Višekriterijsko projektiranje složenih tankostjenih konstrukcija

Prebeg, Pero

Doctoral thesis / Disertacija

2011

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:820996

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-13**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

VIŠEKRITERIJSKO PROJEKTIRANJE SLOŽENIH TANKOSTJENIH KONSTRUKCIJA

DOKTORSKI RAD

Mentor:
Prof. dr. sc. Vedran Žanić

Pristupnik:
Pero Prebeg, dipl. ing.

ZAGREB, 2011.

PODACI ZA BIBLIOGRAFSKU KARTICU

UDK: 629.7.02:629.5.01

Ključne riječi: višekriterijsko projektiranje, optimizacija, tankostjene konstrukcije, dekompozicija, koordinacija, višedisciplinarna optimizacija

Znanstveno područje: TEHNIČKE ZNANOSTI

Znanstveno polje: zrakoplovstvo, raketna i svemirska tehnika

Institucija u kojoj je rad izrađen: Fakultet strojarstva i brodogradnje

Mentor rada: Dr. sc. Vedran Žanić, red. profesor

Broj stranica: 242

Broj slika: 139

Broj tablica: 29

Broj korištenih bibliografskih jedinica: 105

Datum obrane: 26. 12. 2011.

Povjerenstvo: Dr. sc. Zdravko Terze, red. profesor, predsjednik povjerenstva
Dr. sc. Vedran Žanić, red. profesor, mentor
Dr. sc. Damir Kalpić, red. profesor - član povjerenstva,

Institucija u kojoj je rad pohranjen: Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu

ZAHVALA

Iskrene zahvale profesoru Vedranu Žaniću, mom mentoru, na ukazanom povjerenju, strpljenju i prenesenom znanju tijekom svih ovih godina zajedničkog rada. Brojni razgovori i diskusije izbistrili su mi mnoge nedoumice te otvorili pregršt novih pitanja.

Posebna hvala članovima povjerenstva na korisnim sugestijama, primjedbama i komentarima koji su rad učinili boljim i razumljivijim.

Zahvaljujem prof. O. F. Hughesu i AMTC ogranku kompanije DRS na mogućnosti integracije nove verzije programa MAESTRO u ovaj rad te posebice Dr. Ming Ma na hitrom odgovaranju na sve moje zahtjeve.

Hvala prof. K. Schitowskom na ustupanju programa NLPQLP za korištenje u ovom radu. Također hvala Dr. ir. A.J. de Witu na ustupanju programskog koda za analizu nadzvučnog poslovnog zrakoplova.

Najljepša hvala kolegama i prijateljima Boži Važiću, Jerolimu Andriću, Stanislavu Kitaroviću, Marku Stipčeviću, Mati Grgiću, Karlu Piriću na zajedničkom radu i druženju te nesebičnoj tehničkoj pomoći u izradi rada.

Posebno zahvaljujem mojim informatičkim guruima, Gordanu Šikiću i Dariju Mihaljeviću, koji su mi svojim savjetima omogućili da manje vremena potrošim na istjerivanje buba iz programskog koda

Hvala zrakoplovnom društvu sa stanke za ručak na razgovorima koji nemaju veze sa zrakoplovstvom.

Hvala i svim profesorima, kolegicama i kolegama na zajedničkom radu i druženju tijekom proteklih godina.

Posebno zahvaljujem svojim roditeljima na ukazanom povjerenju . Oni su postavili temelje za sve sto je uslijedilo nakon napuštanja roditeljskog doma.

Na kraju, najveće hvala mojoj supruzi Marti na podršci, ljubavi i razumijevanju koje je pokazala na ovom lijepom, ali dugom i iscrpljujućem putu.

Zagreb, prosinac 2011.

Pero Prebeg

S A D R Ž A J

PREDGOVOR vii
SAŽETAK ix
SUMMARY x
Ključne riječi: xi
Key words: xi
POPIS OZNAKA xii
POPIS SLIKA xiv
POPIS TABLICA xviii
1. UVOD 1
HIPOTEZA RADA 3
2. OSNOVNE KARAKTERISTIKE PROJEKTIRANJA SLOŽENIH TANKOSTJENIH KONSTRUKCIJA 4
2.1. Zahtjevi pri projektiranju složenih tankostjenih konstrukcija 5
2.2. Mjere kvalitete pri projektiranju složenih tankostjenih konstrukcija 7
2.3. Osnovne karakteristike matematičkih modela analize u konceptualnom i preliminarnom projektiranju 7
2.4. Pregled istraživanja 9
3. OSNOVE OPTIMIZACIJKOG PROBLEMA I OPTIMIZACIJSKE METODE 17
3.1. Definicija projektnog problema 17
3.1.1. Projektni prostor 19
3.1.2. Atributni prostor 19
3.1.3. Definicija koncepta nedominiranosti 20
3.2. Optimizacijske metode 20
3.2.1. Sekvencijalno linearno programiranje (SLP) 20
3.2.2. Sekvencijalno kvadratično programiranje (SQP) 23
3.2.3. Optimizacija rojem čestica (*Particle swarm optimization- PSO*) 25
3.3. Mjerenje kvalitete Pareto fronte 33
3.3.1. Indikatori kvalitete ostvarene Pareto fronte 35
4. SUROGAT MODELIRANJE 43
4.1. Uvod 43
4.2. Metode planiranja računalnih pokusa 44
4.2.1. Puni faktorski plan pokusa 45
4.2.2. Djelomični faktorski planovi pokusa 46
4.2.3. Optimalni planovi pokusa 49
4.2.4. Popunjavajuće metode uzorkovanja 50
4.3. Surogat modeli 51
4.3.1. Odzivne površine - polinomski surogat modeli 51
4.3.2. Kriging 54
4.3.3. Radijalne bazne funkcije 57
4.3.4. Validacija surogat modela 60
4.3.5. Analiza varijance polinomskih surogat modela 61
4.4. Primjena surogat modeliranja u optimizaciji 63
5. DEKOMPOZICIJA I KOORDINACIJA PROJEKTNOG PROBLEMA 64
5.1. Opis korištene notacije za hijerarhijske sustave 65
5.2. Dekompozicija problema 68

5.2.1. Načini izvođenja dekompozicije ... 69
5.2.2. Klasifikacija interakcije sustava/podsustava prema jačini i količini spregnutosti ... 72
5.3. Koordinacija ... 74
5.3.1. Tri tipa odluka pri koordinaciji ... 74
5.3.2. Načini izvođenja koordinacije ... 76
5.4. Koordinacijske metode ... 78
5.4.1. Tradicionalne jednorazinske MDO metode ... 79
5.4.2. Optimizacija linearnom dekompozicijom ... 83
5.4.3. Linearizirana višerazinska dekompozicija ... 85
5.4.4. Kvazi-odvojiva dekompozicija ... 86
5.4.5. Združena optimizacija ... 87
5.4.6. Analitičko propagiranje meta (ATC - Analytical Target Cascading) ... 88
5.5. Perspektivni smjerovi razvoja optimizacije bazirane na dekompoziciji odnosno multidisciplinarne optimizacije ... 90
6. UNAPRJEĐENJE PROCESA PROJEKTIRANJA SLOŽENIH TANKOSTJENIH KONSTRUKCIJA ... 91
6.1. Diskusija o kriterijima kvalitete Pareto fronte realnih složenih tankostjenih konstrukcija ... 91
6.2. Predložena dekompozicija i rješavanje problema projektiranja složene tankostjene konstrukcije ... 92
6.2.1. Dekompozicija problema ... 93
6.2.2. Sekvenca rješavanja problema ... 97
6.3. Primjena surogat-modela za supstituiranje proračuna odziva modula za analizu konstrukcije ... 101
6.3.1. Odabir odzivnih veličina za surogat model ... 101
6.3.2. Odabir parametara surogat modela ... 103
6.3.3. Analiza utjecajnosti parametara surogat modela ... 108
6.4. Primjena hibridnih optimizacijskih metoda ... 110
7. PROGRAMSKO OKRUŽENJE ZA POTPORU PRI PROJEKTIRANJU ... 112
7.1. Tražene karakteristike okružja za potporu pri projektiranju ... 115
7.2. Opis arhitekture ... 116
7.2.1. Unified Modeling Language (UML) dijagrami ... 117
7.2.2. Osnovne klase okruženja za potporu pri projektiranju ... 118
7.2.3. Generički model za analizu (*DeModel*) ... 119
7.2.4. Optimizacijski poslovi u okruženju za potporu projektiranju ... 126
7.2.5. Klase za definiranje optimizacijskih problema složenih sustava korištenjem dekompozicije na podsustave ... 131
7.2.6. Klase za definiranje komponenti optimizacijskog problema ... 138
7.2.7. Poslovi bazirani na planiranju pokusa (DOE) u okruženju za potporu projektiranju ... 140
7.2.8. Surogat-modeler (Surogator) ... 142
7.2.9. Asinkroni rješavač višestruko ponavljajućih zadataka – DeCluster ... 146
8. PRIMJENA RAZVIJENOG OKRUŽENJA NA TESTNIM PRIMJERIMA ... 154
8.1. Projektiranje tankostjene konstrukcije jednostavnog pontona ... 154
8.1.1. Model za analizu tankostjene konstrukcije pontona ... 155
8.1.2. Formulacija osnovnog optimizacijskog problema ... 155
8.1.3. Primjena surogat modela za aproksimaciju odziva ... 157

8.1.4. Usporedba rješavanja optimizacijskih problema sa i bez surogata strukturnih odziva 164
8.2. Projektiranje tankostjene konstrukcije broda za prijevoz putnika i vozila – ROPAX 171
8.2.1. Model za analizu tankostjene konstrukcije RoPax-a 172
8.2.2. Projektni model RoPax-a 180
8.3. Konceptualno projektiranje nadzvučnog poslovnog zrakoplova – SBJ 188
8.3.1. Model analize 188
8.3.2. Projektni model nadzvučnog zrakoplova 194
9. ZAKLJUČAK RADA 209
ZAKLJUČAK DOKTORSKOG RADA 213
LITERATURA 214
KRATKI ŽIVOTOPIS 223
CURRICULUM VITAE 224

PREDGOVOR

Tijekom proteklih dvaju desetljeća može se primijeti zamjetan porast znanstvenoistraživačkih radova s područja optimizacije i potpore pri odlučivanju. Taj rast uvelike je potpomognut napretkom računala, ali i zahtjevima industrije koja primjenom znanstvenoistraživačkih dostignuća s područja sinteze želi osigurati svoj opstanak na vrlo zahtjevnom tržištu današnjice. U takvu ozračju vrlo je bitno osigurati primjenu odgovarajuće metodologije projektiranja kompleksnih inženjerskih sustava (brod, zrakoplov). Pravilna primjena odgovarajuće metodologije projektiranja može dati kritičnu prevagu na tržištu. Kako bi se omogućila primjena znanstvenoistraživačkih dostignuća s područja teoretske sinteze u realnim procesima projektiranja složenih inženjerskih sustava, potrebno je analizirati zahtjeve promatranog inženjerskog sustava zajedno sa svim njegovim posebnostima. Nakon toga potrebno je primijeniti one metode sinteze koje po svojim osobinama odgovaraju identificiranim zahtjevima i karakteristikama promatranog inženjerskog sustava.

