwiązuje problemy programowania nieliniowego na liczbach całkowitych i rzeczywistych za pomocą zmodyfikowanej metody sekwencyjnego programowania kwadratowego (SQP). Funkcje celu są oceniane tylko w punktach dla wartości całkowitych. Przy założeniu, że zmienne całkowite mają łagodny wpływ na funkcje modelu, tj. że wartości funkcji nie zmieniają się drastycznie podczas zwiększania lub zmniejszania wartości całkowitej zmiennej, stosuje się kolejne przybliżenia kwadratowe. Algorytm jest stabilizowany metodą regionów zaufania, w tym poprawek Yuana drugiego rzędu. [165] Kod ma zastosowanie również do problemów optymalizacji niewypukłej. Hessian funkcji Lagrange'a jest aproksymowany przez aktualizacje algorytmu BFGS (Broyden, Fletcher, Goldfarb, and Shanno) dla zmiennych ciągłych i całkowitych. Wspomniane metody regionów zaufania zostały wynalezione dawno temu jako jedne z pierwszych dla optymalizacji bez ograniczeń, zwłaszcza dla optymalizacji metodą najmniejszych kwadratów, jak np. metoda Powell [121], lub Moré [107]. Dodatkowe modyfikacje tego podejścia zostały opracowane w zakresie nieciągłej optymalizacji, patrz Fletcher [63], a w przypadku optymalizacji z ograniczeniami Celis [41], Powell i Yuan [121], Byrd [39], Toint [146] i wielu innych. Szeroki przegląd metod regionu zaufania jest opisany przez Conna, Goulda i Tointa [45].

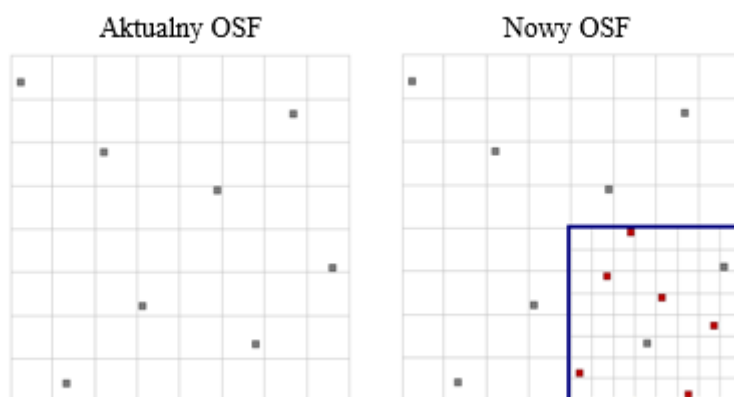
Ta metoda jest przeznaczona przede wszystkim do problemów o niewielkiej liczbie zmiennych, w których ocena funkcji celu wymaga długich obliczeń, a jej pochodne mogą nie być dostępne. W odróżnieniu do innych algorytmów ta metoda może być stosowana dla modeli, w których nie można zrelaksować zmiennych całkowitych tzn., że można traktować zmienne całkowite jako zmienne ciągłe, tj. wartości funkcji można również obliczać pomiędzy kolejnymi punktami całkowitymi. Skutkuje to tym, że symulacje lub oceny funkcji zachodzą tylko wtedy, gdy zmienne mają wartości całkowite. Ponieważ działanie tego algorytmu można porównać do działania algorytmu SQP, można do metody MISQP zastosować funkcję równoległego multi-startu, aby zmaksymalizować prawdopodobieństwo znalezienia globalnego rozwiązania.

Z drugiej strony sekwencyjne programowanie kwadratowe lub metody SQP należą do najczęściej stosowanych przez algorytmy do rozwiązywania praktycznych problemów optymalizacyjnych. Informacje na ten temat można znaleźć w pracach Stoera [137], Fletchera [62] lub Gilla [71]. Jednak sytuacja staje się znacznie bardziej złożona, jeśli należy wziąć pod uwagę dodatkowe zmienne całkowite. Zazwyczaj podejścia te wymagają wypukłych funkcji i ciągłej relaksacji zmiennych liczb całkowitych.

3.9.3. Jednokryterialny algorytm ASO

Adaptive Single-Objective (ASO) to matematyczna metoda optymalizacji, która łączy DOE (Design of Experiments), OSF (*Optimal Space-Filling*), powierzchnię odpowiedzi *Kriging* i wcześniej opisany MISQP. Jest to algorytm oparty na gradiencie funkcji celu oraz na powierzchni odpowiedzi.

Tok postępowania sprowadza się do wygenerowania próbki rozwiązań OSF, którą wykorzystuje się w konstrukcji metody *Kriging*. W oryginalnym OSF liczba wygenerowanych rozwiązań jest równa liczbie podziałów na oś, a na każdy podział przypada jedno rozwiązanie, co pokazano na Rys. 3.24.



Rys. 3.24 Metoda OSF doboru punktów konstrukcyjnych

Alternatywą do OSF jest metoda wyznaczania rozpatrywanych rozwiązań za pomocą algorytmu LHS (*Latin Hypercube Sampling*).[104] Zredukowany OSF ma taką samą liczbę podziałów jak oryginał i utrzymuje istniejące punkty projektowe w nowych granicach. Nowe punkty projektowe są dodawane, dopóki, w każdym podziale zredukowanej dziedziny znajdzie się punkt. W przykładzie przedstawionym na Rys.3.24 zarówno oryginalna jak i zredukowana domena mają osiem podziałów na oś i zawierają osiem punktów projektowych. Aby punkt projektowy występował w każdym podziale, należy dodać sześć nowych punktów projektowych. Następnie dla każdego wyjścia pokazanego dla aktualnego rozkładu OSF tzn. w bieżącym obszarze, stosowana jest metoda odpowiedzi powierzchni RSM (*Response Surface Method*). Dla powierzchni odpowiedzi w celu znalezienia rozwiązania reprezentatywnego wykorzystana jest metoda Krigingu. Kilka procesów MISQP jest uruchamianych w tym samym czasie, z różnymi punktami początkowymi, co w efekcie powoduje wygenerowanie różnych rozwiązań. Wszystkie te rozwiązania są oceniane na podstawie tzw. estymatora błędu Kriginga. Każde rozwiązanie jest poddane ocenie, aby sprawdzić, czy dalsze udoskonalanie powierzchni odpowiedzi wyeliminuje go czy nie. Jeśli jakość rozwiązania jest niska, granice obszaru są zmniejszane. Jeśli obszar wyboru się nie zmienił, to obliczany jest nowy punkt weryfikacji. Jest on wstawiany do bieżącej powierzchni odpowiedzi metodą Krigingu, jako punkt udoskonalania i proces MISQP jest uruchamiany ponownie. Po sprawdzeniu poprawności rozwiązań oblicza się nowe granice obszaru. Jeśli wszystkie rozwiązania znajdują się w tej samej strefie, granice są zwiększane. W przeciwnym razie granice są zmniejszane obejmujące wszystkie rozwiązania. Przy każdej redukcji obszaru generowany jest nowy OSF (z zachowaniem punktów projektowych między nowymi granicami) i generowana jest nowa powierzchnia odpowiedzi na podstawie tego nowego OSF.

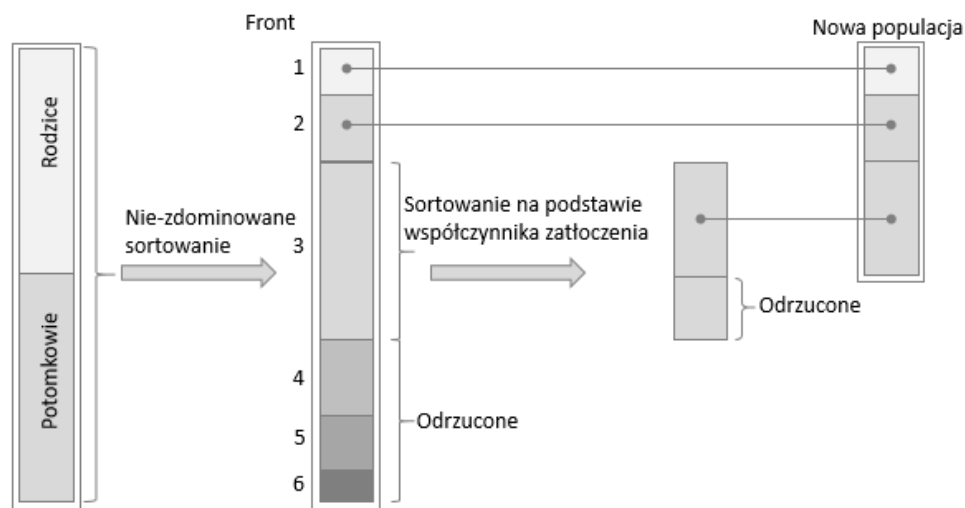
3.9.4. Wielokryterialny algorytm NSGA-II

NSGA (Non-dominated Sorting Genetic Algorithm) [135] jest popularnym algorytmem genetycznym opartym o procedurę niezdominowanego sortowania. Jego zmodyfikowaną odmianą jest algorytm NSGA-II [50], który ma lepszy algorytm sortowania, zawiera elitarną metodę selekcji oraz posiada pewne modyfikacje w porównaniu do swojego pierwowzoru. W algorytmie tym, po zainicjowaniu populacji startowej, populacja ta jest sortowana w oparciu o niezdominowane rozwiązania w każdym froncie. Pierwszy front staje się frontem rozwiązań całkowicie

niezdominowanych w tejże populacji, drugim frontem jest front zdominowanych osobników przez jednostki z pierwszego frontu. Każdemu osobnikowi w każdym froncie przypisywana jest wartość rangi według frontu, do którego należą. Osobniki w pierwszym froncie otrzymują wartość rangi równą 1 i osobnikom w drugim froncie przypisywana jest wartość rangi równą 2. Kolejne fronty otrzymują odpowiednio kolejne wartości.

Oprócz wartości rangi dla każdego osobnika obliczany jest parametr zwany współczynnikiem zatłoczenia. Współczynnik zatłoczenia jest miarą tego, jak blisko dany osobnik znajduje się w przestrzeni rozwiązań od swoich sąsiadów. Duża średnia odległość zatłoczenia powoduje lepszą różnorodność w populacji. Szczegółowo mechanizm działania operatora zatłoczenia opisano w ramach algorytmu MOOPTIM.

Chromosomy rodzicielskie są wybierane z populacji za pomocą binarnej selekcji turniejowej, co odróżnia go od algorytmu MOOPTIM, na podstawie rangi i współczynnika zatłoczenia. Zostaje wybrany ten osobnik, który posiada niższą rangę w porównaniu z pozostałymi lub jeśli współczynnik zatłoczenia jest większy od pozostałych. Następnie na wybranej populacji stosuje się operatory krzyżowania i mutacji, celem wygenerowania populacji potomnej. Po połączeniu populacji aktualnej oraz potomnej, nowo utworzona populacja jest poddana algorytmowi sortowania niezdominowanego. Z niej to wybieranych jest w kolejnym kroku tylko N najlepszych osobników (gdzie N jest wielkością populacji). Wybór opiera się na randze i współczynniku zatłoczenia na ostatnim froncie (Rys. 3.25).



Rys. 3.25 Schemat algorytmu NSGA-II

Działanie algorytmu postępowania dla NSGA-II można przedstawić w następujących punktach:

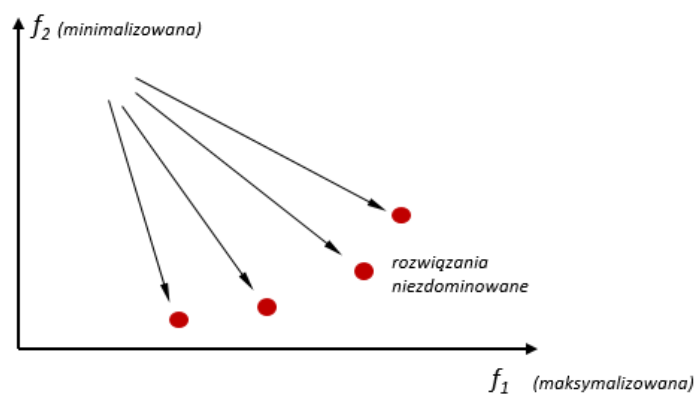
- Inicjalizacja populacji. Populacja jest inicjowana na podstawie zakresu problemu i ewentualnych ograniczeń.
- Sortowanie niezdominowane. Zainicjowana populacja jest poddana algorytmowi sortowania niezdominowanego. Algorytm niezdominowanego sortowania jest opisany w pracy [50]
- Obliczanie współczynnika zatłoczenia. Po zakończeniu sortowania niezdominowanego następuje obliczanie współczynnika zatłoczenia. Ponieważ selekcja bazuje na wartości rangi i współczynniku zatłoczenia, wszystkim osobnikom w populacji przypisywana jest jego wartość. Współczynnik zatłoczenia jest przypisywany dla całego frontu i porównywanie go między dwoma osobnikami w różnych frontach nie ma istotnego znaczenia. Podstawową ideą wyznaczania współczynnika zatłoczenia jest znalezienie odległości Euklidesowej pomiędzy każdym osobnikiem frontu w oparciu o ich m funkcji celu i w n -wymiarowej przestrzeni rozwiązań. Osobniki skrajne, na granicy

frontu, są zawsze wybierane, ponieważ tym osobnikom z definicji przypisano nieskończoną wartość współczynnika zatłoczenia.

- d) Selekcja. Do sortowania osobników wykorzystuje się procedurę sortowania niezdominowanego. Wybór osobników następuje po przypisaniu każdemu z nich wartości współczynnika zatłoczenia. Osobniki są wybierane za pomocą binarnej selekcji turniejowej wykorzystującej operator porównania współczynnika zatłoczenia (*crowdcomparison-operator*).
- e) Operatory genetyczne. Oryginalne algorytmy genetyczne GA używają (SBX-*Simulated Binary Crossover*) [87,28] operatora dla operacji krzyżowania i mutacji wielomianowych [87,124].
- f) Rekombinacja i selekcja. Populację potomną łączy się z populacją obecnego pokolenia. Selekcję przeprowadza się w celu ustalenia osobników następnej generacji. Zapewnienie elitaryzmu wyboru gwarantuje umieszczenie wszystkich poprzednich i obecnych najlepszych osobników w połączonej populacji. Populacja ta następnie poddana jest algorytmowi sortowania niezdominowanego. Nowa generacja jest wypełniana sukcesywnie przez każdy front kolejno, aż do wielkości, gdy liczba osobników z kolejnego frontu, po ich dodaniu przekroczyłaby wielkość populacji aktualnej. Jeżeli dodając kolejne osobniki z frontu F_j populacja przekroczyłaby N , to osobniki z tego frontu F_j są wybierane na podstawie ich współczynnika zatłoczenia w kolejności malejącej, aż do momentu, gdy wielkość populacji wyniesie N (Rys. 3.25). Proces powtarza się, generując kolejne pokolenia.

3.9.5. Wielokryterialny algorytm MOGA

MOGA (*Multi-Objective Genetic Algorithm*) [133] algorytm wykorzystuje zmienny kierunek poszukiwań jak pokazano to na Rys. 3.26. Stosuje jednokryterialną wersję algorytmu ewolucyjnego opisaną w rozdziale 3.10.2 i „skalaryzuje” problem optymalizacji wielokryterialnej do problemu optymalizacji jednokryterialnej. Podczas procedury selekcji, ten algorytm używa sumy wag wszystkich kryteriów i przetwarza je w jednowymiarową funkcję celu. Charakterystyczną rzeczą tego algorytmu jest to, że zaimplementowane wagi dla każdego kryterium nie są stałe, lecz zmienne, losowo wyznaczone dla każdej selekcji [104]. Dlatego też kierunek poszukiwań w algorytmie MOGA jest zmienny [110]. Podczas działania, algorytm MOGA zachowuje tymczasowy zestaw optymalnych rozwiązań Pareto. Pewna liczba osobników w tym zbiorze jest dziedziczona do następnego pokolenia jako jednostki elitarne.



Rys. 3.26 Kierunek poszukiwań w wielokryterialnym algorytmie optymalizacyjnym MOGA

Operacje genetyczne w algorytmie MOGA obejmują również etap elitarniej selekcji. Jednym ze sposobów przekształcenia funkcji wielu kryteriów w jedną skalarną funkcję jest metoda sum ważonych. W przypadku, gdy wszystkie kryteria będą maksymalizowane, wtedy

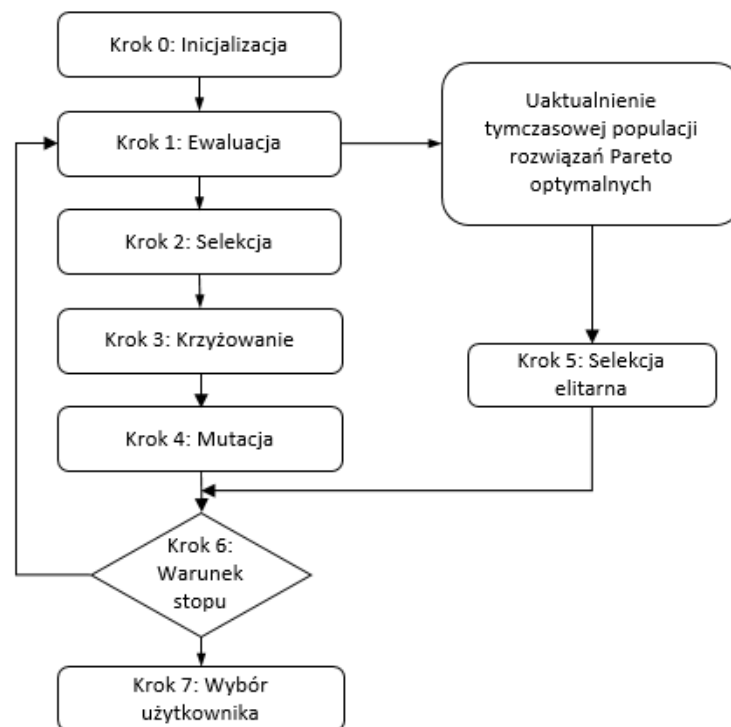
$$f(x) = w_1 \cdot f_1(x) + \dots + w_i \cdot f_i(x) + \dots + w_n \cdot f_n(x) \quad (3.89)$$

gdzie: x to jest osobnik z populacji,
 $f(x)$ jest połączoną funkcją celu,
 $f_i(x)$ jest i -tą funkcją kryterialną,
 w_i – jest stałą wagową dla $f_i(x)$, oraz n jest liczba kryteriów.

Dla stałych wartości wag w_i , kierunek poszukiwan w algorytmie genetycznym jest ściśle określony jak to pokazuje (Rys. 3.26). MOGA stosuje procedurę selekcji rozwiązań pareto optymalnych ze zmiennymi losowo wagami by kierunek poszukiwan mógł być też zmienny. W momencie kiedy para osobników jest wybrana do operacji krzyżowania następuje przypisanie losowo wygenerowanych rzeczywistych wartości wag wg poniższego wzoru:

$$w_i = \frac{random_i(\cdot)}{\sum_{j=1}^n random_j(\cdot)} \quad i = 1, 2, \dots, n, \quad (3.90)$$

gdzie $random_j(\cdot)$ jest dodatnia losowo wyslosowaną wartością. Z równania 3.92 wynika, że w_i jest liczbą rzeczywistą z zakresu $[0,1]$. Dla kolejnej pary wybranych osobników zostaną wygenerowane nowe wagi, inne niż da poprzedników wg wzoru (3.90). W kolejnym kroku algorytmu MOGA, z każdą iteracją, jest generowana i uaktualniana populacja próbna rozwiązań Pareto optymalnych. Pewna liczba osobników określona jako N_{elite} jest losowo wybierana z każdej populacji. Te osobniki są traktowane jako osobniki elitarne. Ta elitarna strategia utrzymywania grupy osobników ma duży wpływ na utrzymanie zmienności w każdej populacji generowanej przez MOGA. Rys. 3.27 przedstawia schemat blokowy algorytmu MOGA.



Rys. 3.27 Schemat blokowy algorytmu MOGA

3.10. Zewnętrzne algorytmy optymalizacji (niewbudowane w oprogramowanie MES)

W przypadku algorytmów zewnętrznych wykorzystano podobnie jak opisano w poprzednim rozdziale zarówno algorytmy optymalizacji jednokryterialnej jak i algorytmy optymalizacji wielokryterialnej. Wykorzystano cztery algorytmy. Trzy z nich wykorzystane są w optymalizacji jednokryterialnej oraz jeden w optymalizacji wielokryterialnej. Do grupy algorytmów optymalizacji jednokryterialnej należy algorytm Nelder-Meada, algorytm oparty o metodę ewolucji różnicowej L-SHADE oraz klasyczny algorytm ewolucyjny. Do grupy algorytmów optymalizacji wielokryterialnej należy natomiast algorytm MOOPTIM.

3.10.1. Jednokryterialny algorytm Nelder-Meada

Algorytm Nelder-Meada opracowano w 1965 roku i jest często nazywana metodą pełzającego sympleksu. Jest jedną z metod numerycznych służących do minimalizacji funkcji w wielowymiarowej przestrzeni. Zasada działania metody opiera się na utworzeniu w przestrzeni $E_n + 1$ n - wymiarowego sympleksu o $n + 1$ wierzchołkach.[97] Simpleks ten powinien być tak zdefiniowany by można było go wpisać w powierzchnię reprezentującą badaną funkcję celu. Najprostszym jednowymiarowym sympleksem jest odcinek o dwóch wierzchołkach. Przykładowo sympleksem dwuwymiarowym jest trójkąt. Ogólnie sympleksem n - wymiarowym o $n+1$ wierzchołkach jest wielościan rozpięty na $n+1$ wektorach bazowych. Współrzędne punktów sympleksu oznaczone są jako x_i . Zaletą sympleksu jest to, że wymaga dużo mniejszej ilości punktów niż hipersześcian. Jest to szczególnie widoczne dla wielu wymiarów. Sympleks w N -wymiarowej przestrzeni ma $N + 1$ wierzchołków. Jest to minimalna ilość wierzchołków umożliwiającą poszukiwanie we wszystkich możliwych kierunkach N -wymiarowej przestrzeni. Istotnym jest, aby sympleks nie tworzył figury o zerowej objętości w N -wymiarowej przestrzeni, czyli na przykład dla funkcji dwuwymiarowej (dwie zmienne) trzy punkty sympleksu nie mogą leżeć na jednej linii, a w przypadku funkcji trójwymiarowej (trzy zmienne) cztery punkty sympleksu nie mogą leżeć na jednej płaszczyźnie.[70]

Procedura rozpoczyna się od obliczania współrzędnych punktów wierzchołkowych sympleksu x_i (dla $i = 1, \dots, n + 1$) przy czym, zakłada się pewną minimalną odległość pomiędzy tymi wierzchołkami. W kolejnych iteracjach następuje przekształcanie sympleksu tak długo, aż odległość pomiędzy jego wierzchołkami w pobliżu poszukiwanego ekstremum będzie mniejsza od założonej dokładności obliczeń ϵ . To właśnie zostało przyjęte jako kryterium zbieżności dla tej metody.

Działanie metody najlepiej widać w pseudokodzie algorytmu:

1. Dobierz wartości współczynników $\alpha, \gamma, \delta, \rho$ oraz parametr zakończenia ϵ .

Są to odpowiednio współczynniki: odbicia, ekspansji, kontrakcji lub zmniejszenia.

Ich domyślne wartości wynoszą:

$$\begin{aligned}\alpha &= 1 \\ \gamma &= 2 \\ \delta = \rho &= \frac{1}{2}\end{aligned}\tag{3.91}$$

Stwórz sympleks początkowy o $N+1$ wierzchołkach.

Oblicz wartość funkcji celu dla tych wierzchołków.

2. Uporządkuj je wg wartości.

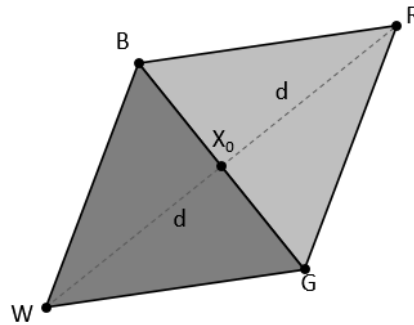
$$f(x_B) \leq \dots \leq f(x_G) \dots \leq f(x_w) \quad (x_B = x_1, x_w = x_{n+1})\tag{3.92}$$

3. Tzn. znajdź punkt najgorszy (W), punkt najlepszy(B) oraz drugi w kolejności z najgorszych (G). Następnie wyznacz punkt x_0 względem, którego następuje odbicie punktu najgorszego W.

$$x_0 = \frac{1}{N} \sum_{i=1, i \neq h}^{N+1} x_i \quad (3.93)$$

4. Wyznacz punkt odbicia x_R

$$x_R = x_0 + \alpha(x_0 - x_W) \quad (3.94)$$



przy czym x_0 jest środkiem ciężkości figury, za wyjątkiem wężła W.

Jeśli $f(x_G) > f(x_R) \geq f(x_B)$ to wyznaczamy nowy simpleks z użyciem punktu R w miejsce W.

Dla ekspansji i kontrakcji oraz tzw. ‘zmniejszania’ w zależności od relacji pomiędzy wartościami funkcji celu w punktach: B, G, W, punkt R jest wyznaczany w różny sposób. I tak,
a) jeśli po wyznaczeniu nowego położenia punktu R,

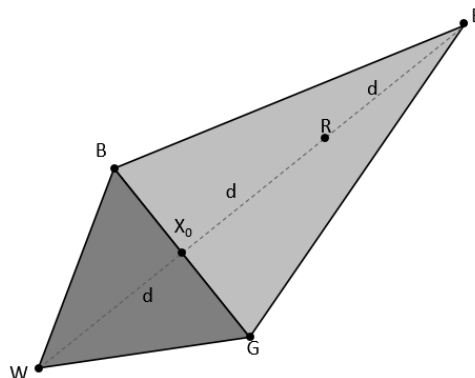
$$f(x_R) < f(x_B) \quad (3.95)$$

to nowy punkt

$$x_R = x_0 + \gamma(x_0 - x_W) \quad (3.96)$$

100

przesuwamy do nowego położenia (punkt E) i mamy do czynienia z ekspansją, kierowaną za pomocą współczynnika γ .



Jeśli $f(x_E) < f(x_R)$ to wyznaczamy nowy simpleks z użyciem punktu E zamiast W. W innym przypadku wyznaczamy simpleks zawierający punkt R w miejsce W.

- b) jeśli po wyznaczeniu nowego położenia punktu R,

$$f(x_R) \geq f(x_G) \text{ i } x_G \neq x_W \quad (3.97)$$

to nowy punkt R wyznaczany jest w miejsce punktu C_1

$$x_{C_1} = x_W + \delta(x_o - x_W) \text{ lub} \quad (3.98)$$

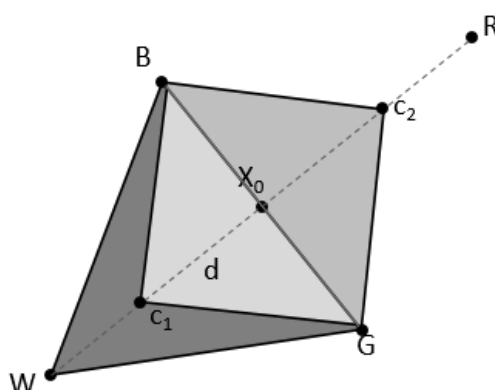
c) jeśli

$$f(x_G) < f(x_R) < f(x_W) \quad (3.99)$$

to nowy punkt R wyznaczany jest w miejsce punktu C_2

$$x_{C_2} = x_o + \delta(x_o - x_W) \quad (3.100)$$

i mamy do czynienia wtedy z kontrakcją.

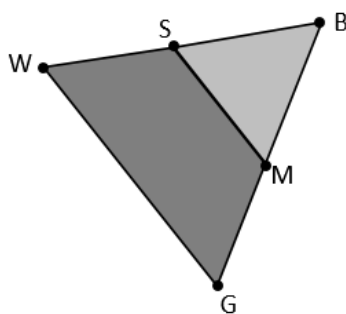


Jeżeli $f(x_C) < f(x_W)$ to wyznaczamy nowy simpleks z użyciem punktu $C_{1,2}$ zamiast W.

5. Jeśli $f(x_C) > f(x_W)$ wtedy dla punktów W i G wyznaczamy nowe współrzędne:

$$x_i = x_B + \rho(x_i - x_B), \quad \text{gdzie: } i \in \{2, \dots, n+1\} \quad (3.101)$$

i mamy do czynienia z tzw. podziałem krawędzi sympleksu ‘*shrinking*-iem’. Nowe punkty S i M znajdują się w połowie długości odpowiednich boków. O dokładnym ich położeniu decyduje współczynnik ρ .



6. Jeśli natomiast,

$$\left\{ \sum_{i=1}^{N+1} \frac{(f(x_i) - f(x_o))^2}{N+1} \right\}^{1/2} \leq \epsilon \quad (3.102)$$

algorytm kończy działanie lub przechodzi do punktu drugiego.

Proces ‘odbicia’ ma na celu znalezienie punktu, dla którego być może wartość funkcji celu będzie mniejsza od punktu W, którego funkcja jest największa. Odbicie następuje względem bryły, powierzchni lub krawędzi uformowanej z pozostałych wierzchołków. W przypadku ekspansji, jeśli odbity punkt okazał się najlepszy spośród pozostałych wierzchołków i stanowi nowe minimum, to spodziewamy się znaleźć jeszcze lepsze wartości funkcji celu dla punktów, które są położone na linii x_w i x_R . Jeżeli chodzi natomiast o kontrakcję, to dla warunku $f(x_R) \geq f(x_G)$ możemy przypuszczać, że lepsza wartość funkcji celu wystąpi wewnątrz simpleksu utworzonego poprzez wszystkie wierzchołki.

Bardzo ważną rzeczą jest wybór początkowego simpleksu, który ma ogromny wpływ na skuteczność przeszukiwania przestrzeni. Dobry zbyt mały może nie znaleźć globalnego, lecz utknie na jednym z lokalnych minimum.

Przy czym, punkt W jest punktem wierzchołkowym simpleksu spośród $n+1$ wierzchołków x_i , w którym wartość badanej funkcji jest największa. Punkt B jest wybranym punktem wierzchołkowym simpleksu spośród $n+1$ wierzchołków x_i , w którym wartość badanej funkcji jest najmniejsza. x_0 to środek symetrii simpleksu z wyłączeniem punktu x_w wyznaczany wg wzoru (3.93), a d jest odległością pomiędzy wierzchołkami W i x_0 . Wartości współczynników $\alpha, \gamma, \delta, \rho$ dobiera się w sposób eksperymentalny, lecz wg autorów metody jako optymalne należy przyjąć $\alpha = 1, \rho = 0,5, \delta = 0,5$ oraz $\gamma = 2$.

3.10.2. Jednokryterialny algorytm ewolucyjny EA

Typowy algorytm ewolucyjny opiera swoje działanie na populacji rozwiązań $P(t)$, której liczebność wynosi N . Wszystkie osobniki są reprezentowane poprzez chromosomy [139,105,55,112]:

$$P(t) = \left\{ \begin{array}{c} ch_t^1 \\ ch_t^2 \\ \vdots \\ ch_t^j \\ \vdots \\ ch_t^N \end{array} \right\} \quad (3.103)$$

gdzie: $t = 1, 2, \dots$ – numer iteracji algorytmu (numer generacji)

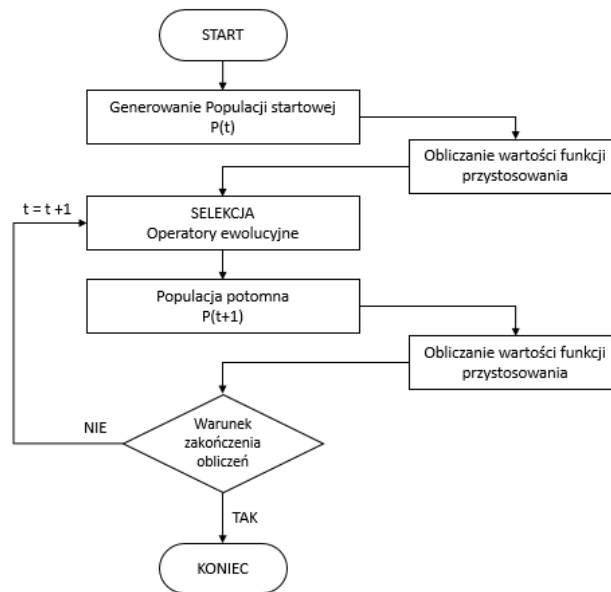
ch_t^1 – chromosom pierwszy w t -tej iteracji

Pojedynczy chromosom w populacji jest wektorem, zawierający n -elementów – genów

$$ch_t^j = [gen_1^j, gen_2^j, \dots, gen_i^j, \dots, gen_n^j] \quad (3.104)$$

Każdy gen może zmieniać swoją wartość w pewnym zdefiniowanym zakresie:

$$[gen_i^j]_{min} \leq gen_i^j \leq [gen_i^j]_{max} \quad (3.105)$$



Rys. 3.28 Schemat blokowy algorytmu ewolucyjnego.

Algorytm ewolucyjny (Rys. 3.28) zaczyna swoje działanie od wygenerowania (zwykle z równomiernym rozkładem losowym) startowej populacji rozwiązań $P(0)$. Generuje liczbę chromosomów stanowiącą o rozmiarze populacji N , dla których losowo dobiera wartości genów z zakresu dopuszczalnego dla każdej zmiennej. Każdy chromosom zawiera tyle genów, ile zostało zdefiniowanych zmiennych dla optymalizowanej funkcji. W kolejnym kroku oblicza się wartość funkcji przystosowania dla każdego chromosomu (osobnika). Następnie z populacji $P(t)$, $t = 1, 2, \dots$, wybiera się odpowiednią metodą selekcji chromosomy do tzw. puli rodzicielskiej $O(t)$. Prawdopodobieństwo wyboru osobników w operatorze selekcji zależy od jego przystosowania (wartości funkcji celu). W kolejnym etapie następuje kojarzenie osobników rodzicielskich z populacji $O(t)$ w pary. W ramach skojarzonych par dokonuje się operacji krzyżowania zgodnie z prawdopodobieństwem krzyżowania p_c . Prawdopodobieństwo to decyduje jaki procent osobników poddany zostanie operacji krzyżowania. Po krzyżowaniu następuje operacja mutacji zgodnie z określonym prawdopodobieństwem p_m . Ono z kolei decyduje o procencie osobników jakie zostaną poddane operacji mutacji. W ten sposób tworzy się nową populację potomną $P(t + 1)$, do której należą osobniki z populacji $O(t)$.

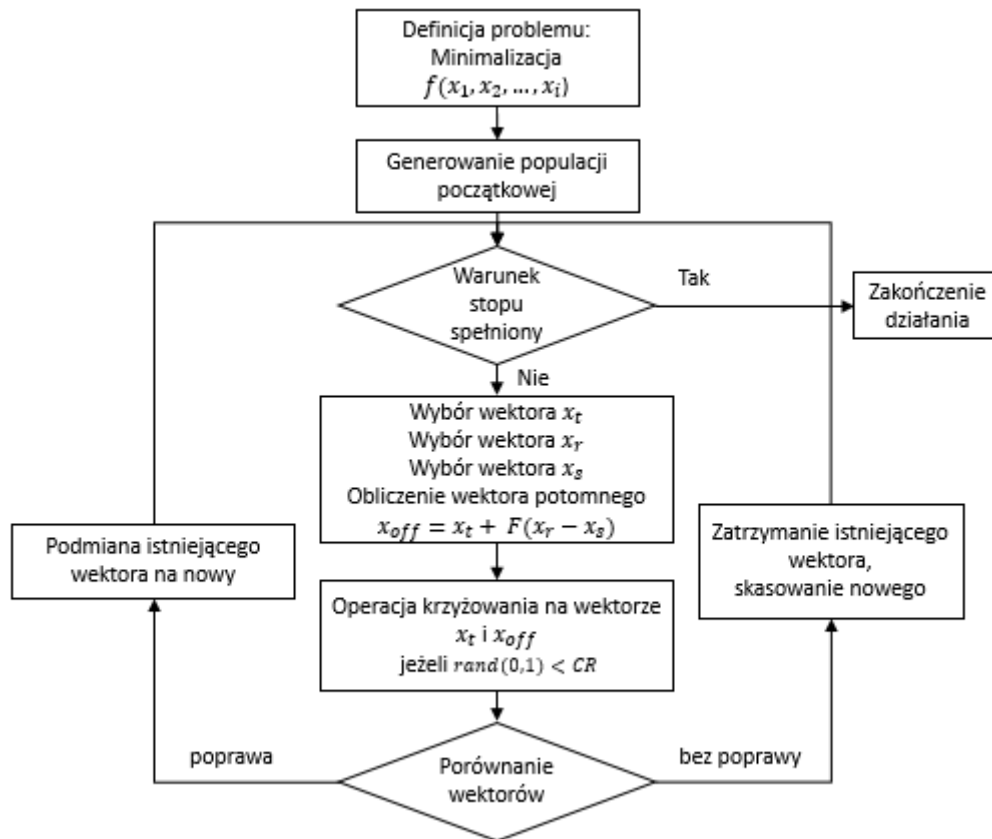
Po stworzeniu populacji potomnej następuje ocena przystosowania jej chromosomów. Po czym algorytm sprawdza warunek zakończenia działania algorytmu. Jeżeli jest spełniony algorytm kończy działanie. Zakończenie działania algorytmu może nastąpić po uzyskaniu żądanej wartości funkcji celu. W praktyce jednak zatrzymanie algorytmu następuje po uzyskaniu wartości z pewną dokładnością lub jeżeli jego dalsze iteracje nie poprawiają już uzyskanej najlepszej wartości. Innymi warunkami stopu dla algorytmu może być przekroczenie zadanej wartości maksymalnego czasu obliczeń lub maksymalnej liczby iteracji algorytmu.

Jeżeli jednak warunek zatrzymania algorytmu nie został spełniony to algorytm przechodzi do następnej iteracji. Wtedy populacja potomna staje się populacją rodzicielską i proces się powtarza.

3.10.3. Jednokryterialny algorytm ewolucji różnicowej L-SHADE

Algorytm L-SHADE reprezentuje rodzinę algorytmów ewolucji różnicowej. Ewolucja różnicowa (*Differential Evolution* – DE) jest zaliczana do zaawansowanych i jednocześnie bardzo skutecznych, ewolucyjnych metod optymalizacji. Jest skuteczną i efektywną alternatywą dla algorytmów genetycznych oraz ewolucyjnych. Został on opracowany w 1995 r. przez Storna i Pricea [138].

W algorytmie DE populacja początkowa jest zbiorem wektorów x_j , wybieranych przeważnie losowo z przestrzeni poszukiwań. W trakcie optymalizacji obliczana jest ważona różnica między dwoma losowo wybranymi osobnikami populacji x_r i x_s , a następnie dodawana, biorąc pod uwagę współczynnik krzyżowania, do trzeciego x_t , tak aby utworzyć nowy wektor zmiennych projektowych. Jeżeli nowy wektor uzyskuje lepszą wartość, zastępuje poprzednika. Zaletą algorytmu DE w porównaniu z wieloma innymi bioinspirowanymi jest jego mała liczba parametrów. Możliwe jest zdefiniowanie tylko trzech parametrów. Są to wielkość populacji (NP), współczynnik skalowania mutacji $F \in [0,2]$ oraz współczynnik krzyżowania $CR \in [0,1]$. Schemat blokowy podstawowej wersji algorytmu ewolucji różnicowej można przedstawić jak na Rys.3.29.



Rys. 3.29 Schemat blokowy ewolucji różnicowej

Współczynniki krzyżowania o niższych wartościach powodują zachowanie większej części rozwiązań rodzicielskich. Współczynnik skali F służy do obliczania różnicy ważonej podczas generowania osobników. Jako warunek zakończenia można przyjąć np. maksymalną liczbę iteracji $Niter$. W pracy [138] autorzy sugerują, przyjmowanie wartości F z przedziału od 0,4 do 1, a wartość CR z przedziału od 0,1 do 0,9. Wartość parametrów użytych na potrzeby badań w tej pracy to wielkość populacji $Npop = 50$, współczynnik krzyżowania $CR = 0,5$ i współczynnik skali $F = 0,8$. W pracy zastosowano zmodyfikowaną i rozbudowaną wersję tego typu algorytmu (L-SHADE).

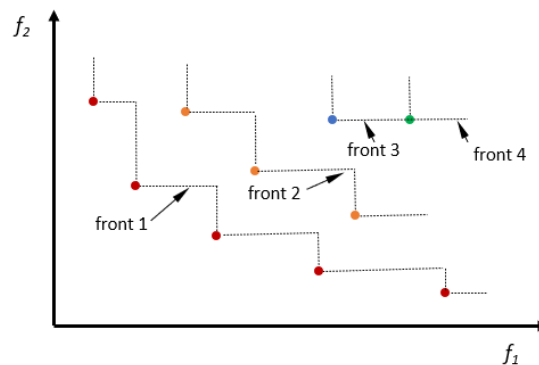
Istnieje wiele metod, które zostały opisane w literaturze, które samodzielnie dostosowują parametry ER. Algorytm SHADE [142] dostosowuje parametry F i CR , podczas gdy L-SHADE [143] to ulepszona wersja SHADE, która dodaje liniowy schemat redukcji wielkości populacji, dostosowując parametr NP. Zamiast stałej wartości dla parametru NP, ma wielkość początkową (NP_0) i końcową (NP_f). Pod koniec każdej generacji g , rozmiar dla następnej generacja jest obliczany jako:

$$NP_{g+1} = \text{round} \left[\frac{NP_f - NP_0}{MAX_NFE} \cdot NFE + NP_0 \right] \quad (3.106)$$

gdzie MAX_NFE jest maksymalną liczbą wywołań funkcji celu dozwolonej dla algorytmu, a NFE to bieżąca liczba wydanych już ocen.