Projektiranje složene tankostjene konstrukcije (STK), poput broda ili zrakoplova, pripada skupini visoko dimenzionalnih (*large scale)* problema s nekoliko projektnih ciljeva, više stotina projektnih varijabli i više tisuća projektnih ograničenja. Postojanje više od jednog kriterija prema kojima se procjenjuje kvaliteta složenog sustava poput STK (u literaturi poznati i kao *Key Performance Indicators - KPI*) onemogućuje donošenje objektivne odluke koja je projektna varijanta najbolja te zahtijeva korištenje neke od tehnika višekriterijskog odlučivanja.

Sa stajališta optimizacije, uz već spomenutu dimenzionalnost problema, vrlo su važne karakteristike modula za analizu problema. U zrakoplovstvu i brodogradnji ti moduli mogu uključivati nelinearnu analizu strukture konačnim elementima, rješavanje nelaminarnih i nestacionarnih problema računalne mehanike fluida te proračune koji uključuju interakciju elastičnog tijela i fluida kroz koji se giba. Njihovo direktno korištenje unutar optimizacijskog procesa često nije moguće jer optimizacija zahtijeva više stotina ili čak tisuća analiza. Takvi problemi mogu se pokušati riješiti pojednostavljenim, često i linearnim, metodama niže razine točnosti, ali i aproksimacijskim, odnosno surogat modelima koji mogu zamijeniti stvarni model u pojedinim dijelovima optimizacijskog procesa.

Uspješno projektiranje složenih inženjerskih sustava uključuje dekompoziciju sustava na određeni broj manjih podsustava, odnosno potproblema sa njihovim ciljevima i

ograničenjima. Pri tome te sustave nije moguće nezavisno optimizirati zbog toga što su takvi sustavi međusobno spregnuti. Optimizacija jednog podsustava, koja ne uzima u obzir efekt njihove međusobne interakcije, može dovesti do snažne degradacije performansi cjelokupnog sustava tako da krajnji rezultat optimizacije može biti lošiji nego na početku.

S obzirom na karakteristike arhitekture današnjih računala, kod kojih se ukupna proračunska snaga više bitno ne povećava povišenjem radnog takta procesora, već povećanjem broja radnih jezgri, jedna od bitnih karakteristika jest mogućnost paralelnog izvršavanja određenih sekvenci naredbi. Prethodno navedene metode sinteze, optimizacijske metode, a posebice surogat modeliranje te dekompozicija na podsustave s koordiniranom optimizacijom, omogućuju paralelizaciju sekvence rješavanja projektnog problema neovisno o tome omogućuju li to moduli za analizu promatranog problema ili ne.

Iako se u znanstvenoistraživačkom radu na fakultetima i institutima, kao i u razvijenijim visoko sofisticiranim ograncima industrije već više desetljeća redovito koriste razni sustavi višeprocesorskih i/ili distribuiranih računalnih sustava, u segmentu malih i kućnih računala veliki potencijal i dalje ostaje neiskorišten. Naime, na fakultetima, u školama, na institutima te u poslovnim uredima često postoji velik broj računala koja se koriste samo tijekom standardnog radnog vremena. Povezivanje tih računala u male radne klastere može pojedinoj instituciji / uredu dati bitan računalni kapacitet iskoristiv izvan standardnog radnog vremena. Jedna od najvećih zapreka u ostvarivanju te ideje jest nepostojanje dovoljno jednostavnih programskih rješenja kojima bi se ta ideja ostvarila. Postojeća programska rješenja najčešće zahtijevaju visokoobrazovanu tehničku podršku informatičke struke samo za povezivanje računala u klastere, dok se programski kod pojedinog algoritma najčešće mora prilagoditi implementiranom programskom rješenju.

SAŽETAK

Višekriterijsko projektiranje složenih inženjerskih sustava metodološki je, vremenski i računalno zahtjevan proces. Pri rješavanju takvih problema koriste se metode sinteze poput optimizacijskih algoritama, dekompozicijsko-koordinacijskih metoda, aproksimacijskog (surogat) modeliranja i metoda za potporu pri odlučivanju. Projektiranje složene tankostjene konstrukcije broda ili zrakoplova spada u skupinu visoko dimenzionalnih (*large scale*) problema s više projektnih ciljeva, više stotina projektnih varijabli i više desetaka tisuća projektnih ograničenja. U radu je provedena klasifikacija koncepata iz domene optimizacije složenih sustava, što obuhvaća formulacije optimizacijskih problema za takve sustave te postupke primjenjive za njihovo rješavanje. Na osnovi analize karakteristika pojedinih metoda sinteze te karakteristika samog projektiranja složenih tankostjenih konstrukcija izvedeno je unapređenje procesa projektiranja tankostjenih konstrukcija dekompozicijom na funkcionalne jedinice -podstrukture (unutar kojih su sadržani elementi čija su svojstva bitnije spregnuta) te povezivanje, odnosno koordinacija podstruktura surogat modelom strukturnih odziva. Korištenjem tehnika objektno-orijentirane paradigme, svojstva i ponašanje optimizacijskih metoda, dekompozicije problema te surogat modeliranja, a u određenoj mjeri i integriranih modula analize, preneseni su u okruženje za potporu pri projektiranju. Time je omogućeno intuitivno korištenje tih metoda putem prezentacijskog sloja računalnog okruženja, pri rješavaju problema projektiranja složenih sustava općenito, pa tako i pri projektiranju tankostjenih konstrukcija.

SUMMARY

Multi-criteria design of complex engineering systems is methodologically and computationally demanding and time consuming process. Optimization algorithms, problem decomposition, surrogate metamodeling and methods for decision support are synthesis methods that are usually employed to solve the design problem. The design of complex thin-walled structure (eg. ship and/or aircraft) falls within the category of large scale problems characterized by several design objectives, hundreds of design variables and tens of thousands of design constraints. The classification of concepts in the complex systems optimization domain, which includes the formulation of optimization problems for such systems and procedures applicable for their solution, has been done first. Based on the undertaken classification and the characteristics of a thin-walled structures design, the process of thin-walled structures design has been improved by decomposition on the functional units - substructures (within which are elements whose properties are strongly coupled), and the coordination between structural substructures by surrogates of a structural response. Using the object oriented programming paradigm techniques, properties and a behavior of optimization methods, a problem decomposition, a surrogate modeling and, to some extent, integrated analysis modules, have been transferred into the design support environment. That enables an intuitive use of these methods through the presentation layer of the computing environment, in solving the problem of complex systems design in general, including the design of complex thin-walled structures.

Ključne riječi:

– višekriterijsko projektiranje, strukturna optimizacija, tankostjene konstrukcije, surogat modeli, dekompozicija problema

Key words:

– multi-criteria decision making, structural optimization, thin-walled structure, surrogate modeling, problem decomposition

POPIS OZNAKA

(zasnovan na osnovnoj taksonomiji OCTOPUS sustava)

d, d⁰, d⁻ n-torka (svih, osnovnih) deskriptora,
d = {d_i}={**x**, $\mathbf{d}^-$}; d_i = {$d_{i\,MEAN}$, statistics $_i$}, i=1,..n**.**

D ^{k} projektna varijanta k (ili projekt $\mathbf{P}^k$) = {**d, y, m, l** }^{k}

m n-torka normaliziranih atributa; točka $\mathbf{m}^k$ u **M,** $\mathbf{m}^{ideal}$**=**{m_i^{max}}

M metrički atributni prostor razapet između/određen sa atributa m_i

l n-torka kompozitnih atributa; točka $\mathbf{l}^k$ u **L**

L atributni prostor razapet između/određen sa kompozitnih atributa l_i

x n-torka projektnih varijabli x_i, i=1,.n_x; točka $\mathbf{x}^k$ u **X,**
x = {$\mathbf{x}^{Topologija}$, $\mathbf{x}^{Geometrija}$, $\mathbf{x}^{Materijal}$, $\mathbf{x}^{Scantlings}$}

X, $\mathbf{X}^{\geq}$, $\mathbf{X}^{ND}$ projekt, podoban, nedominirani projekt određen sa x_i,

y n-torka atributa; točka $\mathbf{y}^k$ u **Y**

Y atributni prostor razapet između/određen sa atributa y_i

a n-torka atributnih funkcija a_i: **(d, z)** → y_i

g n-torka funkcija ograničenja g_i: **(d, z)** → I_{gi} (indikator zadovoljenja ograničenja), $\mathbf{X}^{\geq}$ = {**x** | I_{gi} **=** zadovoljenje, za svaki *i*};

o n-torka projektnih funkcija o_i = određen sa a_i, u_i, l_i

p n-torka **c** funkcija temeljenih na teoriji vjerojatnosti e.g.:
REL: **g**(**d, z**) → $p_{failure}$; ROB: a_i (**d, z**) → mjera robusnosti

r n-torka funkcija odziva (e.g. 3D FEM) r_i: **d**→ $\mathbf{z}^i$

u n-torka subjektivno normaliziranih atributnih funkcija

u_i (y_i,$\mathbf{P}^u$)→m_i; $\mathbf{P}^u$=unutrašnje / vanjske preference dane od strane projektanta

v n_v-ntorka vrijednosnih funkcija v_i: (**m,** $\mathbf{P}^v$**)**→l_i, ili alternativno:
v_i (**u** (ROB (**a** (**d, z**), $\mathbf{P}^u$), $\mathbf{P}^v$) = l_i.; e.g. v_i= L_p =Σ|$\mathbf{m}^*$-**m**|^p) $^{1/p}$; $\mathbf{m}^* \subseteq \mathbf{D}^{target}$;

α podobnosni podsustav (modul za podobnost); podskup modula u modelu za analizu (AM) koji sadržava sigurnost (npr. klasifikacijska pravila), iznos funkcija ograničanje $g_{i.}$. Izlazne vrijednosti: $\mathbf{l}_{gi}$**, g-**values**.**

Γ **(**Interaktivni modul) Skup modula za sintezu u modelu za sintezu (SM) za optimizaciju (koristeći podatke $\mathbf{P}^u$, $\mathbf{P}^v$ za subjektivnu definiciju **u** i **v**)**,** interakcija projektanta sa procesom projektiranja, filtriranje projekata i vizualizacija **X, Y,M,L** prostora. Izlazne vrijednosti: **m, l**.