3.10.4. Wielokryterialny algorytm MOOPTIM

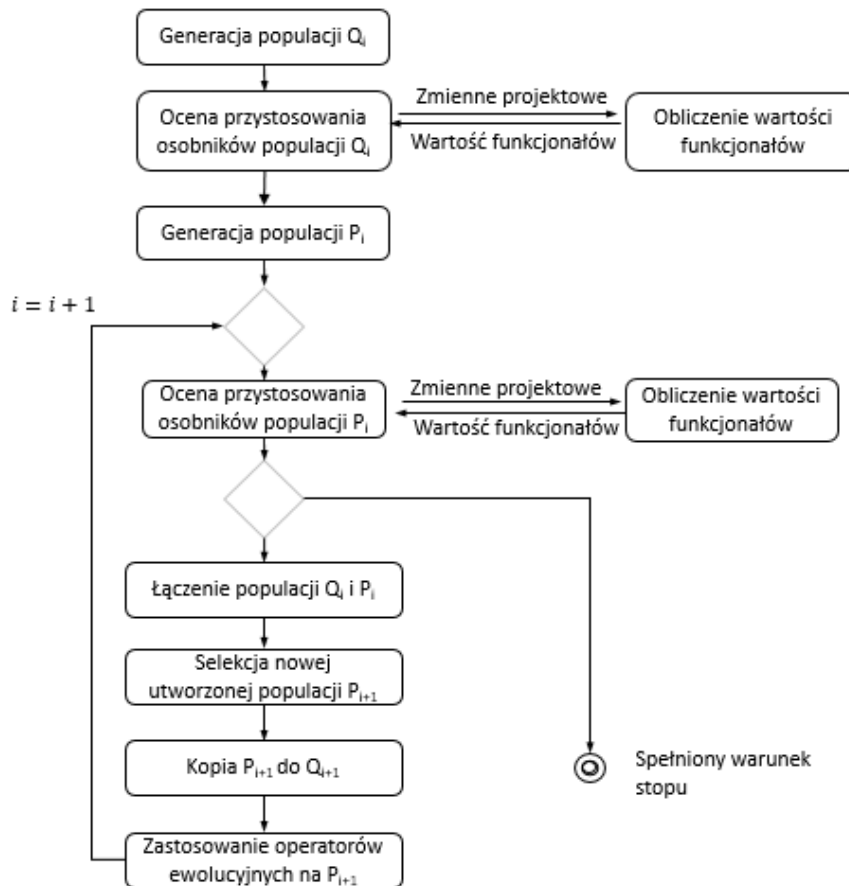
Algorytm MOOPTIM jest autorską implementacją wielokryterialnego algorytmu ewolucyjnego wykorzystującego procedurę niezdominowanego sortowania, zmodyfikowane operatory genetyczne oraz zmodyfikowany mechanizm selekcji [54]. Rys. 3.30 przedstawia schemat niezdominowanego sortowania dla przypadku minimalizacji. Algorytm bazuje na algorytmie NSGA-II, który został opisany w rozdziale 3.9.4. W algorytmie MOOPTIM tworzenie frontów osobników i przypisywanie odpowiednich rang następuje w identyczny sposób jak w NSGA-II.



Rys. 3.30 Schemat procedury niezdominowanego sortowania algorytmie MOOPTIM

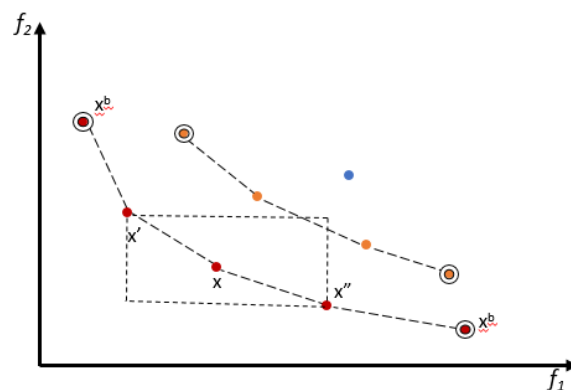
Algorytm MOOPTIM składa się z dwóch części. Pierwszą stanowi inicjalizacja, a drugą tzw. pętla główna. Na Rys. 3.31 przedstawiono schemat działania algorytmu.

Algorytm w pierwszym kroku generuje przy pomocy generatora liczb losowych dwie populacje Q_i i P_i . Każda z nich liczy N osobników. Ponadto, w części inicjalizacyjnej zostają zdefiniowane parametry algorytmu. Następnie są obliczane wartości funkcjonałów dla każdego z osobników w populacji Q_i . W pętli głównej algorytmu po obliczeniu wartości funkcjonałów dla osobników populacji P_i , jest sprawdzany warunek zakończenia działania algorytmu. Jest nim maksymalna liczba iteracji zdefiniowana przez użytkownika. W kolejnym kroku następuje połączenie populacji Q_i oraz P_i w jedną populację R_i . Rozmiar populacji R_i wynosi $2N$. Z populacji tej wybieranych jest tylko N osobników. Druga połowa jest odrzucona. Osobniki do populacji potomnej P_{i+1} są wybierane na podstawie procedury niezdominowanego sortowania. Wybór dokładnie N osobników do populacji P_{i+1} , wymaga zastosowania kryterium współczynnika zatłoczenia, który zastosowano w algorytmie NSGA-II. Dla osobników przynależących do tego samego frontu wybierane są te, które mają mniejszą wartość tego współczynnika.



Rys. 3.31 Schemat działania algorytmu MOOPTIM

Współczynnik zatłoczenia dla danego osobnika, jego wartość, jest wyznaczana w przestrzeni funkcji przystosowania w odniesieniu do położenia „sąsiadów” rozpatrywanego osobnika. W roli „sąsiadów” mogą występować wyłącznie osobniki z tego samego frontu. Rys. 3.32 przedstawia sposób wyznaczania współczynnika zatłoczenia, w tym przypadku, dla dwóch kryteriów. Dla rozwiązania x jest to pole prostokąta, którego wierzchołkami są rozwiązania znajdujące się najbliżej x , czyli x' oraz x'' . Wartość współczynnika zatłoczenia jest wyznaczana dla wszystkich osobników frontu. Wyjątek stanowią osobniki skrajne (x^b). W ich przypadku, dla każdego frontu, są im przypisywane duże wartości współczynnika zatłoczenia. Działanie to sprzyja maksymalizacji rozpiętości frontu Pareto.



Rys. 3.32 Schemat wyznaczania współczynnika zatłoczenia

Można dostrzec plusy takiego postępowania. Pierwszym z nich jest zachowanie różnorodności w populacji. Wtedy, kiedy zdecydowana większość populacji przynależy do jednego frontu. Drugim, nie mniej ważnym, jest mechanizm, który zapewnia przetrwanie osobnikom skrajnym dla danych frontów. Pozwala to uzyskać maksymalną rozpiętość frontu Pareto. Dla większej liczby kryteriów ten sam mechanizm można zastosować w przestrzeni n-wymiarowej.

W kolejnym kroku algorytmu osobniki populacji P_{i+1} są kopiowane do populacji Q_{i+1} . Osobniki należące do populacji P_i , są poddawane działaniu operatorów mutacji i krzyżowania i tworzą populację P_{i+1} . Zastosowano dwa rodzaje mutacji: równomierną i z rozkładem Gaussa. W przypadku operatora krzyżowania, zdecydowano się na krzyżowanie proste i arytmetyczne. Ostatnia populacja wygenerowana przez algorytm zawiera zbiór rozwiązań Pareto optymalnych. W algorytmie MOOPTIM, w porównaniu z algorytmem NSGA-II, zrezygnowano z binarnej selekcji turniejowej. Ponadto, algorytm MOOPTIM zawiera więcej operatorów ewolucyjnych. Selekcja natomiast odbywa się wyłącznie na podstawie przynależności do danego frontu (procedura niezdominowanego sortowania) oraz współczynnika zatłoczenia. Zastosowanie mutacji z rozkładem Gaussa w algorytmie MOOPTIM pozwoliło znacząco poprawić skuteczność działania tego algorytmu [54].

3.11. Kryteria optymalizacji

W pracy rozpatrzono jednowarstwową osłonę cylindryczną z przetłoczeniami, której geometrię przedstawiono na Rys. 4.3. Wybrane cechy geometryczne osłony oraz sposób jej mocowania (warunki brzegowe) są przedmiotem rozpatrywanych zadań optymalizacji jednokryterialnej oraz wielokryterialnej. Osłona termiczna posiada trzy otwory montażowe oraz dwa przetłoczenia przechodzące przez jej większą i mniejszą część. Zakładając, że średnice części cylindrycznej są narzucone i ustalone ze względów technologicznych, optymalne zaprojektowanie własności mechanicznych tego typu osłon będzie polegało na odpowiednim ukształtowaniu przetłoczeń, doborze grubości blachy oraz lokalizacji otworów montażowych. Do rozwiązania zagadnień wytrzymałościowych wykorzystano oprogramowanie ANSYS. Osłona została poddana numerycznej analizie strukturalnej i modalnej. Zgodnie z wymaganiami opisanymi w rozdziale 2.1 sformułowano dla przypadku optymalizacji jednokryterialnej i zaimplementowano numerycznie następujące kryterium wraz z ograniczeniami:

1. Minimalizacja maksymalnej wartości przemieszczenia wypadkowego (f_1)

$$\min_x f_1 = \max(u_{res}) \quad (3.107)$$

2. Ograniczenie w postaci maksymalnej wartości naprężenia zredukowanego von Misesa

$$\max(\sigma_{eq}) < \sigma_{dop} \quad (3.108)$$

3. Ograniczenie w postaci minimalnej wartości pierwszej częstotliwości drgań własnych osłony

$$f_{res} < f_{dop} \quad (3.109)$$

gdzie:

σ_{dop} – maksymalne naprężenia dopuszczalne równe 450MPa

f_{dop} – minimalna wartość pierwszej częstotliwości drgań własnych osłony równa 250Hz

W przypadku optymalizacji wielokryterialnej zdefiniowano i zaimplementowano numerycznie następujące kryteria:

1. Minimalizacja maksymalnej wartości przemieszczenia wypadkowego (f_1)

$$\min_x f_1 = \max(u_{res}) \quad (3.110)$$

2. Minimalizacja maksymalnej wartości naprężenia zredukowanego von Misesa (f_2)

$$\min_x f_2 = \max(\sigma_{eq}) \quad (3.111)$$

3. Maksymalizacja wartości pierwszej częstotliwości drgań własnych osłony (f_3) w odniesieniu do częstotliwości referencyjnej

$$\max_x f_3 = |f_{res} - f_{ref}| > \Delta f \quad (3.112)$$

gdzie: f_{ref} - częstotliwość referencyjna jest równa 230Hz

Δf – zdefiniowana różnica pomiędzy $|f_{res} - f_{ref}|$ równa 20Hz

Pierwsze kryterium wyraża sztywność układu, która jest istotnym czynnikiem ze względu na możliwość kontaktu osłony z układem wydechowym, jeśli odległość między tymi elementami nie jest duża. Drugie, poprzez zmniejszenie maksymalnych wartości naprężeń zredukowanych wyznaczonych z hipotezy Hubera-Misesa, związane jest z trwałością układu (należy zaznaczyć, że nie uwzględniono analizy zmęczeniowej w pracy). Natomiast trzecie kryterium pozwala na dobranie cech geometrycznych osłony w taki sposób, aby nie występowały w niej drgania rezonansowe. Zazwyczaj takie kryterium definiuje się jako różnicę między najniższą częstotliwością rezonansową a wartością odniesienia. W niniejszej pracy przyjęto wartość odniesienia równą 230 Hz, ze względu na częstotliwość wymuszenia powodowaną przez silnik spalinowy.

Wektor \mathbf{x} oznacza zestaw zmiennych projektowych opisanych szczegółowo w rozdziale 4.1. Sformułowanie i jednoczesne uwzględnienie więcej niż jednego kryterium prowadzi do problemu optymalizacji wielokryterialnej [170]. Funkcjonały f_1 i f_2 będące ograniczeniami w optymalizacji jednokryterialnej, rozpatrywane jednak jako kryteria pozwalają na analizę frontów Pareto i badanie relacji między funkcjonalami.

Kryteria (3.110) i (3.111) oblicza się na podstawie rozwiązania numerycznego liniowej analizy sprężystej w podwyższonej temperaturze. Macierzowe równanie MES takiego problemu wyraża się wzorem 3.62.

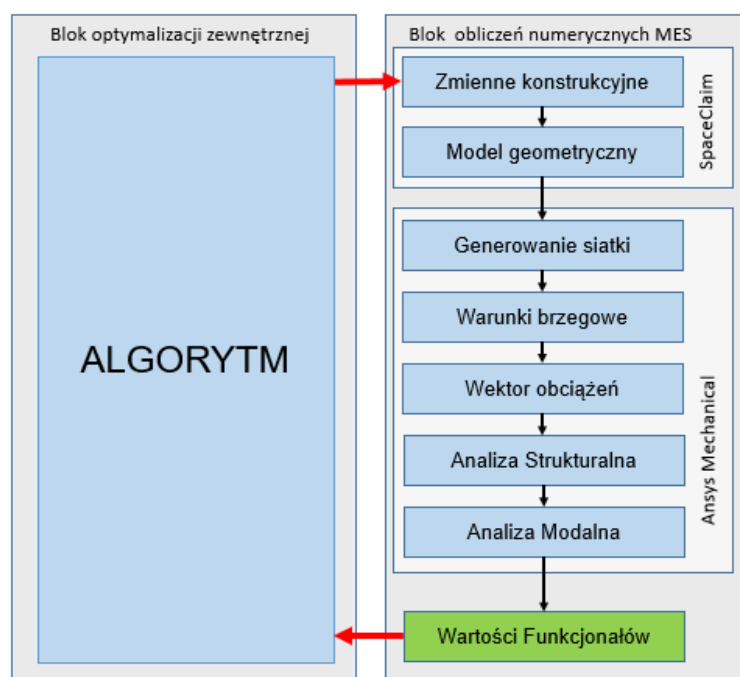
Kryterium (3.112) oblicza się na podstawie wyników analizy modalnej MES układu konstrukcyjnego bez tłumienia wg wzoru 3.84.

Warunek ograniczenia maksymalnych wartości naprężeń poniżej 450MPa wynika z rodzaju materiału użytego do wykonania osłony. Dla zastosowanej stali wg. PN, typu 0H18N9 (wg DIN 1.4301, wg AISI 304), jest to stal austenityczna, dla której wartość maksymalnych naprężeń R_m waha się pomiędzy 500 - 700MPa, dla której wartość 450MPa jest wartością bezpieczną, wybraną z dolnego pułapu wartości granicznych. Wartość ta została tak dobrana, by zminimalizować prawdopodobieństwo powstawania uszkodzeń osłony podczas pracy układu. Natomiast warunek ograniczający częstotliwość drgań własnych do wartości powyżej 250Hz wynika z typu silnika spalinowego z którym współpracuje układ

wydechowy, w którym zastosowano osłonę termiczną. Obroty silnika spalinowego możemy wyrazić za pomocą widma częstotliwościowego. Silnik spalinowy czterocylindrowy o zapłonie iskrowym pracuje w zakresie (z reguły) od 600 do 6900obr/min. Oznacza to, że zgodnie z wzorem (2.10) w widmie częstotliwościowym zakres pracy tego samego silnika zawiera się pomiędzy 20 a 230Hz. W ramach pracy zakres górny dodatkowo rozszerzono o 20Hz jako margines bezpieczeństwa, by wyeliminować graniczne przypadki powstawania zjawiska rezonansu na granicach pasma częstotliwościowego, górnego zakresu pracy silnika.

3.12. Opracowany schemat optymalizacji jedno i wielokryterialnej

W przypadku wykorzystania algorytmów zewnętrznych dla optymalizacji jednokryterialnej i wielokryterialnej użyto tego samego schematu postępowania [82]. Jedną z istotnych różnic była odpowiednio dla każdego typu optymalizacji ilość aktywnych kryteriów i ewentualnie zdefiniowanych ograniczeń.



Rys. 3.33 Schemat obliczeń dla procesu optymalizacji z wykorzystaniem algorytmów zewnętrznych

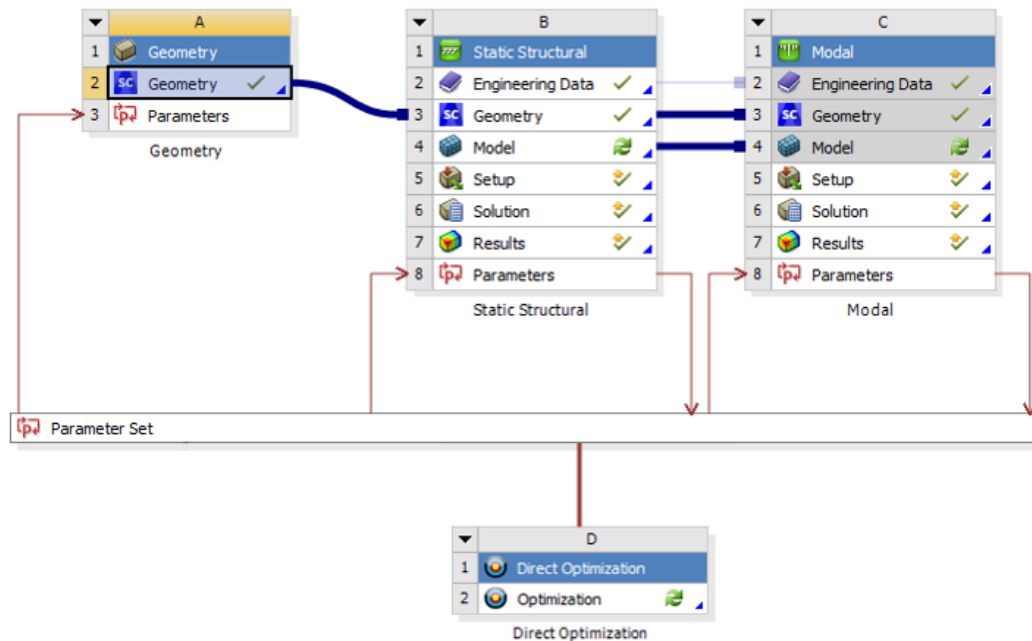
Zastosowanie zewnętrznego algorytmu optymalizacyjnego wymagało opracowania i zaimplementowania metod automatycznego tworzenia i rozwiązywania problemów brzegowych opisanych wzorami (3.62) i (3.84). Na Rys. 3.33 przedstawiono sposób sprzężenia bloku zewnętrznego algorytmu optymalizacyjnego z blokiem obliczeń MES.

Dla każdego chromosomu w algorytmie zewnętrznym przygotowywana jest geometria modelu osłony. Krok ten jest realizowany w oparciu o zakodowaną wartość genów w oprogramowaniu SpaceClaim, dla którego opracowano odpowiednie procedury w C++ i języku skryptowym IronPython (zaimplementowany w SpaceClaim). Po wygenerowaniu geometrii modelu, przy pomocy preprocesora w oprogramowaniu Ansys generowana jest siatka elementów skończonych, definiowane są odpowiednie warunki brzegowe oraz wektor obciążeń. Następnie moduł obliczeniowy Ansysa rozwiązuje zdefiniowany problem. Na podstawie plików wynikowych generowanych przez Ansys formułowane wcześniej funkcje celu są obliczane przez dodatkowe procedury napisane w C++ i

przekazywane do algorytmu. Po analizie wartości funkcjonałów algorytm generuje kolejny wektor danych wejściowych i rozpoczyna się kolejna pętla obliczeń. Pętla optymalizacyjna wykonywana jest do momentu spełnienia warunku zakończenia obliczeń.

Wspomniane procedury napisane w C++ i IronPythonie sprzęgają automatycznie algorytm z modułami Ansys'a. Modułem obliczeniowym wykorzystanym do obliczeń MES w oprogramowaniu ANSYS był moduł *Mechanical*. Ponadto, dla skrócenia czasu obliczeń jednej pętli, wszystkie obliczenia były przeprowadzone w trybie wsadowym (*batch mode*).

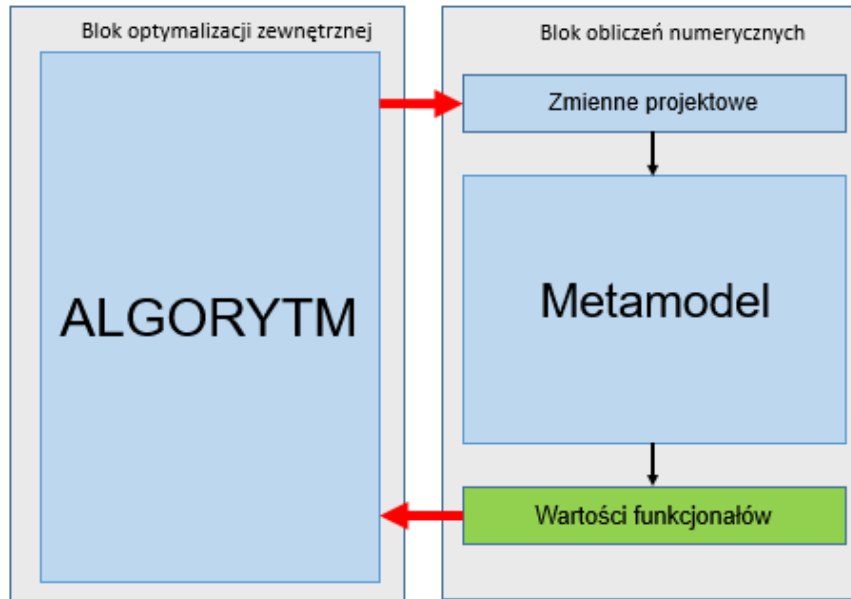
W przypadku obliczeń optymalizacyjnych z wykorzystaniem algorytmów zaimplementowanych w oprogramowanie ANSYS, schemat procesu optymalizacji przedstawia Rys. 3.34



Rys. 3.34 Schemat procesu optymalizacji w środowisku ANSYS

W przypadku wykorzystania algorytmów zewnętrznych w połączeniu z metamodeliem schemat połączenia modułów różni się od schematu z rys. 3.31 tym, że w miejsce wykorzystywanego wcześniej oprogramowania Ansys jest wbudowany metamodel oparty o SSN.

W identyczny sposób jak poprzednio algorytm generuje wektor wejściowy, lecz pomijany jest krok budowy modelu geometrycznego (Rys. 3.35). Wektor zmiennych projektowych wygenerowanych przez algorytm jest wejściowym wektorem danych do metamodelu. Na podstawie metamodelu wyznaczane jest przybliżone rozwiązanie symulacji MES, wyznaczana jest wartość funkcjonałów, a następnie przekazywana do algorytmu przy pomocy opracowanych procedur w C++. Po analizie wartości funkcjonałów algorytm generuje kolejny wektor danych wejściowych i rozpoczyna się kolejna pętla obliczeń. Pętla optymalizacyjna wykonywana jest do momentu spełnienia warunku zakończenia obliczeń.



Rys. 3.35 Schemat obliczeń dla procesu optymalizacji z wykorzystaniem metamodelu

4. Model geometryczny oraz numeryczny osłony – budowa oraz weryfikacja numeryczna

4.1. Model geometryczny

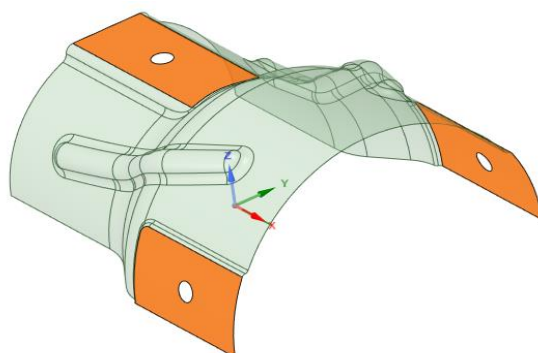
Uwarunkowania prawidłowego montażu osłon mocowanych na płaszcach katalizatorów lub filtrów cząstek stałych w układach wydechowych silników spalinowych, wiążą się ze spełnieniem określonych warunków konstrukcyjnych, wytrzymałościowych i technologicznych. Zabudowa kilku różnego rodzaju katalizatorów w jednej obudowie zwane ‘*canning-iem*’ z reguły wymagają skomplikowanych kształtów geometrycznych osłon termicznych, wynikających z konieczności zwiększania sztywności, tak aby wyeliminować zjawisko rezonansu. Ponadto układy takie pracują w zakresie podwyższonych temperatur. W praktyce osłony tego typu posiadają liczne przetłoczenia w celu zwiększenia sztywności samej osłony oraz redukcji maksymalnych wartości przemieszczeń.

Prawidłowo zaprojektowane osłony muszą spełniać wiele kryteriów związanych z ich charakterystykami wytrzymałościowymi, cieplnymi, akustycznymi, itp. Dzięki wysokiej odporności cieplnej nie ulegają one uszkodzeniu nawet w wysokich temperaturach. W ramach pracy utworzono wstępny model geometryczny osłony w postaci fragmentu powierzchni o zadanej krzywiznie wraz z przetłoczeniami oraz otworami montażowymi. Osłony te zostały poddane numerycznej analizie modalnej we wcześniejszej fazie analizy zdefiniowanego problemu. Wykonano analizę wariantową uwzględniającą częstotliwości wymuszenia (drżania generowane przez silnik) i opisano ją w artykule [83]. Zauważono niewielki wpływ przetłoczeń umiejscowionych w płaszczyznach symetrii. Natomiast analiza postaci drgań własnych osłony (pierwsze pięć postaci) niosła cenne informacje gdyż pozwalają one poprawnie dobrać miejsce dla przetłoczeń usztywniających i mocowania osłony.

Kontynuując badania opracowano kolejny model geometryczny osłony posiadający symetrycznie rozłożone cztery przetłoczenia oraz cztery otwory montażowe. Szczegółowe wyniki analizy przedstawiono w artykule [84].

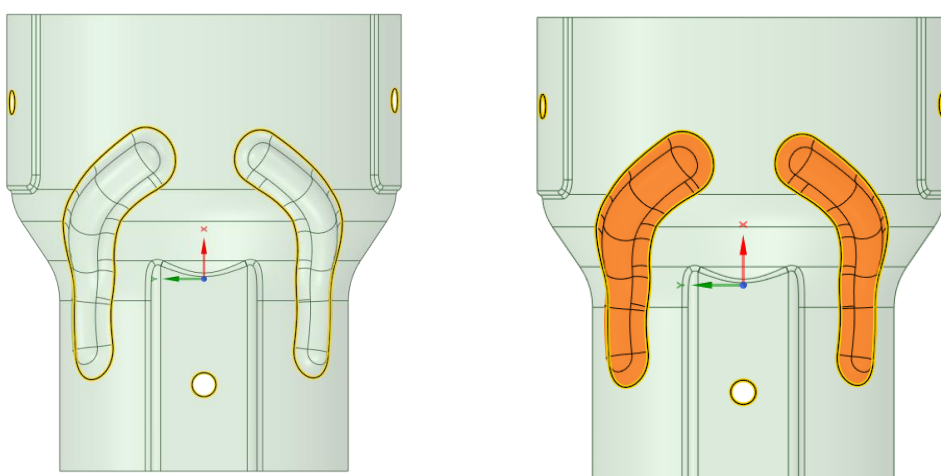
W ramach pracy, w wyniku przeprowadzonych analiz i wstępnych obliczeń utworzono aktualny model geometryczny osłony wraz z przetłoczeniami oraz otworami montażowymi, który został wykorzystany w procesie optymalizacji jedno, i -wielokryterialnej oraz do budowy metamodelu. Jest to w pełni sparametryzowany model geometryczny. Sterowanie parametrami tego modelu zostało zaprogramowane w skrypcie napisanym w języku IronPython, który stanowi wewnętrzny język programowania środowiska ANSYS. Do tworzenia samej geometrii użyto ‘*SpaceClaim*’, modułu wewnętrznego do tworzenia geometrii w oprogramowaniu ANSYS.

Opracowano odpowiednie modele geometryczne w różnych konfiguracjach, jeżeli chodzi o ilość otworów montażowych i ilości przetłoczeń. Ostatecznie wybrano model geometryczny osłony pokazany na (Rys. 4.1). Jest to jednowarstwowa osłona cylindryczna z przetłoczeniami. Jak wcześniej wspomniano, założono, że średnice części cylindrycznej są narzucone i ustalone ze względów technologicznych, optymalne zaprojektowanie właściwości mechanicznych tego typu osłon polegało na odpowiednim ukształtowaniu przetłoczeń przy jednoczesnym doborze grubości blachy oraz położeniu otworów montażowych. Jest to model w pełni sparametryzowany. Posiada dwa niesymetryczne przetłoczenia oraz trzy otwory montażowe. Otwory montażowe mogą być usytuowane ze względów technologicznych tylko na wybranych do tego celu powierzchniach, pokazanych na Rys. 4.1.



Rys. 4.1 Położenie otworów montażowych

Przetłoczenia natomiast mogą przybierać dowolną trajektorię na całym obszarze osłony poza obszarami przeznaczonymi dla otworów montażowych. Są predefiniowane za pomocą 7-miu parametrów każdy.



Rys. 4.2 Model geometryczny osłony

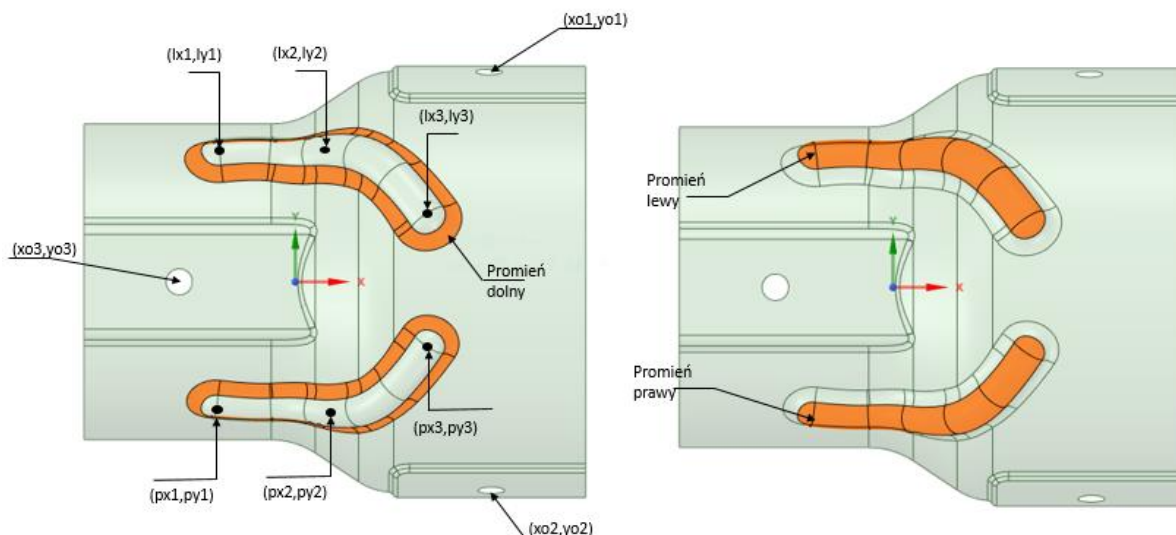
Obydwa przetłoczenia nie są ze sobą powiązane geometrycznie tzn. mogą przyjmować dowolny kształt niezależnie od siebie. Są sparametryzowane niezależnie, co prowadzi do niesymetrycznych wyników i zwiększa zakres możliwości poszukiwań projektowych.

Jak pokazano na

Rys. 4.2 wszystkie zaznaczone kolorem pomarańczowym elementy geometrii osłony zostały sparametryzowane. Chodzi o położenie otworów montażowych oraz położenie i kształt każdego z przetłoczeń usztywniających. Geometria osłony jest rezultatem wartości wszystkich parametrów zaimplementowanymi w modelu parametrycznym zdefiniowanych względem globalnego układu współrzędnych. Tak przygotowane parametryczne modele geometryczne, były podstawą przeprowadzenia późniejszych zadań optymalizacji.

Na Rys. 4.3, przedstawiono wszystkie parametry – zmienne konstrukcyjne modelu. Zaliczamy do nich:

- współrzędne otworów montażowych – 6 parametrów,
- współrzędne punktów kontrolnych krzywych typu „spline”, które definiują kształt i trajektorię przetłoczeń (x, y) – 12 parametrów
- promień przekroju przetłoczeń, które odpowiadają za szerokość i głębokość przetłoczenia – 2 parametry.



Rys. 4.3 Parametry modelu geometrycznego

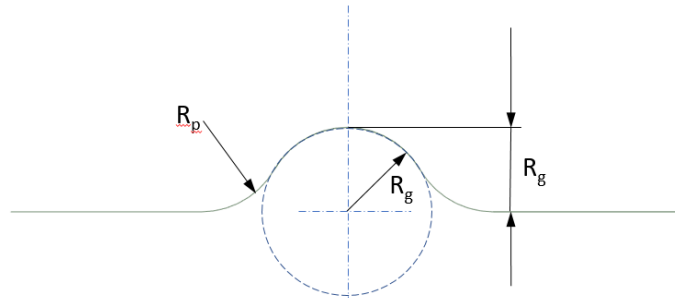
Zakres zmienności wszystkich parametrów został tak dobrany, by parametry leżące po przeciwnych stronach (lewa, prawa) miały możliwość zmieniać swoje wartości w tym samym zakresie. Stwarza to możliwość uzyskania dowolnych kształtów przetłoczeń i położenia otworów montażowych po obu stronach osłony (przypadek szczególny).

Na każdą zmienną projektową nałożone zostały ograniczenia. Zakres zmienności dla wszystkich 20 zmiennych projektowych zawarto w tabeli 4.1, gdzie wartości są zdefiniowane w odniesieniu do globalnego układu współrzędnych.

Tabela 4.1 Opis, oznaczenia oraz zakres zmienności zmiennych projektowych

Parametr		Zakres zmienności		Opis parametru
		dolny	górny	
1	xo1	4	14	Współrzędne otworów montażowych
2	yo1	4	36	
3	xo2	-8	19	
4	yo2	-9	2	
5	xo3	-17	10	
6	yo4	-6	5	
7	RR	2.5	4	Promień przetłoczenia dla lewego i prawego przetłoczenia usztywniającego
8	LR	2.5	4	
9	rx1	-2	10	Współrzędne punktów konstrukcyjnych krzywej <i>spline</i> dla lewego i prawego przetłoczenia
10	ry1	-14	0	
11	rx2	-1	0	
12	ry2	15	16	
13	rx3	-24	-5	
14	ry3	45	65	
15	lx1	-12	0	
16	ly1	-79	-65	
17	lx2	-2	-1	
18	ly2	-52	-51	
19	lx3	5	24	
20	ly3	-20	0	

Oprócz otworów montażowych elementami decydującymi o sztywności osłony są przetłoczenia. Ich trajektoria, profil przekroju, głębokość przetłoczenia, długość i położenie na powierzchni osłony mają istotny wpływ na wartość funkcjonałów (3.110, 3.111, 3.112). Profil przekroju jak i trajektoria zostały sparametryzowane za pomocą siedmiu parametrów niezależnych i jednego zależnego. Parametrem zależnym jest wartość promienia u podstawy przetłoczenia (promień przejścia pomiędzy główną powierzchnią osłony i powierzchnią przetłoczenia) oznaczony R_p . Wszystkie parametry przetłoczenia pokazano na Rys. 4.4.



Rys. 4.4 Przekrój przetłoczenia

Promień R_p jest promieniem zależnym od promienia głównego R_g odpowiedzialnego za szerokości głębokość przetłoczenia. Jego wartość wynosi $R_g - 0.5$ [mm].

4.2. Model MES osłony układu wydechowego

Osłony termiczne są mocowane do płaszcza tłumików bądź katalizatorów na różne sposoby. Rodzaj mocowania zależy od rodzaju osłony, jej przeznaczenia, ale też od możliwości montażowych. Czasem brak wystarczającego miejsca wymusza sposób montażu. Mocowanie osłon realizujemy na kilka sposobów. W najnowszych rozwiązaniach stosuje się opaski mocujące szczególnie do osłon jednowarstwowych z matą ochronną, czasem do powłokowych. Osłony dwuwarstwowe mocuje się poprzez zgrzewanie lub spawanie. W miejscach, w których wymagany jest dostęp ze względów serwisowych często osłony są przykręcane, by były elementami wielokrotnego użytku. Model osłony będący przedmiotem rozważań tej rozprawy jest przykręcany do obudowy tłumika. Połączenie to zostało zamodelowane w modelu numerycznym przy pomocy elementów kontaktowych. W różnych oprogramowaniach dedykowanych do MES zaimplementowane są różne sposoby definicji elementów kontaktowych. Oprócz elementów „frictionless” zastosowanych w modelu, rozróżniamy jeszcze kilka innych typów elementów kontaktowych, m.in. „bonden”, „no separation” symulujące sklejanie elementów bez ich wzajemnego przemieszczenia się. Kolejnym typem elementu kontaktowego jest „rough” modelujący doskonale szorstki kontakt cierny, w którym nie ma poślizgu oraz „Frictional” czy „Forced Frictional Sliding”, w których można przenosić naprężenia ścinające do określonej wielkości, zanim zaczną się przesuwać względem siebie [104].

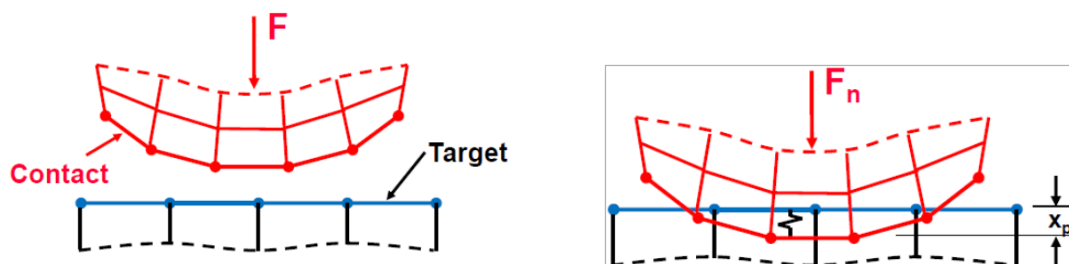
Frictionless to typ połączenia, zastosowany w modelu osłony termicznej. Modeluje standardowy kontakt jednostronny, to znaczy ciśnienie, normalne jest równe zero, jeśli nastąpi separacja połączenia. W ten sposób w modelu mogą powstawać luki między ciałami w zależności od obciążenia. To rozwiązanie jest nieliniowe, ponieważ powierzchnia styku może się zmieniać w miarę wzrostu obciążenia. Ansys oferuje kilka sposobów definicji połączeń kontaktowych. Pierwszym z nich jest „*On Gauss Point*”, dla którego lokalizacja wykrywania kontaktu znajduje się w punktach całkowania Gaussa. Jest opcją domyślną pod warunkiem, że algorytm użyty przez oprogramowanie do obliczania określonej pary kontaktowej występuje w wersji *Pure Penalty* lub *Augmented Lagrange*. Następnymi są „*Nodal-Normal From Contact*” oraz *'Nodal-Normal To Target'*, dla których miejsce wykrywania kontaktu

znajduje się w punkcie węzłowym, w którym normalna kontaktu jest prostopadła odpowiednio do powierzchni 'contact body' albo 'target surface'. Ostatnim sposobem definicji połączeń kontaktowych jest metoda „Nodal-Projected normal From Contact”, w której miejsce wykrywania kontaktu znajduje się w punktach węzłowych styku w nakładającym się obszarze powierzchni 'contact' i 'target' [104]. Ponieważ stykające się ciała nie przenikają się, oprogramowanie musi ustalić relację między dwiema powierzchniami, aby zapobiec ich przechodzeniu przez siebie podczas analizy. Zapobieganie wzajemnemu przenikaniu nazywana jest wymuszeniem „kompatybilności kontaktów”. W celu wymuszenia kompatybilności w połączeniu kontaktowym, Ansys oferuje kilka różnych formuł kontaktowych. Formuły te określają zastosowaną metodę rozwiązania. Zaliczamy do nich metodę „Pure Penalty”, która jest opcją domyślną. Ponadto metodę „Augmented Lagrange”, „Normal Lagrange”, „MPC” oraz „Beam”.

W przypadku nieliniowego kontaktu ciał stałych z powierzchniami, a takim jest połączenie osłony termicznej z płaszczem monolitu, można zastosować formuły *Pure Penalty* lub *Augmented Lagrange*.

$$F_{\text{Normal}} = k_{\text{Normal}} \cdot x_{\text{Penetration}} \quad (4.1)$$

Skończona siła kontaktu, F_n , jest pojęciem sztywności kontaktu, k_{Normal} . Im wyższa sztywność styku, tym mniejsza penetracja, x_p , jak pokazano na Rys. 4.5.



Rys. 4.5 Interpretacja połączenia kontaktowego

Dla przypadku idealnego, gdy k_{Normal} jest nieskończenie duże uzyskujemy zerową penetrację. Wspomniane metody oczywiście różnią się między sobą. Główna różnica polega na tym, że metoda *Augmented Lagrange* zwiększa obliczenia siły kontaktu (ciśnienia) o współczynnik λ :

$$F_{\text{Normal}} = k_{\text{Normal}} \cdot x_{\text{Penetration}} + \lambda \quad (4.2)$$

Ze względu na dodatkowy składnik λ metoda rozszerzonego Lagrange'a jest mniej wrażliwa na wielkość sztywności k_{Normal} . Ostatecznie zdecydowano się na zastosowanie metody *Pure Penalty* i detekcję punktów kontaktowych opartą o metodę *On Gauss Point* [104].

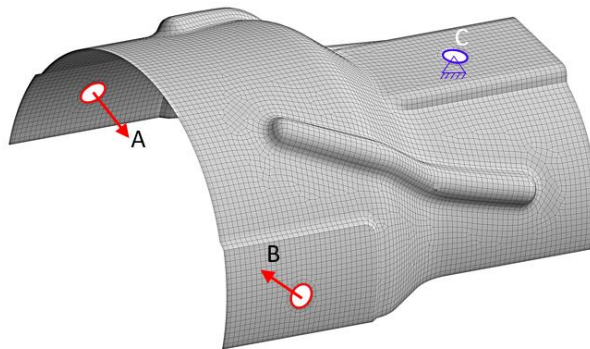
Model numeryczny osłony dla określonej geometrii generowany w zadaniach optymalizacyjnych zbudowany jest z przewagą elementów typu QUAD4. Domyślnie wybierane były elementy typu kwadratowego przy ustawionej opcji „Program controlled = default” tzn. były wybierane elementy z kwadratową funkcją kształtu. W dalszej części rozdziału przedstawiono przeprowadzaną analizę wpływu typu elementów na dokładność wyznaczonych wartości przemieszczeń i naprężeń zredukowanych von Misesa.

Przemieszczenie wstępne o tej samej wartości zostało zadane po obu stronach osłony, w punkcie A i B jak pokazano na Rys. 4.6.

Liczba elementów dla zmiennej geometrii i przy ustawionych parametrach siatki jak,

- wielkość elementu (*size element*) = 1.2mm,
- wielkość elementu dla obszarów niepłaskich (*defeaturing value*) = 1.mm,
- otoczka, jeden lub więcej rzędów elementów regularnych wokół otworu (*wisher value*) = 1.0mm,
- typ elementu (quad, tria)

zmienia się w niewielkim zakresie i wynosi około 8600 elementów. Materiałem osłony jest standardowa stal austenityczna o module Younga $2,1 \cdot 10^5$ MPa, współczynnika Poissona 0,3 i gęstości $7.75 \cdot 10^{-9} \frac{kg}{m^3}$. Osłonę utwierdzono w jednym otworze montażowym (punkt C). W pozostałych dwóch (punkty A i B) zdefiniowano wstępne przemieszczenie początkowe (składowa pozioma) równe 0,25 mm (Rys. 4.6).

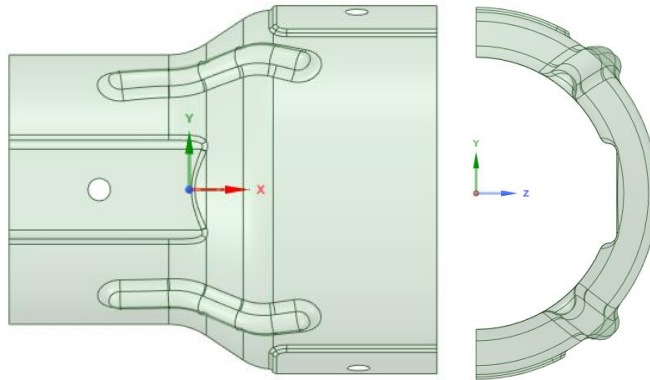


Rys. 4.6 Model numeryczny z warunkami brzegowymi

Do rozwiązania zadań brzegowo-początkowych zaadaptowane zostało komercyjne oprogramowanie ANSYS. Analiza stanu odkształcenia i naprężenia pomiędzy osłoną a obudową wymusiła opracowanie odpowiednich modeli numerycznych umożliwiających rozwiązanie klasycznej analizy sprężystości. Aby uzyskać rozwiązania dla sprzężonej analizy termomechanicznej, zagadnień kontaktu oraz analiz dynamicznych rozbudowano aktualny model osłony o dodatkowe parametry i warunki brzegowe. Ponadto gradient temperatury w całym modelu przyjęto $\Delta T = 60^\circ C$, co dało w równaniu (3.62) stałe wartości wektora siły węzłowej F_T i nie wymagało dodatkowego rozwiązywania problemu przewodzenia ciepła. Na Rys. 3.34 pokazano schemat definiujący pełną pętlę obliczeń rozpoczynając od wygenerowania geometrycznego modelu parametrycznego, poprzez ciąg symulacji strukturalnych i modalnych, aż do końcowego wyznaczenia wartości poszukiwanych funkcjonałów. W przypadku wykorzystywania tego schematu w procesie optymalizacji jedno- lub wielokryterialnej, dodatkowo wykorzystano moduł „*Direct Optimization*”.

4.3. Wariant podstawowy - referencyjny

Jako model wyjściowy wybrano model pokazany na Rys. 4.7, który jest modelem porównawczym do całego procesu optymalizacji kształtu i warunków brzegowych.



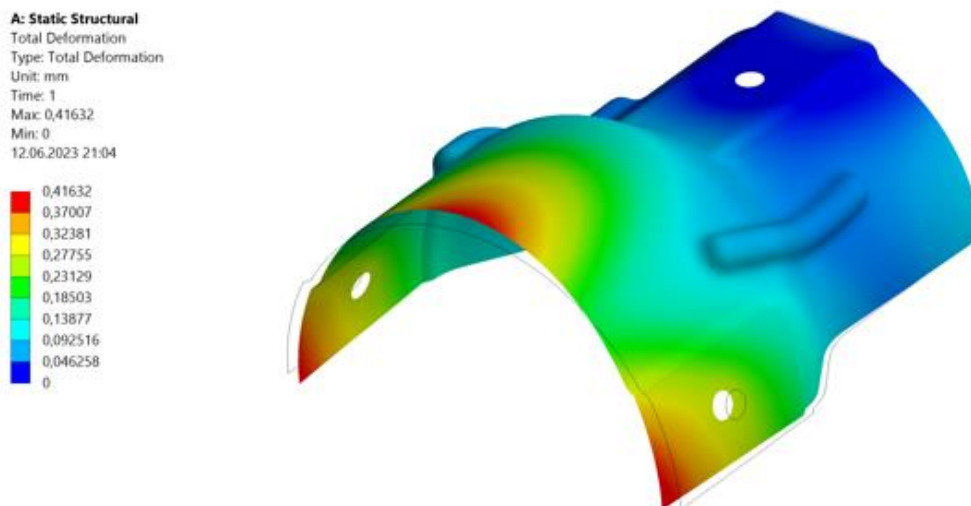
Rys. 4.7 Model geometryczny osłony termicznej - wariant podstawowy

Wartości zmiennych konstrukcyjnych dla tego modelu ujęto w tabeli 4.2. Rozkład przemieszczeń i naprężeń natomiast pokazano na Rys.4.8 i Rys. 4.9.

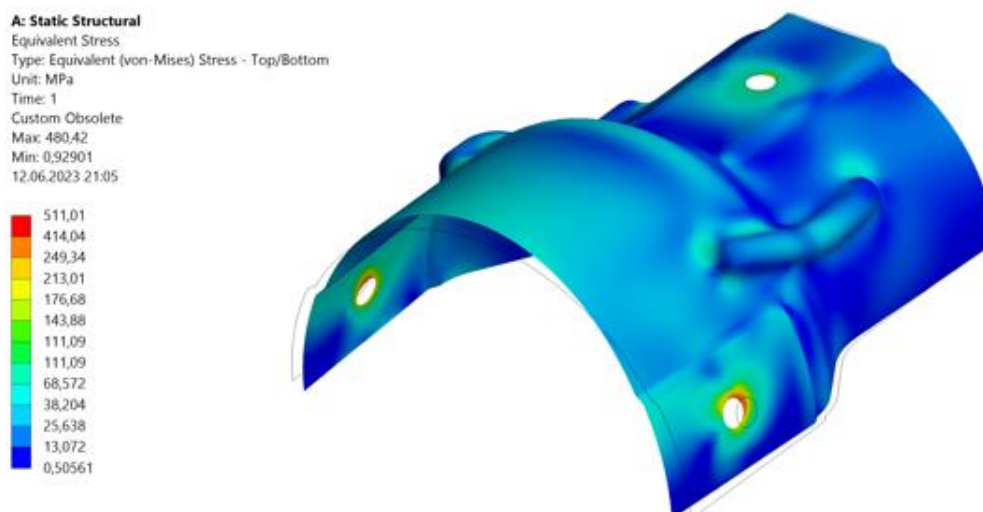
Tabela 4.2 Wartość parametrów dla modelu wyjściowego

Wartości zmiennych konstrukcyjnych dla wariantu podstawowego - referencyjnego (1-10) [mm]									
xo1	yo1	xo2	yo2	xo3	yo3	RR	RL	x1pp	y1pp
9	25	6	-1	-4	1	3	3	-2	0

Wartości zmiennych konstrukcyjnych dla wariantu podstawowego - referencyjnego (11-20) [mm]									
x2pp	y2pp	x3pp	y3pp	x1pl	y1pl	x2pl	y2pl	x3pl	y3pl
-1	16	-5	45	0	-65	-1	-52	5	-20



Rys. 4.8 Osłona termiczna - wariant podstawowy – mapa przemieszczeń



Rys. 4.9 Osłona termiczna - wariant podstawowy – mapa naprężeń redukowanych

Maksymalne naprężenia występujące w modelu referencyjnym wynoszą 511,01MPa, maksymalne przemieszczenia natomiast wynoszą 0,416mm. Wartość pierwszej częstotliwości drgań własnych osłony dla tego modelu jest równa 617,66Hz. Na Rys. 4.8 pokazano odkształcenie osłony pod wpływem wstępnego przemieszczenia (zarys konturu modelu). Kryterium nadrzędnym w procesie optymalizacji jednokryterialnej jest wartość minimalna maksymalnych przemieszczeń osłony przy zdefiniowanych ograniczeniach w postaci maksymalnych naprężeń redukowanych von Misesa i minimalnej wartości pierwszej częstotliwości drgań własnych.

4.4. Dyskretyzacja i dobór siatki

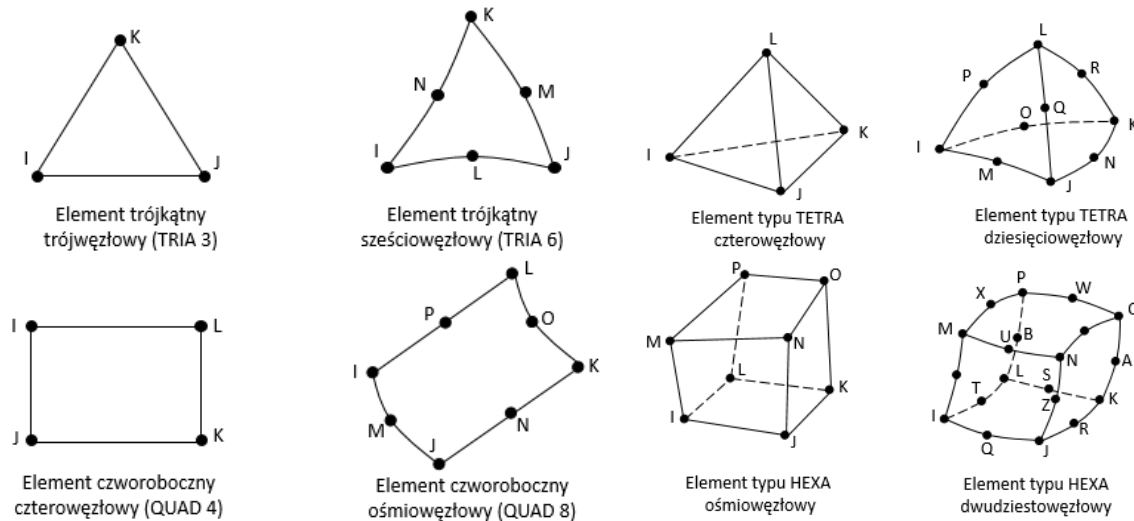
W metodzie elementów skończonych analizowany jest obiekt, który jest obiektem ciągłym, poddanym procesowi dyskretyzacji. Wynikiem dyskretyzacji są elementy skończone, które mogą być różnego typu. Najczęściej stosuje się elementy typu liniowego posiadające tylko węzły w swoich punktach narożnych (na końcach swoich krawędzi) lub można użyć elementów typu nieliniowego, posiadające również węzły w środku każdej swojej krawędzi, a niektóre z nich nawet w środku elementu.

Typ elementów skończonych można opisać za pomocą liczby wymiarów, którymi opisujemy element, kształtu geometrycznego elementu, typu i stopnia wielomianu założonej funkcji kształtu elementu skończonego, liczby węzłów w elemencie oraz rodzaju więzów ogólnych nałożonych na element skończony.

Przemieszczenia punktów wewnętrznych elementu mogą być opisane funkcjami liniowymi lub funkcjami wyższych rzędów. Decydując się na opis równaniami liniowymi uzyskujemy liniowo zmienne wartości przemieszczeń oraz stałe wartości odkształceń i naprężeń w całym obszarze elementu. Dokładniejsze wyniki pozwalają uzyskać elementy posiadające opis przemieszczeń funkcjami kwadratowymi. Jeden dodatkowy węzeł umieszczony w środku długości każdej jego krawędzi pozwala generować zmiennie-kwadratowe przemieszczenia, liniowo-zmienne odkształcenia oraz naprężenia.

Dodatkowe węzły na krawędzi elementu umożliwiają aproksymację pola przemieszczeń wzdłuż tej krawędzi przez wielomian wyższego stopnia. Dla elementu z dwoma węzłami na krawędzi, które są ulokowane tylko na końcach krawędzi elementu, składowe wektora przemieszczenia zmieniają się liniowo, a odkształcenia i naprężenia są stałe. W związku z tym, by uzyskać wymaganą dokładność obliczeń należy wykorzystać większą liczbę elementów skończonych o prostych funkcjach kształtu lub

odwrotnie, mniejszą liczbę elementów o bardziej złożonych funkcjach modelujących. Okazuje się jednak, że stosowanie bardzo złożonej aproksymacji na przykład za pomocą wielomianów stopnia czwartego i wyższych jest niekorzystne. Dlatego też w praktyce, w trudniejszych, nieliniowych zadaniach zalecane jest wykorzystywanie elementów skończonych o prostych funkcjach kształtu. Praktyka pokazuje również, że znacznie łatwiej jest uzyskać dobrą dokładność obliczeń przybliżonych MES dla pola przemieszczeń niż dla naprężeń. Potwierdzają to również zamieszczone wyniki dla analizowanej osłony termicznej. Na Rys. 4.10 pokazano przykładowe elementy wykorzystane w modelu numerycznym.



Rys. 4.10 Wybrane rodzaje elementów skończonych dla zagadnień dwu, i trzymiwymiarowych

Dla każdego elementu skończonego zdefiniowana jest tzw. funkcja kształtu [167], która przybiera różną formę w zależności od typu elementu skończonego, który w procesie dyskretyzacji modelu ma istotny wpływ na uzyskanie poprawnych wyników.

Dla przykładowego elementu trójkątnego, funkcja kształtu wraz z wartościami przemieszczeń w węzłach elementu opisuje poszukiwane pola przemieszczeń wewnątrz elementu skończonego:

$$\{u\} = [N(x, y, z)] \{q\}_e \quad (4.3)$$

gdzie:

$\{q\}_e$ - jest wektorem przemieszczeń węzłowych

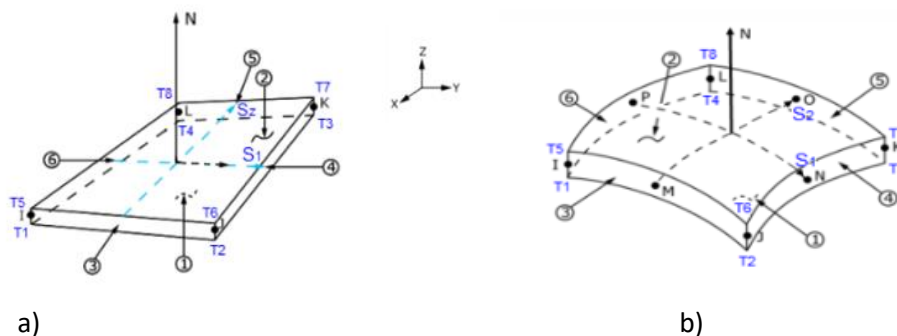
$[N]$ – nazywana jest macierzą kształtu

$$\begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix} = \begin{bmatrix} N_1(x, y) & 0 & N_2(x, y) & 0 & N_3(x, y) & 0 \\ 0 & N_1(x, y) & 0 & N_2(x, y) & 0 & N_3(x, y) \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \quad (4.4)$$

Trochę bardziej rozbudowaną postać przyjmuje (funkcja kształtu) dla popularnego elementu jakim jest QUAD, czterowięzłowy element płaski typu SHELL. W różnych programach obliczeniowych nazywany jest różnie, w środowisku ANSYS nosi nazwę SHELL181 i SHELL281.

Do celów obliczeniowych zastosowano elementy płaskie (SHELL) jak i przestrzenne (HEXA,PENTA). Trzy z nich zostały wybrane do analizy porównawczej. Pierwszym z nich jest element typu SHELL181, który nadaje się do analizy cienkich i średnio grubych struktur skorupowych. Jest to element czterowęzłowy z sześcioma stopniami swobody w każdym węźle: przemieszczenia w kierunkach x, y i z oraz obroty wokół osi x, y i z. Jeśli używana jest opcja membrany, element ma tylko translacyjne stopnie swobody. Element doskonale nadaje się do zastosowań liniowych, o dużym obrocie i/lub nieliniowych zastosowaniach o dużych odkształceniach. Zmiana grubości powłoki jest uwzględniana w analizach nieliniowych. Element może być używany w aplikacjach warstwowych do modelowania modeli kompozytowych lub konstrukcji warstwowych.

Rys. 4.11a przedstawia geometrię, lokalizację węzłów, kierunek normalny dla tego elementu.



Rys. 4.11 Geometria, lokalizacja węzłów i kierunek normalny elementu typu SHELL181 i SHELL281

Bliźniaczy element z podwojoną liczbą węzłów oznaczony jako SHELL281 posiada kwadratową funkcję kształtu. Posiada 8 węzłów. Sam element można opisać podobnie jak SHELL181, co zostało pokazane na Rys. 4.11b.

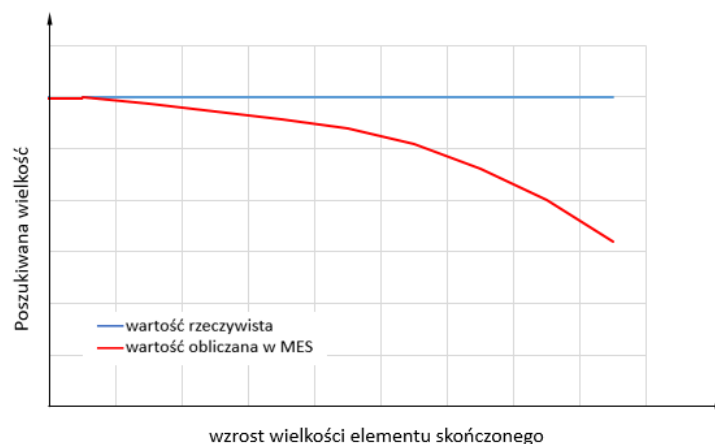
Trzecim typem elementów, który został użyty w analizie porównawczej to element typu TRIA. W ramach analizy zastosowano zarówno element trójkątny z trzema jak i sześcioma węzłami jak pokazano na Rys. 4.10. Do obliczenia naprężenia w dowolnym punkcie elementu niezbędna jest znajomość pola przemieszczeń, a nie ich wartości dyskretnych w węzłach. Stąd bierze się konieczność interpolacji przemieszczeń pomiędzy węzłami elementu i do tego używane są funkcje kształtu. Dla elementu trójkątnego w i-tym węźle bazując na równaniu 4.3 przyjmuje postać:

$$N_i = [x_j y_k - x_k y_j + (y_j - y_k)x + (x_k - x_j)y] / (2\Delta) \quad (4.5)$$

gdzie: x i y to współrzędne węzła, a Δ jest powierzchnią trójkąta o wierzchołkach i, j, k.

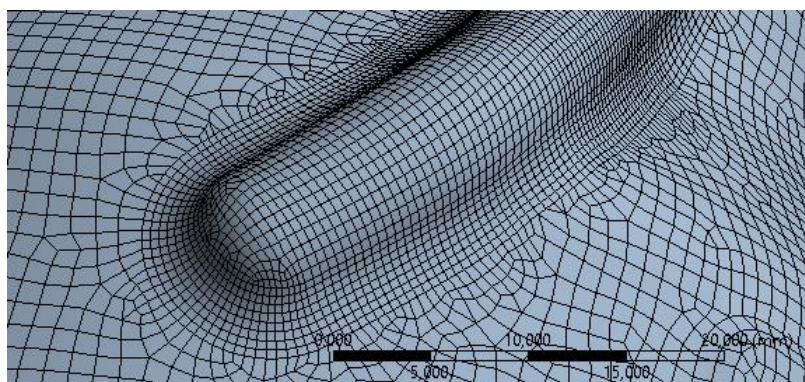
W zagadnieniach liniowych i związanych z nimi symulacjami dokładność obliczeń zależy między innymi od wielkości zdefiniowanego elementu skończonego (Rys. 4.12). Z reguły jest tak, że im wielkość elementu skończonego jest większa, tym dokładność obliczeń mniejsza. Zmniejszanie wielkości elementu powoduje niestety wzrost ich liczby, co skutkuje wydłużeniem czasu obliczeń. Jest to powiązanie z ilością równań algebraicznych, które należy rozwiązać. Ilość ta jest wprost proporcjonalna do ilości węzłów w modelu. Jednym ze sposobów optymalizacji wielkości elementu skończonego w zagadnieniach statycznych jest rozpoczęcie obliczeń z utworzoną siatką elementów skończonych złożonej z niewielkiej ilości elementów. Następnie podwajamy ilość elementów i ponawiamy obliczenia. Jeżeli wyniki po kolejnych dwóch pętlach zagęszczania siatki niewiele się różnią to przyjmuje się wielkość elementu za wystarczająco dokładną. Różnica w uzyskanych wynikach pomiędzy dwoma pętlami obliczeń w procesie poszukiwania optymalnej wielkości elementu powiązana jest odpowiedzialnością konstrukcji. Dla modeli obciążonych wysokim współczynnikiem

bezpieczeństwa (6-10) oczywiście ta różnica nie powinna być większa niż 1%. Dla modeli o ‘mniejszej odpowiedzialności’ konstrukcyjnej akceptowane są również różnice w granicach 3-5%, a czasem nawet wyższe. Poniższy Rys. 4.12 przedstawia ogólną zależność wielkości elementu skończonego od dokładności obliczeń. Jak można zauważyć, nie jest to zależność liniowa. Stąd, każdy proces zagęszczania siatki niesie ze sobą ryzyko ‘przewymiarowania’ modelu numerycznego, który zostanie obarczony zbyt dużą ilością elementów i węzłów skutkujące zbyt dużym czasem obliczeń i nieefektywnym jego wykorzystaniem.



Rys. 4.12 Zależność dokładności wyników od wielkości elementów

Równomierne zagęszczanie siatki dyskretyzacyjnej jest nieefektywne, ponieważ wzrost liczby stopni swobody modelu dyskretnego prowadzi do zwiększania kosztów obliczeń a również do wzrostu błędu obliczeń ϵ_e . Dlatego należy rozważyć opcję zróżnicowania wielkości elementu skończonego w zależności od przewidywanych gradientowych pól naprężeń. Podział na elementy skończone można wtedy tak przeprowadzić, by gęstość dyskretyzacji obszaru odpowiadała przewidywanym gradientom pola naprężeń. W praktyce oznacza to, że w miejscach i w kierunkach, gdzie przewidujemy szybkie zmiany stanu naprężenia, wymiary elementów skończonych powinny być najmniejsze. Dla osłony termicznej takimi miejscami dostosowanymi do przewidywanego rozkładu pola naprężeń są promienie przejścia w obrębie modelowanych przetłoczeń, które zostały pokazane na Rys. 4.13



Rys. 4.13 Dyskretyzacja modelu numerycznego

Aby optymalnie dobrać wielkość i rodzaj elementu skończonego dla modelowanej osłony termicznej, i być pewnym, że powyżej wymienione czynniki nie wpływają ujemnie na wyniki obliczeń przeprowadzono wstępne obliczenia dla dwóch zdefiniowanych modeli M1 i M2 (zdefiniowane w

rozdziale 4.5), dla których porównano zredukowane naprężenia von Missesa i przemieszczenia dla osłony. Dodatkowo została obliczona również częstotliwość drgań własnych osłony, ponieważ w ramach optymalizacji jednokryterialnej jest ono kryterium ograniczającym. Obliczenia przeprowadzono dla kilku wariantów różniących się między sobą wielkością elementów skończonych, rodzajem elementów (QUAD lub TRIA) rodzajem funkcji kształtu (liniowa lub kwadratowa) oraz wartościami dwóch parametrów, charakterystycznych dla generowanej siatki elementów, a mianowicie maksymalną wielkością elementów na krzywiznach modelu (*'defeaturing size'*) oraz wielkością elementów wokół otworów (*'whisher'*).

Wyniki obliczeń zostały zestawione w tabeli 4.3.

Tabela 4.3. Wyniki dla różnej gęstości siatki i typów elementów skończonych

Wariant	Rodzaj elementu	σ [MPa]	s [mm]	f [Hz]	$ \frac{\Delta\sigma}{\sigma} $	$ \frac{\Delta s}{s} $	$ \frac{\Delta f}{f} $	Liczba elementów	Czas obliczeń [s]
A (bazowy)	Q (8 węzłów)	351,14	0,52765	671,01				8568	25
A	L(4 węzły)	333,91	0,526	666,86	4,91%	0,35%	0,62%	8575	28
B	L	340,36	0,527	667,48	3,07%	0,09%	0,53%	32644	220
	Q	348,5	0,5281	671,89	0,75%	0,08%	0,13%	32637	331
C	L	281,94	0,5182	665,31	19,71%	1,79%	0,85%	2324	15
	Q	382,72	0,526	667,3	8,99%	0,39%	0,55%	2326	20
D	L (3 węzły)	333,91	0,525	666,85	4,91%	0,50%	0,62%	8570	23
	Q(6 węzłów)	351,15	0,527	671,08	0,00%	0,12%	0,01%	8575	27

gdzie:

- A – wielkość elementu = 1.2mm, *defeaturing* = 0.8mm, *wisher* = 1mm, 8560 elementów
- B - wielkość elementu = 0.6mm, *defeaturing* = 0.3mm, *wisher* = 0.5mm, 32644 elementów
- C - wielkość elementu = 2.4 mm, *defeaturing* = 1mm, *wisher* = 1mm, 2324 elementów
- D - wariant A z elementami typu TRIA, 8575 elementów,
- L – element z liniową funkcją kształtu, (czterowęzłowy dla elementu typu QUAD lub trzywęzłowy dla elementu typu TRIA)
- Q – element z kwadratową funkcją kształtu, (ośmiowęzłowy dla elementu typu QUAD lub sześciowęzłowy dla elementu typu TRIA)

Wartości podane w tabeli 4.3 obliczono wg wzoru:

$$\left| \frac{\Delta x}{x} \right| = \frac{x_{AQ} - x_i}{x_{AQ}} \quad (4.6)$$

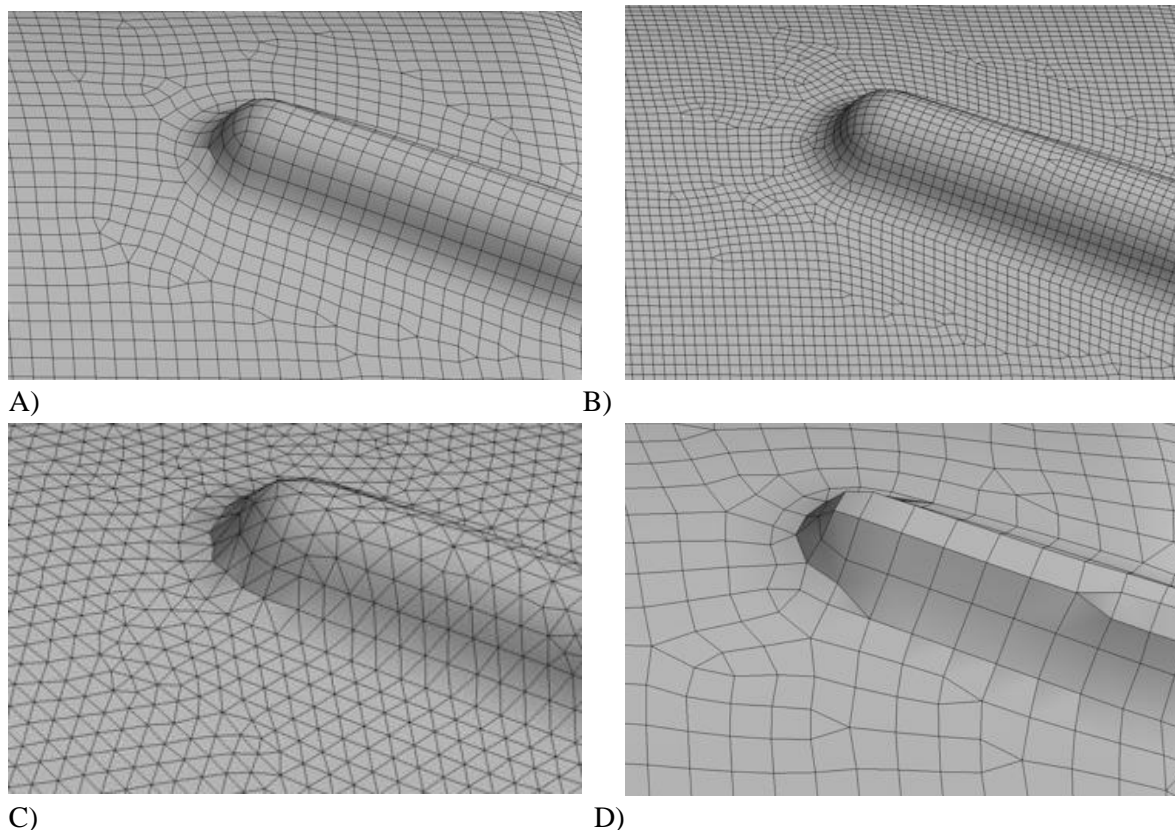
gdzie:

x_{AQ} – wartość dla wariantu A z elementami typu Q (wariant bazowy)

x_i – wartość dla danego wariantu

Jak można zauważyć w tabeli 4.3, dla obliczonego maksymalnego naprężenia von Misesa, maksymalnego przemieszczenia i wartości pierwszej częstotliwości drgań własnych różnice kształtują się pomiędzy 0% - 19,71%. Największe różnice można zauważyć w wartościach naprężeń. Najmniejsze są dla przemieszczeń. Najwyższe różnice dla wszystkich trzech funkcjonalów występuje dla wariantu C, w którym zastosowano elementy dwukrotnie większe od przyjętych ostatecznie w modelu numerycznym. Potwierdza to przypuszczenia, że dla zbyt dużych elementów w stosunku do gabarytów modelu, kształtu i jego cech konstrukcyjnych (promień przejścia, przetłoczenia) jakość siatki jest

niezadawalająca (zbyt duża ilość zniekształconych elementów), a co za tym idzie, wyniki są obarczone wyższym błędem.



Rys. 4.14 Siatka elementów skończonych dla wariantów A, B, C i D.

Wartość funkcjonałów dla modelu z elementami dwukrotnie mniejszymi (wariant B, Rys. 4.14) z kwadratową funkcją kształtu są prawie identyczne z modelem bazowym A z elementami nieliniowymi. Różnica w wynikach jest mniejsza niż 1%. W przypadku elementów liniowych, różnica pomiędzy wartościami maksymalnych naprężeń wynosi już 3,07%. Wadą modelu B był stosunkowo długi czas obliczeń w porównaniu z innymi wariantami. Dla modelu z elementami nieliniowymi – wariant B, czas obliczeń był aż 9-12 krotnie dłuższy w stosunku do wariantu podstawowego. Dla funkcjonału przemieszczeniowego, można stwierdzić, że zaistniałe różnice są bardzo małe i pomijając wariant C, różnica ta nie przekracza 0.5%. Przy czym należy zaznaczyć, że te 0.5% występuje w modelu z elementami trójkątnymi. Kiedy weźmiemy pod uwagę tylko modele z elementami typu QUAD, różnica ta wynosi zaledwie 0,35%. Dla modelu z elementami trójkątnym wartość obliczonych naprężeń jest identyczna z modelem referencyjnym. Po analizie powyższych wyników zdecydowano, że w dalszym procesie optymalizacyjnym model numeryczny będzie oparty o elementy typu QUAD (przeważająca ilość) z kwadratową funkcją kształtu, takich jak w modelu referencyjnym. W porównaniu z wariantem B dodatkowo wykazał się znacznie krótszym czasem obliczeń. Ma to znaczenie w procesie optymalizacji tejże osłony, w którym funkcja celu jest wywoływana co najmniej kilka tysięcy razy. W literaturze [136] można spotkać jeszcze inne sposoby oceny siatki elementów skończonych. Jednym z nich jest zasada Melosha, którą opisuje poniższe równanie:

$$F - \tilde{F}_1 \approx c \cdot h_i^r \quad (4.7)$$

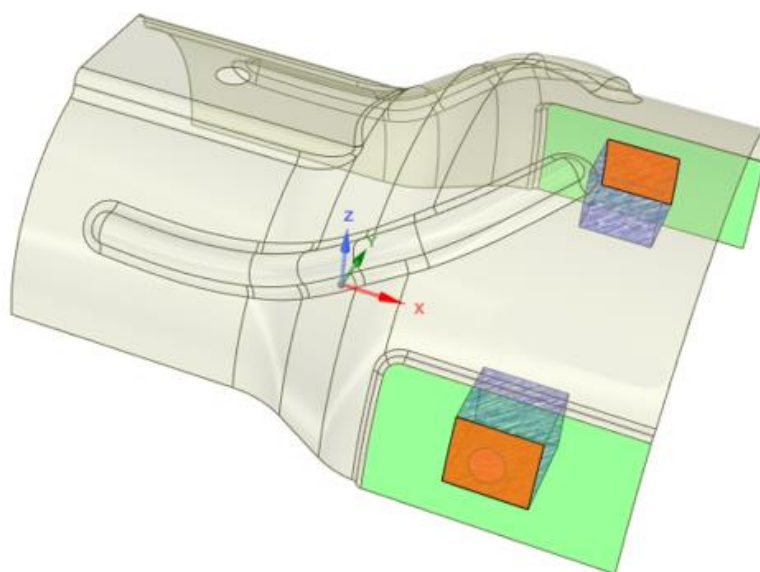
Wyrażenie po lewej stronie równania jest różnicą między rozwiązaniem ścisłym a rozwiązaniem metodą elementów skończonych przy i -tym (na początku $i = 1$) podziale na elementy skończone. Po prawej stronie zależności natomiast jest iloczyn h_i , czyli średniej wielkości największych elementów skończonych badanego fragmentu konstrukcji (h_i rozumianej, jako średnica koła wpisanego w element) podniesionej do potęgi r i wielkości c . Zarówno r , jak i c nie zależą od liczby elementów. r uwzględnia rodzaj elementu skończonego (stopień wielomianów interpolacyjnych), a stała c , w zamierzeniu, zależy od „konfiguracji konstrukcji”. Tak określony błąd metody stanowi podstawę do oszacowania jego wielkości na drodze równomiernego zagęszczania siatki elementów. Pierwszy podział konstrukcji na elementy skończone nie pozwala nam na wyznaczenie błędu, bo nie znamy stałych c i r . Ponowny podział na elementy, tym razem dwa razy mniejsze liniowo, daje możliwość do wyznaczenia tego błędu poprzez ekstrapolację Richardsona:

$$F - \tilde{F}_2 \approx \frac{\tilde{F}_2 - \tilde{F}_1}{\left(\frac{h_1}{h_2}\right)^r - 1} \quad (4.8)$$

Jak pokazują badania eksperymentów numerycznych, wykładnik r osiąga wartości poniżej jedynki. Przyjmując konserwatywnie $r = 1$ zawyżamy błąd, co stanowi pewien margines bezpieczeństwa. Stosunek wymiarów h_1 i h_2 jest więc równy 2. W wyniku czego mianownik równania 4.8 wynosi 1. Metoda ta jednak jest rzadko stosowana, ze względu na trudności w wyznaczeniu stałej c i promienia r .

4.5. Model uproszczony

Model wstępny zdefiniowany do pierwszych poglądowych symulacji uwzględniał wszystkie aspekty techniczne mające wpływ na ostateczne zachowanie się osłony jako modelu. Został opisany za pomocą określonej liczby parametrów. By stwierdzić, czy model wstępny może zostać uproszczony należało przeprowadzić szereg obliczeń numerycznych. Do tego celu utworzono dwa modele M1 i M2. Jeden (M1) zawierający elementy kontaktowe oraz drugi (M2), który ich nie zawierał. Rys. 4.15 przedstawia model wstępny składający się z osłony (powłoka cienkościenna typu shell tzw. ‘target body’), bloków reprezentujących podzespół (bryły typu solid – kolor niebieski, tzw. ‘Contact body’), do którego osłona jest przykręcana.

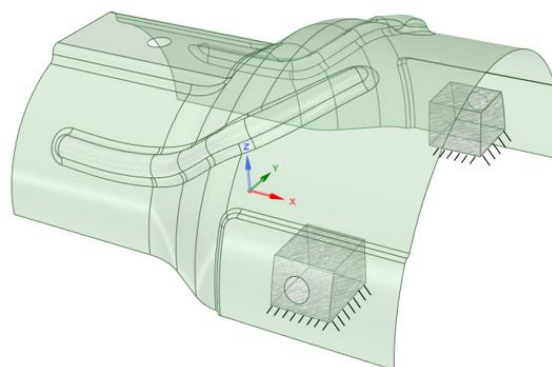
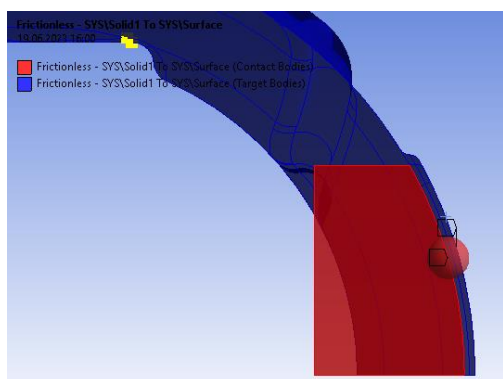


Rys. 4.15 Model geometryczny z elementami kontaktowymi

W pierwszym modelu (M1) zostały zdefiniowane elementy kontaktowe typu „*frictionless*”, które odwzorowały kontakt osłony z podstawą (kolor pomarańczowy).



- a) *target body* – powierzchnia styku osłony (powierzchnia zielona ograniczona żółtą ramką)
 b) *contact body* – powierzchnia styku brył (kolor pomarańczowy)



- c) promień przestrzeni poszukiwanych powiązań
 d) utwierdzenie brył

Rys. 4.16 Definicja elementów kontaktowych w modelu M1

Powierzchnia osłony pokazana na Rys. 4.16a (kolor zielony) definiuje obszar elementów kontaktowych dla - *target body*, a powierzchnia bryły równoległa do niej pokazana na Rys. 4.16b reprezentuje obszar od strony podstawy – tzw. – *contact body*. Pomiędzy nimi jest zdefiniowana szczelina wynikająca z warunków technologicznych. Szerokość przestrzeni pomiędzy elementami skończonymi osłony i brył podpierających, która jest przeszukiwana, zdefiniowano za pomocą parametru „Pinball radius” co pokazano na Rys. 4.16c. Reprezentowany jest za pomocą kuli, której promień określony jest wartością parametru „pinball radius”. W ramach tej pracy wartość tego parametru ustawiono na 2mm.

Warunki brzegowe jak wstępne przemieszczenie, jego wartość i kierunek, miejsce przyłożenia oraz utwierdzenie osłony zostały zdefiniowane w oparciu o rzeczywiste warunki pracy osłony. Utwierdzenie brył pokazano na Rys. 4.16d – zostały im odebrane wszystkie stopnie swobody. Badania polegały na poszukiwaniu sposobu uproszczenia modelu, np. próby opisanie modelu mniejszą ilością parametrów, ewentualnie eliminacji elementów kontaktowych, które w znacznym stopniu wydłużały obliczenia MES. Zważywszy na fakt, że model ten będzie poddany symulacjom obejmującym kilka tysięcy wywołań funkcji celu, badania te nabierają dużego znaczenia. Pozostałe warunki brzegowe: dla obu modeli M1 i M2 były identyczne. Analiza mająca potwierdzić lub zaprzeczyć czy dany model wyjściowy można uprościć sprowadzała się do obliczeń i porównania wartości naprężeń i przemieszczeń modelu osłony dla obu wariantów dla tych samych warunków brzegowych.

Dla tak przygotowanych modeli geometrycznych zdefiniowano modele numeryczne i przeprowadzono analizę strukturalną. Wartości naprężeń i przemieszczeń osłony pokazano w tabeli 4.4.

Tabela 4.4 Wartości naprężeń i przemieszczeń dla modelu M1 i M2

Model	Wstępne przemieszczenie [mm]	Naprężenia [Mpa]	Przemieszczenia [mm]
M1	0,25	753,29	0,45796
M2	0,25	768,08	0,44397
delta		1,93%	3,15%

gdzie:

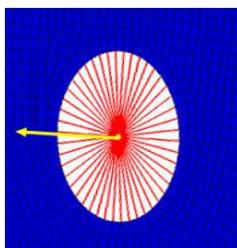
M1 – model zawierający elementy kontaktowe typu ‘*frictionless*’

M2 – model pozbawiony elementów kontaktowych

Błąd względny określony jest wzorem $\Delta = \frac{|M_2 - M_1|}{M_2} \cdot 100\%$

Jak można zauważyć różnice pomiędzy rezultatami analiz numerycznych dla obydwu modeli nie przekraczają paru procent. Różnica wartości naprężeń nie przekracza 2%. Różnica wartości dla przemieszczeń wynosi maksymalnie 3,15%. Zatem można przyjąć do dalszych symulacji model uproszczony, który będzie pozbawiony nieliniowej analizy z uwzględnieniem kontaktu, zachowując jednocześnie wszystkie pozostałe warunki brzegowe.

Wstępne przemieszczenia o wartości 0.25mm w osi Y, jako jeden z warunków brzegowych, zostało zamodelowane za pomocą elementów jednowymiarowych typu MPC (*Multi-Point Constraint*) jak pokazano na Rys. 4.17

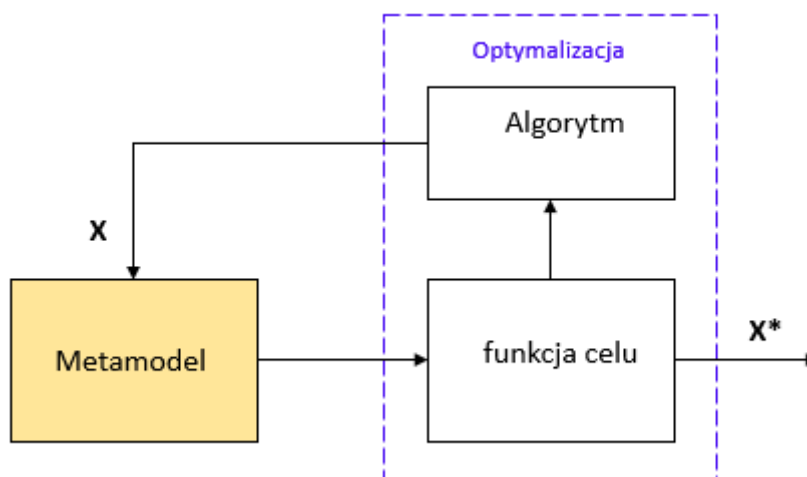


Rys. 4.17 Definicja wymuszenia w postaci numerycznej.

W tego typu połączeniach węzły na obwodzie otworu, zwane węzłami zależnymi (*slave nodes*), są połączone za pomocą elementów typu RBE2 - elementów o nieskończonej sztywności i ‘sprowadzone’ do jednego węzła niezależnego (*master node*), w którym zostało przyłożone wstępne przemieszczenie. Elementy RBE2 to elementy o bardzo wysokiej sztywności.

4.6. Metamodel odpowiedzi numerycznej z zastosowaniem MES

Jednym ze skutecznych sposobów skrócenia czasu obliczeń związanych z rozwiązywaniem zadań optymalizacji rzeczywistych problemów technicznych jest zastosowanie szybko rozwijającej się w ostatnim czasie, strategii opartej na metamodelowaniu. W strategii tej analizowany model zdefiniowanego problemu zastępowany jest metamodelem. (Rys. 4.18).