Δ skup modula za definiciju optimizacijskog problema (odabir varijabli **x** i kriterijskih funkcija **a** i **g**)

ε modul za opisivanje i zadavanje stanja okoliša - (opterećenja, ekonomski uvjeti na tržištu); podskup modela u AM sa generatorima podataka E: $\mathbf{d}^{\varepsilon}$= {$\mathbf{d}^{tlakoviLC}$, $\mathbf{d}^{akceleracijeLC}$, $\mathbf{d}^{mase}$, $\mathbf{d}^{cijena}$} = E($\mathbf{d}^0$) ⊆ **d**

Φ strukturni **(**fizički) sustav; sa podskupom modula/modelera u AM/SM sa generatorom podataka: $\mathbf{d}^{\Phi}$ = {$\mathbf{d}^{topologija}$, $\mathbf{d}^{geometrija}$, $\mathbf{d}^{materijal}$, $\mathbf{d}^{scantlings}$} = F($\mathbf{d}^0$) ⊆ **d**

π metasustav za pozdanost/ robusnost, podskup AM sadržava module bazirane na: **REL/ROB** ($\mathbf{d}_{MEAN}$, **z**) funkcijama. Izlaz: $p_{failure}$ mjera pouzdanosti

ρ meta-sustav za odziv; podskup AM, sadržavajući module za FEM procedure r_i.
Output: **z** (load effects).

Σ skup optimizacijski algoritama (Sekvencijalno linearno programiranje- (SLP), Frakcionirani faktorski pokusi (FFE), Višeciljna optimizacija rojem čestica (MOPSO), Višeciljni genetski algoritmi (MOGA), Evolucijaka strategija - adaptivni *Monte Carlo* (ES-AMC) , itd.) koji generiraju optimalna rješenja ili Pareto frontu $\{\mathbf{x}^k, \mathbf{y}^k\}^{ND}$ tako što filtriraju projekte u $\mathbf{X}^{\geq}$ U $\mathbf{Y}^{\geq}$ koji su bazirani na **o**.

Ω meta-sustav za kvalitetu projekta; podskup AM/SM sadržavajući funkcije a_i. Izlaz: **y**.

POPIS SLIKA

Sl. 2.1. Mogućnost utjecaja na ukupne troškove zrakoplova kroz njegove životne faze [1] 4
Sl. 2.2. Faze projektiranja složenih tankostjenih konstrukcija 4
Sl. 3.1. Osnovna formulacija potpore procesu odlučivanja (DSP) [11]. 18
Sl. 3.2. Primjer sekvencijalne linearizacije ograničenja [15] 22
Sl. 3.3. Dijagram toka osnovnog PSO algoritma 27
Sl. 3.4. Osnovni princip Sigma metode [21] 31
Sl. 3.5. Kvaliteta Pareto fronte 34
Sl. 3.6. Grafički prikaz odnosa dominiranosti kod minimuma za rješenja i skupove rješenja 36
Sl. 3.7. Ilustracija indikatora raširenost Pareto fronte 38
Sl. 3.8. Ilustracija indikatora vanjskog promjera Pareto fronte 38
Sl. 3.9. Ilustracija indikatora srednje udaljenosti od NADIR-a 39
Sl. 3.10. Ilustracija indikatora najveće greške Pareto fronte 41
Sl. 3.11. Ilustracija Epsilon indikatora 42
Sl. 4.1. Osnovni koraci surogat modeliranja 44
Sl. 4.2. Puni faktorski plan uzoraka za tri faktora na tri nivoa 46
Sl. 4.3. Centralni kompozitni plan uzoraka 47
Sl. 4.4. Box – Behnken plan uzoraka 47
Sl. 4.5. D- optimalni plan uzoraka s tri faktora za puni kvadratni polinomski surogat model 50
Sl. 4.6. Usporedba klasičnog D- optimalnog plan pokusa s dva popunjavajća LHS plana pokusa 50
Sl. 4.7. LHTPA- plan uzoraka s tri faktora za puni kvadratni polinomski surogat model 51
Sl. 4.8. Modeliranje slučajnog procesa krigingom [29] 54
Sl. 4.9. Neke od najčešće korištenih baznih funkcija za RBF surogat modele [35] 58
Sl. 5.1.Primjeri složenih konstrukcija s područja zrakoplovstva i brodogradnje 64
Sl. 5.2. Hijerarhijski složeni sustav s tri razine hijerarhije 66
Sl. 5.3. Spreg (coupling) odziva sustava 66
Sl. 5.4. Zajednički deskriptori sustava 67
Sl. 5.5.Primjeri dekompozicije problema projektiranja krila zrakoplova 68
Sl. 5.6.Spregnutost odziva dvaju sustava 69
Sl. 5.7. Dekomponiranje sustava uvođenjem ograničenja konzistentnosti 69
Sl. 5.8. Dekomponiranje hijerarhijskih sustava jakom formulacijom (ograničenja jednakosti) 70
Sl. 5.9. Dekomponiranje hijerarhijskih sustava slabom formulacijom (relaksacija ograničenja) 70
Sl. 5.10. Dekomponiranje nehijerarhijskih sustava jakom formulacijom (ograničenja jednakosti) 71
Sl. 5.11. Dekomponiranje nehijerarhijskih sustava slabom formulacijom (relaksacija ograničenja) 71
Sl. 5.12. Mogući pristupi za dekompoziciju složenih sustava 72
Sl. 5.13. Utjecaj mjera spregnutosti sustava na primjenu dekompozicijsko-koordinacijskih tehnika 73
Sl. 5.14. Odjeljive odluke pri koordinaciji 75
Sl. 5.15. Neodjeljive odluke pri koordinaciji 75

Sl. 5.16. Neodjeljive i spregnute odluke pri koordinaciji 75
Sl. 5.17. MDF arhitektura [60] 80
Sl. 5.18 IDF arhitektura [60] 81
Sl. 5.19 AAO arhitektura [60] 83
Sl. 5.20. Horizontalni i vertikalni razvoj optimizacije bazirane na dekompoziciji 90
Sl. 6.1. Funkcionalne cjeline nekih složenih tankostjenih konstrukcija 92
Sl. 6.2. Sprega disciplina funkcionalnih komponenti preko zavisnih deskriptora 94
Sl. 6.3. Sprega disciplina funkcionalnih komponenti preko operatora preslikavanja odziva 94
Sl. 6.4. Dekompozicija funkcionalne komponente strukture hijerarhijski odozgo 95
Sl. 6.5. Dekompozicija funkcionalne komponente strukture hijerarhijski odozgo 96
Sl. 6.6. Dijagram toka generiranja Pareto rješenja 99
Sl. 6.7. Uključivanje promjene distribucije odziva po podstrukturi 102
Sl. 6.8. Nekorigirane, momentom inercije normalizirane vrijednosti parametra koje opisuje jaki nosač palube pontona 105
Sl. 6.9. Nekorigirana normalizacija složenog deskriptora grednog nosača 106
Sl. 6.10. Korigirana normalizacija složenog deskriptora grednog nosača 107
Sl. 6.11. Prikaz sučelja OCTOPUS Designera za provjeru utjecajnosti parametara korištenjem polinomskih surogat modela 108
Sl. 6.12. Efikasna rješenja generirana SLP-om za različite omjere važnosti ciljeva [78] 111
Sl. 6.13. Pareto fronta generirana MOPSO algoritmom [78] 111
Sl. 7.1. Komponente OCTOPUS DESIGNER 114
Sl. 7.2. Osnovne klase i koncepti sučelja okruženja za potporu pri projektiranju 119
Sl. 7.3. Osnovni koncepti sučelja generičkog modela za analizu 121
Sl. 7.4. Generička implementacija osnovnih koncepata sučelja generičkog modela za analizu 123
Sl. 7.5. Klasa zadužena za dinamičko povezivanje korištenih modula za analizu 124
Sl. 7.6. Dijagram klasa zaduženih za povezivanje generičkog modela za analizu s optimizacijskim komponentama 125
Sl. 7.7. Klase za definiciju optimizacijskih zadataka (`OptimizationJob`) 126
Sl. 7.8. Klase za detaljno i fleksibilno definiranje optimizacijske sekvence (`OptimizationFlow`) 128
Sl. 7.9. Klase za definiciju optimizacijskih zadataka (`OptimizationJob`) 130
Sl. 7.10. Klasa `ComplexSystemOptimizationModel` 132
Sl. 7.11. Dijagram klasa dekompozicijskog modela 133
Sl. 7.12. Dijagram klasa koordinacijskog modela 136
Sl. 7.13. Dijagram klasa za definiranje komponenti optimizacijskog problema 139
Sl. 7.14. Dijagram klasa za definiranje poslova baziranih na planiranju pokusa 141
Sl. 7.15. Dijagram klasa za definiranje surogat modelera 143
Sl. 7.16. Dijagram klasa za definiranje općenitog polinomskog surogat modela 145
Sl. 7.17. Organizacija memorije standardne Win32 aplikacije i .NET aplikacije [93] 147
Sl. 7.18. Arhitektura DeCluster sustava 148
Sl. 7.19. Dijagram klase za upravljanje rješavanja skupine ponavljajućih zadataka (TaskManager) 150
Sl. 7.20. Dijagram klasa za definiranje izvođenja DOE simulacija pomoću DeCluster sustava 151
Sl. 7.21. Dijagram klasa rješavača pojedinog zadataka (TaskWorker) 152
Sl. 7.22. Prikaz sučelja DeCluster sustava pri rješavanju pokusa potrebnih za generiranje surogat modela odziva 153