Rys. 4.18 Schemat blokowy optymalizacji opartej na metamodelu

Jednym z podstawowych zadań metamodelu jest znaczne zredukowanie liczby szczegółowych symulacji analizowanego zadania, które są wymagane przy obliczeniach klasycznymi procedurami optymalizacji. Odpowiednio przygotowany metamodel pozwala na szybkie obliczenie wartości funkcji celu co pozwala na skrócenie czasu obliczeń, bez uszczerbku dla samej prowadzonej analizy i przy zachowaniu wymaganej dokładności. Wykorzystując metamodel, analizowana funkcja celu jest jedynie przybliżeniem rzeczywistej funkcji celu, więc bardzo ważne jest oszacowanie tego przybliżenia. Istnieją różne wskaźniki oceny modelu. Opisano kilka najczęściej stosowanych do oceny Sztucznych Sieci Neuronowych (SSN) wskaźników, do których możemy zaliczyć współczynnik determinacji R^2 , MSE (*Mean Squared Error*) oraz RMSE (*Root Mean Squared Error*). Współczynnik determinacji czasami zwany R-kwadrat jest statystycznym miernikiem oceny modelu określającym dobroć modelu regresyjnego. Jest on więc miarą stopnia, w jakim model pasuje do próby. Przyjmuje wartości z przedziału $[0,1]$. Do estymacji parametrów wykorzystuje metodę najmniejszych kwadratów. Wartość zbliżona do zera oznacza słabą jakość modelu, a wartość zbliżona do jedynki, przeciwnie, oznacza jego bardzo dobre dopasowanie. Czasami może nawet uzyskiwać wartości ujemne, co oznacza, że model jest nawet gorszy od modelu bazowego. Często jego wartości są wyrażane w procentach. Opisany jest wzorem:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \quad (4.9)$$

gdzie:

SSR – suma regresyjna kwadratów odchyłeń, tzn. suma różnic pomiędzy przewidywaną wartością a wartością średnią danych

SSE – suma resztowa kwadratów odchyłeń, tzn. suma różnic pomiędzy wartością pierwotną a wartością przewidywaną

SST – suma całkowita kwadratów odchyłeń, tzn. suma różnic pomiędzy wartością pierwotną a wartością średnią danych

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \geq 0 \quad (4.10)$$

gdzie:

y_i – i-ta obserwacja zmiennej y ,

\hat{y}_i – wartość teoretyczna zmiennej

\bar{y} – średnia arytmetyczna wartości zmiennej

Współczynnik MSE natomiast jest zdefiniowany jako kwadrat zmiany pomiędzy wartościami pierwotnymi i przewidywanymi a ich średnią dla wszystkich wartości:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2 \quad (4.11)$$

gdzie:

y – jest wartością pierwotną, natomiast \hat{y} jest wartością przewidywaną.

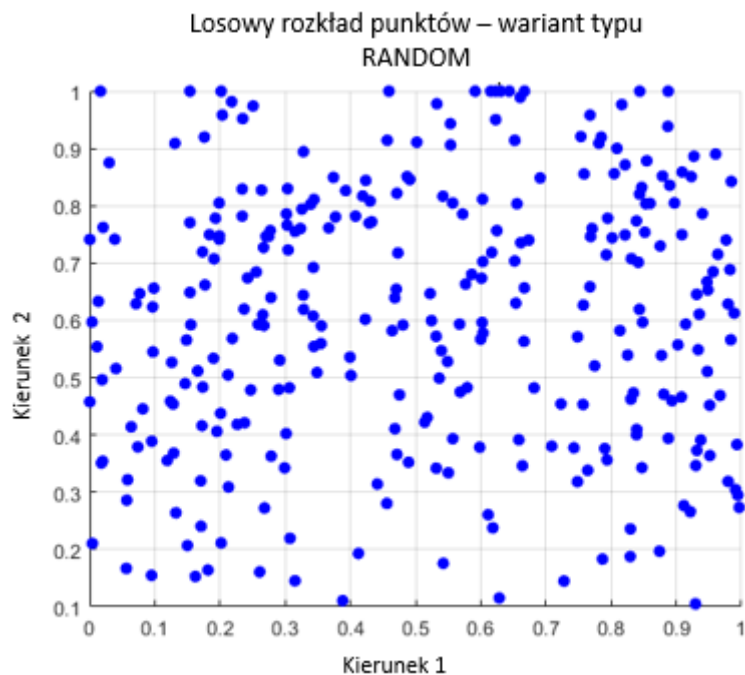
Z kolei RMSE obliczany jest jako pierwiastek z błędu średniokwadratowego. Jak sama nazwa wskazuje obliczamy go pierwiastkując błąd MSE:

$$RMSE = \sqrt{MSE} \quad (4.12)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2} \quad (4.13)$$

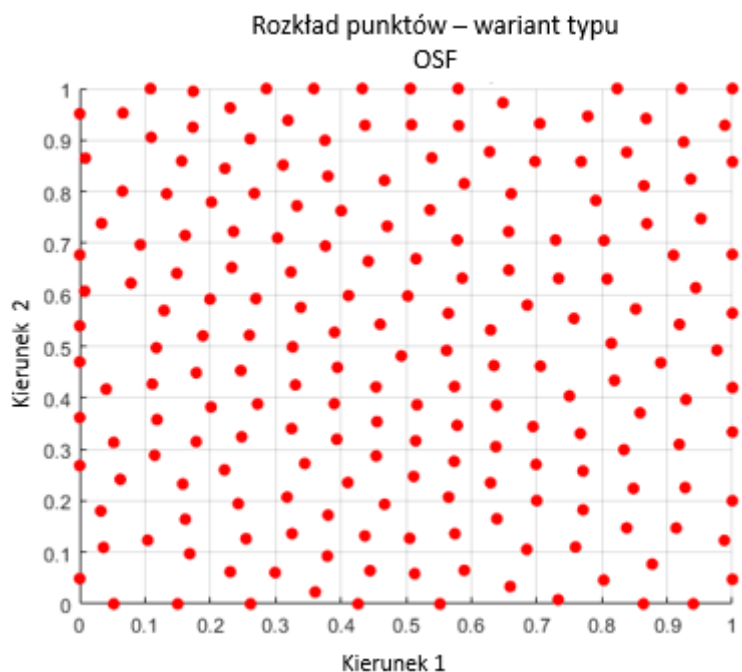
Do budowy metamodelu wykorzystuje się różne metody aproksymacji, które z jednej strony zapewniają uzyskanie założonej dokładności, a z drugiej zaś pozwalają na skrócenie czasu obliczeń symulacyjnych. Przykładowo można wykorzystać analizę regresyjną, polegającą na dopasowaniu do zbioru danych pomiarowych funkcję wybraną z pewnej grupy funkcji, jednakże najczęściej wykorzystuje się techniki sztucznej inteligencji, a w szczególności sztuczne sieci neuronowe (SSN). Odpowiednio nauczona SSN potrafi prawidłowo zinterpretować sygnały wejściowe i dać prawidłowy sygnał wyjściowy. Dzięki temu, może pełnić rolę metamodelu analizowanego obiektu. Czas odpowiedzi jest znacząco krótszy od symulacji numerycznych z wykorzystaniem MES.

Budowa metamodelu opartego na SSN składa się z kilku etapów. Pierwszym krokiem jest projektowanie sieci. W tym kroku definiowana jest topologia sieci, funkcja celu oraz zbiór rozwiązań dopuszczalnych. Następnie określa się początkowy zbiór składający się z N wektorów wejściowych (próbnych). Każdy wektor zbudowany jest z n zmiennych optymalizacji. Równoległe do zbioru początkowego budowany jest zbiór odpowiedzi. Każdemu wektorowi wejściowemu odpowiada jedna poprawna odpowiedź bazująca na rzeczywistej funkcji celu opisującej problem. Zbiór ten następnie jest dzielony – w zależności od podejścia – na dwa lub trzy podzbiory: uczący, testujący i walidujący w odpowiednich proporcjach. Z reguły zbiór uczący jest największy i obejmuje 70% wszystkich wektorów zbioru początkowego. Natomiast pozostałe dwa są stosunkowo mniejsze i obejmują zazwyczaj po 15% całości zbioru początkowego. W kolejnym kroku następuje proces uczenia sieci. Etap uczenia i testowania sieci neuronowej wymagają zbiorów danych uczących i testujących, przygotowanych w poprzednim kroku. Dane te, w przypadkach analizy rzeczywistych problemów pochodzą z pomiarów lub symulacji komputerowych. W ramach zakresu pracy przygotowano zbiór danych opierając się na dwóch wariantach: typu RANDOM i typu OSF (*Optimal Space Filling = UNIFORM*). Każdy z nich zawiera 1000 chromosomów. W przypadku wariantu typu RANDOM zbiór wektorów danych, został wygenerowany w sposób losowy, tzn. zmienne konstrukcyjne mogły przyjmować wartości z całego swojego zakresu, lecz ich wartość w konkretnym wektorze została ustalana losowo. Statystycznie rozkład wylosowanych chromosomów odpowiadał rozkładowi równomiernemu, a ich rozmieszczenie w przestrzeni rozpatrywanej pokazano na Rys. 4.19



Rys. 4.19 Losowy rozkład punktów dla wariantu typu RANDOM w zbiorze początkowym

W wariantcie typu OSF, zbiór początkowy został przygotowany tak, by wszystkie tzw. ‘Design Points’ (wektory zmiennych konstrukcyjnych) znajdowały się w maksymalnej możliwej odległości od siebie. Maksymalizacja minimalnej odległości pomiędzy poszczególnymi punktami pozwala na równomierne ‘pokrycie’ całego obszaru rozwiązań dopuszczalnych. Przykładowo dla trójwymiarowego obszaru rozwiązań dopuszczalnych efekt rozmieszczenia punktów dla wariantu typu OSF pokazano na Rys. 4.20. Przy pomocy funkcji OSF wbudowanej w oprogramowanie Matlab, został wygenerowany zbiór początkowy składający się z 1000 punktów o 20-tu wymiarach.



Rys. 4.20 Rozkład punktów dla wariantu typu OSF w zbiorze początkowym

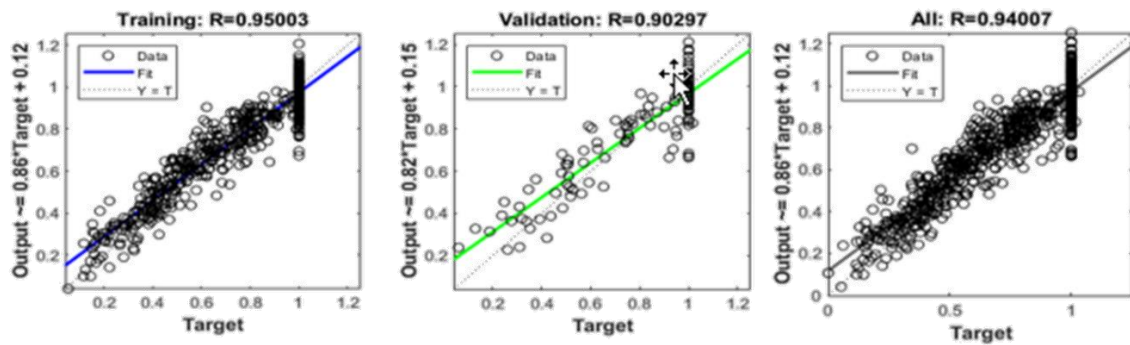
Proces uczenia sieci można uznać za zakończony, jeżeli odpowiedź sieci na wartości ze zbioru testowego i walidacyjnego są zadawalające. Oznacza to, że błąd nie przekracza pewnej założonej wartości i sieć można uznać za metamodel analizowanego problemu. Mając przygotowany i wytrenowany metamodel poszukiwanie rozwiązania optymalnego \mathbf{x}^* funkcji celu (Rys. 4.18) otrzymanej za pomocą metamodelu wykorzystuje się odpowiednią technikę optymalizacji. W ramach pracy wykorzystany został klasyczny algorytm ewolucyjny i algorytm L-SHADE. Wyniki obliczeń przedstawiono w rozdziale 5.3.

Pierwszy metamodel bazuje na zbiorze danych wygenerowany dla 20 parametrów wejściowych, jednego sygnału wyjściowego (przemieszczenie) oraz dla ograniczeń opisanych w rozdziale 3.11. Drugi bazuje na zbiorze danych wygenerowany dla 21 parametrów wejściowych, trzech sygnałów wyjściowych (przemieszczenie, naprężenie, częstotliwość). Celem przygotowania metamodeli jest zastąpienie wykorzystywanych systemów MES w procesie optymalizacji jedno, i wielokryterialnej. Dla metamodelu 'jednokryterialnego', w którym dwa kryteria stanowią ograniczenia metamodel zbudowany jest na zbiorze początkowym danych, wygenerowany dla dwóch wariantów: typu RANDOM i typu OSF. Odpowiednio dla każdego wektora wejściowego został zdefiniowany wektor wyjściowy. W przypadku metamodelu z jednym sygnałem wyjściowym jest to wartość przemieszczenia, a dla metamodelu z trzema sygnałami wyjściowym są to wartości funkcjonałów (3.110, 3.111, 3.112). Dokładność oczekiwanych wyników używając metamodelu zależy od dokładności przygotowania samego modelu. Jakość SSN będąca podstawą metamodelu oceniania jest m.in za pomocą wskaźnika RMSE.

Poszukiwanie odpowiedniej struktury SSN wiąże się z poszukiwaniem optymalnych wartości parametrów sieci, by wartości wyjściowe przez nią generowane były obciążone jak najmniejszym błędem. Lista parametrów, które były dobierane obejmuje:

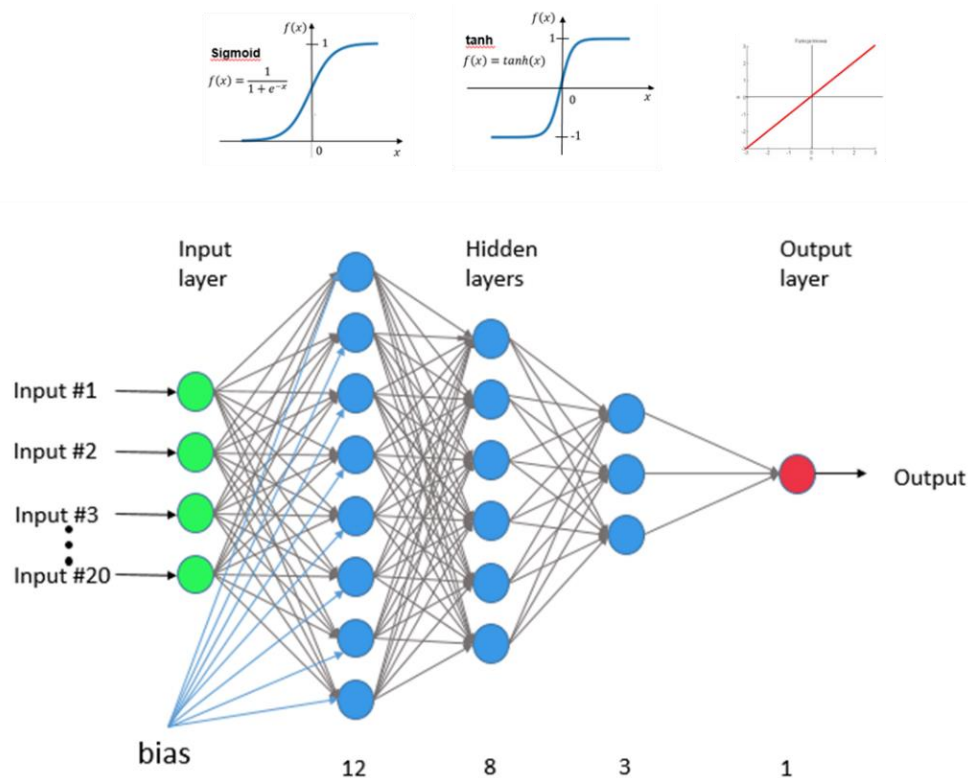
- ilość warstw ukrytych
- ilość neuronów w warstwie ukrytej
- liczbę epok
- typ funkcji aktywacji
- wartość błędu RMSE
- współczynnik regresji
- algorytm uczenia

W przypadku z 20-toma sygnałami wejściowymi i jednym sygnałem wyjściowym budowanie struktury SSN rozpoczęto od jednej warstwy ukrytej i kilku neuronów. Tym samym była to prosta sieć dwuwarstwowa. Model wyjściowy SSN, który należało zbudować posiadał funkcję aktywacji typu 'sigmoid', warstwę wejściową z 20 parametrami, jedną warstwę ukrytą z 10 neuronami oraz warstwę wyjściową z jednym parametrem, wartością przemieszczeń. Liczbę epok jako parametr procesu uczenia sieci ustawiono na wartość 1000. Zbiór danych początkowych podzielono na dwie grupy, pierwsza tzw. trenująca stanowiła 85% wszystkich wektorów wejściowych, a część walidująca 15. Wartość błędu RMSE wynosiła 0,117 dla danych trenujących i 0,1113 dla danych walidujących. Współczynnik regresji pokazano na Rys. 4.21 Jego wartość dla wszystkich danych wynosi 0.94007.



Rys. 4.21 Wartości współczynnika. regresji dla wstępnego modelu SSN

Pomimo akceptowalnej zgodności, w wyniku dalszych modyfikacji utworzono sieć o jeszcze lepszym dopasowaniu. Udoskonalony model SSN został zdefiniowany jako model czterowarstwowy (Rys. 4.22).



Rys. 4.22 Model SSN z jednym sygnałem wyjściowym

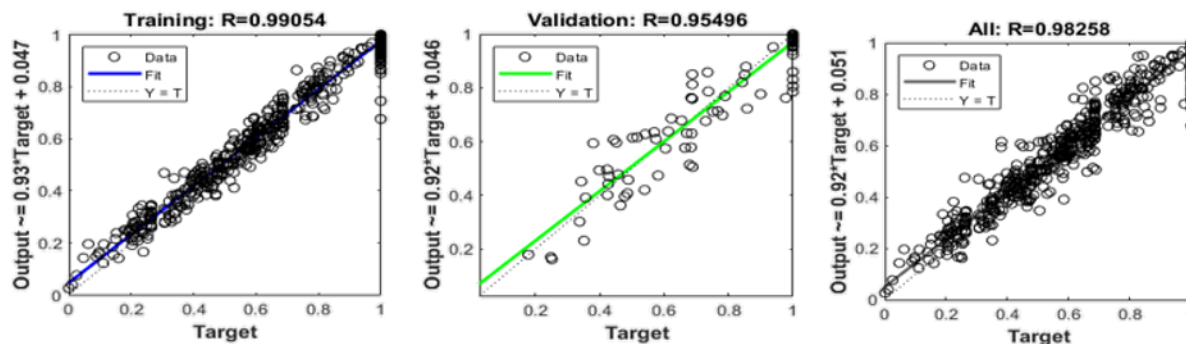
Zbudowany jest z trzech warstw ukrytych, z których każda posiada odpowiednio liczbę neuronów 12, 8, 3. Dla pierwszej warstwy ukrytej wybrano funkcję aktywacji typu 'sigmoid', pozostałe dwie warstwy ukryte posiadają funkcje aktywacji typu 'tanh' (tanges hiperboliczny). Warstwa wyjściowa posiada liniową funkcję aktywacji. Zbiór danych początkowych, tak samo jak poprzednio, podzielono na dwie grupy, pierwsza tzw. trenująca stanowiła 85% wszystkich wektorów wejściowych, a część walidująca 15%. Dane wejściowe poddano normalizacji. Jest to standardowa procedura, kiedy zakresy zmiennych wejściowych różnią się znacznie między sobą. Zakres normalizacji przyjęto [0 1]. Tak przygotowana struktura sieci została poddana procesowi uczenia. Do uczenia sieci zastosowano algorytm Levenberga-Marquardta opisanego w rozdziale 3.4. Uczenie przeprowadzono dla dwóch wariantów. W pierwszym przypadku bazowano na zbiorze początkowym wygenerowanym dla wariantu typu RANDOM i drugi raz dla wariantu typu OSF. W wyniku procesu uczenia otrzymano modele SSN, dla których wartości

współczynnika regresji oraz wartości błędu RMSE dla danych znormalizowanych pokazano w tabeli 4.5.

Tabela 4.5 Wartości współczynnika regresji i błędu RMSE dla wyuczonych SSN

	Wartość współczynnika regresji			Wartość błędu RMSE	
	Dane trenujące	Dane walidacyjne	Cały zbiór danych	Dane trenujące	Dane walidacyjne
RANDOM	0,99054	0,95496	0,98258	0,0087	0,0089
OSF	0,99243	0,96415	0,99174	0,0034	0,0051

Wartości współczynnika regresji dla modelu wytrenowanego dla wariantu typu RANDOM pokazano na Rys. 4.23. Wartości błędów dla wariantu typu OSF są dla każdej grupy danych mniejsze od błędów z wariantu typu RANDOM. Ta sama topologia SSN w połączeniu ze zbiorem początkowym bazującym na danych typu OSF generuje dokładniejsze wyniki niż dla danych typu RANDOM. Różnice te jednak nie są duże i patrząc na wartości błędów rzeczywistych, różnią się o 0.0013mm dla danych walidujących i o 0.0034mm dla danych trenujących.



Rys. 4.23 Wartość współczynnika regresji dla wytrenowanego modelu – typu RANDOM

W tabeli 4.6 pokazano wartości błędu RMSE dla wartości znormalizowanych i rzeczywistych generowane przez zdefiniowaną SSN.

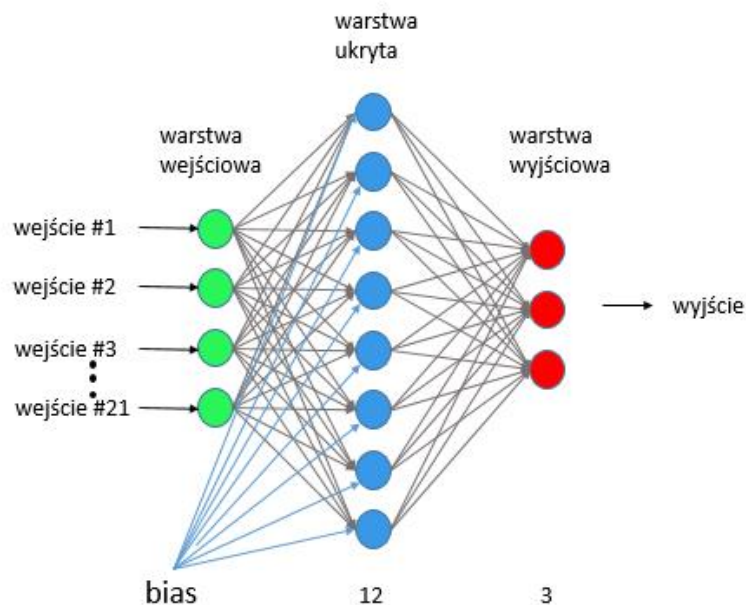
Tabela 4.6 Porównanie wyników metamodelu z MES dla dwóch wariantów, OSF i RANDOM.

		RANDOM	OSF (UNIFORM)
Dane trenujące	RMSE	0,0087	0,0034
	RMSE_rz [mm]	0,0133	0,0099
Dane walidujące	RMSE	0,0089	0,0051
	RMSE_rz [mm]	0,0182	0,0169

Wielkość błędu rzeczywistego poniżej 0,01mm dla danych trenujących (wariant OSF) uznano za satysfakcjonujący.

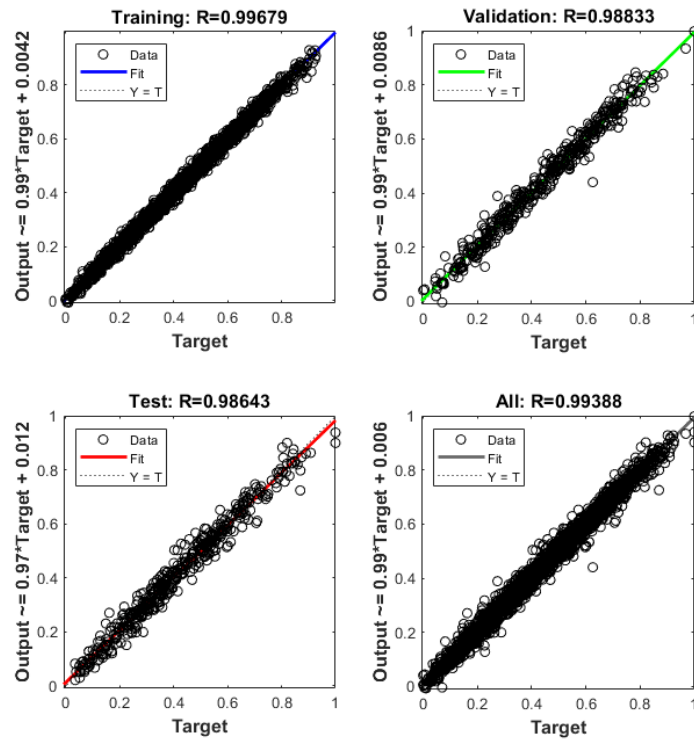
W przypadku metamodelowania dla trzech sygnałów wyjściowych model SSN został opracowany według tego samego algorytmu postępowania co poprzednio. Wybrany został model dwuwarstwowy. Posiada 21 sygnałów wejściowych i jedną warstwę ukrytą z dwunastoma neuronami i sigmoidalną funkcją aktywacji. Warstwa wyjściowa posiada trzy sygnały wyjściowe i liniową funkcję aktywacji (Rys. 4.24). Zbiór danych początkowych podzielono na trzy grupy, pierwsza tzw. trenująca stanowiła 70% wszystkich wektorów wejściowych, część testująca 15%, a część walidująca również 15%. Dane wejściowe poddano normalizacji. Zakres normalizacji przyjęto [0 1]. Tak przygotowana struktura sieci

została poddana procesowi uczenia. Do uczenia sieci zastosowano algorytm Levenberga-Marquardta. W wyniku tego procesu otrzymano model SSN, dla którego współczynnik regresji wyniósł 0,9938.

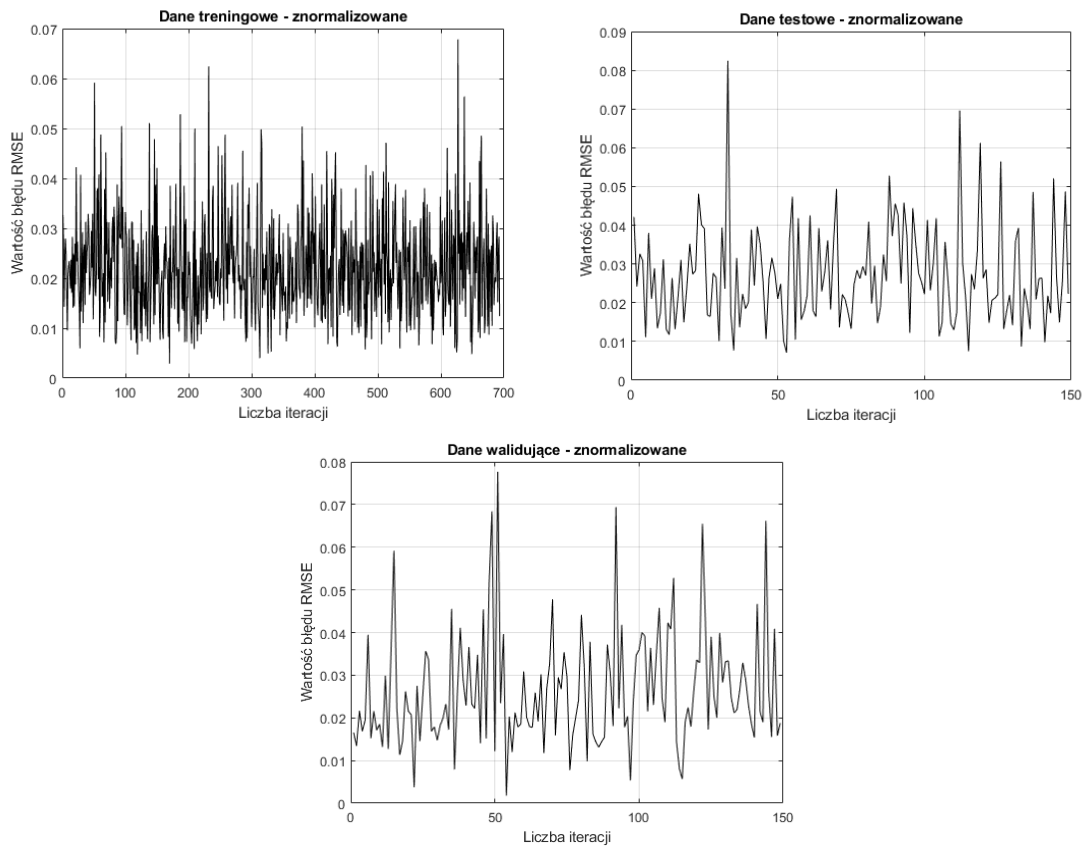


Rys. 4.24 Struktura SNN dla modelu z trzema sygnałami wyjściowymi

Dane zbioru początkowego zostały wygenerowane przy pomocy oprogramowania Matlab [72] i komercyjnego oprogramowania Ansys. Dane te zostały wygenerowane dla wariantu RANDOM. Liczba wektorów wejściowych wynosi 1000. Wartości współczynnika regresji dla wszystkich grup danych pokazano na Rys. 4.25. Jak można zauważyć zgodność modelu jest na poziomie powyżej 99,67% dla danych trenujących i 99,38% całego zbioru. Dla danych testujących i walidujących przekracza 98,6%. Wartość błędu RMSE dla danych trenujących wynosi 0,0124, dla danych testujących 0,0223, a dla danych walidujących jest równa 0.0188. Wartości błędu RMSE dla wartości znormalizowanych dla sieci uwzględniającej równocześnie wszystkie trzy sygnały wyjściowe pokazano na Rys. 4.26.

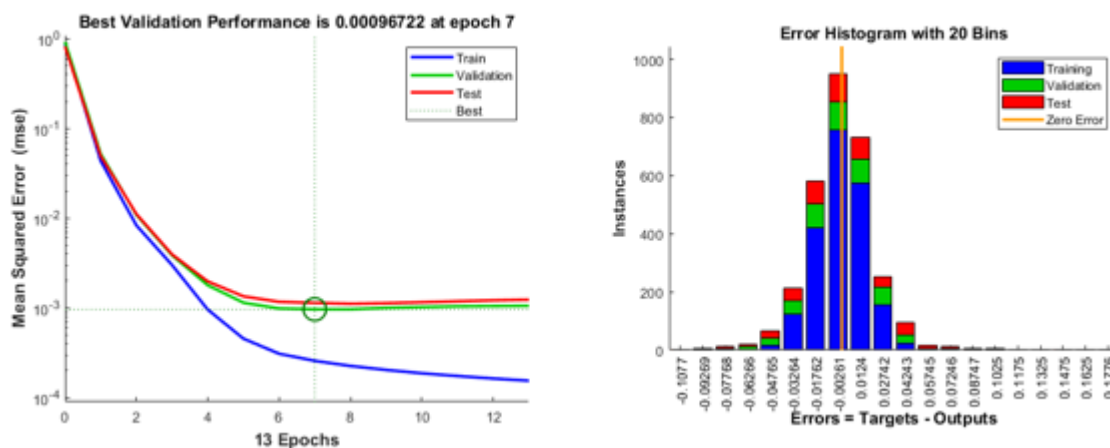


Rys. 4.25 Współczynnik regresji dla SSN z trzema kryteriami



Rys. 4.26 Błąd RMSE dla SSN z trzema kryteriami

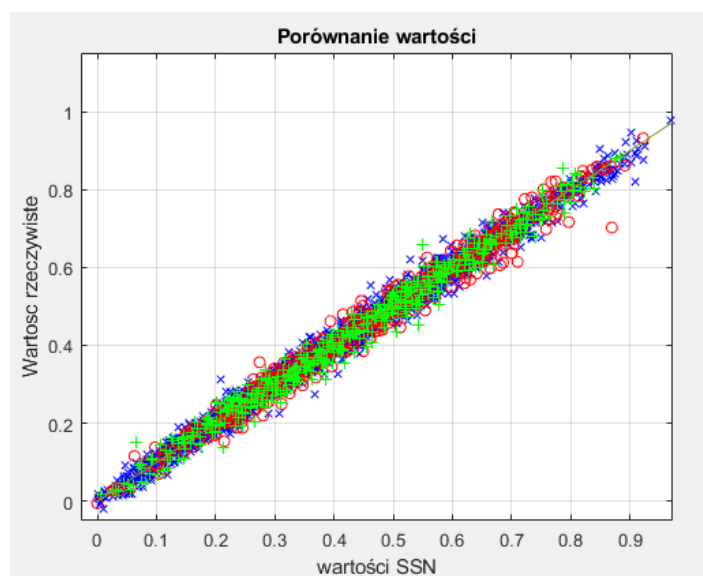
Wartości błędu MSE w procesie uczenia sieci oraz histogram błędu (różnica pomiędzy wartościami wyznaczonymi przez SSN i docelowymi) dla wartości znormalizowanych pokazano na Rys. 4.27.



Rys. 4.27 Wartość błędu MSE w procesie uczenia sieci z trzema sygnałami wyjściowymi

Ponieważ powyższe wartości błędu odniesione są do danych znormalizowanych, nie można bezpośrednio odczytać rzeczywistych wartości błędu. Można to zrobić po odkodowaniu danych i obliczeniu maksymalnego i średniego błędu względnego. Obliczono maksymalny błąd względny dla przemieszczeń rzeczywistych, który wynosił dla każdej z grup danych odpowiednio: treningowe 25%, testujące 28%, walidujące 27%. Wskaźnikiem bardziej miarodajnym w ocenie jakości sieci jest średni błąd względny obliczony dla danych rzeczywistych (odkodowanych). Wynosił on odpowiednio 6.57%, 3,13% oraz 2.44%. Niskie wartości średniego błędu względnego pozwalają stwierdzić, że SSN została poprawnie wytrenowana.

Potwierdzeniem tego jest porównanie danych wyjściowych ze zbioru początkowego (dane docelowe) z danymi generowanymi przez SSN, co pokazano na Rys. 4.28.



Rys. 4.28 Porównanie danych docelowych z przewidywanymi z SSN

Bardzo dobra zgodność danych rzeczywistych z danymi generowanymi przez SSN pozwala sądzić, że opracowany metamodel może być wykorzystany w procesie optymalizacji osłony termicznej.

5. Rezultaty optymalizacji

5.1. Optymalizacja jednokryterialna

Jak wspomniano w ramach pracy wykorzystano sześć algorytmów optymalizacji jednokryterialnej. Trzy z nich są wbudowane w oprogramowanie Ansys, a trzy są algorytmami zewnętrznymi. Do algorytmów wbudowanych należą NLQPL (Nonlinear Programming by Quadratic Lagrangian), MISQP (Mixed-Integer Sequential Quadratic Programming) oraz ASO (Adaptive Single Objective). Natomiast z grypy algorytmów zewnętrznych wybrano algorytm oparty o algorytm sympleksu Nelder-Meada, algorytm ewolucji różnicowej L-SHADE oraz klasyczny algorytm ewolucyjny (EA). Niektóre z tych algorytmów wymagają zdefiniowania tzw. punktu startowego (NM, NLQPL, MISQP).

Dla każdej optymalizacji z wykorzystaniem wszystkich algorytmów populacyjnych zdefiniowano wielkość populacji równą 25, liczba iteracji równą 20 oraz liczbę parametrów wynoszącą 20.

Dla algorytmu ewolucyjnego zdefiniowano zakres mutacji Gaussa równej 0.9, prawdopodobieństwo mutacji Gaussa 0.7, prawdopodobieństwo mutacji jednorodnej 0.1, prawdopodobieństwo krzyżowania prostego 0.1 oraz prawdopodobieństwo krzyżowania arytmetycznego 0.1.

W przypadku algorytmów wbudowanych w oprogramowanie Ansys można było dodatkowo sterować współczynnikiem zbieżności algorytmu. Stanowił on jeden z wielu zdefiniowanych kryteriów stopu.

W każdym przypadku liczba wywołań funkcji celu wynosiła 500 ± 20 . Pełną kontrolę nad parametrami sterującymi algorytmem posiadano tylko dla algorytmów zewnętrznych, co stwarza dodatkowe możliwości obliczeń wariacyjnych. W algorytmach wbudowanych w środowisko Ansys, niektóre parametry zostały ustawione domyślnie i możliwość ich modyfikacji była niemożliwa. Kolejnym parametrem wspólnym dla obu typów algorytmów była zadeklarowana liczba iteracji. To kryterium, powszechnie stosowane, daje pewność zakończenia obliczeń algorytmu.

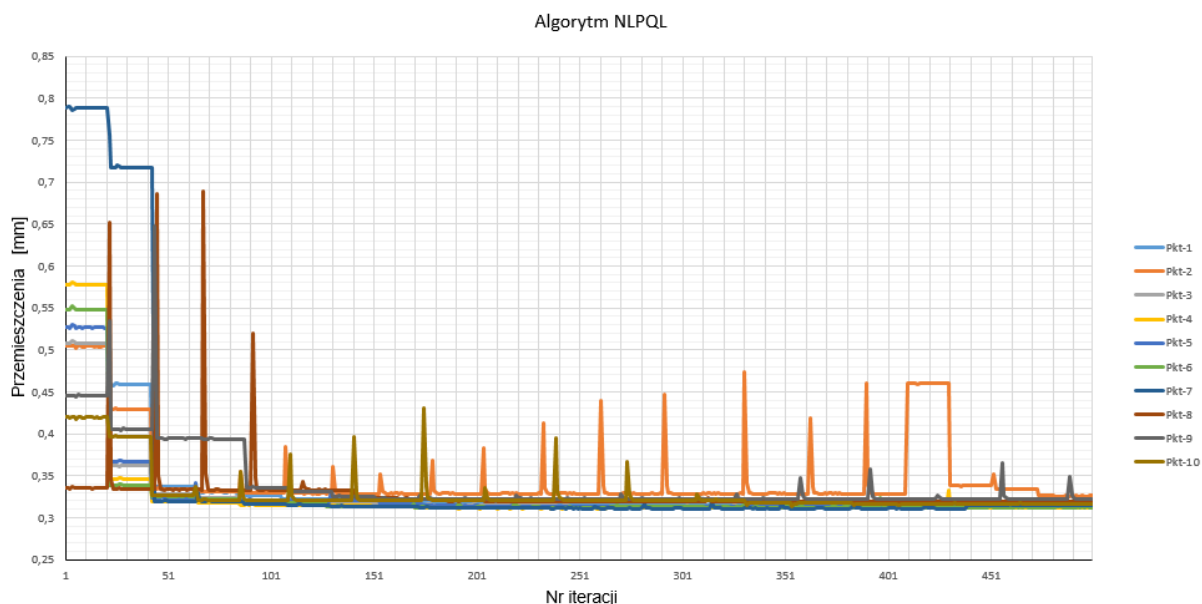
Biorąc pod uwagę cały zakres obliczeń, dla sześciu algorytmów łącznie, przeprowadzono około 20000 pojedynczych symulacji numerycznych. Analizy dotyczyły wyznaczenia wartości minimalnych przemieszczeń, przy założonych dwóch ograniczeniach jak opisano w rozdziale 3.11. Ponadto ustawiono parametr grubości blachy 'g' jako wartość stałą dla całego zakresu obliczeń.

Zdefiniowano 10 punktów startowych. Algorytmy, które wymagały ich zdefiniowania zostały uruchomione 10 razy, za każdym razem z innym punktem początkowym. Pozostałe algorytmy uruchomiono także dziesięć razy. Punkty te zebrano w tabeli 5.1. Pięć z nich (od 1 do 5) zostały dobrane na zasadzie równomiernego rozłożenia w całym swoim zakresie zmienności. Pozostałe pięć (od 6 do 10) zostało losowo wybranych z zakresu zmienności zmiennych.