Sl. 8.1. Jednostavni ponton 154
Sl. 8.2. Prikaz sučelja OCTOPUS Designera za validaciju surogat modela 159
Sl. 8.3. NRMSE greška testiranih surogata za normalno naprezanje σ_x na palubi i dnu 160
Sl. 8.4. RMSE greška testiranih surogata za normalno naprezanje σ_x na palubi i dnu 160
Sl. 8.5. NRMSE greška testiranih surogata za normalno naprezanje σ_y na palubi i dnu 161
Sl. 8.6. RMSE greška testiranih surogata za normalno naprezanje σ_y na palubi i dnu 161
Sl. 8.7. NRMSE greška testiranih surogata za moment savijanja uzdužnih jakih nosača Mz na palubi i dnu 162
Sl. 8.8. RMSE greška testiranih surogata za moment savijanja uzdužnih jakih nosača Mz na palubi i dnu 162
Sl. 8.9. NRMSE greška testiranih surogata za moment savijanja poprečnih jakih nosača Mx na palubi i dnu 163
Sl. 8.10. RMSE greška testiranih surogata za moment savijanja poprečnih jakih nosača Mx na palubi i dnu 163
Sl. 8.11. Sekvenca rješavanja AiO formulacije bez surogata 165
Sl. 8.12. Ostvarene vrijednosti atributa kroz iteracije za AiO bez surogata (bilježeno u A_1) 165
Sl. 8.13. Sekvenca rješavanja dekomponiranog problema bez surogata 166
Sl. 8.14. Ostvarena rješenja dekomponiranog problema bez surogata (Bilježeno u A_1) 166
Sl. 8.15. Sekvenca rješavanja AiO problema sa surogat modelom strukturnog odziva 167
Sl. 8.16.Ostvarena rješenja AiO problema sa surogat modelima odziva (Bilježeno u A_MM_100) 167
Sl. 8.17. Sekvenca rješavanja dekomponiranog problema sa surogat modelima odziva 168
Sl. 8.18. Ostvarena rješenja dekomponiranog problema sa surogatima odziva (Bilježeno u A_1) 168
Sl. 8.19. Sekvenca rješavanja problema koji uključuje lokalno kalibriranje surogat modela odziva 170
Sl. 8.20. Ostvarena rješenja dijela sekvence koji radi s globalnim surogatom (bilježeno u A_1) 170
Sl. 8.21. Ostvarene vrijednosti atributa u dijelu sekvence koji radi s lokalno kalibriranim surogatima odziva (bilježeno u A_303) 170
Sl. 8.22. EU projekt IMPROVE RoPax 171
Sl. 8.23. Metodologija projektiranja strukture višepalubnih brodova [76] 172
Sl. 8.24. Opis integracije sustava MAESTRO s DeMak (OCTOPUS Designer) okruženjem 173
Sl. 8.25. Model strukture RoPax broda s označenom kontrolnom konstrukcijom za optimizaciju 175
Sl. 8.26. Shematski prikaz procedure određivanja amplitude projektnog vala 177
Sl. 8.27. Postignuti (statički+valni) vertikalni moment savijanja i usporedba sa Pravilima za LC2 179
Sl. 8.28. Postignuta (statička+valna) vertikalna poprečna sila i usporedba sa Pravilima za LC2 179
Sl. 8.29. Distribucija vanjskog tlaka (statički +dinamički) duž modela za LC2 179
Sl. 8.30. Definirana svojstva ljuskastih konačnih elemenata na kontrolnoj optimizacijskoj konstrukciji 181
Sl. 8.31. Definirana svojstva grednih konačnih elementa na kontrolnoj optimizacijskoj konstrukciji 181

Sl. 8.32. Osnovne podstrukture glavnog na poziciji glavnog rebra RoPax- 182
Sl. 8.33. Definicija dekompozicijskog problema RoPax-a s virtualnim globalnim sustavom 185
Sl. 8.34. Ostvarene vrijednosti atributa RoPax-a kroz iteracije (bilježeno u A_1) 185
Sl. 8.35. Sekvenca rješavanja problema RoPax-a sa surogat modelima odziva 186
Sl. 8.36. Ostvarene vrijednosti atributa RoPax-a kroz iteracije 187
Sl. 8.37. Ilustrativni prikaz nadzvučnog zrakoplova [99] 188
Sl. 8.38. Tok podataka modela nadzvučnog zrakoplova 189
Sl. 8.39. Tri segmenta modela krila 191
Sl. 8.40. Osnovne dimenzije torzione kutije krila 192
Sl. 8.41. Prikaz sila na zrakoplov u ravnotežnom horizontalnom letu 192
Sl. 8.42. Definicija sekvence rješavanja za SBJ MDF problem u OCTOPUS Designeru 197
Sl. 8.43. Prikaz početnih 100 rješenja zajedno sa završnih 99 MDF rješenja 197
Sl. 8.44. Prikaz 99 konvergiranih rješenja MDF formulacije 198
Sl. 8.45. Definicija sekvence rješavanja za SBJ IDF problem u OCTOPUS Designeru 199
Sl. 8.46. Prikaz početnih 100 rješenja zajedno sa završnih 100 IDF rješenja 199
Sl. 8.47. Prikaz 98 konvergiranih rješenja IDF formulacije 200
Sl. 8.48. Definicija sekvence rješavanja za – SBJ-R-2L-ATC problem u OCTOPUS Designeru 201
Sl. 8.49. Definicija spregnutih podataka između podsustava u OCTOPUS Designeru 202
Sl. 8.50. Konvergencija globalnog sustava (SBJ-PERF) 203
Sl. 8.51. Konvergencija AERO podsustava 203
Sl. 8.52. Konvergencija STRUCT podsustava 204
Sl. 8.53. Konvergencija PROP sustava 204
Sl. 8.54. Definicija sekvence rješavanja za SBJ-R-3L-ATC problem u OCTOPUS Designeru 205
Sl. 8.55. Konvergencija globalnog sustava (SBJ-PERF) 206
Sl. 8.56. Konvergencija AERO podsustava 206
Sl. 8.57. Konvergencija STRUCT podsustava 206
Sl. 8.58. Konvergencija PROP sustava 206
Sl. 8.59. Konvergencija PROP sustava 207
Sl. 8.60. Konvergencija WingRoot sustava 207
Sl. 8.61. Konvergencija WingMidd sustava 207
Sl. 8.62. Konvergencija WingTip sustava 207

POPIS TABLICA

Tbl. 2.1 Prikupljeni podaci o relevantnim aplikacijama optimizacije pri projektiranju složenih tankostjenih konstrukcija – objavljeno u ([12], [13]) 11
Tbl. 3.1 Odnos dominacije rješenja i skupa rješenja 36
Tbl. 4.1 Najčešće korištena ortogonalna polja 48
Tbl. 6.1 Odabrane potencije *p* za korekciju normalizacije prijenosnih funkcija deskriptora grede 107
Tbl. 7.1 Hijerarhijska struktura projektnog sustava OCTOPUS [85] 113
Tbl. 8.1 Propisane gornje i donje granice oplate pontona 155
Tbl. 8.2 Propisane gornje i donje granice grednih dijelova konstrukcije 155
Zadana ograničenja su lokalni faktori podobnosti konstrukcije definirani u Tbl. 8.3 koji se provjeravaju programskim sustavom MAESTRO, a njihov opis moguće je pronaći u [95] 156
Tbl. 8.3 Ograničenja – kriteriji podobnosti konstrukcije pontona 156
Tbl. 8.4 Atributi – mjere kvalitete konstrukcije 156
Tbl. 8.5 Legenda oznaka surogat modela 158
Tbl. 8.6 Osnovne karakteristike MAESTRO programa [5] 174
Tbl. 8.7 Slučajevi opterećenja i glavne komponentne 178
Tbl. 8.8 Definicija donjih i gornjih granica varijabli oplate (plate) i ukrepa (*stiffeners*) 183
Tbl. 8.9 Definicija donjih i gornjih granica varijabli jakih uzdužnih nosača (*longitudinal girders*) 183
Tbl. 8.10 Definicija donjih i gornjih granica varijabli jakih poprečnih nosača (*transverse frames*) 183
Zadana ograničenja su lokalni faktori podobnosti konstrukcije definirani u Tbl. 8.3 koji se provjeravaju programskim sustavom MAESTRO, a njihov opis moguće je pronaći u [95] 184
Tbl. 8.11 Ograničenja – kriteriji podobnosti konstrukcije RoPax-a 184
Tbl. 8.12 Atributi – mjere kvalitete konstrukcije RoPax-a 184
a) Ostvarene vrijednosti atributa nakon dva ciklusa optimizacije s globalnim surogat modelom 187
b) Ostvarene vrijednosti atributa nakon tri ciklusa optimizacije s lokalno umjerenim surogatom 187
Tbl. 8.13 Popis korištenih oznaka u na primjeru projektiranja nadzvučnog poslovnog zrakoplova 190
Tbl. 8.14 Zajedničke varijable 194
Tbl. 8.15 Lokalne varijable podsustava 195
Tbl. 8.16 Korišteni projektni cilj 195
Tbl. 8.17 Projektna ograničenja 196
Tbl. 8.18 Postignuti doleti konvergiranih rješenja MDF formulacije grupirani po iznosu doleta 198
Tbl. 8.19 Spregnute varijable (IDF formulacija) 198
Tbl. 8.20 Postignuti doleti konvergiranih rješenja IDF formulacije grupirani po iznosu doleta 200

1. UVOD

Porast opsega znanstvenoistraživačkog rada iz područja optimizacije i potpore pri odlučivanju uvelike je potpomognut napretkom računala, ali i zahtjevima industrije koja primjenom znanstvenoistraživačkih dostignuća želi osigurati svoj opstanak na zahtjevnom tržištu današnjice. Karakteristika složenih sustava, poput složene tankostjene konstrukcije (STK) broda ili zrakoplova, jest postojanje više ciljeva prema kojima se procjenjuje kvaliteta, što onemogućuje donošenje objektivne odluke koja je projektna varijanta najbolja te zahtijeva korištenje neke od tehnika višekriterijskog odlučivanja. Projektiranje STK spada u skupinu visoko dimenzionalnih (*large scale*) problema s nekoliko projektnih ciljeva, više stotina projektnih varijabli i više desetaka tisuća projektnih ograničenja.