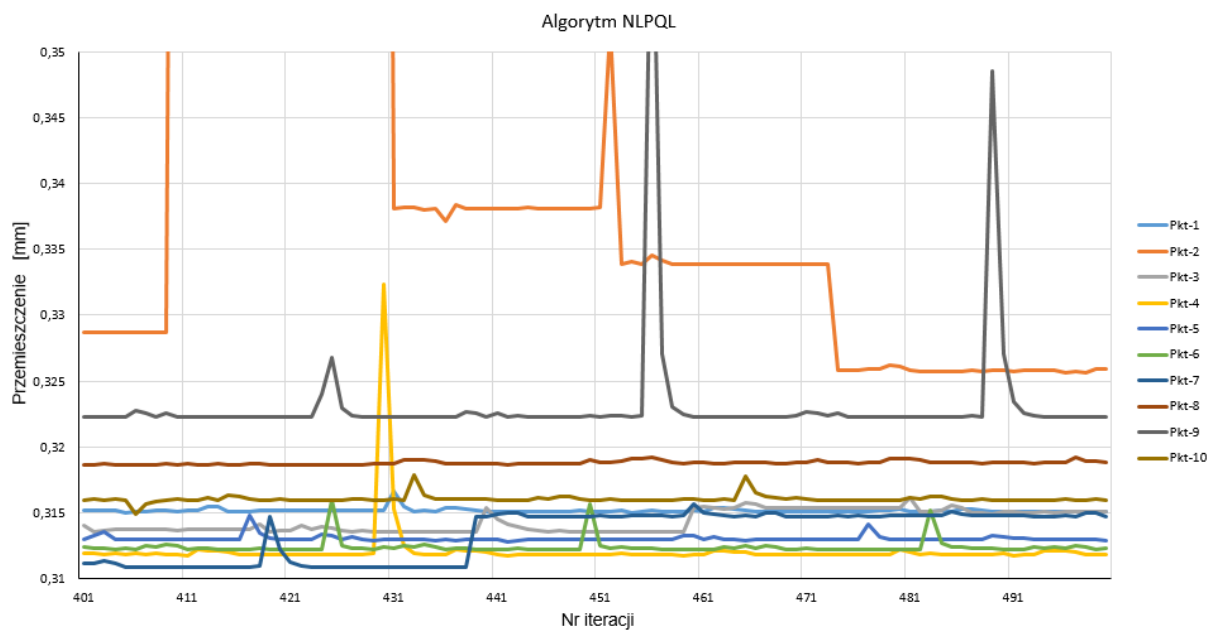
Tabela 5.1 Wartości zmiennych projektowych dla dziesięciu punktów startowych

Punkty startowe										
Parametr	5 punktów zdefiniowanych					5 punktów losowych				
	1	2	3	4	5	6	7	8	9	10
xo1	6,5000	7,3333	9,0000	10,6667	11,5000	9,0000	4,9992	9,5381	14,0000	5,0571
yo1	12,0000	14,6667	20,0000	25,3333	28,0000	25,0000	4,0000	31,1232	36,0000	8,5453
xo2	-1,2500	4,0000	5,5000	10,0000	5,0000	6,0000	19,0000	-8,0000	-8,0000	-3,5056
yo2	-6,2500	-5,3333	-3,5000	-1,6667	-0,7500	-1,0000	2,0000	-9,0000	-9,0000	-2,1695
xo3	-10,2500	-8,0000	-3,5000	1,0000	3,2500	-4,0000	-10,1692	10,0000	-9,1899	-1,5098
yo3	-3,2500	-2,3333	-0,5000	1,3333	2,2500	1,0000	5,0000	-6,0000	-6,0000	-5,4271
gr	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
RR	2,8750	3,0000	3,2500	3,5000	3,6250	3,2500	2,5000	4,0000	2,5000	3,5930
RL	2,8750	3,0000	3,2500	3,5000	3,6250	3,2500	2,5000	2,5000	4,0000	3,6068
x1pp	1,0000	2,0000	4,0000	6,0000	7,0000	-2,0000	10,0000	-2,0000	10,0000	-1,2391
y1pp	-10,5000	-9,3333	-7,0000	-4,6667	-3,5000	-7,0000	0,0000	-5,6196	-14,0000	-1,9538
x2pp	-0,7500	-0,6667	-0,5000	-0,3333	-0,2500	-1,0000	-0,5873	-0,8387	0,0000	-0,0655
y2pp	15,2500	15,3333	15,5000	15,6667	15,7500	16,0000	16,0000	15,8651	15,0000	15,9844
x3pp	-19,2500	-17,6667	-14,5000	-11,3333	-9,7500	-5,0000	-5,0000	-16,6032	-5,0000	-7,6802
y3pp	50,0000	51,6667	55,0000	58,3333	60,0000	45,0000	45,0000	65,0000	45,0000	60,7112
x1pl	-9,0000	-8,0000	-6,0000	-4,0000	-3,0000	-6,0000	0,0000	-12,0000	-12,0000	-5,8395
y1pl	-75,5000	-74,3333	-72,0000	-69,6667	-68,5000	-65,0000	-79,0000	-65,0000	-79,0000	-76,5136
x2pl	-1,7500	-1,6667	-1,5000	-1,3333	-1,2500	-1,0000	-1,9135	-2,0000	-2,0000	-1,6014
y2pl	-51,7500	-51,6667	-51,5000	-51,3333	-51,2500	-52,0000	-52,0000	-51,0000	-52,0000	-51,8661
x3pl	9,7500	11,3333	14,5000	17,6667	19,2500	5,0000	5,0000	5,0000	24,0000	5,5869
y3pl	-15,0000	-13,3333	-10,0000	-6,6667	-5,0000	-20,0000	-17,1046	-13,5341	-20,0000	-1,2172

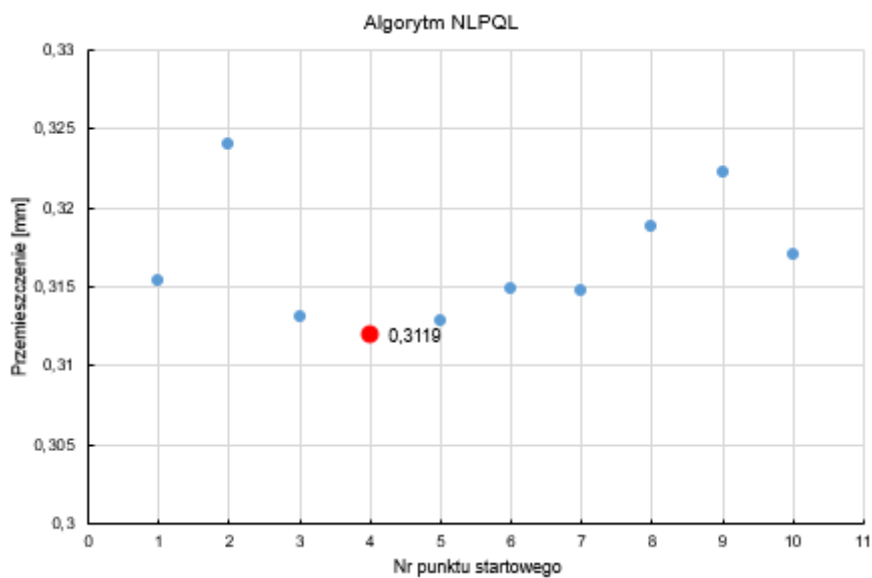
Otrzymano wyniki, które indywidualnie dla każdego algorytmu pokazano na Rys. 5.1 5.1 do 5.16.



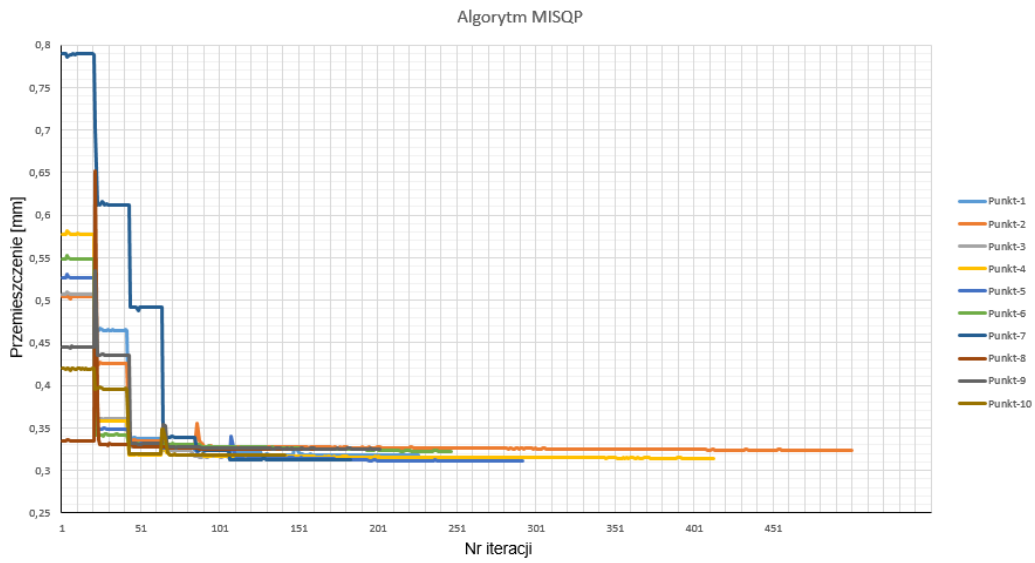
Rys. 5.1 Przebieg optymalizacji dla algorytmu NLPQL



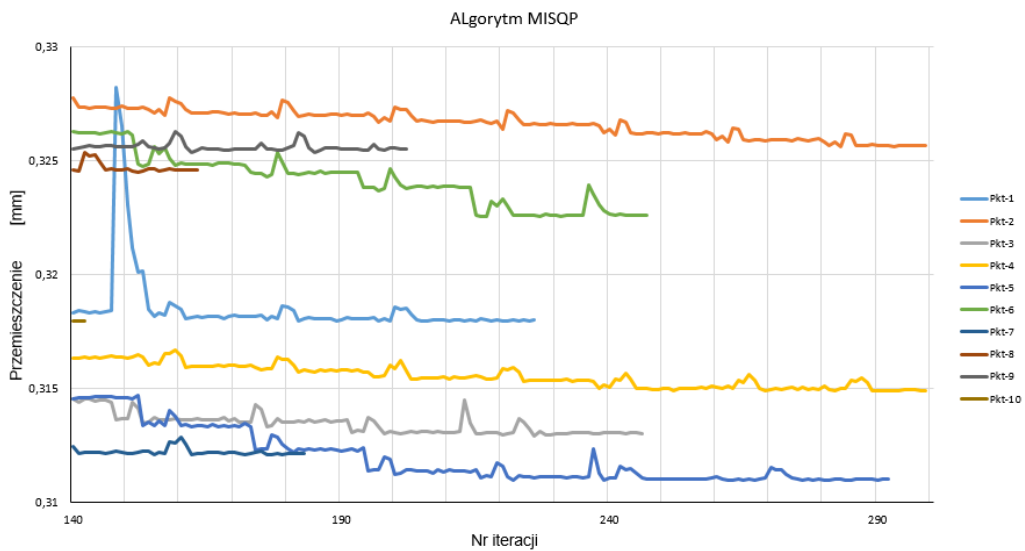
Rys. 5.2 Przebieg optymalizacji dla algorytmu NLPQL - ostatnie 100 iteracji



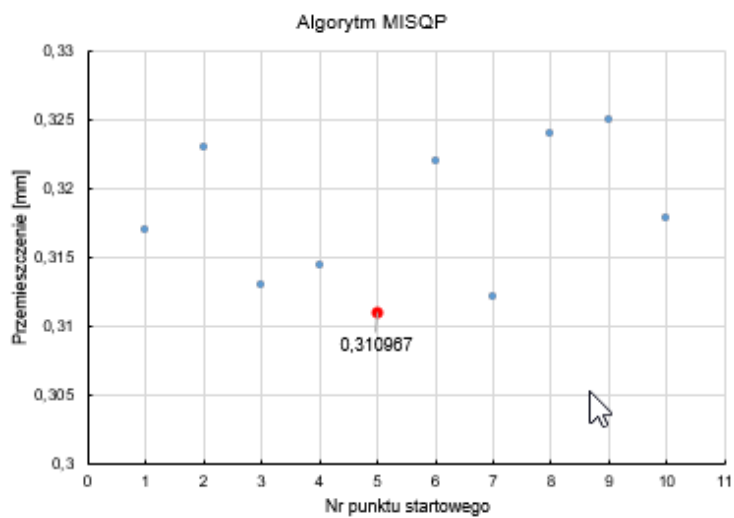
Rys. 5.3 Zestawienie wyników dla algorytmu NLPQL dla 10-ciu uruchomień.



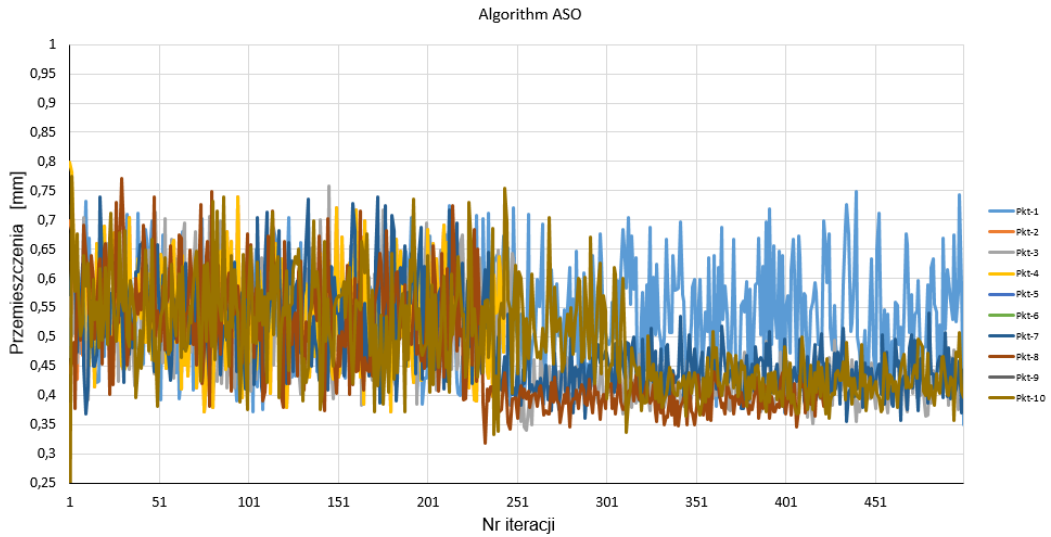
Rys. 5.4 Przebieg optymalizacji dla algorytmu MISQP



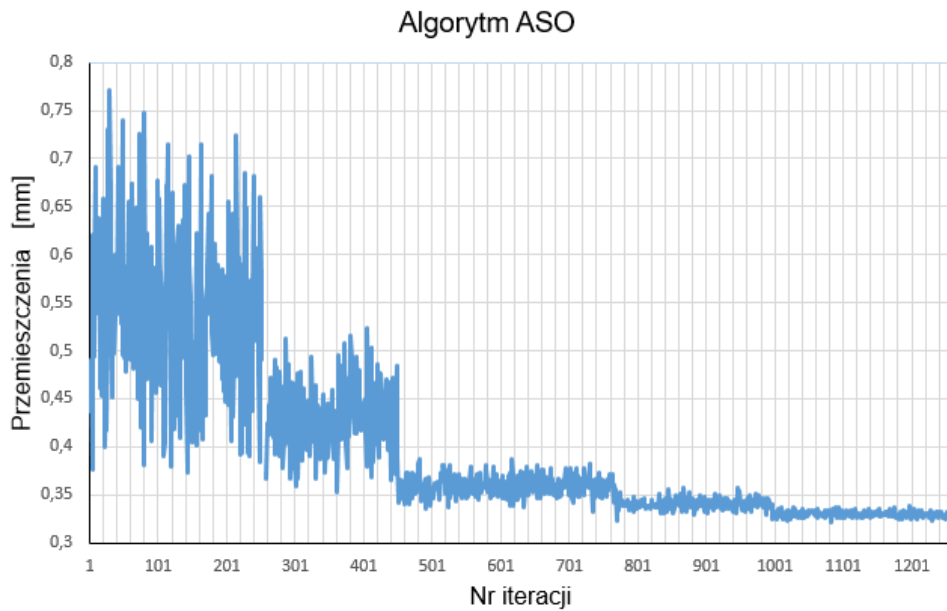
Rys. 5.5 Przebieg optymalizacji dla algorytmu MISQP - ostatnie 100 iteracji



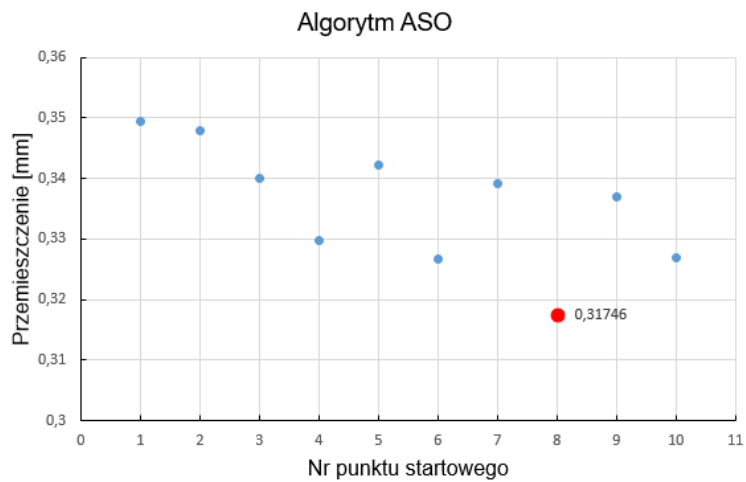
Rys. 5.6 Zestawienie wyników dla algorytmu MISQP dla 10-ciu uruchomień.



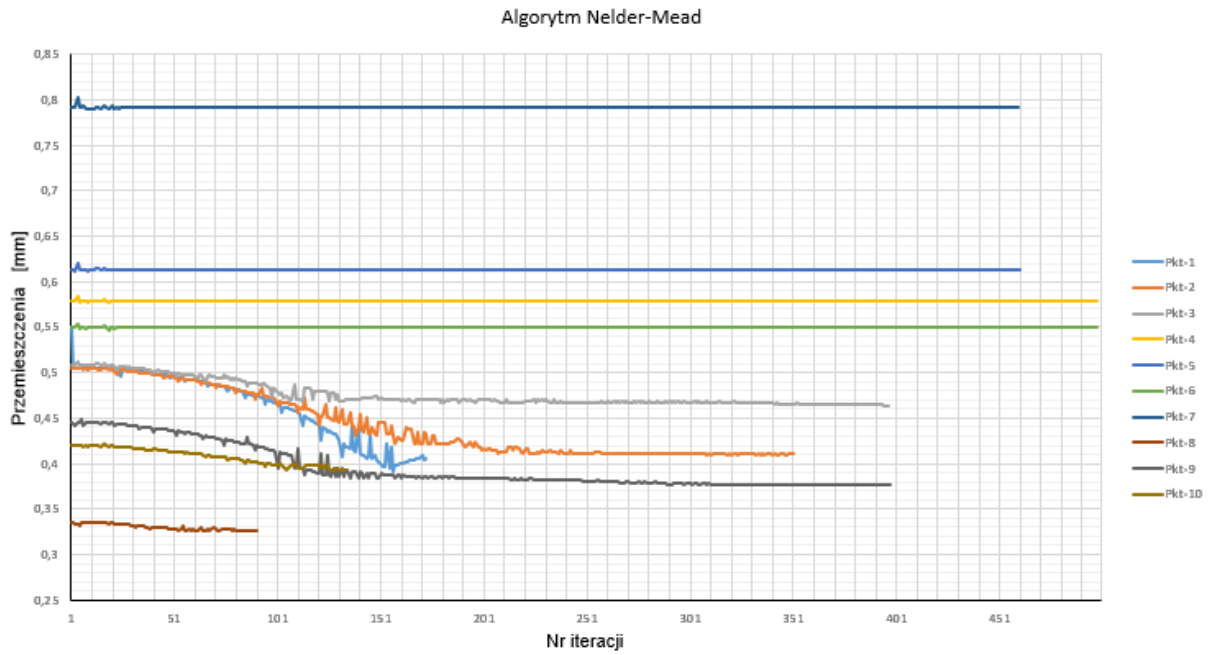
Rys. 5.7 Przebieg optymalizacji dla algorytmu ASO



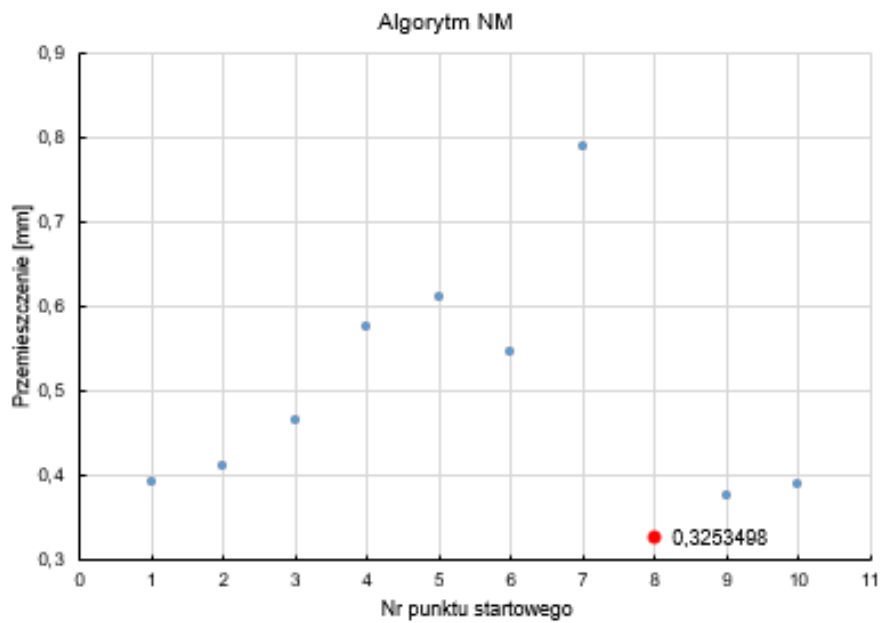
Rys. 5.8 Przebieg optymalizacji dla algorytmu ASO – 1250 iteracji



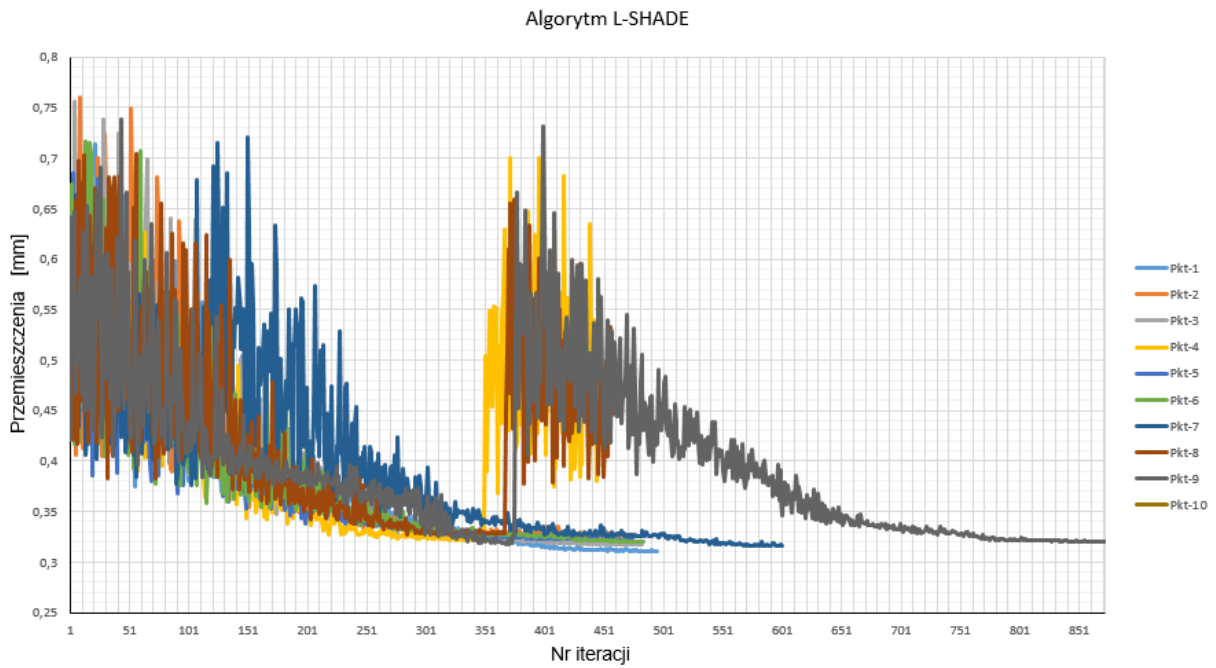
Rys. 5.9 Zestawienie wyników dla algorytmu ASO dla 10-ciu uruchomień.



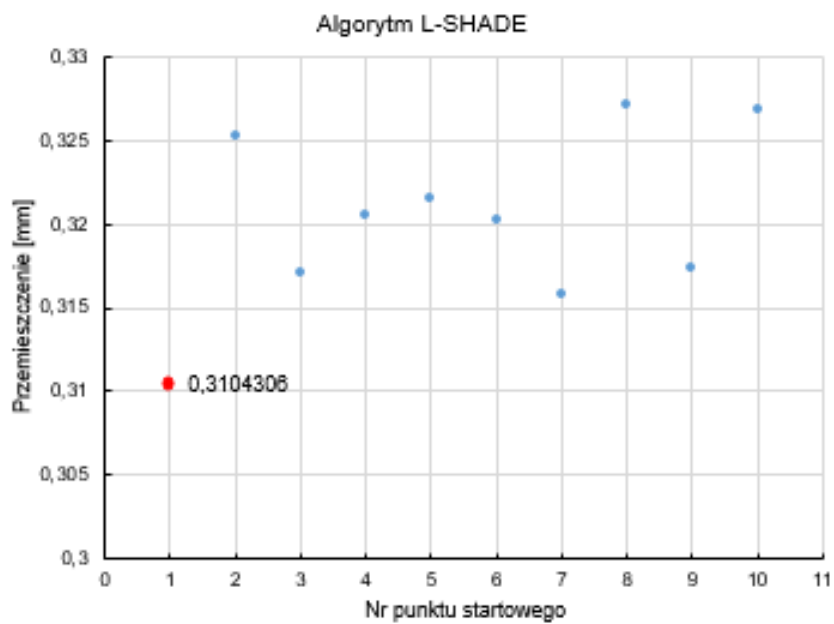
Rys. 5.10 Przebieg optymalizacji dla algorytmu NM



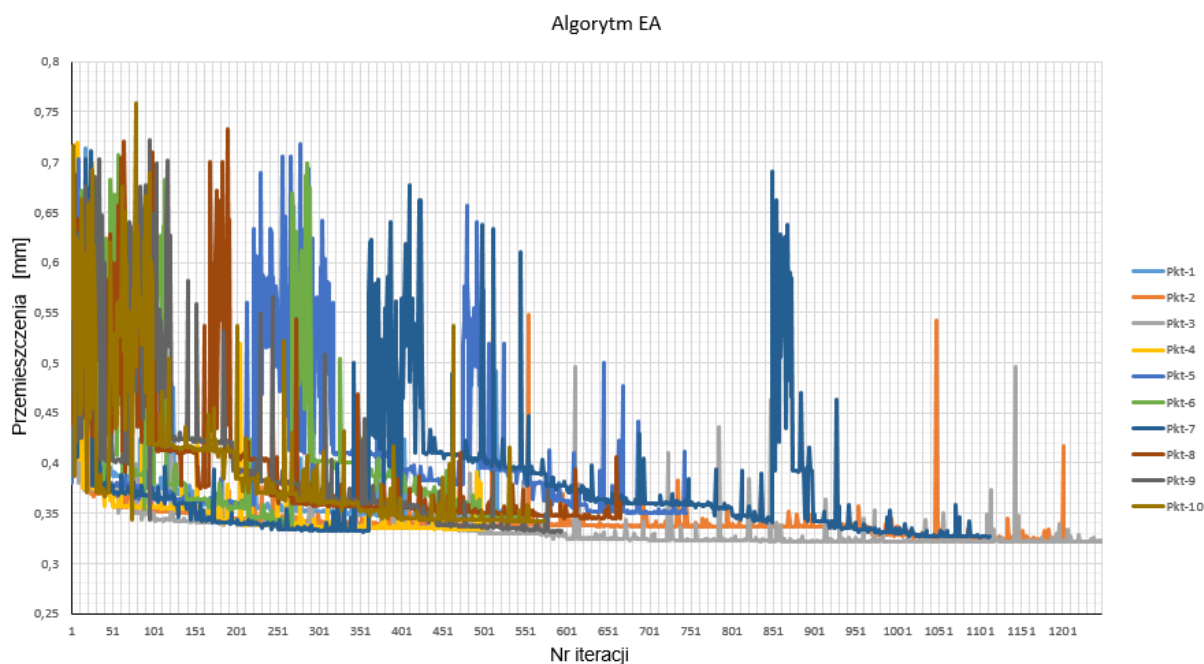
Rys. 5.11 Zestawienie wyników dla algorytmu NM dla 10-ciu uruchomień.



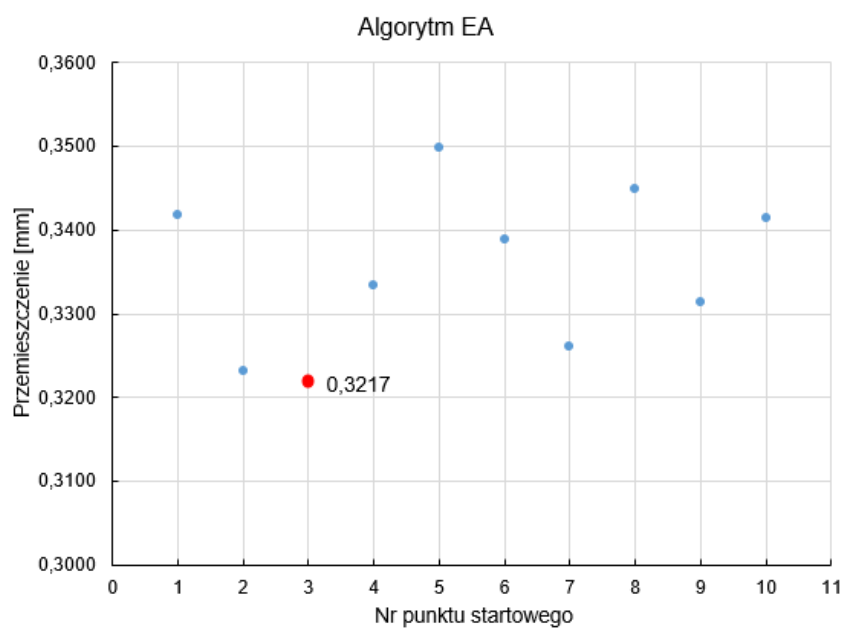
Rys. 5.12 Przebieg procesu optymalizacji dla algorytmu L-SHADE



Rys. 5.13 Zestawienie wyników dla algorytmu L-SHADE dla 10-ciu uruchomień.



Rys. 5.14 Przebieg procesu optymalizacji dla algorytmu EA



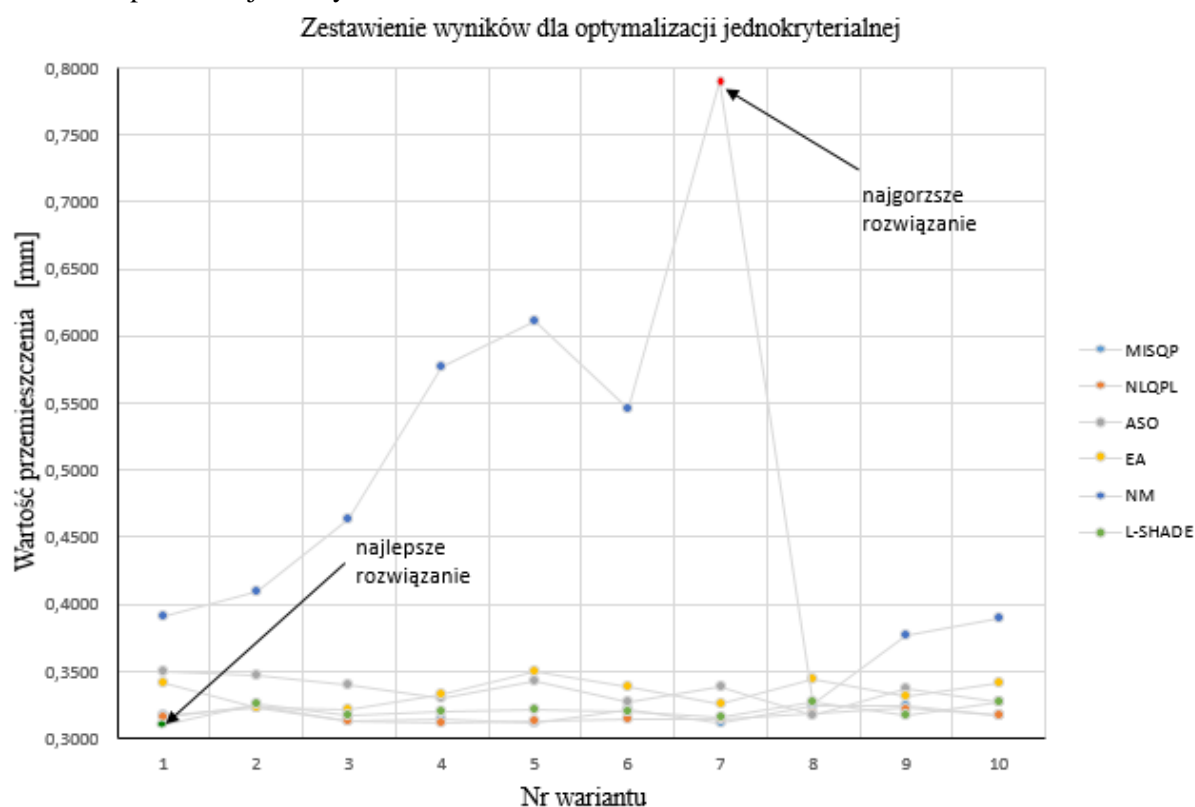
Rys. 5.15 Zestawienie wyników dla algorytmu EA dla 10-ciu uruchomień.

Zestawienie wszystkich optymalnych wartości minimalnych przemieszczeń dla wszystkich algorytmów, dla dziesięciu uruchomień, zestawiono w tabeli 5.2.

Tabela 5.2 Zestawienie najlepszych rozwiązań optymalnych

Algorytm	Punkty początkowe/ nr wariantu (startu)									
	1	2	3	4	5	6	7	8	9	10
MISQP	0,3170	0,3230	0,3130	0,3144	0,3110	0,3220	0,3121	0,3240	0,3250	0,3179
NLQPL	0,3154	0,3240	0,3131	0,3119	0,3129	0,3149	0,3147	0,3188	0,3223	0,3170
ASO	0,3494	0,3478	0,3400	0,3299	0,3422	0,3267	0,3391	0,3175	0,3370	0,3268
EA	0,3419	0,3233	0,3217	0,3335	0,3499	0,3389	0,3262	0,3449	0,3313	0,3415
NM	0,3914	0,4100	0,4638	0,5771	0,6105	0,5457	0,7895	0,3253	0,3762	0,3896
L-SHADE	0,3104	0,3253	0,3171	0,3205	0,3216	0,3202	0,3158	0,3271	0,3173	0,3269

Graficznie pokazano je na Rys. 5.16.



Rys. 5.16 Zestawienie wyników optymalizacji jednokryterialnej

Najlepszy wynik z spośród wszystkich wybranych i użytych algorytmów uzyskano za pomocą algorytmu L-SHADE. Przebieg jego wszystkich 10-ciu uruchomień pokazano na Rys. 5.13. Dla wariantu (uruchomienia) dziewiątego wartość minimalnego przemieszczenia wyniosła 0,3104mm. Jest to wynik lepszy aż o 26% od rezultatów uzyskanych w modelu referencyjnym, w którym wartość minimalnych przemieszczeń wynosiła 0.416mm. Natomiast rezultaty otrzymane za pomocą algorytmu ewolucyjnego (EA), są tylko niewiele gorsze od tych uzyskanych za pomocą LSHADE. Jego najlepszym wynikiem jest wartość minimalnych przemieszczeń wynosząca 0.3232mm uzyskana dla wariantu nr 3. Jest on o 4% gorszy od algorytmu L-SHADE.

Problemy ze znalezieniem minimum globalnego, co można zaobserwować na Rys. 5.10 miał algorytm NM. Algorytm ten najslabiej radził sobie z rozpatrywanym zadaniem optymalizacji. Otrzymane wyniki jednoznacznie potwierdzają, że jego skuteczność zależy od położenia punktu startowego. Dla pięciu z dziesięciu zdefiniowanych punktów startowych rezultaty uzyskane algorytmem NM są ekstremami lokalnymi, co potwierdza jego wrażliwość na prawidłowy dobór punktu startowego. Minimalną wartość

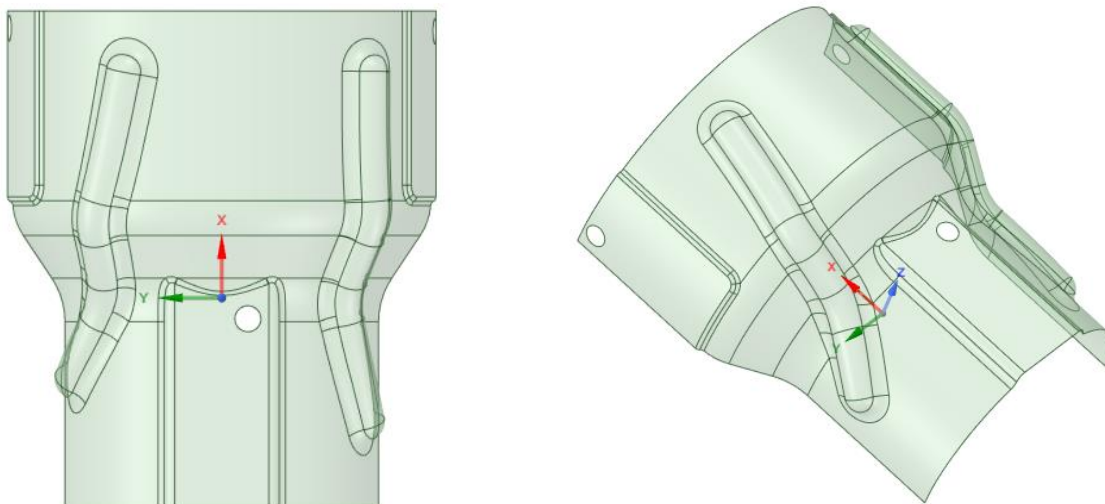
przemieszczenia porównywalną z najlepszymi wynikami pozostałych algorytmów uzyskał tylko dla jednego wariantu, w którym wartość minimalnych przemieszczeń wyniosła 0.3253mm. W pozostałych dziewięciu przypadkach znacząco odbiegał od najlepszych wyników, a wartość minimalnych przemieszczeń wahała się dla nich pomiędzy 0.3762mm a 0.7895mm.

W przypadku algorytmu NLPQL minimalne wartości przemieszczenia wyniosły 0.3108mm. Biorąc pod uwagę tylko ostatnie 100 iteracji, co przedstawiono na Rys. 5.2, można zauważyć, że różnice w uzyskanych wynikach wynoszą maksymalnie o 0,016 mm.

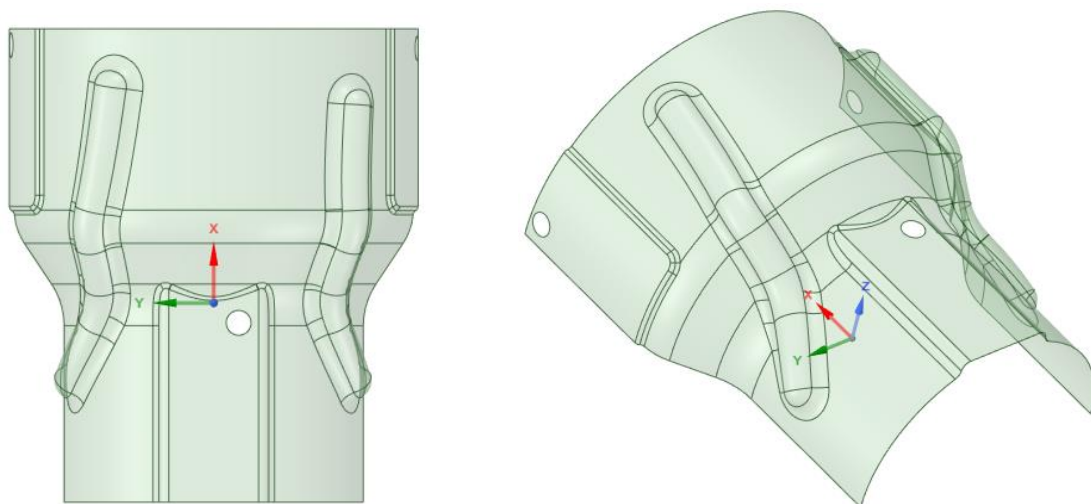
Dla kolejnego algorytmu MISQP wyniki z dziesięciu uruchomień pokazano na Rys. 5.4. Algorytm ten szybko osiągał wartości 0.35mm w pierwszych 100 iteracjach, by później przez kolejne 80% czasu pracy poprawić swój wynik jedynie o kilka procent. Najmniejszą wartość przemieszczenia uzyskano dla wariantu z punktem startowym nr. 5. Wynoszą one 0.3109mm.

Algorytm ASO zbieżność uzyskiwał dopiero dla 1250 iteracji, co pokazano na Rys.5.8. Przebieg jego krzywych optymalizacyjnych jest zmienny. Najniższą wartość minimalnych przemieszczeń uzyskał dla wariantu ósmego. Wynoszą one 0.3174mm.

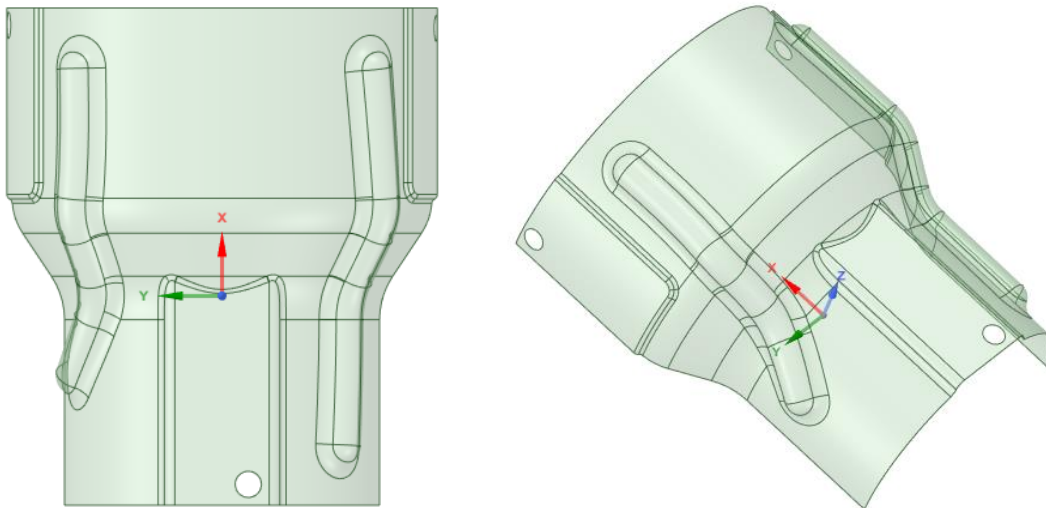
Postać geometryczną modelu osłony termicznej dla najlepszego rozwiązania dla każdego z wykorzystanych do obliczeń algorytmów pokazano na rysunkach 5.17 do 5.22.



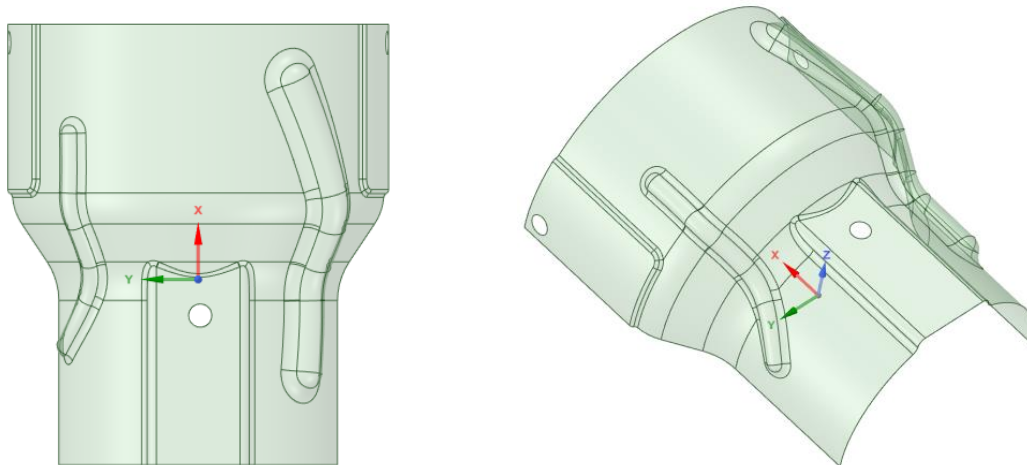
Rys. 5.17 Postać geometryczna najlepszego rozwiązania algorytmu NLPQL



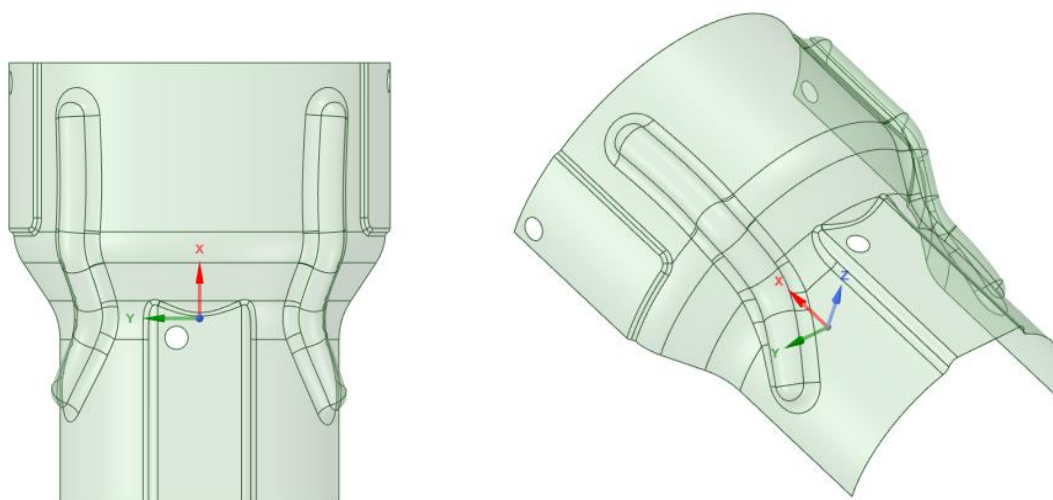
Rys. 5.18 Postać geometryczna najlepszego rozwiązania algorytmu MISQP



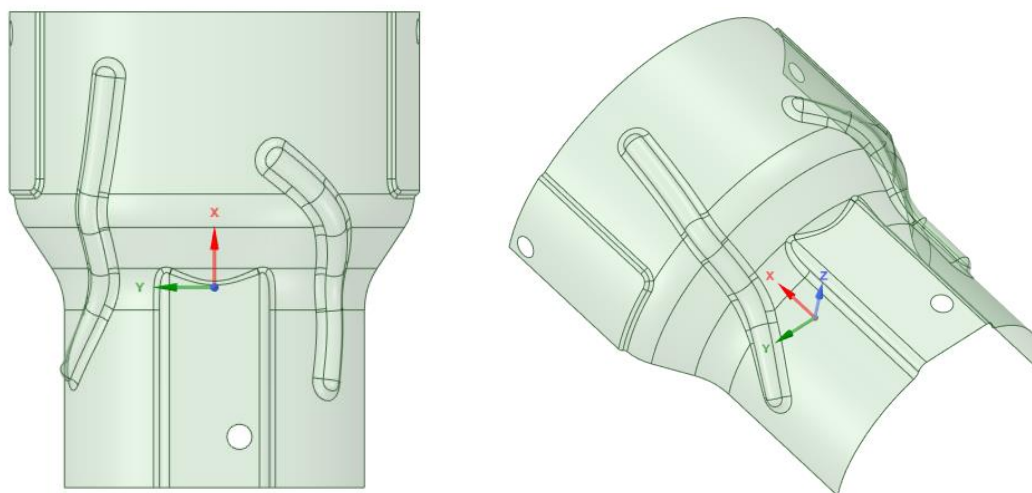
Rys. 5.19 Postać geometryczna najlepszego rozwiązania algorytmu ASO



Rys. 5.20 Postać geometryczna najlepszego rozwiązania algorytmu NM



Rys. 5.21 Postać geometryczna najlepszego rozwiązania algorytmu L-SHADE



Rys. 5.22 Postać geometryczna najlepszego rozwiązania algorytmu EA

W tabeli 5.3 zestawiono wartości parametrów (zmiennych projektowych) najlepszych rozwiązań. Dla każdego algorytmu wskazano numer punktu startowego (wariant), dla którego dany algorytm uzyskał optymalne rozwiązanie.

Tabela 5.3 Wartości zmiennych konstrukcyjnych dla najlepszych rozwiązań dla każdego algorytmu optymalizacji jednokryterialnej

	Wariant	Wartości zmiennych konstrukcyjnych dla najlepszych rozwiązań dla każdego algorytmu (1-10) [mm]									
Algorytm		xo1	yo1	xo2	yo2	xo3	yo3	RR	RL	x1pp	y1pp
MISQP	5	14	36	-9	-9	10	-5,883	4	4	10	-0,258
NLQPL	4	14	36	-9	-9	10	-6	4	4	8,197	-8,887
ASO	8	14	4	-9	-9	10	-6	4	4	-2	-14
EA	3	14	10,149	-8	-10	10	-6	2,636	2,521	-2	-2,717
NM	8	9,440	32,209	-8,332	-9,562	9,917	-5,900	4,132	2,445	-2,005	-5,690
L-SHADE	1	4,017	35,835	-7,989	-9,540	9,999	-5,990	3,997	4,000	9,955	-1,118

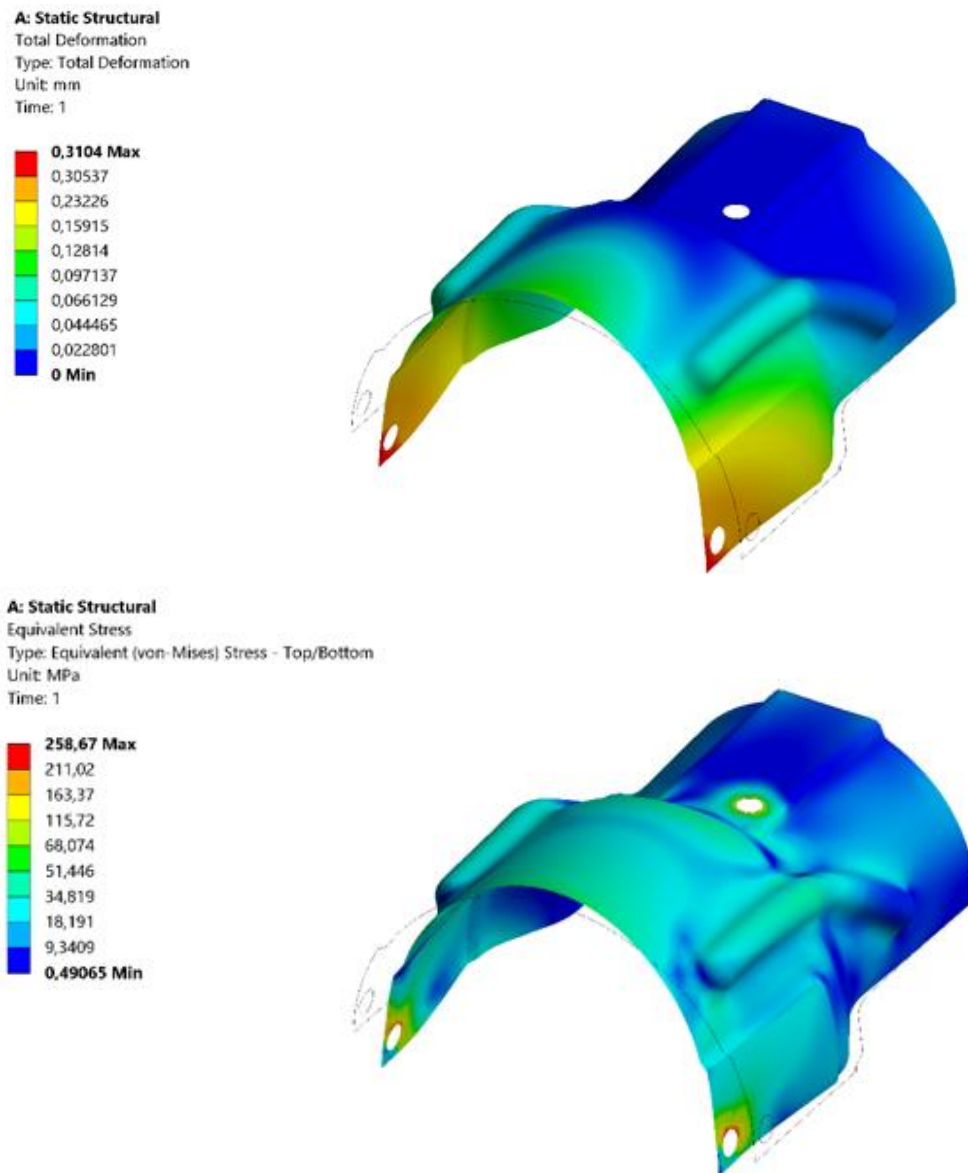
	Wariant	Wartości zmiennych konstrukcyjnych dla najlepszych rozwiązań dla każdego algorytmu (11-20) [mm]									
Algorytm		x2pp	y2pp	x3pp	y3pp	x1pl	y1pl	x2pl	y2pl	x3pl	y3pl
MISQP	5	-0,807	15,393	-5	61,239	-12	-65	-1	-51,953	14,265	0
NLQPL	4	-1	15,498	-5	65	-12	-65	-1	-52	16,385	0
ASO	8	-1	15	-5	65	-10,973	-65	-2	-51,454	5	0
EA	3	-0,954	15,016	-24	45,032	-12	-65	-1,146	-52	12,595	-2,743
NM	8	-0,851	15,780	-15,983	64,566	-11,266	-62,690	-1,988	-53,263	5,029	-13,507
L-SHADE	1	-0,823	15,345	-5,025	64,947	-11,911	-65,212	-1,135	-51,935	5,695	-0,082

Jak można zauważyć, uzyskano bardzo podobne wartości minimalnych przemieszczeń dla różnych zestawów wartości zmiennych konstrukcyjnych. W wielu przypadkach otrzymano minimalną wartość przemieszczeń dla różnych wartości parametrów. Jak pokazano w tabeli 5.3, niektóre z parametrów przyjmują bardzo zbliżoną wartość. Niektóre natomiast, przyjmują wartość z przeciwnego końca swojego zakresu zmienności. Różne wartości zmiennych decyzyjnych mogą świadczyć o tym, że podobne optymalne rozwiązanie można uzyskać dla różnych postaci geometrycznych osłony, wykorzystując różne algorytmy. Nie istnieje więc tylko jedna postać geometryczna osłony, która gwarantuje jedyne optymalne rozwiązanie, przy założeniu pewnego pola tolerancji dla uzyskanych wyników np. $\pm 0.002\text{mm}$.

Różnice te można dostrzec na rysunkach (5.17 – 5.22) pokazujących postać geometryczną osłony dla najlepszych rozwiązań optymalnych. Wyraźnie można zauważyć asymetryczność geometrii osłony dla rozwiązań optymalnych, jeżeli chodzi o trajektorię przetłoczeń, jak i położenie otworów montażowych. W niektórych przypadkach przetłoczenia różnią się nie tylko długością ale również i szerokością (Rys.

5.20). Zupełnie różne kształty przetłoczeń obserwujemy dla rozwiązania optymalnego algorytmu ewolucyjnego (Rys.5.22). Położenie trzeciego otworu montażowego w żadnym z rozwiązań nie jest położeniem symetrycznym względem głównych płaszczyzn symetrii osłony.

Rozkład naprężeń i przemieszczeń dla najlepszego rozwiązania, które uzyskano algorytmem L-SHADE pokazano na Rys. 5.23.



Rys. 5.23 Mapa przemieszczeń i naprężeń dla najlepszego rozwiązania algorytmu L-SHADE

Wartości naprężeń dla tego rozwiązania optymalnego są niższe w porównaniu z wariantem podstawowym o 198%% i wynoszą 258,67MPa.

5.2. Optymalizacja wielokryterialna

Jak wspomniano w rozdziale trzecim w ramach pracy wykorzystano trzy algorytmy optymalizacji wielokryterialnej. Wybrano algorytmy zaimplementowane w środowisko Ansys, do których należy MOGA (Multi Objective Genetic Algorithm) oraz NSGA-II (Non-dominated Sorting Genetic

Algorithm-II). Natomiast z grupy algorytmów zewnętrznych wybrano algorytm MOOPTIM opisany szczegółowo w rozdziale 3.10.4.

Podobnie jak w przypadku optymalizacji jednokryterialnej wszystkie użyte algorytmy uruchamiano przy porównywalnym koszcie obliczeniowym tj. tej samej lub podobnej liczbie wywołań funkcji celu. W odróżnieniu do optymalizacji jednokryterialnej grubość blachy została zdefiniowana jako 21-szy parametr. Biorąc pod uwagę wszystkie zadania optymalizacji z każdym z trzech algorytmów, to łącznie przeprowadzono około 60000 zadań numerycznych.

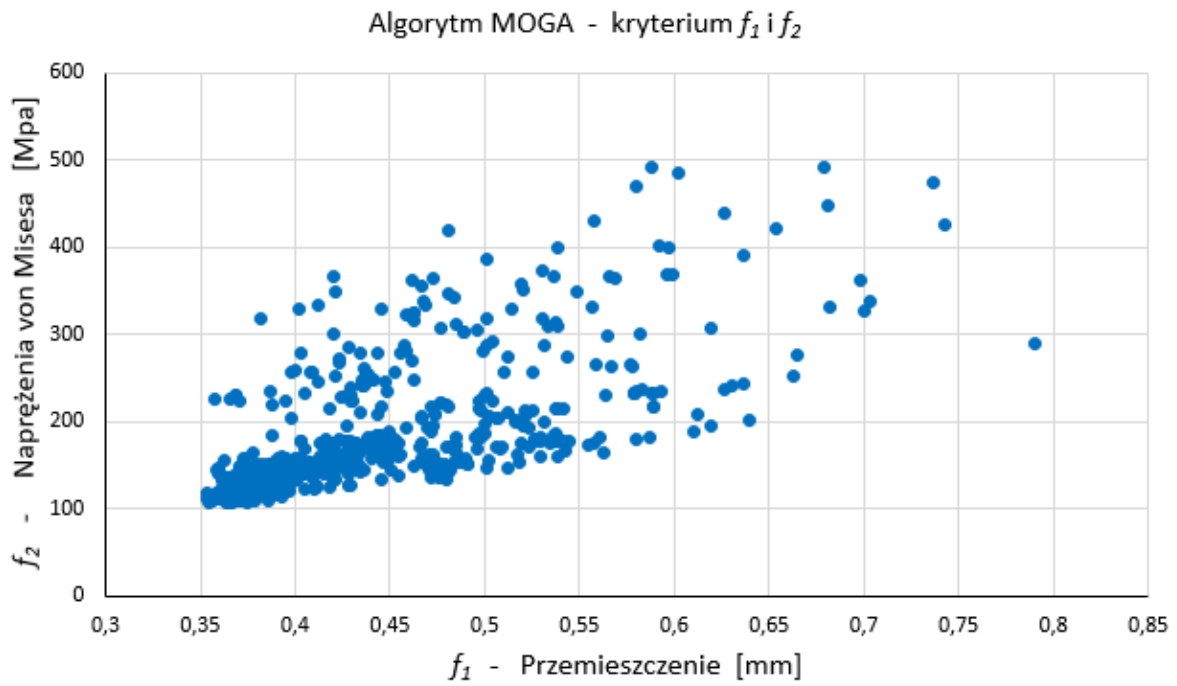
Rozwiązano następujące warianty optymalizacji z wykorzystaniem MOOPTIM, MOGA i NSGA-II:

- Wariant 1 – minimalizacja funkcjonałów (3.110) and (3.111).
- Wariant 2 – minimalizacja funkcjonału (3.110) i maksymalizacja funkcjonału (3.112).
- Wariant 3 – minimalizacja funkcjonału (3.111) i maksymalizacja funkcjonału (3.112).
- Wariant 4 - minimalizacja funkcjonałów (3.110) i (3.111) i maksymalizacja funkcjonału (3.112).

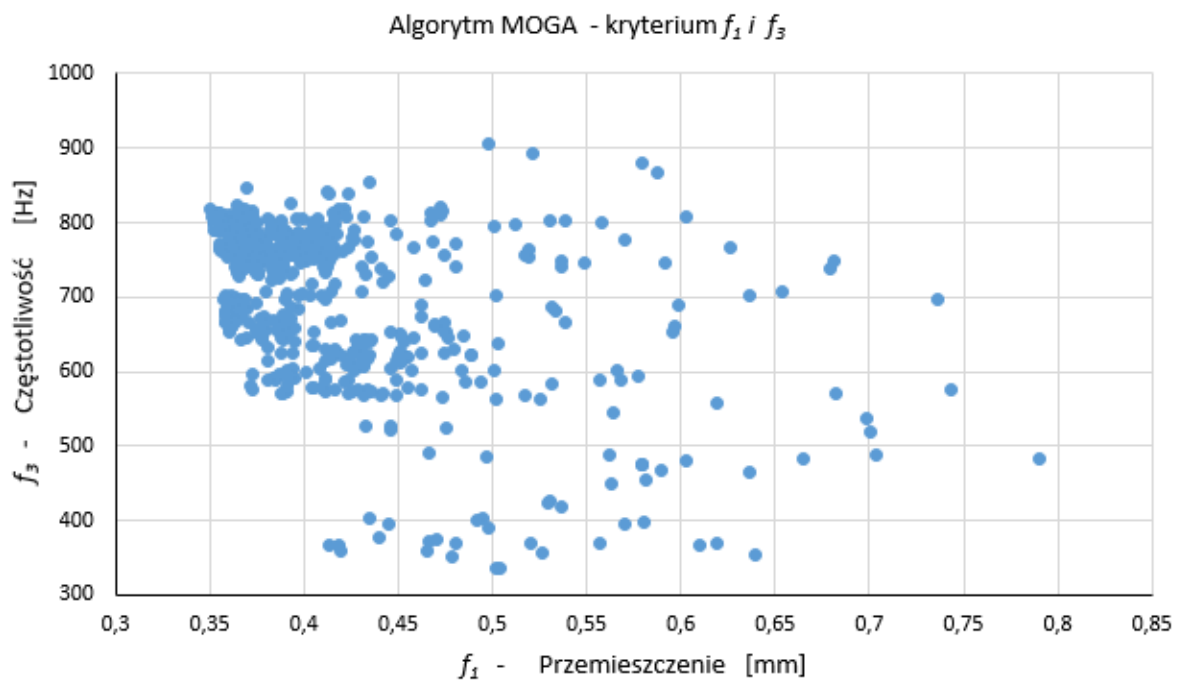
Dla każdego z zastosowanych algorytmów zdefiniowano parametry inicjujące. Dla algorytmu MOOPTIM przyjęto: prawdopodobieństwo mutacji równomiernej 0.1, prawdopodobieństwo mutacji z rozkładem Gaussa 0.7, prawdopodobieństwo krzyżowania prostego 0.1 oraz prawdopodobieństwo krzyżowania arytmetycznego równego 0.1. Natomiast dla algorytmu MOGA i NSGA-II wartość prawdopodobieństwa krzyżowania prostego przyjęto równe 0.9 oraz prawdopodobieństwo mutacji równomiernej 0.1, jak sugerują autorzy w pracach [50,51,52]. Dla tych dwóch algorytmów został zdefiniowany jeszcze dodatkowo jeden parametr; współczynnik zbieżności równy 2%. Ponadto dla wszystkich algorytmów ustawiono te same wartości wielkości populacji równej 50 oraz liczby pokoleń równej 50.

Takie przyjęcie wartości parametrów zewnętrznej procedury optymalizacyjnej i algorytmów wbudowanych w system CAE zapewnia prawidłowe porównanie przy podobnej prędkości obliczeniowej (liczba ocen funkcji przystosowania). Należy pamiętać, że użytkownik w przypadku wbudowanych algorytmów optymalizacyjnych ma ograniczony dostęp do zmiany poszczególnych parametrów algorytmu.

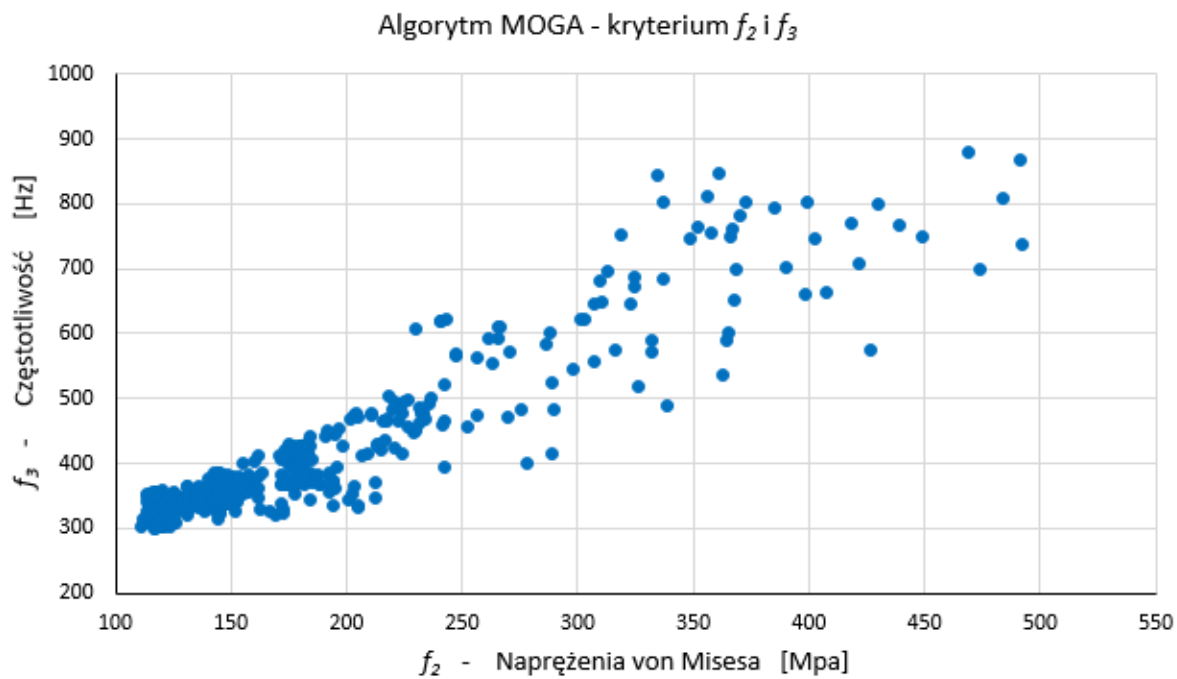
Wyniki optymalizacji dla poszczególnych algorytmów przedstawiono na rysunkach 5.24 do 5.39 oraz w artykule [82]. Na rysunkach 5.24 do 5.27 pokazano otrzymane wyniki optymalizacji dla algorytmu MOGA. Na trzech pierwszych wykresach pokazano wyniki wszystkich możliwych kombinacji trzech zdefiniowanych kryteriów. Na Rys. 5.27 pokazano wszystkie rozwiązania w całej przestrzeni poszukiwań 3D .



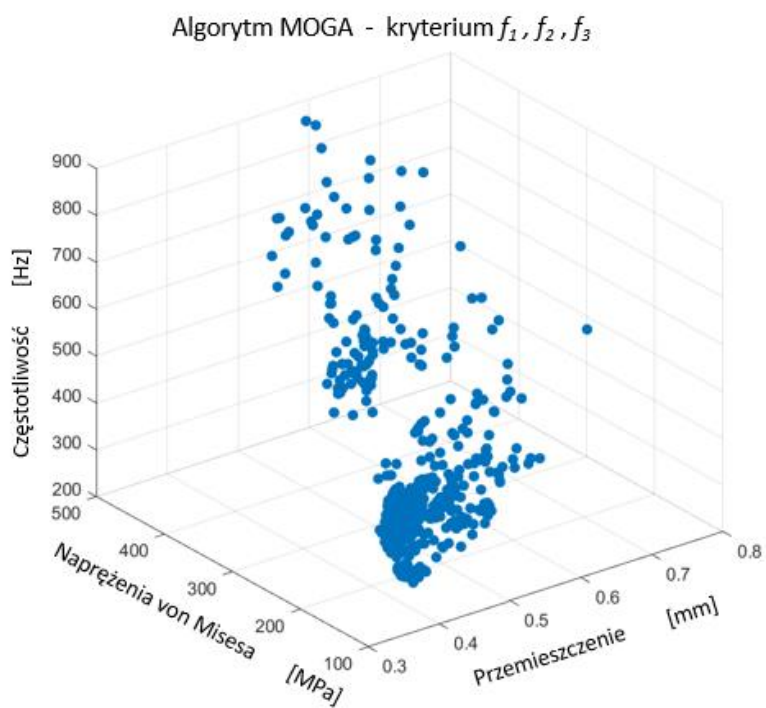
Rys. 5.24 Rozwiązania algorytmu MOGA dla kryteriów f_1 i f_2



Rys. 5.25 Rozwiązania algorytmu MOGA dla kryteriów f_1 i f_3

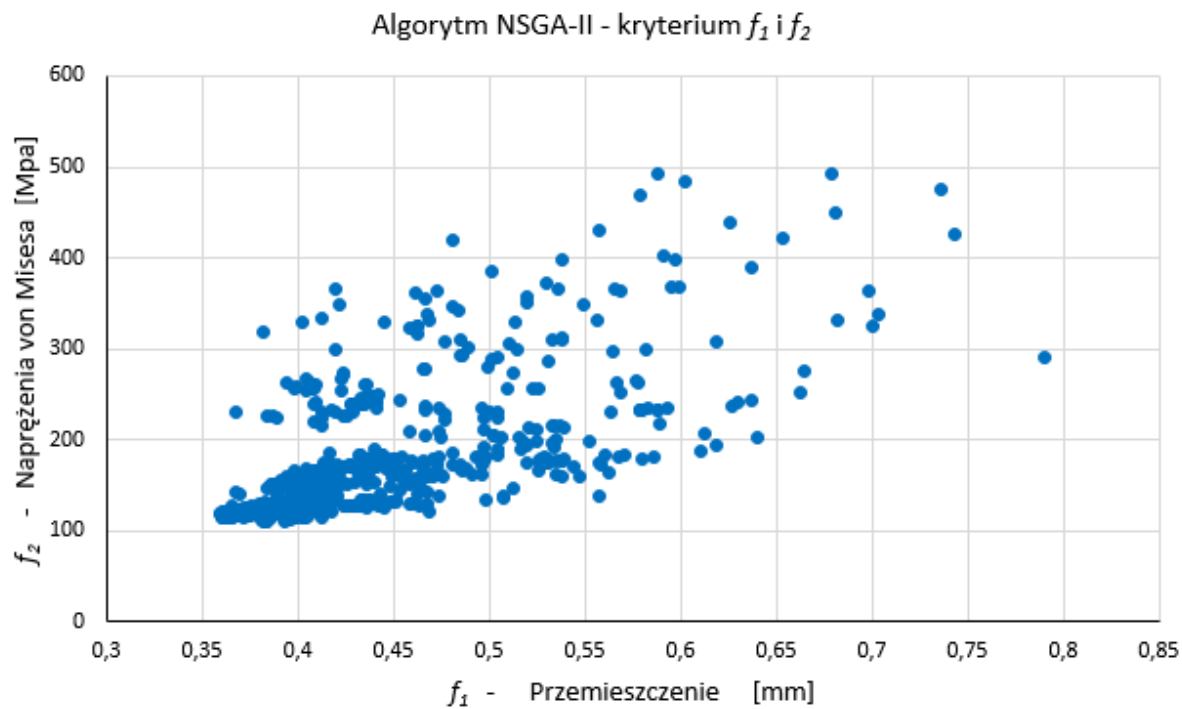


Rys. 5.26 Rozwiązania algorytmu MOGA dla kryteriów f_2 i f_3

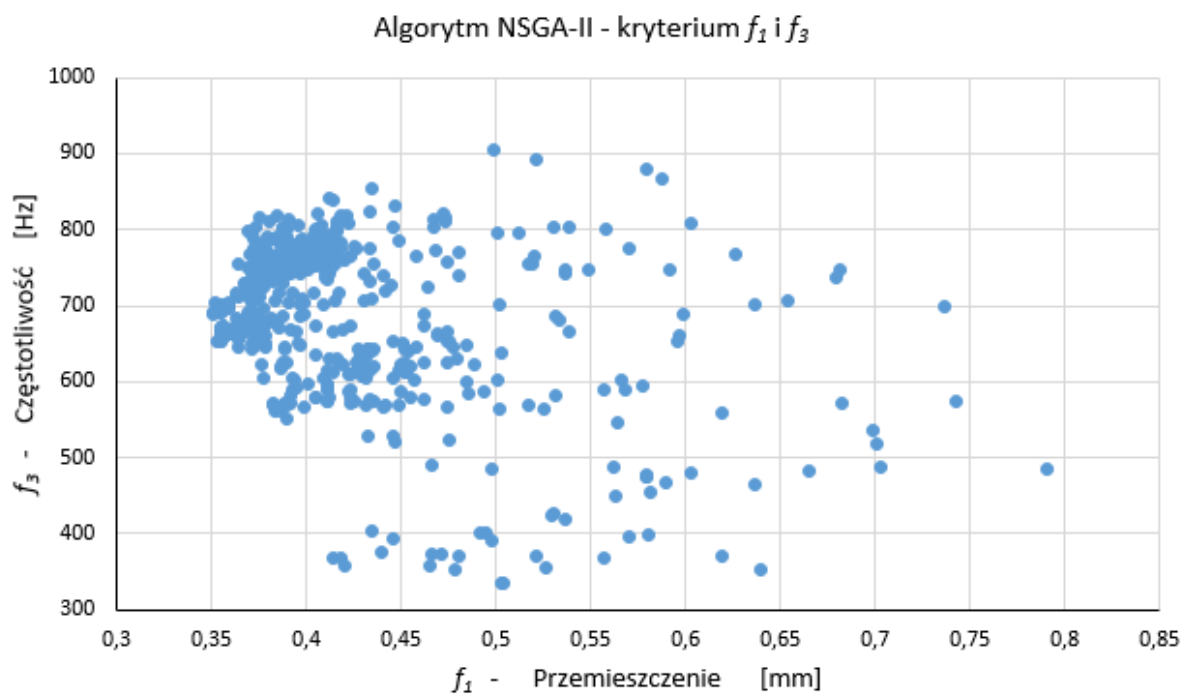


Rys. 5.27 Rozwiązania algorytmu MOGA dla kryteriów f_1, f_2 i f_3

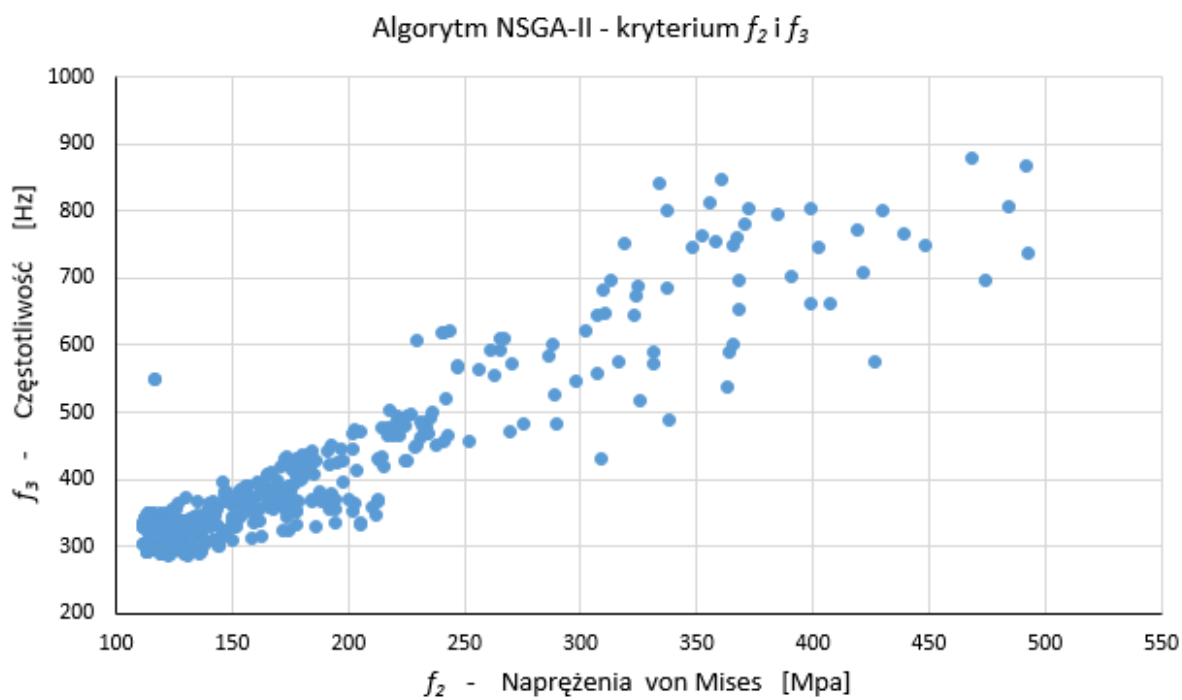
Na rysunkach 5.28 do 5.31 pokazano otrzymane wyniki optymalizacji dla algorytmu NSGA-II.



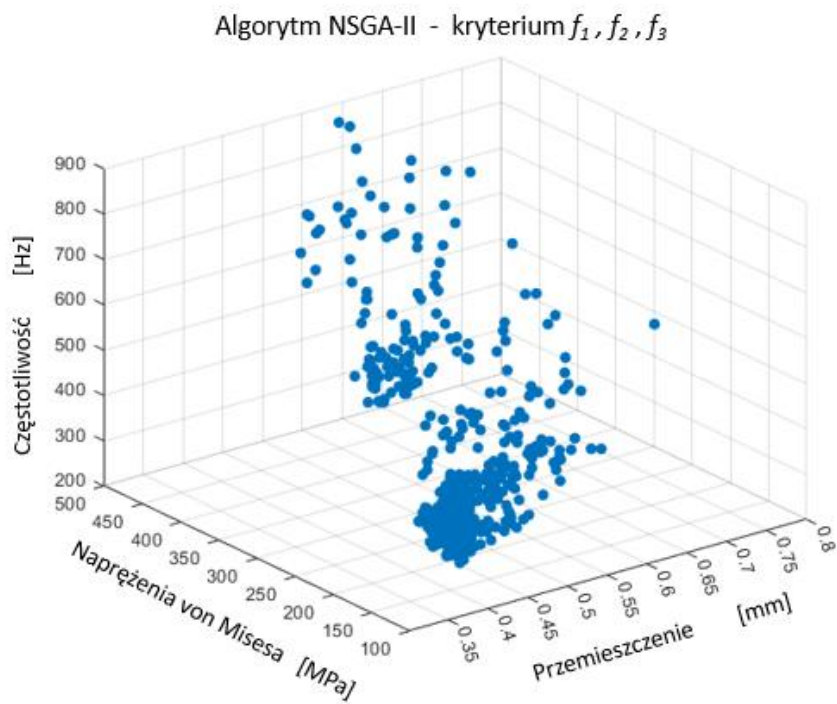
Rys. 5.28 Rozwiązania algorytmu NSGA-II dla kryteriów f_1 i f_2



Rys. 5.29 Rozwiązania algorytmu NSGA-II dla kryteriów f_1 i f_3

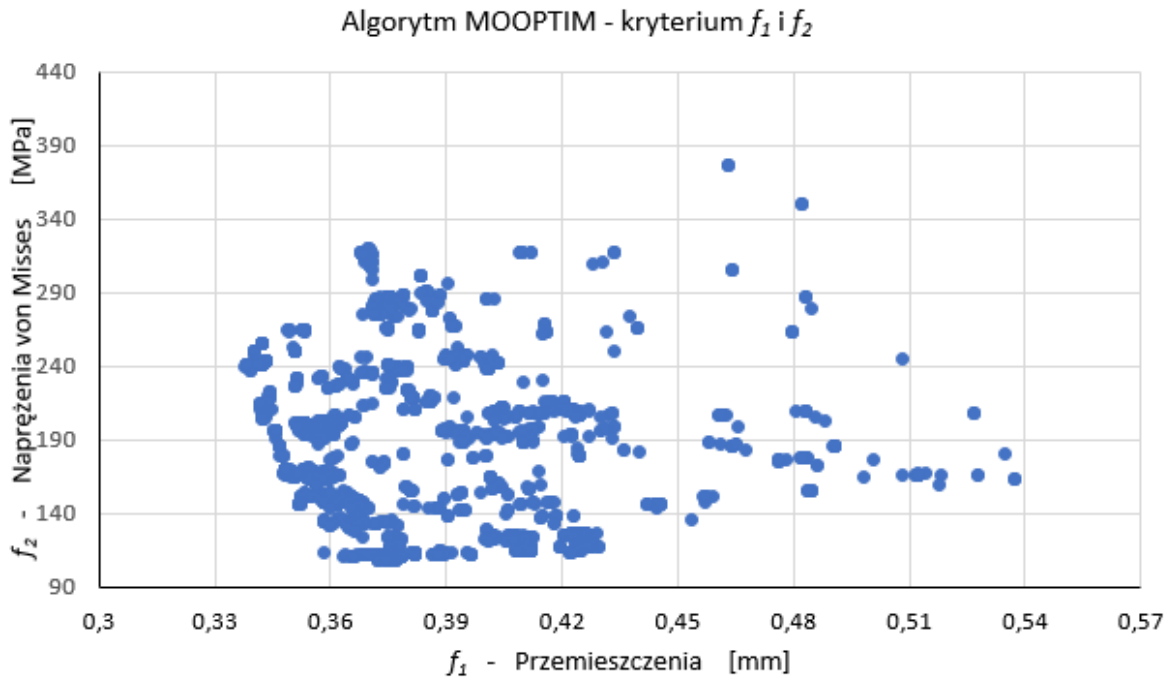


Rys. 5.30 Rozwiązania algorytmu NSGA-II dla kryteriów f_2 i f_3

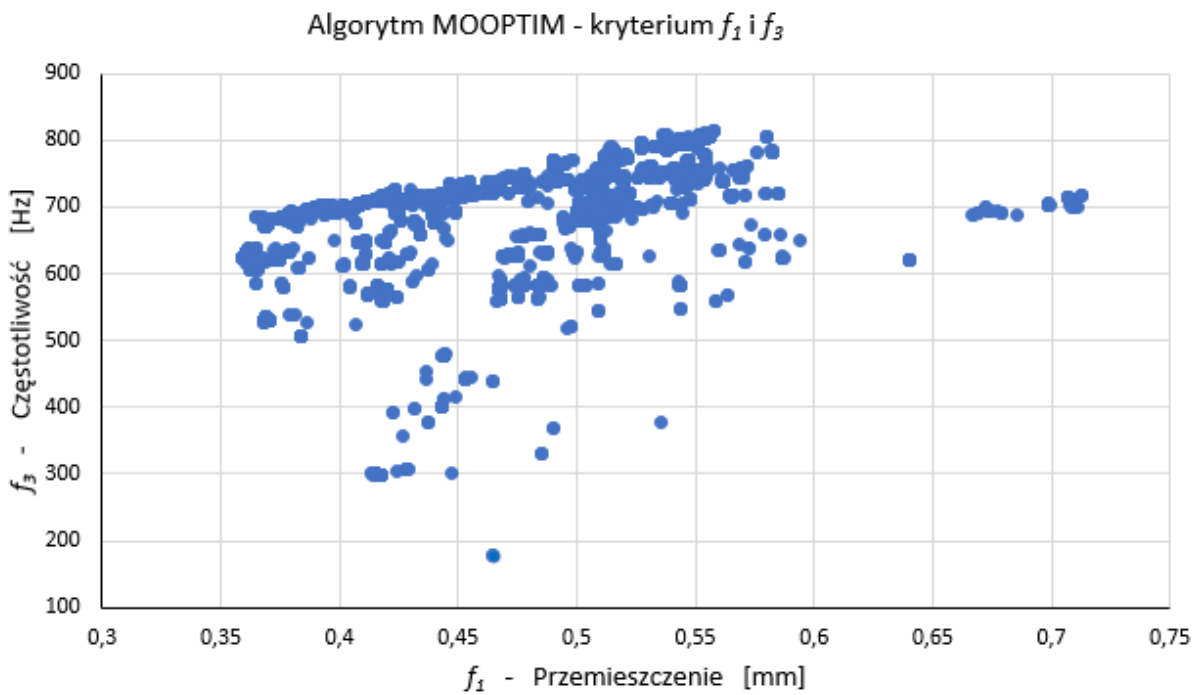


Rys. 5.31 Rozwiązania algorytmu NSGA-II dla f_1, f_2 i f_3

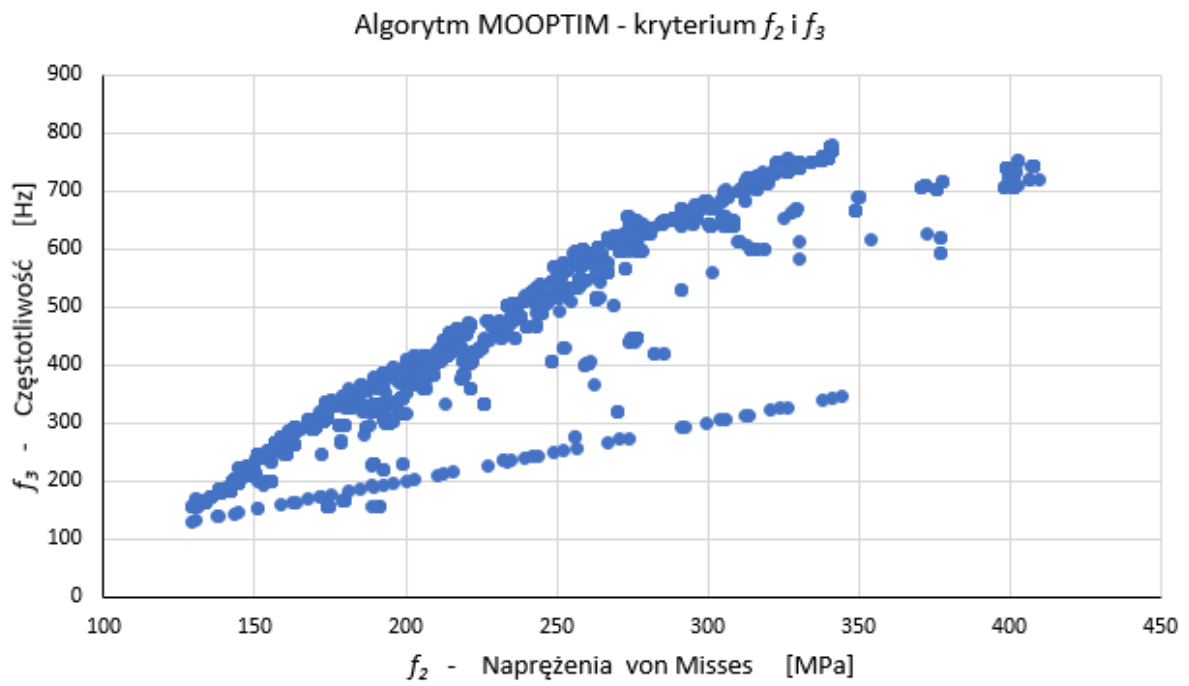
Na rysunkach od 5.32 do 5.35 pokazano otrzymane wyniki optymalizacji dla algorytmu MOOPTIM.