Dodatna otežavajuća okolnosti pri postizanju globalnog optimalnog rješenja cijelog sustava je činjenica da su projektni timovi podijeljeni prema disciplinama npr. u projektiranju zrakoplova nalazimo timove za aerodinamiku, strukturu, motor, buku itd. Nuspojava specijalizacije za određenu disciplinu je projektanti zaduženi za određenu disciplinu često nisu upoznati s projektnim procesom ostalih disciplina. Ti timovi često koriste visokospecijalizirane alate za modeliranje i analizu koji su nerijetko i vlastite izrade. To nameće zahtjev programskom alatu za sintezu da može uključiti više različitih alata za analizu istovremeno.

U *uvodnom poglavlju* ukratko je opisan razmatrani problem i motivacija za istraživanje te je dan pregled rada po poglavljima. Slijedi postavljena hipoteza rada te cilj istraživanja.

U *drugom poglavlju* dan je prikaz osnovnih karakteristika projektiranja složenih tankostjenih konstrukcija, poput zrakoplova i broda, u kontekstu zahtijevane sigurnosti koja je regulirana mjerodavnim tijelima. Opisane su glavne karakteristike alata za analizu tankostjene konstrukcije u konceptualnoj i preliminarnoj fazi projektiranja prema provedenom istraživanju postojećih radova s tog područja.

U *trećem poglavlju* dana je generalna formulacija optimizacijskog problema te pri istraživanju korištene optimizacijske metode sposobne rješavati probleme iz inženjerske prakse.

U *četvrtom poglavlju* dane su osnove aproksimacijskog, tj. surogat-modeliranje odziva modula za analizu čijim je korištenjem umjesto zahtjevnih modula moguće reducirati ukupno vrijeme projektiranja složenog problema.

U *petom poglavlju* dan je opis metode dekompozicije složenog problema, gdje se originalni problem dijeli na više povezanih, jednostavnijih potproblema koji se mogu optimizirati paralelno na višeprocesorskim računalima ili klasterima računala. Opisane su glavne

karakteristike složenih problema koje utječu na izbor načina dekompozicije te odabir odgovarajuće metode koordinacije.

U *šestom poglavlju*, na osnovi analize karakteristika pojedinih metoda sinteze te karakteristika projektiranja složenih tankostjenih konstrukcija opisana je predložena primjena surogat modeliranja i dekompozicije složenog problema na potprobleme pri projektiranju složenih tankostjenih konstrukcija. Istaknuta je važnost sagledavanja problema projektiranja tankostjene konstrukcije sa stajališta količine i jačine spregnutosti pojedinih dijelova strukture te je predložena alternativna metoda za njihovo mjerenje.

U *sedmom poglavlju* opisana je arhitektura interaktivnog okruženja za potporu pri projektiranju OCTOPUS Designer, koje je prošireno za metode sinteze odabrane kroz poglavlja tri, četiri i pet, čime je omogućeno njihovo intuitivno korištenje tijekom procesa projektiranja. Također je opisana arhitektura i implementacija novorazvijene komponente koja omogućuje paralelno izvršavanja određenih sekvenci naredbi korištenjem osnovnih biblioteka .NET Framework-a 2.0.

Osmo poglavlje obuhvatilo je testiranje implementiranih projektnih modula te predloženih unaprijeđenija projektiranja složene tankostjene konstrukcije na primjerima koji uključuju konceptualno projektiranje nadzvučnog poslovnog zrakoplova te konceptualno projektiranje broda za prijevoz robe i putnika.

U *devetom poglavlju* dani su opći zaključci te prijedlozi za daljnje istraživanje koja će se provesti u sklopu odobrenog znanstvenog projekta (MZOŠ).

U *desetom poglavlju* kratko su sumirani znanstveni doprinosi i spoznaje ostvarene kroz rad.

HIPOTEZA RADA

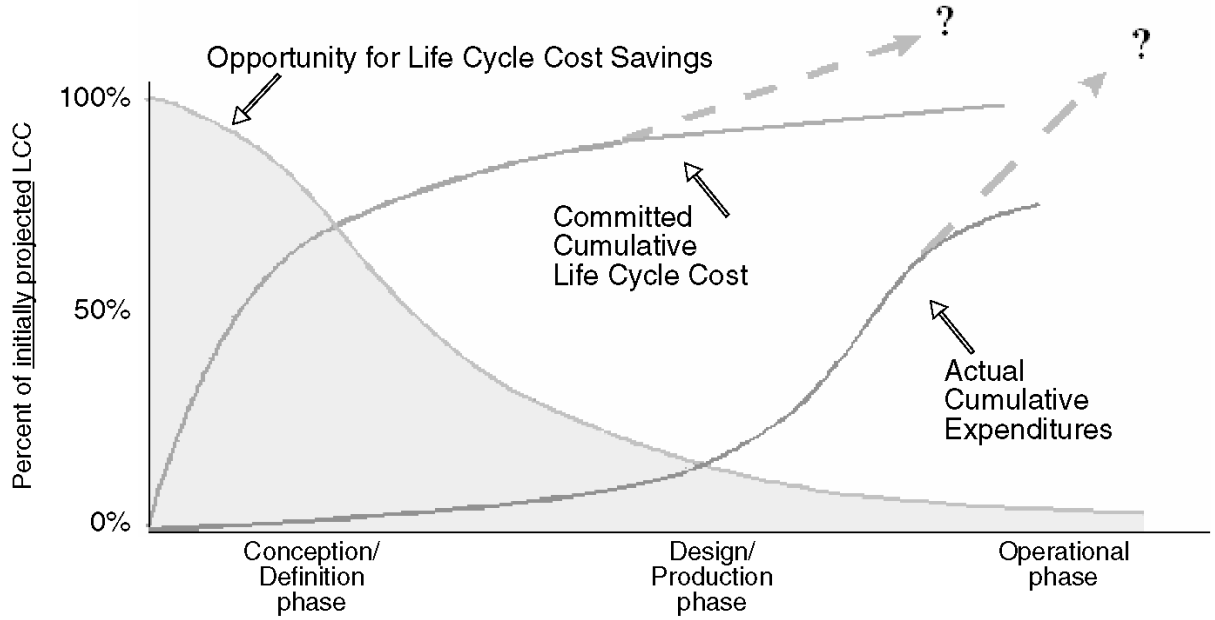
Cilj istraživanja je unapređenje višekriterijskog projektiranja složene tankostjene konstrukcije, kao visoko dimenzionalnog projektnog problema, primjenom i prilagodbom metoda dekompozicije, surogat-modeliranja i hibridnih optimizacijskih metoda.

Hipoteze rada:

- unaprijeđena procedura višekriterijskog projektiranja složene tankostjene konstrukcije omogućit će iznalaženje kvalitetnijih projektnih rješenja unutar ograničenog vremenskog okvira
- uporabom razvijenog računalnog okruženja za višekriterijsko projektiranje omogućit će se nova razina interakcije projektanta s projektnim problemom, cjelovitiji uvid u sve faze projektnog procesa te efikasniji odabir preferiranih projektnih rješenja.

2. OSNOVNE KARAKTERISTIKE PROJEKTIRANJA SLOŽENIH TANKOSTJENIH KONSTRUKCIJA

Složene tankostjene konstrukcije poput zrakoplova i broda traže primjenu novih tehnologija kako u gradnji i opremanju tako i u procesu projektiranja. Da bi bili konkurentni na svjetskom tržištu proizvođači/projektni uredi moraju moći brzo generirati različite projektne varijante i proizvesti kvalitetne projekte u ranoj projektnoj fazi. Pri tome je potrebno imati na umu da projektanti u ranim fazama projektiranja raspolažu vrlo ograničenom količinom deskriptora (parametara) projekta koji se tijekom procesa projektiranja povećava. Istovremeno, upravo u ranim fazama projektiranja donosi se najveći broj odluka vezanih za ukupne, cjeloživotne troškove proizvoda (LCC - *life cycle cost*), vidi (Sl. 2.1).



Sl. 2.1. Mogućnost utjecaja na ukupne troškove zrakoplova kroz njegove životne faze [1]

Kao i za druge tipove proizvoda, projektiranje složenih tankostjenih konstrukcija može se podijeliti na faze prikazane na Sl. 2.2. Vrlo je bitno prikladnom metodologijom i alatima omogućiti projektantima / projektnim uredima što bolji uvid utjecaja deskriptora na ključne mjere kvalitete kroz sve faze projektiranja, a zbog osobitog utjecaja na ukupne troškove, naročito u konceptualnoj fazi.