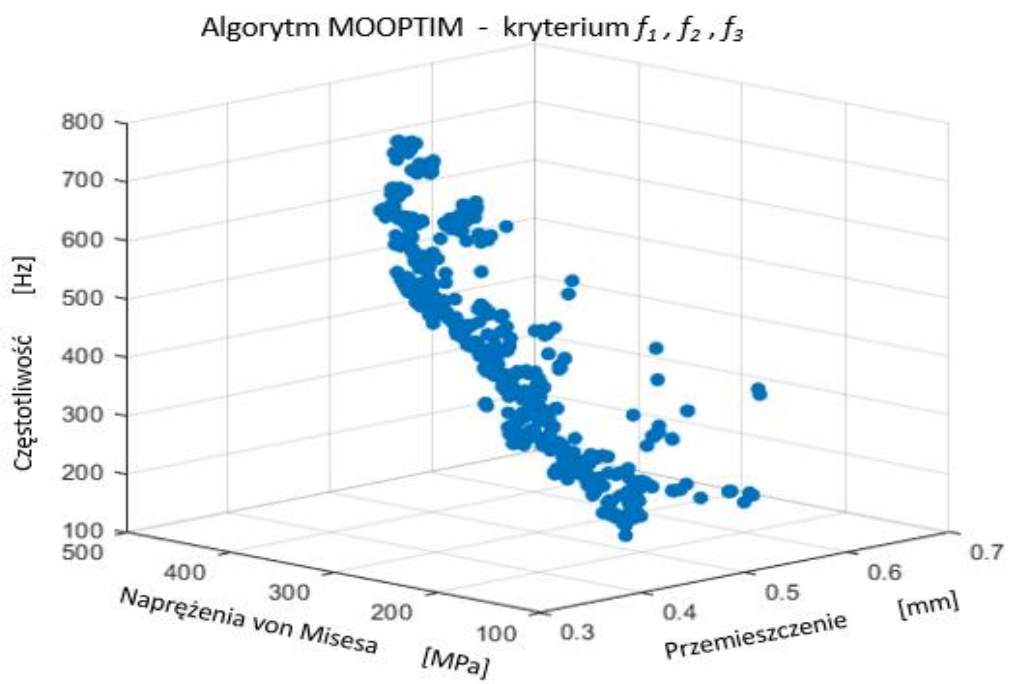
Rys. 5.32 Rozwiązania algorytmu MOOPTIM dla kryteriów f_1 i f_2



Rys. 5.33 Rozwiązania algorytmu MOOPTIM dla kryteriów f_1 i f_3

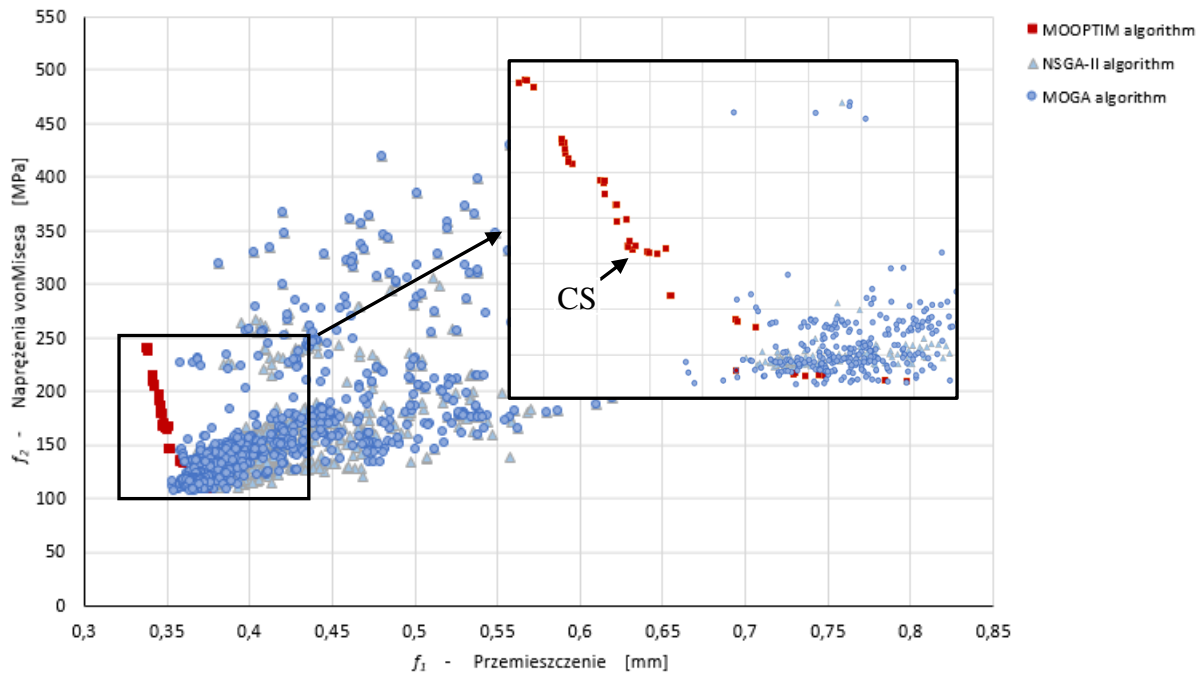


Rys. 5.34 Rozwiązania algorytmu MOOPTIM dla kryteriów f_2 i f_3

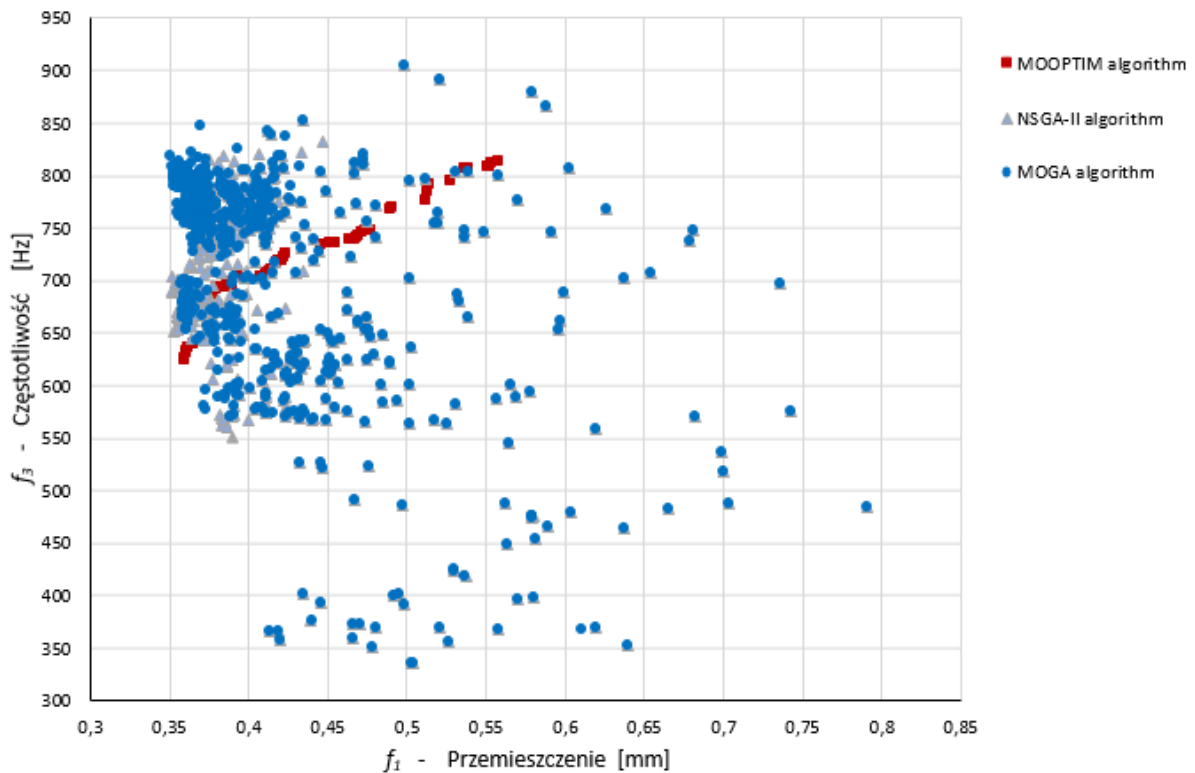


Rys. 5.35 Rozwiązania algorytmu MOOPTIM dla f_1, f_2 i f_3

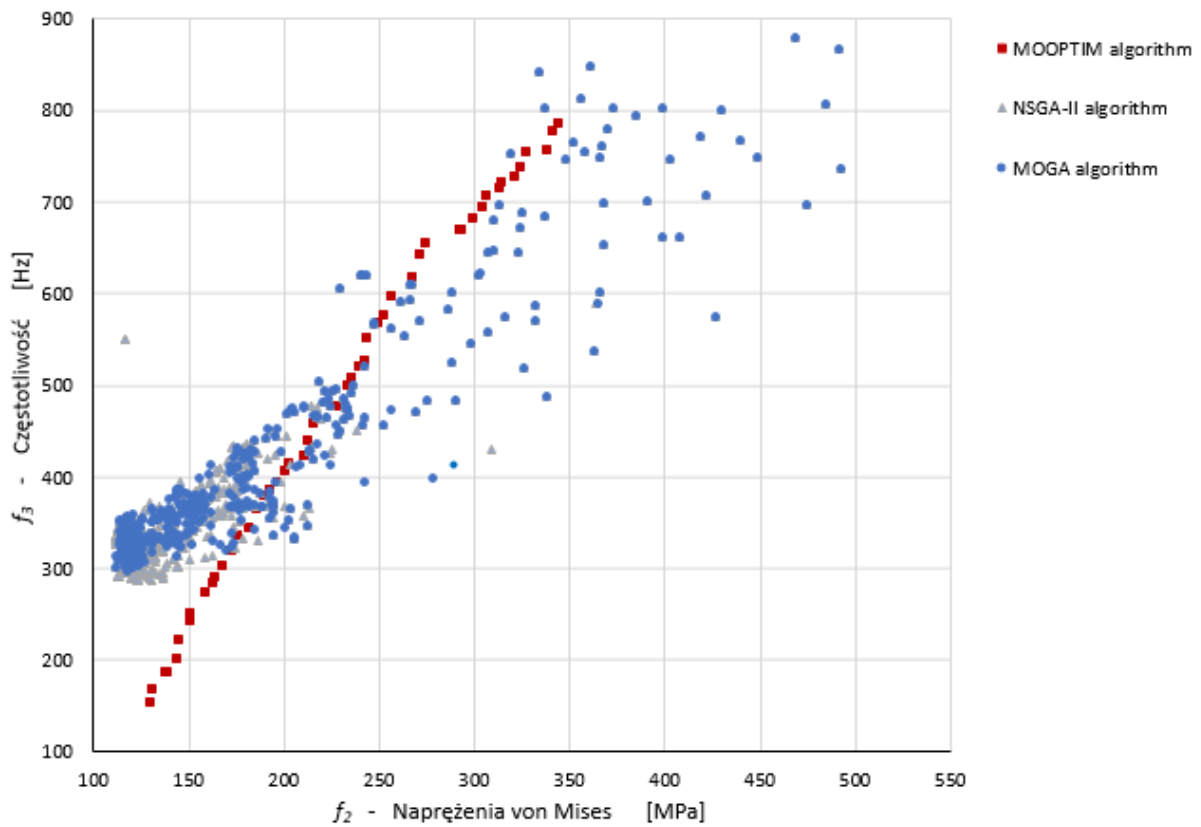
Porównanie wyników optymalizacji (zbiór rozwiązań optymalnych w sensie Pareto) dla wszystkich trzech algorytmów i dla każdego rozpatrywanego wariantu przedstawiono na rysunkach od 5.36 do 5.39.



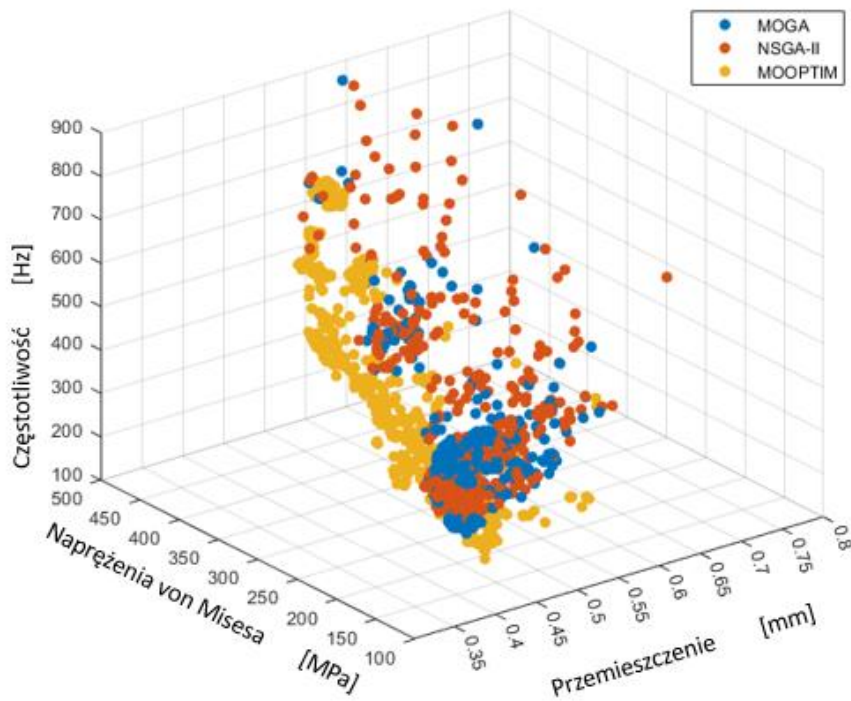
Rys. 5.36 Front Pareto-optymalnych rozwiązań dla wariantu 1



Rys. 5.37 Front Pareto-optymalnych rozwiązań dla wariantu 2



Rys. 5.38 Front Pareto- optymalnych rozwiązań dla wariantu 3



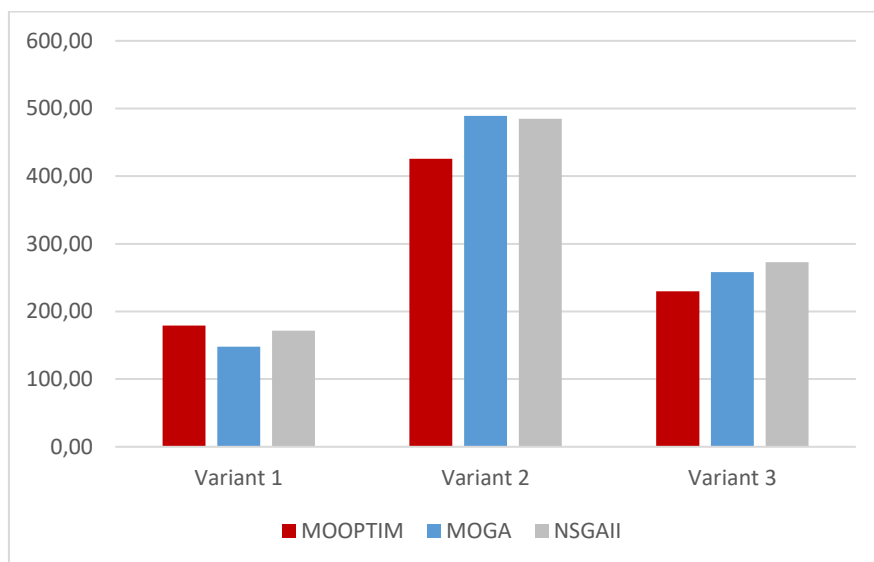
Rys. 5.39 Front Pareto- optymalnych rozwiązań dla wariantu 4

Dla każdego z zadań optymalizacji przeprowadzono kilka niezależnych uruchomień wszystkich trzech algorytmów. Dla otrzymanych rozwiązań obliczono metryki HV opisane dokładnie w rozdziale 3.3.

Ponieważ rozwiązanie powyższych zadań optymalizacyjnych nie jest pojedynczym rozwiązaniem, a zbiorem rozwiązań, przy ocenie jakości otrzymanego zbioru należy wziąć pod uwagę więcej czynników, a mianowicie: odległość znalezionych rozwiązań Pareto- optymalnych od rzeczywistego Pareto przód (TPF); równomierność rozkładu rozwiązań na froncie Pareto i rozpiętości frontu Pareto, opisanych w rozdziale 3.3. Oceny tych cech można dokonać metodą wizualną, jednak ocena taka może być niedokładna, zwłaszcza jeśli weźmie się pod uwagę wszystkie wymienione powyżej cechy. Spośród kilku różnych miar oceny jakości otrzymanych frontów wybrano metrykę hiperobjętości (HV), ze względu na to, że do rozwiązania zadania nie jest znana lokalizacja TPF. Ponadto metryka HV uwzględnia wszystkie trzy wymienione cechy [149]. Obliczone wartości HV podsumowano i porównano w tabeli 5.4 oraz na wykresie na Rys. 5.40.

Tabela 5.4 Wartości metryki hiperobjętości HV oraz liczba nie-zdominowanych rozwiązań dla rozważanych wariantów optymalizacji używając algorytmów MOOPTIM, MOGA i NSGA-II

	MOOPTIM		MOGA		NSGA-II	
	HV	NoNS	HV	NoNS	HV	NoNS
Variant 1	179.43	33	148.22	4	171.65	8
Variant 2	425.55	50	488.98	5	485.05	13
Variant 3	230016.7	50	258460.8	38	272812.1	43



Rys. 5.40 Porównanie metryki HyperVolume dla zastosowanych algorytmów.

W pierwszym wariantcie optymalizacji zewnętrzny algorytm MOOPTIM okazał się lepszy od wbudowanych algorytmów w systemie Ansys. Pokazuje to zarówno Rys. 5.40, jak i porównanie uzyskanych wartości metryki HV w tabeli 5.4. Dla wariantów 2 i 3 MOGA i NSGA-II okazały się lepsze w porównaniu z MOOPTIM, jednak dla trzeciego wariantu rozwiązania uzyskane przez MOOPTIM w niektórych obszarach dominują nad rozwiązaniami uzyskanymi przez MOGA i NSGA-II. Należy zauważyć, że moduł optymalizacyjny Ansys prezentuje wyniki wielokryterialnych zadań optymalizacyjnych jako zbiór wszystkich dotychczas znalezionych rozwiązań. Taki zbiór zawiera zarówno rozwiązania zdominowane, jak i niezdominowane. Na podstawie otrzymanych zestawów rozwiązań, po wyeksportowaniu danych i zastosowaniu procedury sortowania niezdominowanego, można wyświetlić wyłącznie rozwiązania niezdominowane. Tabela 5.4 przedstawia również porównanie liczby rozwiązań niezdominowanych znalezionych przez każdy z trzech algorytmów. Dla

wszystkich rozważanych wariantów optymalizacji algorytm MOOPTIM znalazł w zbiorze wyników największą liczbę rozwiązań niezdominowanych (NoNS). Z jednej strony te dodatkowe rozwiązania wymagają odpowiedniej interpretacji przy podejmowaniu decyzji o ostatecznym wyborze rozwiązania optymalnego, z drugiej strony kształt i rozkład rozwiązań na froncie Pareto przynosi dodatkowe informacje o rozpatrywanym problemie.

Dla szerszego zobrazowania spektrum rozwiązań Pareto-optymalnych, jako przykład przedstawiono graficznie kilka rozwiązań kompromisowych (CS). Rozwiązanie pierwsze otrzymano dla wariantu 1 z algorytmem MOOPTIM (oznaczone jako CS na Rys. 5.36), które leży najbliżej punktu utopijnego (UP), czyli przecięcia osi układu współrzędnych dwóch kryteriów.

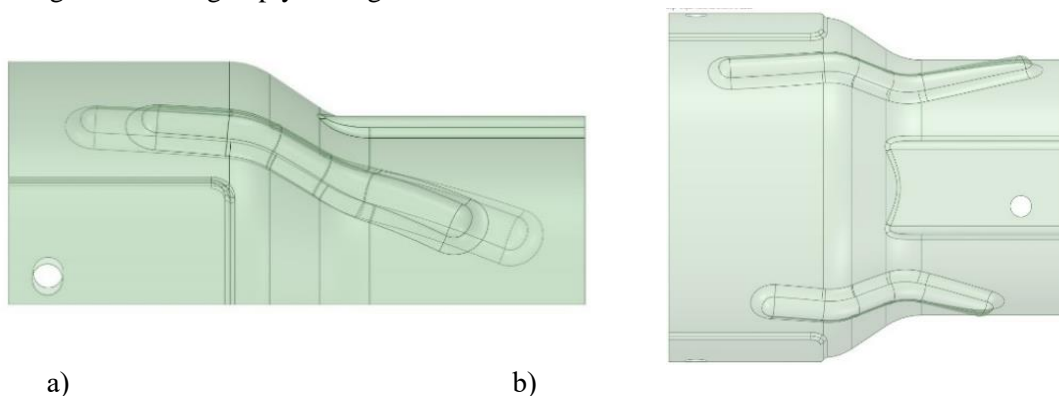
Tabela 5.5 zawiera wyniki optymalizacji (parametry projektowe i funkcjonały celu) dla tych rozwiązań (CS) w porównaniu z rozwiązaniem początkowym (IS), graficznie przedstawione na Rys. 5.41 i Rys. 5.42.

Dla Initial Solution (IS) – wcześniej oznaczone jako rozwiązanie referencyjne (rozdział 4.3), wraz ze wzrostem grubości osłony rośnie sztywność – kryterium (3.110) maleje, ale jednocześnie wzrastają maksymalne naprężenia i pierwsza częstotliwość rezonansowa. Zastosowanie w pracy zaproponowanych i opracowanych metod optymalizacyjnych pozwoliło na znalezienie rozwiązania kompromisowego, dla którego możliwe było znaczne poprawienie kryteriów (3.110) i (3.111) oraz wyznaczenie wartości trzeciego kryterium (3.112), dla którego wartość pierwszej częstotliwości rezonansowej znajduje się w bezpiecznej odległości od założonej częstotliwości referencyjnej opisanej w rozdziale 3.11.

Tabela 5.5 Porównanie optymalnych wartości dla wybranych rozwiązań kompromisowych i różnych przypadków rozwiązań początkowych

	DP1	DP2	...	DP7 (g) [mm]	...	DP20	DP21	f ₁ [mm]	f ₂ [MPa]	f ₃ [Hz]
IS	9	25	...	0.25	...	5	-20	0.61	195.5	303.4
	9	25	...	0.5	...	5	-20	0.58	310.7	501.1
	9	25	...	0.75	...	5	-20	0.57	399.2	659.3
	9	25	...	1.0	...	5	-20	0.56	468.2	799.4
	9	25	...	0.46	...	5	-20	0.59	294.3	476.2
CS	4.6	12.0	...	0.46	...	5.4	-13.6	0.34	166.2	516.9

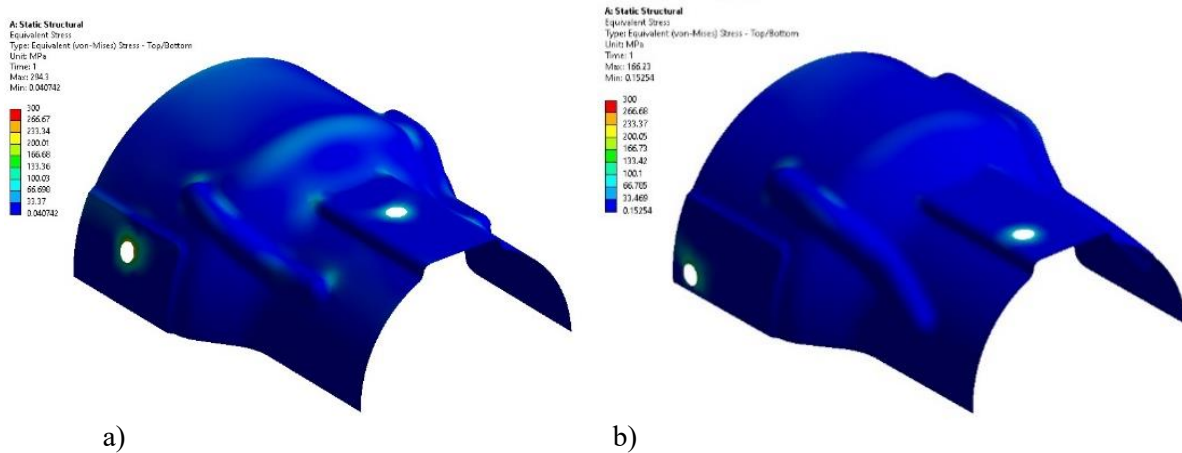
Znaleziony kompromis charakteryzuje się asymetrią zarówno wytłoczenia, jak i umiejscowienia górnego otworu montażowego. Rys. 5.41 przedstawia geometrię (widok z przodu i z góry) dla pierwszego omawianego optymalnego CS.



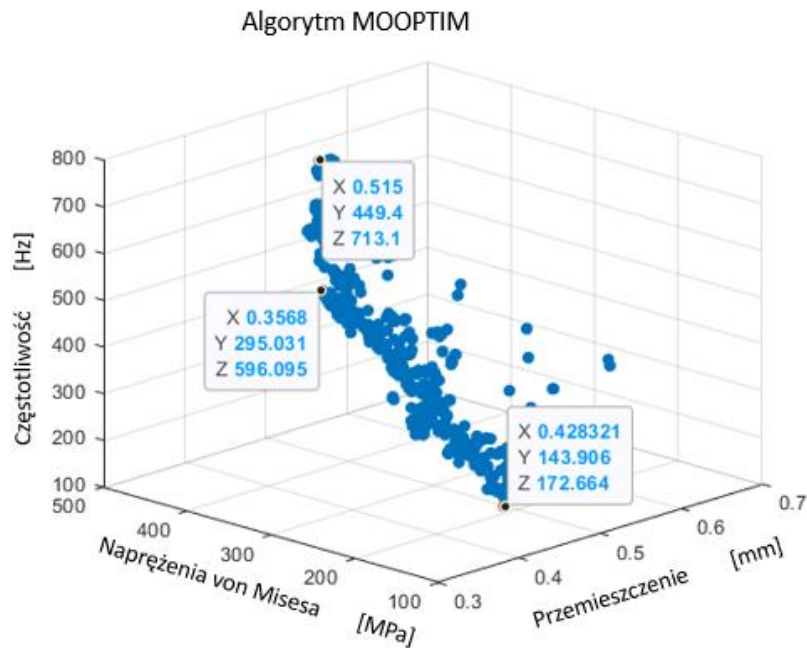
Rys. 5.41 Model geometryczny osłony termicznej dla wybranego rozwiązania kompromisowego

Ostatni wiersz dla rozwiązań IS w tabeli 5.5 zawiera wartości dla rozwiązania geometrycznie identycznego z wyjściowym, ale z grubością blachy równą wybranemu otrzymanemu rozwiązaniu

kompromisowemu. Porównując te rozwiązania, obserwujemy poprawę wartości funkcjonałów (3.110, 3.111, 3.112) osłony dla rozwiązania kompromisowego. Rys. 5.42 zawiera kolorowe mapy zredukowanych naprężeń von Misesa dla tych rozwiązań.



Rys. 5.42 Rozkład naprężeń w modelu dla wybranych rozwiązań: wyjściowego i kompromisowego. Kolejne trzy rozwiązania zostały wybrane z przestrzeni rozwiązań algorytmu MOOPTIM jak pokazano na Rys. 5.43.



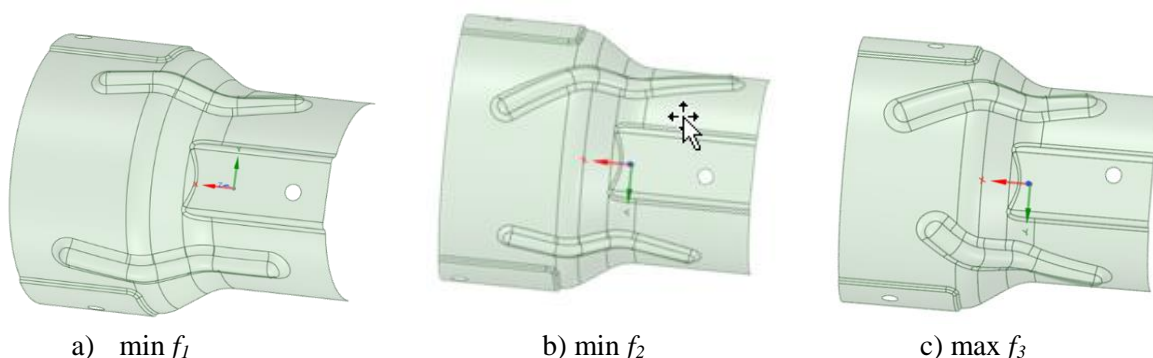
Rys. 5.43 Wybrane DPs dla przestrzeni rozwiązań algorytmu MOOPTIM

Zostały one zestawione w tabeli 5.6. Są to rozwiązania leżące najbliżej punktu UP (*ang. Utopian Point*), biorąc pod uwagę tylko jedno kryterium optymalizacyjne. Pozycja pierwsza tabeli zawiera wartości zmiennych konstrukcyjnych i wartości funkcjonałów dla najlepszego rozwiązania, kiedy kryterium nadrzędnym jest kryterium minimalizacji maksymalnych przemieszczeń $f_1 \rightarrow \min$. Druga dla kryterium minimalizacji maksymalnych naprężeń zredukowanych von Misesa $f_2 \rightarrow \min$, a trzecia dla kryterium maksymalizacji pierwszej częstotliwości drgań własnych $f_1 \rightarrow \max$.

Tabela 5.6 Optymalne wartości zmiennych projektowych oraz funkcjonałów dla najlepszych wariantów rozwiązań.

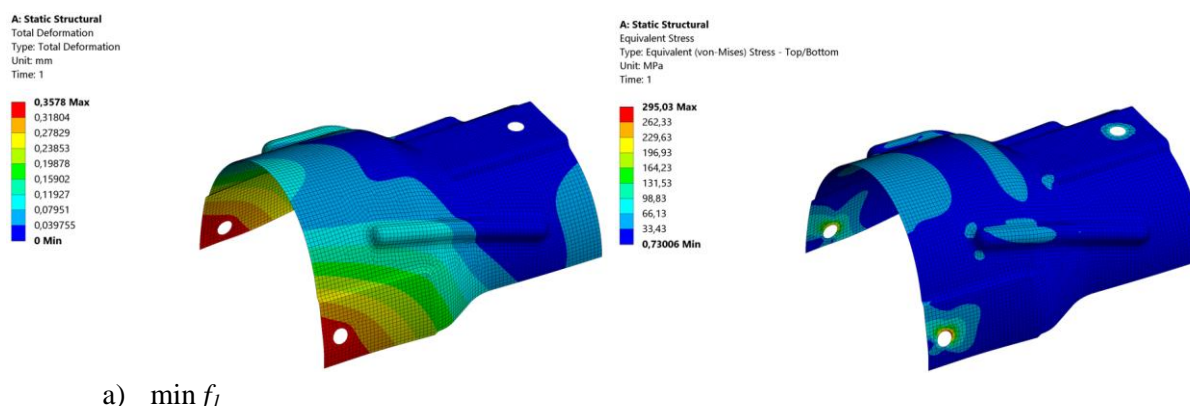
P1	P2	...	P6	P7 'g' [mm]	P8	...	P20	P21	P22 Przemieszczenie (mm)	P23 Napężenia von Misesa (MPa)	P24 Częstotliwość (Hz)	
9,917	10,449	...	-5,761	0,976	3,651	...	6,722	-11,979	0,357	295,031	596,095	<----- min f_1
6,985	14,404	...	-5,345	0,297	2,908	...	6,78	-11,941	0,428	143,906	172,664	<----- min f_2
13,249	11,947	...	-0,976	0,967	3,835	...	18,788	-19,257	0,515	449,401	713,146	<----- max f_3

Na Rys. 5.44 przedstawiono wygenerowane modele geometryczne dla wariantów z tabeli 5.6. Jak można zauważyć, modele różnią się między sobą położeniem otworów montażowych oraz obustronne przetłoczenia posiadają różną szerokość i trajektorie.

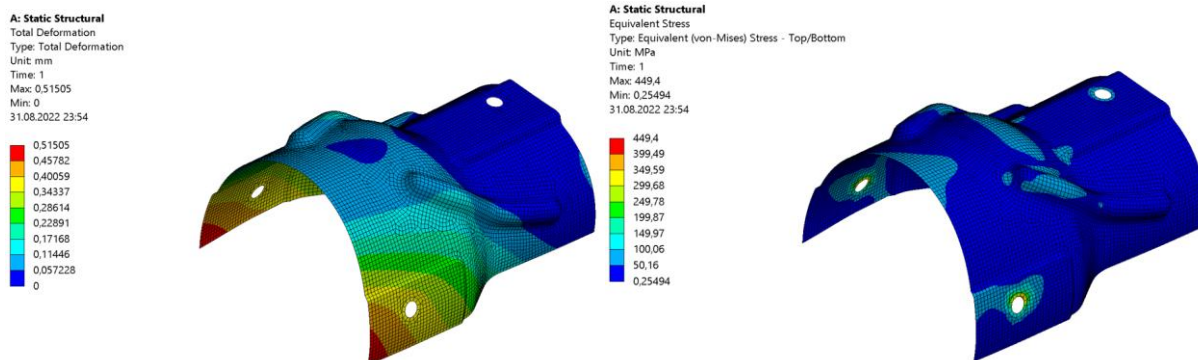


Rys. 5.44 Modele geometryczne dla wariantów z Tabeli nr.4

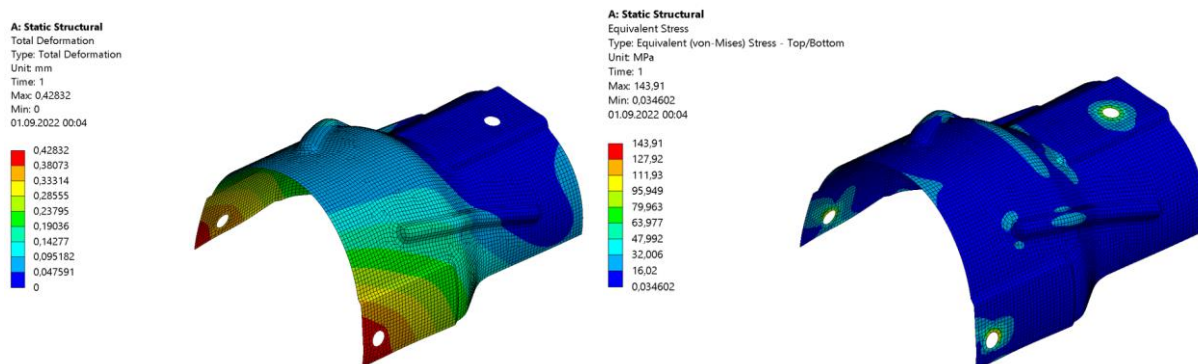
Położenie otworów, trajektoria przetłoczeń i grubość osłony mają wpływ m.in. na sztywność i wartość częstotliwości drgań własnych osłony. Wybranie jednego kryterium jako kryterium nadrzędnego ma wpływ na postać geometryczną osłony. Otrzymany kształt, położenie otworów montażowych, lokalizacja przetłoczeń, ma wpływ na wartość optymalizowanych funkcjonałów. Na Rys. 5.45 a, b i c przedstawiono odpowiednio wygenerowaną barwną mapę rozkładu przemieszczeń oraz naprężeń zredukowanych dla wariantów z tabeli 5.6.



a) min f_1



b) $\min f_2$



c) $\max f_3$

Rys. 5.45 a), b), c), Barwna mapa rozkładu przemieszczeń oraz naprężeń zredukowanych dla wariantów z tabeli 5.6

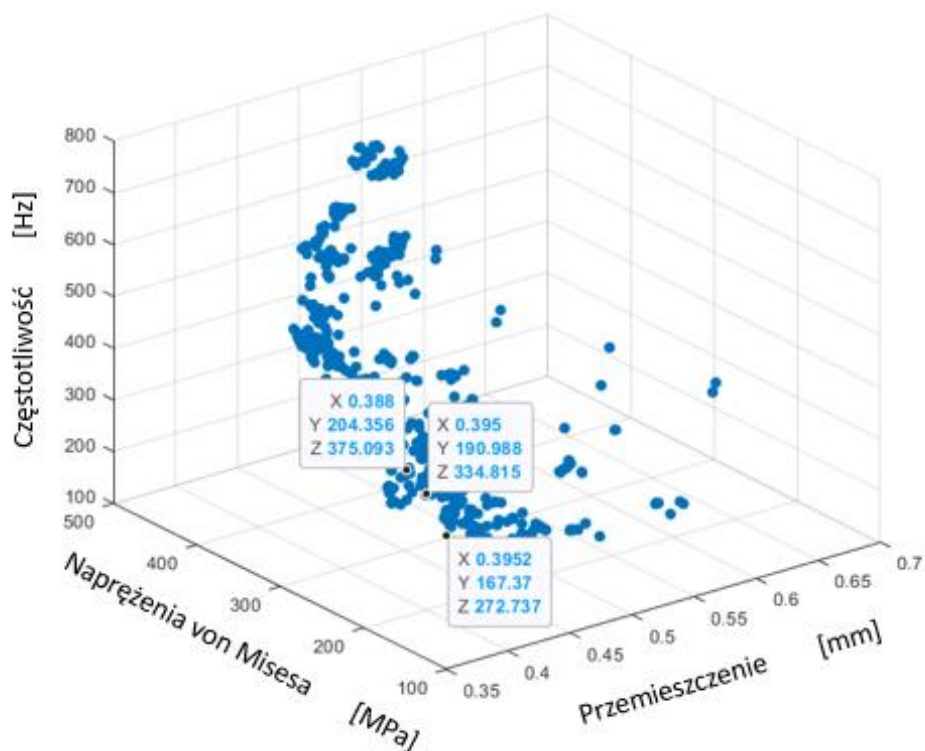
Jak pokazano w tabeli 5.6 i na Rys. 5.45, wartość minimalnego przemieszczenia waha się pomiędzy 0.357mm a 0.515mm. Wartość minimalnych naprężeń zredukowanych zmienia się pomiędzy 143,906 a 449,401 MPa. Natomiast wartość pierwszej częstotliwości drgań własnych osłony mieści się w zakresie od 172Hz do 713Hz.

Przeanalizowano i porównano dodatkowo trzy dowolnie wybrane rozwiązania kompromisowe (Rys. 5.46) z opisanymi powyżej przypadkami. Są to punkty z przestrzeni rozwiązań, które nie są optymalnymi rozwiązaniami pod kątem jednego z wybranych kryteriów, ale są to punkty, które się oznaczają poprawnymi wartościami funkcjonałów. Jak pokazuje tabela 5.7, wartości poszczególnych funkcjonałów dla tych rozwiązań kompromisowych różnią się od najlepszych rozwiązań pokazanych w tabeli 5.6. Wartość minimalnych przemieszczeń dla rozwiązań kompromisowych jest wyższa o 7.9%, wartość minimalnych naprężeń jest wyższa o 14%, a wartość pierwszej częstotliwości drgań własnych jest niższa aż o 47.4%. Dane te pokazują, że rozwiązania kompromisowe w pewnych sytuacjach, szczególnie na potrzeby przemysłu, mogą swobodnie konkurować z rozwiązaniami Pareto-optymalnymi dla optymalizacji wielokryterialnej.

Tabela 5.7 Wartość funkcjonałów dla rozwiązań kompromisowych

	P1	P2	...	P6	P7 'g' [mm]	P8	...	P20	P21	P22 Przemieszczenie (mm)	P23 Naprężenia von Misesa (MPa)	P24 Częstotliwość (Hz)
CS1	7,091	14,753	...	-5,587	0,403	3,079	...	6,723	-11,979	0,395	167,371	272,737
CS2	8,271	10,389	...	-4,301	0,525	3,015	...	17,051	-13,918	0,388	204,356	375,093
CS3	7,238	13,23	...	-5,567	0,468	3,123	...	14,182	-8,586	0,395	190,988	334,815

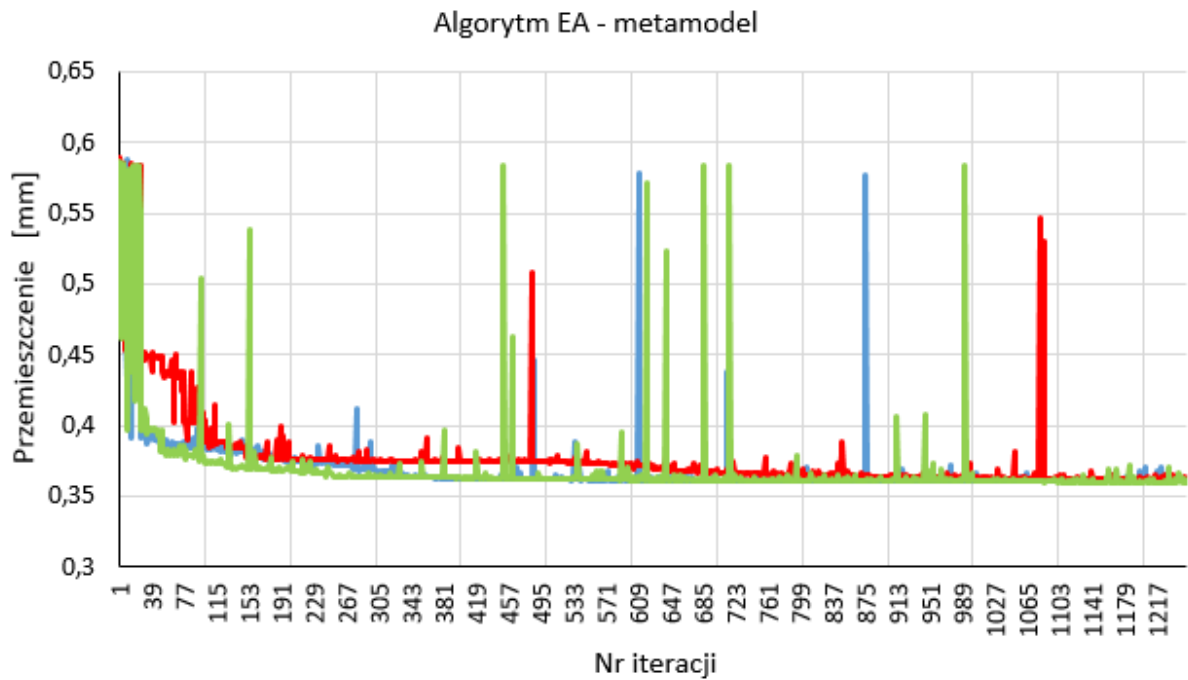
Powyższa prawie ośmioprocentowa różnica w przypadku funkcjonału przemieszczeniowego, przy wyznaczonej maksymalnej wartości minimalnego przemieszczenia w optymalizacji wielokryterialnej wynoszącej 0.569mm, daje nam odchyłkę rzędu 0.045mm, co w warunkach przemysłowych jest w większości przypadków akceptowalne.