Sl. 2.2. Faze projektiranja složenih tankostjenih konstrukcija

2.1. Zahtjevi pri projektiranju složenih tankostjenih konstrukcija

Ekonomski i ekološki aspekti gradnje i uporabe složenih proizvoda za prijevoz ljudi i roba (npr. brod, zrakoplov, most, itd.) uvjetuju sve izraženije zahtjeve za smanjenjem potrošnje materijala i energije, tj. za iznalaženjem racionalnijih i ekonomičnijih projektnih varijanti, koje stoga karakteriziraju sve manje rezerve sigurnosti. Istovremeno, u analitičkim procedurama obuhvaćenim modernim pristupima višekriterijskom projektiranju složenih tankostjenih konstrukcija znatno su više zastupljene metode zasnovane na numeričkom modeliranju i simulacijama nego eksperimentalna ispitivanja modela u naravnoj i/ili reduciranoj veličini. Uzročnici takve situacije, velike dimenzije te masa i cijena, uvjetuju nezavidnu poziciju zrakoplovne i/ili brodograđevne industrije u odnosu na automobilsku i slične, kod kojih je u svrhu verifikacije sigurnosti (novih) konstrukcija moguće provesti ekstenzivne testove prototipova u naravnoj veličini na razne tipove mogućih oštećenja. Iako se za svaki novoizgrađeni zrakoplov/brod provodi pokusni let/plovidba, meritornu procjenu karakteristične razine sigurnosti konstrukcije niti tada nije moguće provesti, s obzirom da se pokusni let/plovidba ne provodi u ekstremnim uvjetima djelovanja okoliša. Zahtjevi pri projektiranju složenih tankostjenih konstrukcija mogu se podijeliti u dvije osnovne grupe: sigurnosne i ekonomske.

U [2] nalazi se pregled relevantne sigurnosne regulative koja je definirana kroz nekoliko razina organizacija i ustanova. Na najvišoj razini regulative nalaze se Međunarodna civilna zrakoplovna organizacija (*ICAO- International Civil Aviation Organization*) te Međunarodna pomorska organizacija (*IMO - International Maritime Organization*). Bitno je spomenuti i Međunarodna organizacija za standardizaciju (*ISO - International Standardization Organization*) Aktualna obvezujuća pravila i standarde propisuju regulatorne agencije odnosno klasifikacijska društva u brodogradnji.

U zrakoplovstvu su najznačajnije regulatorna tijela Europska zrakoplovna sigurnosna agencija (EASA – *European Aviation Safety Agency*) te Federalna zrakoplovna administracija (FAA - *Federal Aviation Administration*) za teritorij Sjedinjenih Američkih Država. EASA putem Specifikacija certifikacije (CS- *Certification Specification*), a FAA putem Federalnih zrakoplovnih regulativa (FAR – *Federal Aviation Regulation*), reguliraju minimalne zahtjeve za plovidbenost pojedine kategorije zrakoplova.

U brodogradnji su klasifikacijska društva koja direktno propisuju regulativu udružena kroz Međunarodnu asocijaciju klasifikacijskih društava (IACS - *International Association of Classification Societies*). Zahtjevi koje se moraju zadovoljiti podijeljeni su prema tipovima

brodova i to najčešće prema tipu tereta koji se prevozi te duljini ili istisnini broda. Za neke tipove brodova kroz zadnje desetljeće osnovana su zajednička regulatorna pravila koje je prihvatilo više regulatornih agencija, npr. CSR (*Common Structural Rules*) za tankere i brodove za prijevoz rasutog tereta (bulkers).

Regulativa i u zrakoplovstvu i u brodogradnji obuhvaća proces projektiranja kao i eksploataciju proizvoda. Propisuju se opterećenja, kriteriji podobnosti, zaštita konstrukcije od djelovanja okoline kao i razni aspekti vezani za eksploataciju poput perioda održavanja te dopuštenih utjecaja na okolinu poput onečišćenja nastalih izgaranjem goriva pogonskih sustava, buke pri slijetanju zrakoplova ili ispuštanja balastnih voda brodova.

Prema Kitarović [2], pristupi za osiguranje sigurnosti tankostjene zasnovani su na razmatranju graničnih stanja (*limit states*), tj. situacija u kojoj razmatrana konstrukcija, neki njen dio ili spoj više ne može vršiti pridruženu funkciju. Evaluaciju sigurnosti konstrukcije potrebno je pri tome provesti za nekoliko različitih relevantnih aspekata:

- granično stanje nosivosti (*Ultimate limit state*),
- granično stanje uporabljivosti (*Serviceability limit state*),
- granično stanje zamora (*Fatigue limit state*),
- granično stanje akcidentnosti (*Accidental limit state*).

Tradicionalni pristup, tj. pristup dopuštenog naprezanja (*allowable stress*) zasniva se na usporedbi izračunate i dopuštene razine odziva konstrukcije (na projektna opterećenja), pri čemu se izračunati odziv nastoji svesti ispod dopuštene razine - određene na temelju uspješnih iskustava iz prošlosti primjenom (klasičnih) faktora sigurnosti. S obzirom da na taj način nije moguće odrediti stvarnu karakterističnu rezervu sigurnosti razmatrane konstrukcije, u novije vrijeme prevladavaju pristupi zasnovani na razmatranju graničnih stanja, koji omogućavaju eksplicitno, tj. direktno razmatranje relevantnih načina oštećenja konstrukcije istovremeno opterećene različitim tipovima opterećenja, čije razine mogu varirati od uobičajene do ekstremne (ovisno o ambijentalnim i uporabnim okolnostima). Kako bi se kod realnih nosivih konstrukcija minimizirala mogućnost pojave oštećenja tokom eksploatacije, pri projektiranju konstrukcije treba racionalno uzeti u obzir kako stohastičku narav svih relevantnih aspekata njene gradnje i uporabe, tako i razinu kvalitete analitičkih i sintetskih modela koji se pri tome koriste. U tom smislu, različite nesigurnosti uslijed prirodnih varijabilnosti, nesavršenosti različitih aspekata procesa gradnje i (ne)točnosti korištenih metoda za opis projektnih opterećenja, njihovih posljedica te izdržljivosti konstrukcije, mogu se korištenjem metoda analize pouzdanosti sažeto izraziti kroz parcijalne faktore sigurnosti.

Ekonomski zahtjevi bazirani su na trenutnom stanju tržišta te projekcijama tržišnih kretanja. Postavljanjem zahtjeva vezanih za kako nabavnu cijenu a tako i operativne troškove naručitelj osigurava konkurentnost na tržištu. Pri tome se sve više kao ključni zahtjev, naročito u zrakoplovstvu, koriste cjeloživotni troškovi LCC, a ne samo proizvodna odnosno nabavna cijena. U ekonomske zahtjeve na određeni način ulaze i performanse jer osiguravaju izvršavanje misija za koje je proizvod namijenjen, odnosno osiguravaju pokrivanje određenog segmenta tržišta.

2.2. Mjere kvalitete pri projektiranju složenih tankostjenih konstrukcija

Najčešće korištene mjere kvalitete proizvoda pri projektiranju vezane su za performanse kao i za ekonomske aspekte. Ekonomske mjere kvalitete u današnje vrijeme više su vezane u direktne operativne troškove i cjeloživotne troškove nego za proizvodnu / nabavnu cijenu. Isto tako može se primijetiti značajan trend, naročito u zrakoplovstvu, uračunavanja troškova kvarova odnosno gubitaka funkcionalnosti pojedinih komponenti proizvoda, uključujući i dijelove nosive konstrukcije, kroz pouzdanost odnosno rizik. Pri tome rizik ne uključuje samo troškove popravaka kvarova, već i gubitke vezane za neplanirano otkazivanje dogovorenih obaveza zbog vremena popravka, a isto tako i mnogo značajnije gubitke vezane za isplatu odgovarajućih naknada u slučaju onečišćenja okoliša te stradavanja putnika ili posade zrakoplova / broda.

Korištenje odgovarajuće mjere kvalitete u određenoj fazi projektiranja, osim o preferencama naručitelja ili proizvođača ovisi i o mogućnostima korištenog matematičkog modela. To je jedan od osnovnih razloga zašto je vrlo raširena primjena mase složene konstrukcije kao jedne od glavnih mjera kvalitete u konceptualnoj i preliminarnoj fazi projektiranja. Korištenje tako jednostavnog kriterija pri projektiranju je u potpunosti opravdana jer masa tankostjene konstrukcije ima veliki udio ne samo u proizvodnoj cijeni složene tankostjene konstrukcije, već i u operativnim troškovima gdje smanjenje mase znači direktno smanjenje potrošnje goriva ili povećanje korisnog tereta.

2.3. Osnovne karakteristike matematičkih modela analize u konceptualnom i preliminarnom projektiranju

Kako bi se osiguralo zadovoljavanje propisanih zahtjeva te poboljšanje značajnih mjera kvalitete, nužno je koristiti matematičke modele čija točnost je primjerena trenutnoj fazi projektiranja. Uz prethodno navedeno, važno je da omogući i fleksibilnu promjenu deskriptora kako bi se olakšalo generiranje projektnih varijanti odnosno pretraga projektnog prostora. Korištenje matematičkih modela visoke točnosti u ranim fazama projektiranja najčešće

nepotrebno usporava tu fazu projektiranja ne samo zbog toga većih računalnih zahtjeva, već i zbog najčešće značajno veće količine vremena potrebne za generiranje takvog modela. Osim toga matematički modeli više točnosti su najčešće i vrlo nefleksibilni glede promjene deskriptora zbog razine detalja koji se unose.

U ranim fazama projektiranja tankostjene konstrukcije najčešće se koriste modeli koji određenom metodom računaju linearni strukturni odziv konstrukcije (pomake i naprezanja), dok se podobnost provjerava unošenjem izračunatog odziva u nelinearne, semi-analitičke formule kojima se provjeravaju razni strukturni zahtjevi vezani za čvrstoću (različiti kriteriji popuštanja i izvijanja), krutost, zamor i vibracije.

Za proračun strukturnog odziva u najranijim fazama projektiranja najčešće se koriste jednostavne analitičke metode nauke o čvrstoći kojima se struktura idealizira odrađenim matematičkim modelom (štap, greda, itd.). Proračun metodom konačnih elemenata (MKE) nudi mnogo veću fleksibilnost jer se dijelovi strukture modeliraju s odgovarajućim konačnim elementima te se njihovom kombinacijom mogu modelirati i najsloženiji dijelovi strukture. Korištenje određenih tipova konačnih elemenata također ovisi o fazi projektiranja. Tako se u konceptualnoj fazi projektiranja najčešće koriste linijski konačni elementi (štapni, gredni), dok se u preliminarnoj najčešće koriste dvodimenzijski konačni elementi (ploče, ljuske) u kombinaciji s linijskim konačnim elementima. U detaljnoj fazi projektiranja, osim navedenog, za provjeru konstrukcijskih detalja koriste se i 3D konačni elementi.

Za provjeru različitih kriterija podobnosti često se koriste izrazi koji imaju oblik:

$$D < C \quad ili \quad \frac{C}{D} > 1 \tag{2.1}$$

gdje je D (*demand*) ukupni odziv konstrukcije na projektna opterećenja općenito definiran jednadžbom (2.2), a C (*capability*) izdržljivost konstrukcije općenito definirana jednadžbom (2.3).

$$D = \gamma_0 \sum_{i=1}^{n} D_{ji}(F_{ji}, \gamma_{fi}) \tag{2.2}$$

$$C = \frac{C_j}{\gamma_m \gamma_c} \tag{2.3}$$

U jednadžbama (2.2) i (2.3) γ_0 predstavlja parcijalni faktor sigurnosti kojim se ponderira suma karakterističnih mjera odziva za razmatrano (*j*-to) granično stanje s obzirom na moguće ekonomske i društvene posljedice i/ili posebne okolnosti u slučaju kolapsa konstrukcije, pri

čemu se obzir uzima namjena konstrukcije, vrsta prevoženog tereta, interakcija razmatranog graničnog stanja s ostalima, itd. D_{ji} je karakteristična mjera odziva za *i*-ti tip opterećenja, F_{ji} karakteristična mjera *i*-tog tipa opterećenja, dok je γ_{fi} parcijalni faktor sigurnosti kojim se obuhvaćaju nesigurnosti vezano uz *i*-ti tip opterećenja. C_f je karakteristična mjera izdržljivosti s obzirom na razmatrano (*j*-to) granično stanje, γ_m je parcijalni faktor sigurnosti kojim se obuhvaćaju nesigurnosti vezano uz relevantna svojstva materijala, dok je γ_c parcijalni faktor sigurnosti kojim se obuhvaćaju nesigurnosti vezano uz čimbenike koji utječu na izdržljivost konstrukcije (npr. kvaliteta gradnje, korozija, točnost metode za određivanje izdržljivosti, itd.).

Najtočniju procjenu podobnosti konstrukcije postiže se primjenom metode konačnih elemenata na trodimenzionalnim (3D) diskretiziranim potpunim modelom cijele konstrukcije, te provođenjem geometrijski i materijalno nelinearne (NL) analize kako bi se simulirao i evaluirao kvazistatički odziv konstrukcije na ekstremna projektna opterećenja. Vrlo intenzivna dinamika razvoja računalnih platformi omogućava značajna smanjenja vremena potrebnog za izvršavanje i najzahtjevnijih numeričkih simulacija, međutim, angažman potreban za uspješno obavljanje cjelokupnog postupka NLMKE analize (naročito za pred/post-procesiranje) još uvijek zahtijeva značajnu količinu vremena, znanja i iskustva. Ove značajke onemogućavaju automatizaciju u implementaciji svih faza NLMKE analize, te je čine neadekvatnom za primjenu u konceptualnoj fazi projektiranja, što je naročito izraženo ukoliko je se želi inkorporirati unutar optimizacijske petlje koja podrazumijeva višestruka i automatska izvršavanja analize za mnogobrojne i različite varijante projekata.

2.4. Pregled istraživanja

U [3] sustavno su prikazani dodatni aspekti i problemi pri projektiranju sustava u zrakoplovstvu koji zahtijevaju korištenje metoda poput dekompozicije i surogat-modeliranja. Pod vodstvom V. Žanića od 1990. na FSB-u je u razvoju metodologija koja je ugrađena u računalno okruženje OCTOPUS Designer za višekriterijsko odlučivanje (*multi-criteria decision making* – MCDM). Ona se bazira na primjeni MADM-a (*multi-attribute* DM) inačice MCDM-a u okviru kojega je omogućeno i paralelno procesiranje simulacija po potproblemima. U radu [4] metodologija je primijenjena za projektiranje poprečne strukture SWATH broda na osnovi pouzdanosti. U [5] prikazana je metodologija projektiranja strukture primijenjena na brodove za prijevoz prirodnog plina (LNG) i roba (vozila i putnika) – RoPax, te kemikalija.

U literaturi se može pronaći mnoštvo primjena optimizacijskih algoritama na probleme s područja strojarstva, brodogradnje i zrakoplovstva. U [6] višekriterijski je optimizirana

kompozitna tlačna posuda korištenjem NSGAII višeciljnog genetskog algoritma. Isti optimizacijski algoritam korišten je za optimizaciju automobilske strukture u [7]. U [8] koriste evolucijske strategije i sekvencijalno kvadratično programiranje za rješavanje problema optimizacije kompozitne strukture *blended wing body* zrakoplova. U [9] prikazana je optimizacija sendvič konstrukcije pomičnih palubica broda, korištenjem genetskih algoritama. Optimizacija strukture broda za prijevoz vagona korištenjem hibridnog optimizacijskog algoritma (SLP + MOGA) prikazana je u [11].

U Tbl. 2.1, koja je pripremljena za potrebe ISSC – odbora te je ovdje dana u obliku kako je objavljena u ISSC, [12], [13], može se vidjeti primjena optimizacije za projektiranje pretežito tankostjenih konstrukcija. U tablicama su posebno prikazane komponente optimizacijskog problema, korišteni moduli analize te moduli sinteze.

Tbl. 2.1 Prikupljeni podaci o relevantnim aplikacijama optimizacije pri projektiranju složenih tankostjenih konstrukcija – objavljeno u ([12], [13])

REF.	PROBLEM IDENTIFICATION			ANALYSIS (MODELS M1-M4)*		SYNTHESIS (MODELS M5-M6)	
	VARIABLES $\{\mathbf{x}\}$	CONSTRAINTS $\{\mathbf{g}\}$	OBJECTIVES $\{\mathbf{a}\}$	RESPONSE (M1-M3)*	FEASIBILITY (M4)	OBJECTIVE DM (M5-M6)	SUBJECTIVE DM (M6)
(Seo et al. 2003)	Steel panel scantlings→ Naval ship	DNV(yield and buckling) and U.S. Navy strength criteria	-min. structural weight	Beam theory + grillage analysis	Analytical formulas	Search method between standard variables from database	
(Rigo 2003)	Steel panel scantlings(9 variables per stiff. panel = 278 total → midship section- LNG ship	- technological(min-max) - geometrical (linear) - structural (nonlinear-yield, buckling, deflection)	-min. global cost (including operational cost)	First order sensitivities Analytical solution of diff. equation using Fourier series	Analytical nonlinear formulas	Convex Linearization Dual Approach	
(Zanic et al. 2003, 2005)	stiff. panel characteristics - up to 300variables ROPAX : 3-D partial model Tank-car carrier: full-ship model	- technological(min-max) - geometrical (linear) - structural (nonlinear-yield, buckling, deflection)	-min. structural weight -max relative safety -min structural cost	3D FEM	-Analytical nonlinear formulae	Hybrid approach to Pareto frontier: Sequential Linear Program. (SLP) + Fractional Factorial Experiments (FFE)	Multi-criteria appr.: Saaty weighting fact. Normalisation via fuzzy functions Lp metrics
(Klanac, Kujala 2004)	Steel sandwich panel- 7 variables	technological(min-max) - geometrical (linear) - structural (nonlinear-yield, buckling, deflection)	-min. structural weight -min. production cost	Analytical solution of diff. equation + grillage analysis	Analytical nonlinear formulas	Genetic Algorithm	
(Parsons, Scott 2004)	General ship design (6 variables → main dimensions- L, B…)	Functional criteria, min-max criteria→ ratio between main dimensions	-min. transportation cost -min. lightship weight -max. annual cargo capacity	Analytical formulas	Analytical formulas	Conjugate gradient method Goal programming	Saaty weighting fact. Utility functions L_{p}-norms
(Boulougouris, Papanikolaou 2003)	General ship design (compartmentation, no. of transverse bulkheads, etc.…)	max sea states criteria compartmentation criteria	-max. survivability index -min. weight -min. probability of damage	Quasi-static approach to probabilistic damage stability	Analytical formulas Enhanced survivability criteria, vulnerability	Multi-objective Genetic Algorithms (MOGA)	Pareto frontier selection
(Meyer, Whitecomb 2004a,b)	General ship design parameters	min-max criteria→ displacement. KG, performance function	-max. performance with optimal margin selection	Analytical formulas	Analytical formulas	Hybrid approach: GA- global level. Gradient search- local level	