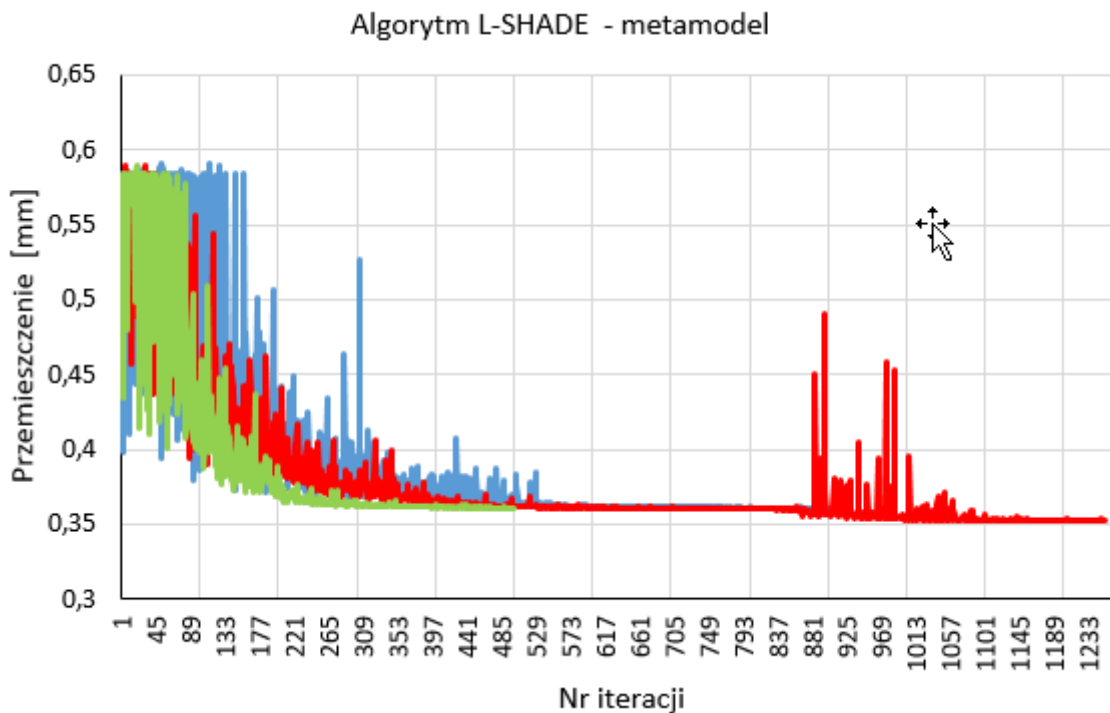
Rys. 5.46 Rozwiązania kompromisowe CS

5.3. Optymalizacja z zastosowaniem metamodelowania

Metamodel został stworzony w oparciu o wytrenowaną SSN dla przypadku z 20-toma sygnałami wejściowymi i jednym wyjściowym. Funkcjonał przemieszczeniowy był funkcjonałem poddanym optymalizacji. Wyuczony model SSN (typu RANDOM) został połączony wg schematu z rys.4.18 z algorytmem optymalizacji jednokryterialnej. Wykorzystano dwa algorytmy heurystyczne do przeprowadzenia zadań optymalizacji z metamodelem. Pierwszym z nich był klasyczny algorytm ewolucyjny EA, a drugim algorytm bazujący na metodzie ewolucji różnicowej L-SHADE. Każdy z nich został uruchomiony 3 razy. Wyniki przeprowadzonych optymalizacji pokazano na Rys. 5.47 i Rys. 5.48.



Rys. 5.47 Metamodel z algorytmem EA



Rys. 5.48 Metamodel z algorytmem L-SHADE

Uzyskane wyniki dla optymalizacji z wykorzystaniem metamodelu różnią się nieznacznie od wyników dla optymalizacji z oprogramowaniem MES. Minimalna wartość przemieszczenia uzyskana w obliczeniach z metamodelem wynosi dla EA 0,343mm, a dla algorytmu L-SHADE 0,334mm. W przypadku optymalizacji z wykorzystaniem MES minimalna wartość przemieszczeń dla algorytmu EA wynosiła 0,321, a dla algorytmu L-SHADE 0,311. Wartości przemieszczenia są niższe dla wariantu optymalizacji z wykorzystaniem MES odpowiednio dla EA o 0,022mm, a dla L-SHADE o 0,023mm.

W ujęciu procentowym różnica wynosi dla obu algorytmów 9.3% . Porównanie rezultatów optymalizacji z wykorzystaniem metamodelu i z wykorzystaniem MES pokazano w tabeli 5.8.

Tabela 5.8 Porównanie rezultatów metamodelu z MES

		EA		L-SHADE	
1	Wartość optymalna: algorytm + MES	$f_1 \min$	0,321	$f_1 \min$	0,311
2	Wartość optymalna: algorytm + Metamodel	$f_1 \min$	0,343	$f_1 \min$	0,334
3	Wartość obliczona: MES	$f_1 \min$	0,328	$f_1 \min$	0,322
Różnica w [%] (porównanie MES i Metamodel)			4,37%		3,60%

Następnie, dla tego samego wektora wejściowego zmiennych decyzyjnych, dla którego wariant optymalizacji z metamodelem uzyskał najlepsze rozwiązanie, obliczono wartości przemieszczeń dla modelu z wykorzystaniem MES (poz.3, tabela 5.8). Wartości przemieszczeń wynoszą dla EA 0,328mm oraz dla algorytmu L-SHADE 0,322mm. Różnica pomiędzy wartością minimalnego przemieszczenia obliczonego z wykorzystaniem metamodelu i za pomocą MES dla algorytmu EA wynosi 0,015mm. Dla algorytmu L-SHADE 0,012mm. W ujęciu procentowym wartości różnic pomiędzy nimi wynoszą odpowiedni 4,37% i 3,6%. Dla wartości rzeczywistych różnica przemieszczenia w granicach 5%, w praktyce, w przypadku osłon jest jak najbardziej akceptowalna.

Zastosowanie metamodelu przyspieszyło znacznie rozwiązywane zadania optymalizacji. Czas potrzebny do obliczenia jednego zadania brzegowego z wykorzystaniem MES wynosił 275s, a w przypadku użycia metamodelu tylko 9s. Uwzględniając liczbę wywołań funkcji celu dla jednego pełnego zadania optymalizacji (500 iteracji), to całkowity czas obliczeń wynosi dla wariantu z MES ponad 38 godzin, podczas gdy dla wariantu z wykorzystaniem metamodelu zaledwie 75 minut. (1h15'). Zatem, zastosowanie metamodelowania w rozwiązywaniu rozpatrywanych zadań optymalizacji, umożliwiło ponad 30-krotnie skrócenie czasu potrzebnego na otrzymanie rozwiązania.

6. Podsumowanie, wnioski oraz kierunki dalszych badań

Niniejsza rozprawa doktorska przedstawia problematykę związaną z metodami optymalnego projektowania osłon termicznych stosowanych w układach wydechowych silników spalinowych. Wyjaśniono oraz opisano szczegółowo problemy związane z praktycznymi aspektami i wymaganiami dotyczącymi procesu projektowego tego typu układów. Zaproponowana w pracy metoda optymalizacji znacznie wykracza poza obszar standardowego wykorzystania systemów do symulacji inżynierskich opartych na MES (większość systemów CAE). Pomimo możliwości rozwiązywania zadań optymalizacji z użyciem wbudowanych algorytmów optymalizacji w niektóre tego typu systemy, w ramach pracy opracowano koncepcję oraz zaimplementowano numerycznie metody pozwalające zastosować zarówno zewnętrzne algorytmy optymalizacji, jak i zewnętrzne procedury umożliwiające tworzenie modeli zastępczych MES. Otrzymane rezultaty optymalizacji z wykorzystaniem zewnętrznych procedur optymalizacji pozwalają na uzyskanie rozwiązań konkurencyjnych, a w niektórych przypadkach również lepszych w porównaniu do metod wbudowanych w istniejące oprogramowanie. Zostało to opisane i omówione w pracy.

Opracowanie właściwego modelu numerycznego poprzedzone zostało analizami numerycznymi uwzględniającymi kontakt w miejscu montażu osłony. Opracowano uproszczony model numeryczny zapewniający wymaganą dokładność, w którym analiza kontaktu nie była wymagana. Jest to szczególnie istotne w kontekście rozwiązywania zadań optymalizacji, ponieważ czas pojedynczej symulacji numerycznej jest w tym przypadku krytyczny. Ponadto przeprowadzono wiele analiz porównawczych dotyczących wpływowi rodzaju użytego elementu skończonego oraz wielkości siatki na dokładność wyznaczania przemieszczeń, naprężeń zredukowanych oraz częstotliwości drgań własnych osłony. Do zadań optymalizacji wykorzystano opracowany model numeryczny o akceptowalnej dokładności obliczeniowej.

W niniejszej rozprawie opracowano i rozwiązano kilka wariantów praktycznych zadań optymalizacji zarówno jedno, jak i wielokryterialnej. Sformułowano i zaimplementowano numerycznie kryteria jakości oraz ograniczenia w postaci funkcji celu zależnych od wielkości uzyskanych z symulacji numerycznych MES dla zagadnień termosprężystych oraz analizy modalnej. Kryteria te oraz ograniczenia wyrażają realne aspekty projektowe tego typu układów stosowane w przemyśle samochodowym. Zastosowano w sumie 9 różnych metod i algorytmów optymalizacji z czego 5 wbudowanych w oprogramowania ANSYS oraz 4 zewnętrzne. Zadania optymalizacji wymagały opracowania odpowiednich procedur umożliwiających automatyczne tworzenie zadań brzegowo-początkowych oraz automatycznego obliczania wartości funkcjonałów na podstawie rezultatów analiz bezpośrednich. To z kolei dla modelu osłony termicznej, który dla praktycznych zadań optymalizacji zależny jest od dużej liczby zmiennych projektowych, wymagało opracowania odpowiednich procedur w oprogramowaniu CAD. Ten etap pracy skutecznie zrealizowano za pomocą programów i skryptów opracowanych w języku programowania C++ oraz Iron Python (w oprogramowaniu SpaceClaim).

Opracowana metoda optymalizacji umożliwiająca zastosowanie zewnętrznych algorytmów optymalizacji znacząco rozszerza możliwości projektowe zarówno tradycyjnych metod projektowania, jak i projektowania z wykorzystaniem algorytmów wbudowanych w oprogramowanie MES. Algorytmy te, jak np. algorytm optymalizacji wielokryterialnej NSGA-II wbudowany w użyte oprogramowanie ANSYS jest algorytmem opracowanym wiele lat temu. Pomimo jego popularności oraz powszechności stosowania (np. biblioteki Matlab) w wielu zastosowaniach posiada również ograniczenia. Tak jest w przypadku jednoczesnego rozpatrywania liczby kryteriów większej niż 3. Problem ten jest znany i opisany w literaturze. Co prawda w ramach pracy nie rozpatrywano większej liczby kryteriów jednocześnie, jednakże zaproponowana i opracowana metoda optymalizacji pozwala na zastosowanie znacznie wydajniejszych algorytmów optymalizacji w tym przypadku, jak np. następcą tego algorytmu - NSGA-III lub algorytmów wykorzystujących elementy teorii gier oraz wielu innych.

Ponadto w ramach pracy przedstawiono i opracowano również sposób tworzenia modeli zastępczych MES z użyciem sztucznych sieci neuronowych. Proces doboru topologii sieci oraz jej uczenia pozwolił na opracowanie metamodelu charakteryzującego się bardzo dobrą jakością, dla którego pomimo relatywnie dużej liczby zmiennych projektowych błędy odpowiedzi są w pełni akceptowalne z praktycznego punktu widzenia. Opracowane metamodely użyte zostały również w rozwiązaniu wybranych zadań optymalizacji, rozwiązanych uprzednio przy wykorzystaniu wyłącznie analiz numerycznych MES. Takie podejście do optymalizacji choć poprzedzone nieco czasochłonnym procesem przygotowania modelu sieci neuronowych, pozwala na skrócenie czasu zadań optymalizacji nawet kilkudziesięciokrotnie.

Podjęte w ramach pracy badania warto kontynuować w przyszłości. Autor pracy widzi takie możliwości m.in. w zakresie:

- Opracowania bardziej złożonych modeli geometrycznych oraz numerycznych osłon dla modeli wielowarstwowych oraz wykonanych z innych materiałów.
- Opracowania modeli numerycznych większych komponentów w ramach których stosowane są osłony termiczne oraz przystosowanie ich do zadań optymalizacji.
- Opracowanie innych kryteriów oraz ograniczeń związanych uwzględnieniem wielkości pochodzących również z innych zjawisk fizycznych, jak np. obliczeniowej mechaniki płynów (ang. CFD) oraz interakcji pomiędzy nimi (zagadnienia sprzężone).
- Opracowanie i rozwiązanie zadań optymalizacji wielokryterialnej przy uwzględnieniu większej liczby kryteriów.
- Dalszych badań związanych z opracowaniem dokładniejszych metamodeli dla modeli MES.

Bibliografia

- [1] Altenberg L.: The schema theorem and price's theorem.
- [2] Aminuddin J., Wihantoro, Bilalodin, Sunardi, Rauf N., Designing of Muffler Part for Car Exhaust System with Low Emission and Noise using Conjugate Gradient Method, Journal of Physics: Conference Series 1494, 2020.
- [3] Anderson J.A., Rosenfeld E., Neurocomputing - Foundation of Research, MIT Press, Cambridge, Mass., 1988.
- [4] Antonisse H.J.: A new interpretation of the schema notation that overturns the binary encoding constraint. ICGA – 3, Morgan Kaufmann Publishers, 1989.
- [5] Arabas J.: Metoda ewolucyjna jako narzędzie przeszukiwania przestrzeni Banacha. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2005.
- [6] Atmar W.: The philosophical errors that plague both evolutionary theory and simulated evolutionary programming. ACEP -1, pp. 27 – 34, 1992.
- [7] Bäck Th., Eiben A.E.: Generalizations of intermediate recombination in evolution strategies. Angeline P.J., Porto W.V. (ed.) CEC, 2, pp 1566 – 1573, The IEEE Press, 1999.
- [8] Bäck Th., Fogel D.B., Michalewicz Z. (ed.): Handbook of Evolutionary Computation. Oxford University Press, 1997, New York, and Institute of Physics Publishing, 1997.
- [9] Bäck Th., Fogel D.B., Whitley L.D., Angeline P.J.: Search operators: mutation. Bäck Th., Fogel D.B., Michalewicz Z. (ed.): HEC, C3.2:1 – C3.2:14, Institute of Physics Publishing and Oxford University Press, 1997.
- [10] Bäck Th., Schefel P.H., Männer R.: An overview of evolutionary algorithms for parameter optimization. JEC, Vol. 1, pp. 1 – 23, 1993.
- [11] Bäck Th.: A genetic algorithm approach to the maximum independent set problem. Bäck Th., Fogel D.B., Michalewicz Z. (ed.): HEC, chapter G9.2, pp. 1 – 6, Oxford University Press, New York, and Institute of Physics Publishing, 1997.
- [12] Bäck Th.: Adaptive learning by evolutionary algorithms within search spaces of different complexity. Bock P., Loew M., Radermacher F.J., Richter M.M. (ed.): Proceeding of 3rd International FAW Workshop Adaptive Learning and Neural Networks, pp. 1 – 17, 1992.
- [13] Bäck Th.: Evolution strategies: An alternative evolutionary algorithm. Alliot J.M., Lutton E., Ronald E., Schönauer M., Snyers D. (ed.): Artificial Evolution: European Conference, Vol. 1063 of Lecture Notes in Computer Science, pp. 3 – 20, 1996.
- [14] Bäck Th.: Evolutionary algorithms in theory and practice. Oxford University Press, 1996.
- [15] Bäck Th.: Evolutionary algorithms: comparison of approaches. Paton R. (ed.): Computing with Biological Metaphors, chapter 14, pp. 227 – 243, Chapman & Hall, 1994.
- [16] Bäck Th.: On the behavior of evolutionary algorithms in dynamic environments. Fogel D.B., Schefel P.H., Yao x. (ed.), WCCI – 2 & IEEE / ICEC – 5, pp. 446 – 451, IEEE Press, 1998.
- [17] Bäck Th.: Optimization by means of genetic algorithms. Koehler E. (ed.): 36th International Scientific Colloquium, pp. 163 – 169, Technical University of Ilmenau, 1991.
- [18] Bäck Th.: The interaction of mutation rate, selection, and self – adaptation within a genetic algorithm. Männer R., Manderick B. (ed.), PPSN – 2, pp. 85 – 94, Free University of Brussels, 1992.
- [19] Bader J.: *Hypervolume-Based Search for Multiobjective Optimization*. Theory and Methods, PhD dissertation, Eidgenössische Technische Hochschule, Zuerich 2009.
- [20] Bąk R., Burczyński T., Wytrzymałość materiałów z elementami ujęcia komputerowego, WNT, Warszawa 2013.

- [21] Barbieria S.G., Giacomina M., Mangerugaa V., Mantovani S., A design strategy based on topology optimization techniques for an additive manufactured high performance engine piston, *Procedia Manufacturing* Vol. 11, 641 – 649, 2017.
- [22] Bergheau J., Fourtunier R., *Finite Element Simulation of Heat Transfer*, ISTE Ltd and John Wiley 7 Sons, Inc., London 2008.
- [23] Bergman A., Feldman W. M.: *Recombination dynamics and the fitness landscape*. PhD. Dissertation, 56, pp. 57-67, 1992.
- [24] Bethke A.: *Genetic algorithms as function optimizers*. PhD. Dissertation, Computer and Communication Sciences, university of Michigan, 1980.
- [25] Beyer H.G.: *Local performance measures: Evolution strategies and evolutionary programming*. Bäck Th., Fogel D.B., Michalewicz Z. (ed.): HEC, chapter B2.4, pp. 1 – 20, Oxford University Press, New York, and Institute of Physics Publishing, 1997.
- [26] Beyer H.G.: Some aspects of the “evolution strategy” for solving tsp – like optimization problems appearing at the design studies of a 0.5 tev – linear collider. Maener R., Manderick B. (ed.), *PPSN – 2*, pp.361-370, 1992.
- [27] Beyer H.G.: *Toward a theory of evolution strategies: self – adaptation*. *Evolutionary Computation*, pp.2.2:311 -347, 1995.
- [28] Beyer H.G. and Kalyanmoy Deb. *On Self-Adaptive Features in Real-Parameter Evolutionary Algorithm*. *IEEE Transactions on Evolutionary Computation*, 5(3):250 – 270, June 2001.
- [29] Bledsoe W.W.: *The use of biological concepts in the analytical study of systems*.
- [30] Booker L.B., Fogel D.B., Whitley L.D., Angeline P.J.: *Search operators: recombination*. Bäck Th., Fogel D.B., Michalewicz Z. (ed.): HEC, pp. C3.3:1 – C3.3:27, Institute of Physics Publishing and Oxford University Press, 1997.
- [31] Box G. E. P.: *Evolutionary operation: A method for increasing industrial productivity*. *Journal of Royal Statistical Society C*, 6 92):81 – 101, 1957.
- [32] Bremermann J.H.: *Optimization through evolution and recombination*. Yovits C. M., Jacobi T. G., Goldstein D. G. (ed.): *Self – organizing systems*, pp. 93 – 106, Spartan Books, 1962.
- [33] Brownlee J.: *Clever algorithms: nature-inspired programming recipes*. Jason Brown-lee, 2011.
- [34] Brzoska Z., *Wytrzymałość materiałów*. PWN, Warszawa , 1972.
- [35] Budynas R. G., Nisbett J. K., and Shigley J. E., *Shigley’s mechanical engineering design*, Tenth edition. New York, NY: McGraw-Hill Education, 2015.
- [36] Bui L., Alam S.: *Multi-objective Optimization in Computational Intelligence. Theory and Practise*, Information Science Reference IGI Global, 2008.
- [37] Burczyński T., Kuś W., Długosz A. [i in.]: *Solid Mechanics and Its Applications*, vol. 261, 2020, Springer, 237 s., ISBN 978-3-030-34159-6
- [38] Burczyński, T., Kuś, W., Beluch, W., Długosz, A., Poteralski, A., Szczepanik, M., *Intelligent Computing in Optimal Design , Solid Mechanics and Its Applications*, Series Volume 261, Springer International Publishing 2020.
- [39] Byrd R., Schnabel R.B., Schulz G.A. (1987): *A trust region algorithm for nonlinearly constrained optimization*, *SIAM Journal of Numerical Analysis*, Vol.24,1152-1170
- [40] Carter J. Booker J., *Finite Element Analysis of Coupled Thermoelasticity*, *Computer and Structures*, vol. 31, No. 1, pp73-80, 1989.
- [41] Celis M.R. (1983): *A trust region strategy for nonlinear equality constrained optimization*, Ph.D. Thesis, Department of Mathematics, Rice University, USA
- [42] Chandrupatla T.R. i A. D. Belegundu, *Introduction to Finite Elements in Engineering*, New Jersey, Prentice Hall, 2002.
- [43] Childs P. R. N., *Mechanical design engineering handbook*. Amsterdam: Butterworth-Heinemann, 2014.

- [44] Coello Coello C.A.: A Comprehensive Survey of Evolutionary-Based Multi-objective Optimization Techniques. *Knowledge and Information Systems*, vol. 1(3), pp. 129 – 156, 1999.
- [45] Conn A.R., Gould I.M., Toint P.L. (2000): *Trust-Region Methods*, SIAM, Philadelphia.
- [46] Cui Q., Ward M., Bundensteiner E.: Enhancing scatterplot matrices for data with ordering or spatial attributes. *Proc. SPIE 6060, Visualization nad Data Analysis*, vol. 6060, 2006.
- [47] Dayi Ou, Cheuk Ming Mak Modification of boundary condition for the optimization of natural frequencies of plate structures with fluid loading. August 24, 2018.
- [48] De Jong K. A., Spears W. M.: A formal analysis of the role of multi – point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5 pp. 1- 26, 1992.
- [49] Deb K. *Multi-Objective using Evolutionary Algorithms*. John Wiley & Sons, Ltd, 2001.
- [50] Deb K., Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182 – 197, April 2002.
- [51] Deb K.: *Evolutionary Multi-Objective Optimization Without Additional Parameters*. *Studies in Computational Intelligence*, vol. 54, Springer, Berlin 2007.
- [52] Deb K.: *Multi-Objective Genetic Algorithm. Problem Difficulties and Construction of Test Problems*. vol. 7(3), pp. 205 -230, 1999.
- [53] Dietrich M. et al., *Podstawy konstrukcji maszyn*. 1 1. Warszawa: Wydawnictwo Naukowe PWN, 2017.
- [54] Długosz A., „Optymalizacja wielokryterialna w problemach pól sprzężonych” WPS 2013,
- [55] Długosz A., Burczyński T., Kuś W.: Parallel evolutionary algorithms in shape optimization of heat exchangers under thermomechanical loading. *Evolutionary computation and global optimization 2006*.
- [56] Długosz A., Optimization in multiscale thermoelastic problems, *Computer Methods in Materials Science*, Vol. 14(1), 86-93, 2014.
- [57] Długosz A., Multiobjective evolutionary optimization of MEMS structures, *Computer Assisted Mechanics and Engineering Sciences*, Vol.17(1), 41-50, 2010.
- [58] Dyląg Z., Jakubowicz A., Orłoś Z.: *Wytrzymałość materiałów tom I i II*. WNT, Warszawa, 2000.
- [59] Ebrahimi-Nejad S., Kheybari M., Borujerd S., Multi-objective optimization of a sports car suspension system using simplified quarter-car models, *Mechanics & Industry* 21, 412, 2020.
- [60] Eiben A. E. et. al.: Orgy in the computer: multi – parent reproduction in genetic algorithms. *ECAL – 5*, 1995.
- [61] Elmqvist I., Dragicevi P., Fekete J., Rolling the dice. Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transaction on Visualization and Computer Graphics (TVCG)*, vol. 14, No. 6, pp. 1141 – 1148, 2008
- [62] Fletcher R.: *Practical Methods of Optimization. Volume 2, Constrained Optimization*, Wiley, Chichester, 1981.
- [63] Fletcher R. (1982): Second order correction for nondifferentiable optimization, in: Watson G.A. (Hrsg.): *Numerical analysis*, Springer Verlag, Berlin, 85-114
- [64] Fogel D. B., Atmar J.W.: Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear search. *Biological Cybernetics*, 63, pp. 111 – 114, 1990.
- [65] Fogel D.B.: An analysis of evolutionary programming. *Proceedings of the 1st Annual Conference on Evolutionary Programming*, pp. 43 – 51. Evolutionary Programming Society, 1992.
- [66] Fogel L.J., Owens J. A., Walsh M. J.: *Artificial intelligence through simulated evolution*. John Wiley & Sons Publishing, 1966.
- [67] Friedman J. G.: Digital simulation of an evolutionary process. *General Systems Yearbook*, 4, pp. 171 – 184, 1 959.

- [68] Gardea A.M., Valenzuela J.C., Topological optimization of automotive structures under impact using robust design, *Computer-Aided Design and Applications*, 33-47, 2015.
- [69] Gavin H. P., The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering Duke University, 2023.
- [70] Gawrylczyk K.M., *Metody optymalizacji*. Zachodniopomorski Uniwersytet Technologiczny, Szczecin, 2011.
- [71] Gill P.E., Murray W., Wright M.: *Practical Optimization*, Academic Press, New York, 1981.
- [72] *Global Optimization Toolbox. User's Guide*. The MathWorks. Inc.
- [73] Gogól W., *Wymiana ciepła: tablice i wykresy*, Wydawnictwo Politechniki Warszawskiej, Warszawa 1972.
- [74] Goldberg D., Deb K.: A comparative analysis of selection schemes used in genetic algorithms. Rawlings G. (ed.), *FOGA*, pp. 69 – 93, Morgan Kaufmann Publishers, 1991.
- [75] Grünz L., Beyer G. H.: Some observation on the interaction of recombination and self-adaptation in evolution strategies. Angeline P.J., Porto V. W. (ed.) *CEC*, 1, pp. 639 – 645, IEEE Press, 1999.
- [76] Gryko-Nikitin A., Niektóre osobliwości algorytmów genetycznych na przykładzie zagadnień logistycznych Certain peculiarities of the genetic algorithms based on logistic issues Politechnika Białostocka, Wydział Zarządzania, Katedra Informatyki Gospodarczej i Logistyki
- [77] Hebb D., *The Organization of behaviour*, New York, Wiley & Sons, 1949.
- [78] Holland J. H.: *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
- [79] Holland J. H.: *Genetic algorithms*. *Scientific American*, pp. 114 – 116, 1992.
- [80] <https://eur-lex.europa.eu/> - prawo legislacyjne Parlamentu Europejskiego dotyczące norm emisji spalin.
- [81] Jagdeesh H.K., Manjunatha K., Reddy M., Numerical and Experimental Investigation on Thermal Behavior of Exhaust Heat Shield, *International Journal of Innovative Science, Engineering & Technology*, Vol. 2, Issue 11, 2015.
- [82] Jarosz J., Długosz A., Shape optimization of the muffler shield with regard to strength properties. *Engineering Transactions*, 2023.
- [83] Jarosz J., Długosz A., Numeryczna analiza modalna osłony układu wydechowego, Gliwice, Politechnika Śląska, 2020, s.41-44, ISBN 978-83-951185-1-7
- [84] Jarosz J., Długosz A., Optymalizacja kształtu przetłoczeń osłony układu wydechowego silnika spalinowego, Gliwice, 2021, Politechnika Śląska, s.45-48, ISBN 978-83-951185-2-4
- [85] Jendo S., Paczkowski W.: Discrete polyoptimization method for finding the sets of nondominated solutions and their evaluation. *proc. in 3rd World Congress of Structural and Multidisciplinary Optimization*, pp. 335 – 337, Buffalo, New York, USA 1999.
- [86] Jendo S., Paczkowski W.: Multicriteria discrete optimization of large-scale truss systems. *Structural Optimization*, vol. 6, Issue 4, pp. 238 – 249, 1993.
- [87] Kalyanmoy Deb and R. B. Agarwal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9:115 – 148, April 1995.
- [88] Kaufmann W., Spears W. M., De Jong A. K.: On the virtues of uniform crossover. *ICGA – 4*, pp. 230 – 236, Morgan Kaufmann Publishers, 1991.
- [89] Knapczyk J., Maniowski M., Optimization of 5-rod car suspension for elastokinematic and dynamic characteristics, *Archive of Mechanical Engineering*, Vol. 57, No. 2, 133-147, 2010.
- [90] Kohonen T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, vol. 43(1), pp. 59-69, 1982.
- [91] Kong Y.S., Abdullah S., Omar M.Z., Haris S.M., Topological and Topographical Optimization of Automotive Spring Lower Seat, *Latin American Journal of Solids and Structures*, 13 (7), 2016.

- [92] Koza R. J., Bennett III H. F., Andre D., Keane M. A.: Genetic programming: biologically inspired computation that creatively solves non – trivial problems. Evolution as Computation Workshops, 1999.
- [93] Koza R. J.: Genetic programming: on the programming of computers by means of natural selection. The MIT Press, 1993.
- [94] Kucharski T.: Drgania mechaniczne. WNT, Warszawa, 2015, wydanie drugie.
- [95] Kusiak J., Danielewska-Tulecka A., Oprocha P.: Optymalizacja – wybrane metody z przykładami zastosowań. PWN, 2009.
- [96] La Jolla.: Recombination distributions for genetic algorithms. FOGA, Morgan Kaufmann Publishers, 1992.
- [97] Lewandowski – metody gradientowe– M. Lewandowski, Metody optymalizacji - teoria i wybrane algorytmy . 2012, skrypt
- [98] Liu X., Deng Y.D., Chen S., Wang W.S., Xu Y., Su C.Q., A case study on compatibility of automotive exhaust thermoelectric generation system, catalytic converter and muffler, Case Studies in Thermal Engineering 2, 62–66, 2014.
- [99] Marcelin J.L.. Optimization of the boundary conditions by genetic algorithms. International Review of Mechanical Engineering, pp.50-54. 2012.
- [100] McClelland T.L., Rumelhart D.E., and the PDP Research Group, Paralell Distributed Processing, MIT Press, Cambridge, Mass. 1986.
- [101] McCulloch Warren, Pitts Walter, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematica Biophysics, No 5, 1943, s. 115-133
- [102] Merkisz J. „Ekologiczne problemy silników spalinowych”. T.2 Poznań, WPP, 1999, ISBN 83-7143-039-6
- [103] Merkisz J. „Emisja cząstek stałych przez silniki spalinowe o zapłonie samoczynnym – wybrane zagadnienia”, Poznań, WPP 1997, ISBN 83-7143-162-7
- [104] Mescos, „ANSYS DesignXplorer – optymalizacja w ANSYS Workbench,” [Online] <https://mescos.com.pl>
- [105] Michalewicz Z. Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT, 2003.
- [106] Miettinen K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publisher, Boston 1999.
- [107] Moré J.J. (1983) Recent developments in algorithms and software for trust region methods, in: Bachem A., Grötschel M., Korte B. (eds.): Mathematical Programming: The State of the Art, Springer. Berlin, 258-287
- [108] Muhammad A. Mohammed A. H., Shanono I., Design optimization of a diesel connecting rod, Materials Today Proceedings Vol. 22, 1600–1609, 2020.
- [109] Muraszkievicz m., R. Nowak i inni, Sztuczna inteligencja dla inżynierów – metody ogólne, Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej, 2022.
- [110] Murata T. and Ishibuchi H., MOGA: Multi-objective genetic algorithms, Proc. of 1995 IEEE International Conference on Evolutionary Computation, pp. 289-294, Perth, Australia, November 1995.
- [111] Murawski K., Arciszewski T., De Jong K. A.: Evolutionary computation in structural design. Special issue on Computer Aided Design. Journal for simulated-based engineering, submitted, 2000.
- [112] Murawski K., Obliczenia ewolucyjne – geneza i zastosowanie. Cybernetics Faculty, Military University of Technology, Warszawa, 2001.
- [113] Niezgodziński M.E., Niezgodziński T., Wytrzymałość materiałów. PWN, Warszawa, 2002.
- [114] Nowacki W., Teoria sprężystości, PWN, Warszawa 1970.
- [115] Nowacki W., Termosprężystość, Ossolineum, Wrocław 1972.

- [116] Oden J., Kross A., Analysis of General Coupled Thermoelasticity Problems by the Finite Element Method. Proc. in 2nd Conference of Matrix Methods in Structural Mechanics, Ohio, USA, 1968.
- [117] Orłósz Z. (red.), Naprężenia cieplne, WNT, Warszawa 1991.
- [118] Piegl L.A., Tiller W., The NURBS book, Springer; 2nd edition, 1996
- [119] Płonka S.: Wielokryterialna optymalizacja procesów wytwarzania części maszyn. WNT, Warszawa 2011
- [120] Pohlheim H.: GEATbx: genetic and evolutionary algorithm toolbox for use with MATLAB, Ver. 1.92, 1997/98.
- [121] Powell M.J.D., Yuan Y. (1991): A trust region algorithm for equality constrained optimization, Mathematical Programming, Vol. 49, 189-211.
- [122] Radcliffe J. N., George W. A. F.: A study of set recombination. Forest S. (ed.), ICGA - 5, pp. 23 – 30, Morgan Kaufmann Publishers, 1993.
- [123] Rafiq M, G. Bugmann, D. Easterbrook, Neural network design for engineering applications, Computers and Structures, nr 79, pp. 1541-1552, 2001.
- [124] Raghuvanshi M.M. and O. G. Kakde. Survey on multiobjective evolutionary and real coded genetic algorithms. In Proceedings of the 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems, pages 150 – 161, 2004.
- [125] Ram Krishna Rathore, Kaushal Sharma, Amit Sarda, „An Adaptive Approach for Single Objective Optimization” Journal of Engineering Research and Applications ISSN:2248-9622. Vol4, Issue2(Version 1), February 2014, pp. 737-746
- [126] Rana B. S.: Examining the role of local optima and schema processing in genetic search. PhD. Dissertation, Colorado State University, 1999.
- [127] Rechenberg I.: Evolutionstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann – Holzboog, 1973.
- [128] Rosenblatt F.: The perceptron a perceiving and recognizing automation, Project PARA, Report No. 85-460-1, Cornell Aeronautical Laboratory, INC. Buffalo, N.Y., January 1957.
- [129] Rusiński E., J. Czmochoński, T. Smolnicki, Zaawansowana metoda elementów skończonych w konstrukcjach nośnych, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2000.
- [130] Rutkowska D. I inni. Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. PWN, ISBN 83-01-12304-4, 1999.
- [131] Schaffer D. J., Caruana A. R., Eshelman J. L., Das R.: A study of control parameters affecting online performance of genetic algorithms for function optimization. Schaffer J. D.(ed.), pp. 51 – 60, Morgan Kaufmann Publishers, 1989.
- [132] Schaffer D. J., Eshelman J. L.: On crossover as an evolutionarily viable strategy. Belew R. K., Booker L. B. (ed.), ICGA - 4, pp. 61 – 68, 1991.
- [133] Schaffer J.D.. “Multiple objective optimization with vector evaluated genetic algorithms”, Proc of 1st ICGA, pp 930100, 1985.
- [134] Spears W., De Jong K.A.: On the virtues of parameterized uniform crossover. Belew R.K., Booker L.B. (ed.), ICGA – 4, pp. 230 – 236, Morgan Kaufmann Publishers, 1991.
- [135] Srinivas N. and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, 2(3):221 – 248, 1994.
- [136] Śródka W. Trzy lekcje metody elementów skończonych - Materiały pomocnicze do przedmiotu wytrzymałość materiałów, Oficyna Wydawnicza Politechniki Wrocławskiej (PW) Wrocław 2004
- [137] Stoer J. (1985): Foundations of recursive quadratic programming methods for solving nonlinear programs, in: K. Schittkowski (ed.): Computational Mathematical Programming, NATO ASI Series, Series F: Computer and Systems Sciences, Vol.15, Springer

- [138] Storn R., Price K., ‘Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces’, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997, doi: 10.1023/A:1008202821328.
- [139] Szczepanik M., *Optymalizacja układów powierzchniowych z wykorzystaniem algorytmów ewolucyjnych*, Rozprawa doktorska, Katedra Wytrzymałości Materiałów i Metod Komputerowych Mechaniki, Politechnika Śląska, Gliwice, 2003.
- [140] Tadeusiewicz R., *Sieci neuronowe*, wyd. II, Akademicka Oficyna Wydawnicza RM, Warszawa, 1993.
- [141] Tan K., Lee T., Khor E.: Evolutionary algorithms for multi-objective optimization performance assessment and comparisons. Proc. in IEEE Congress of Evolutionary Computation, Seoul, Korea, 2001.
- [142] Tanabe R. and A. Fukunaga. 2013. Success-history based parameter adaptation for Differential Evolution. 2013 IEEE Congress on Evolutionary Computation (June 2013), 71-78 <https://doi.org/10.1109/CEC.2013.6557555>
- [143] Tanabe R. and A.S.Fukunaga, 2014. Improving the search performance of SHADE using linear population size reduction. 2014 IEEE Congress on Evolutionary Computation (CEC) (July 2014), 1658-1665, <https://doi.org/10.1109/CEC.2014.6900380>
- [144] Tarnowski W., Krzyżyński T., Maciejewski I., Oleśkiewicz R.: Poly-optimization a paradigm in engineering design in mechatronics. *Archive of Applied Mechanics*, vol. 81, Issue 2, pp. 141 – 156, 2011.
- [145] Tenneco Automotive Polska sp. z o.o., Praca własna. Rybnik, 2023.
- [146] Toint P.L. (1997): A nonmonotone trust-region algorithm for nonlinear optimization subject to convex constraints, *Mathematical Programming*, Vol. 77, 69-94
- [147] Vose M. D., Liepins G. E.: Schema disruption. *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 237 – 242, Morgan Kaufmann, 1991.
- [148] Wajand J. Werner J. „Silniki spalinowe średniej i małej mocy” WNT W-wa 1971.
- [149] While L., Hingston P., Barone L., Huband S., ‘A faster algorithm for calculating hypervolume’, *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, 29–38, 2006.
- [150] Wiśniewski S., Wiśniewski T.S., *Wymiana ciepła*, WNT, Warszawa 2000.
- [151] www.intercars.com.pl, [online], (dostęp: 03.3.2023)
- [152] www.dpf.mielec.pl/regeneracja/katalizatory-i-scr, [online], (dostęp: 03.3.2023)
- [153] www.toyotanews.eu, [online], (dostęp: 10.3.2023)
- [154] www.tarkom.eu, [online], (dostęp: 10.3.2023)
- [155] www.otomatic.de, [online], (dostęp: 21.3.2023)
- [156] www.mappingair.meteo.uni.wroc.pl/2020/05/normy-czystosci-spalin, [online], (dostęp: 21.3.2023)
- [157] www.peugeotnigeria.com, [online], (dostęp: 21.3.2023)
- [158] www.fiatprofessional.com/pl/NEDC, [online], (dostęp: 21.3.2023)
- [159] www.fiatprofessional.com/pl/wltp, [online], (dostęp: 10.4.2023)
- [160] www.continental-industry.com, [online], (dostęp: 10.4.2023)
- [161] www.htgetrid.com, [online], (dostęp: 25.4.2023)
- [162] www.autoneum.com, [online], (dostęp: 03.5.2023)
- [163] www.zabezpieczenia.com.pl/nowe-technologie, [online], (dostęp: 01.5.2023)
- [164] www.natural.pl, [online], (dostęp: 31.3.2023)
- [165] Yuan Y.: A review of trust region algorithms for optimization. *Iciam*. p. 271-282, 2000.
- [166] Zhang J. and A.C. Sanderson, 2009 JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE Transactions on Evolutionary Computation* 13, 5 (Oct 2009).
- [167] Zienkiewicz O.C., *Metoda elementów skończonych*. WNT Arkady Warszawa 1972.

- [168] Zienkiewicz O.C., Taylor R.L., The Finite Element Method, vol. 1-3, Butterworth, Oxford, 2000.
- [169] Zitzler E., Deb K., Thiele L.: Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, vol. 8(2), pp. 173 – 195, 2000.
- [170] Zitzler E., Thiele L., Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, 3, 4, 257–271, 1999.

Streszczenie

Rozprawa dotyczy zagadnień związanych z optymalnym doбором wybranych cech konstrukcyjnych osłon termicznych stosowanych w układach wydechowych silników spalinowych. W ramach pracy opracowano metodę optymalizacji kształtu oraz warunków brzegowych z wykorzystaniem parametrycznych modeli osłon termicznych. Do rozwiązania zagadnień związanych z symulacjami numerycznymi, zastosowano metodę elementów skończonych (MES) dla zagadnień termosprężystych oraz analizy modalnej. Opracowano procedury numeryczne oraz zaadoptowano komercyjne oprogramowane MES umożliwiające rozwiązywanie zadań optymalizacji z wykorzystaniem zarówno wewnętrznych, jak i zewnętrznych algorytmów optymalizacji, zwiększając tym samym znacząco możliwości rozwiązywania tego typu problemów w zastosowaniach praktycznych. Opracowano oraz zaimplementowano numerycznie kryteria jakości związane z wybranymi cechami wytrzymałościowymi, które wynikają z realnych wymagań przemysłowych, stawianych tego typu układom. Wykazano skuteczność oraz elastyczność zaproponowanej metody.

Ponadto w ramach pracy zaproponowano oraz opisano sposób tworzenia modeli zastępczych MES dla osłony termicznej (tzw. metamodeli), który umożliwia w zaproponowanej metodzie optymalizacji znacząco skrócić czas rozwiązywania zadań. Metamodelowanie przeprowadzono przy wykorzystaniu sztucznych sieci neuronowych o odpowiedniej topologii. Opracowane metamodele cechuje wysoka dokładność w stosunku do rozwiązań otrzymywanych z symulacji MES.

Przedstawione w pracy rezultaty badań w postaci rozwiązanych praktycznych zadań optymalizacji zarówno z zastosowaniem metamodelowania jak i bez niego, potwierdzają skuteczność oraz efektywność opracowanej metody.

Summary.

The dissertation deals with issues related to the optimal design of the selected features of heat shields used in exhaust systems of internal combustion engines. The dissertation developed a method for optimizing the shape and boundary conditions using parametric models of heat shields. The finite element method (FEM) for thermo-elastic issues and modal analysis were used to solve numerical simulations. Numerical procedures were developed and commercial FEA software was adopted to solve optimization tasks using both internal and external optimization algorithms, thus significantly increasing the ability to solve such problems in practical applications. Quality criteria related to selected strength characteristics, which are derived from real industrial requirements for this type of systems, were developed and numerically implemented. The effectiveness and flexibility of the proposed method were demonstrated.

In addition, the work proposes and describes a method of creating FEM surrogate models for the heat shield (so-called metamodels), which makes it possible in the proposed optimization method to significantly reduce the time of solved tasks. Metamodeling was carried out using artificial neural networks with appropriate topology. The developed metamodels are characterized by high accuracy in relation to the solutions obtained from FEM simulations.

The research results presented in the paper, in the form of solved practical optimization tasks both with and without metamodeling, confirm the effectiveness and efficiency of the developed method.