REF.	PROBLEM IDENTIFICATION			ANALYSIS (MODELS M1-M4)*		SYNTHESIS (MODELS M5-M6)	
	VARIABLES {**x**}	CONSTRAINTS {**g**}	OBJECTIVES {**a**}	RESPONSE (M1-M3)*	FEASIBILITY (M4)	OBJECTIVE DM (M5-M6)	SUBJECTIVE DM (M6)
(Barone, Bertorello 2004)	General ship design- TRIMARAN (1 variable→ longitudinal outrigger position)	Functional criteria, min-max criteria	-min. resistance -min. generated loads -max. stability and seakeeping characteristics	Analytical formulas + different numerical codes for seakeeping, loads, stability,..	Analytical formulas	No formal method → variation of 21 possible position	Ma-Fa-Hang attribute weight assignment L_p norms
(Peri, Campana 2003)	Ship hull form optimization→ Naval surface combatant	min-max criteria→ min. volume for Sonar equipment,..	-min. resistance -max. seakeeping characteristics	- potential flow theory + ITTC formulae - strip theory	Analytical formulas	Gradient based search method → LPτ-sequence	
(Arai et al. 2003)	Structural shapes (size of transverses, and size, position & number of struts)	minimum plate thickness, minimum depth of transverses and struts	-min. structural weight -min. stress	3D FEM, Standard wave levels, Response Surface Approximation (RSM)	Linear formulae	Quasi-Newton's method	Pareto sets expression of weight and stress
(Chen et al. 2003)	- Fleet optimization - Optimal number of 4 types of barges	Min. annual freight amounts 4 type of barges with the different max. DWT	-min. average annual cost	Regression formulas	Analytical formulas	Genetic Algorithm	
(Jacobs et. al. 2004)	ten bar truss cross section areas	stress and displacement constraints	-min. mass	FEM	Formulas	Sequential Quadratic Programming on Surrogate (SAO - PR)	None (One Objective Function)
(Venter, Sobieszezanski-Sobieski 2003)	cantilevered beam cross section geometry	stress constraint	-min. material volume	FEM (Genesis)	Analytical formulas	Particle Swarm Optimization	None (One Objective Function)
(Venter, Sobieszezanski-Sobieski 2004)	wing geometry and structural scantlings (MDO)	stress and local buckling ,	-max. range -min. material volume	Analytical for aerodyne. and FEM for structural optimization	Nonlinear formulas	Particle Swarm Optimization	Value Function (Range)
(Gantois, Morris 2004)	wing geometry and structural scantlings (MDO)	yield and buckling	-min. weight -min. cost (manuf included) -min. drag	3D FEM, CFD	Nonlinear failure criterias (ADAS),	Newton based algorithm (STARS software)	Value function (Cost)
(Venter, Sobieszezanski-Sobieski 2005)	wing structural scantlings	Von Mises stress and displacement constraints	-min. weight -min. tip rotation -minl first bend mode freq.	3D FEM	Formulas	GENESIS Software	Weighted sum of normalized objectives

REF.	PROBLEM IDENTIFICATION			ANALYSIS (MODELS M1-M4)*		SYNTHESIS (MODELS M5-M6)	
	VARIABLES {**x**}	CONSTRAINTS {**g**}	OBJECTIVES {**a**}	RESPONSE (M1-M3)*	FEASIBILITY (M4)	OBJECTIVE DM (M5-M6)	SUBJECTIVE DM (M6)
(McAllister et al. 2005)	general design variables of a racecar	racecar design constraints, deviation function (MDO)	-min. lap time of racecar -max. weight distribution -min. rear downforce	Analytical formulas	Analytical formulas	Colaborative Optimization – Physical programming	
(Jang et al. 2005)	cross-section dimension (I) of two member hub frame	max stress, buckling	-min. volume and displacement	Analytical formulas	Analytical nonlinear formulas	AAMOGA on Surrogate (Classification + Kriging model)	Weighted sum of normalized objectives
(Barski, Kruzelecki 2005)	shell geometry	buckling constraints	-min. volume of shell material	Analytical nonlinear formulas	Analytical nonlinear formulas	Simulated annealing	None (One Objective Function)
(Khajehpour, Grierson 2003)	high-rise office building structural design variables	dimensional restrictions, availability limitations, performance requirement	-min. initial capital cost -min. annual operating cost -max. annual income revenue	Analytical formulas	Analytical formulas	Multi – criteria Genetic Algorithm	Visualization + selection
(Youn, Choi, Du 2005)	car crashworthiness design variables	reliability feasibility constraints	-max. performance of multicrash modes -min. weight	Nonlinear FE simulation	Nonlinear formulas	LHS + response surface + SQP	
(Ji, 2007)	-Stiffened panel scantlings: thicknesses, spacing and type of longitudinals, (n_v=21) - Oil tanker, 76 000 DWT→ midship section	IACS JTP Rules - Hull Girder Bending and Shear Strength - Local Strenght	- min. structural weight	Beam theory + Analytical formulas	Analytical formulas	-Relative Difference Quotient Algorithm (RDQA)	
(Jastrzebski et al. 2007)	-Number of web frames, girders, longitudinals → n_v=8 -Container structure - partial model 7 parameters- defining spatial arrangement of structural elements	GL Rules - Strength constraints -Min. distance between girders	- min. structural weight - min. length of welds - min. area of structural elements→ maintenance - min. vert. bending moment	- FEM calculation - Longitudinal strenhgt calculation	Analytical formulas	-Parametric optimization analysis- only 11 variants were examined of total number of 126 00 possible variants.	Weighting factors for transformation of multi-objective into single objective

REF.	PROBLEM IDENTIFICATION			ANALYSIS (MODELS M1-M4)*		SYNTHESIS (MODELS M5-M6)	
	VARIABLES $\{\mathbf{x}\}$	CONSTRAINTS $\{\mathbf{g}\}$	OBJECTIVES $\{\mathbf{a}\}$	RESPONSE (M1-M3)*	FEASIBILITY (M4)	OBJECTIVE DM (M5-M6)	SUBJECTIVE DM (M6)
(Klanac and Jelovica, 2007)	-Longitudinal stiffened panel variables- (n_v=28) - Fast ferry→ midship section	-technological(min-max) - Geometrical (linear) - DNV Rules (yield, buckling) - 2 long. LC (crest and hollow landing)	-min. structural weight	Beam theory + Analytical formulas	Analytical formulas	-Genetic Algorithm(GA) -Scalar and vectorized problem formulation	-Pareto frontier
(Klanac et al. 2008)	Longitudinal stiffened panel variables- (n_v=94) -Chemical tanker, 40 000 DWT→ midship section	-Technological(min-max) - Geometrical (linear) - Structural (yield, buckling) -2 load cases (crest and hollow landing)	-min. total structural weight -min. duplex steel weight -max. of adequacy of deck strakes	Couple Beam method + Analytical formulas	Analytical formulas	-Genetic Algorithm(GA) -Vectorized optimization problem formulation	-Pareto frontier
(Nakamori et al., 2008)	- Number, size, spacing, angle of longitudinals in for/aft part of ship structures- n_v=21	-Technological constraint (narrow working space, angle deviation) - NK Rules(strength, distance, angle)	- min. weight of longitudinals - min. number of knuckle joints - min. welding length-cost -	Analytical formulas	Analytical formulas	-Genetic Algorithm(GA)	
(Okada et al., 2007)	- Double bottom height, double side breadth, cross deck breadth, -Large container ship 12 000 TEU→ midship section	- Long. strength (section modulus) - Hatch opening deflection -Number of containers	-min. structural weight -min. building cost -max. number of containers -penalty function	Analytical formulas	Analytical formulas	-Genetic Algorithm(GA)	-Penalty function
(Richie et al., 2006)	- Longitudinal panel scantlings (9 variables per stiff. panel, n_v= 243 -VLCC ship- midship section	- technological(min-max) - geometrical (linear) - DNV structural (yield, buckling) - 8 symmetric load cases	-min. production cost	First order sensitives Analytical solution of diff. equation using Fourier series	Analytical formulas	Convex Linearization And Dual Approach	
(Richie et al., 2006)	- Longitudinal panel scantlings (9 variables per stiff. panel, n_v= 600 -Passenger ship- midship section	- technological(min-max) - geometrical (linear) - BV structural (yield, buckling) - 5 symmetric load cases	-min. production cost (basic and advance cost module)	First order sensitives Analytical solution of diff. equation using Fourier series	Analytical formulas	Convex Linearization And Dual Approach	

REF.	PROBLEM IDENTIFICATION			ANALYSIS (MODELS M1-M4)*		SYNTHESIS (MODELS M5-M6)	
	VARIABLES {**x**}	CONSTRAINTS {**g**}	OBJECTIVES {**a**}	RESPONSE (M1-M3)*	FEASIBILITY (M4)	OBJECTIVE DM (M5-M6)	SUBJECTIVE DM (M6)
(Romanoff, Klanac, 2007)	-Thicknesses of (top face, web plate, bottom face), core height, Stiffener space (n_v=5) - Steel sandwich hoistable car deck	-DNV Guidelines for laser weld sandwich panel (strength and deflection criteria) - technological(min-max)	- min. structural deck weight	Homogenized Plate Theory	Analytical formulas	-Genetic Algorithm (GA) -Vectorized optimization problem formulation -Enumeration	-Pareto frontier
(Sobey et al., 2008)	- Stiffened FRP panel -Plate thicknesses, stiffener spacing, stiffener charachteristic, (n_v=8) -Composite boat hull structure	- LR Rules for special service craft (strength and deflection criteria) - technological(min-max)	-min. structural weight -min. building cost	-Grillage analysis for stiffeners - Third order shear deformation theory for plates	Analytical formulas	-Genetic Algorithm (GA)	-Pareto frontier
(Zanic et al., 2007a)	-Topology parameters (size of side openings, number of trans BHD, height of long. BHD, etc) -n_p=8. - Stiffened panel scantlings (plate thicknesses, stiffener spacing, stiffener and transverse frame characteristics)- n_v=110 - Passenger ship-midship section	- technological(min-max) - geometrical (linear) - structural (nonlinear-yield, buckling, deflection) - 2 symmetric load cases	Investigation of topology arrangement with min. stresses variation in upper decks. Scantling optimization: -min. structural weight - min. production cost - max. hull girder ultimate moment - max. structural safety	-Full ship FEM for deck efficiency coefficient -FEM- primary and secondary stresses Analytical formulas for tertiary stresses	Analytical formulas	-Toplogy optimization Taguchi approach -Scantling optimization: Hybrid approach: Sequential Linear Program. (SLP) + Fractional Factorial Experiments (FFE) + Genetic Algorithm (GA)	-Pareto frontier
(Zanic et al., 2007b)	-Stiffened panel scantlings (plate thicknesses, stiffener spacing, stiffener and transverse frame characteristics)- n_v=79 - Wagon carrier - midship section	- technological(min-max) - geometrical (linear) - structural (nonlinear-yield, buckling, deflection) - 3 load cases according Russian Register	- min. structural weight - max. structural safety	FEM- primary and secondary stresses Analytical formulas for tertiary stresses	Analytical formulas	Hybrid approach: Fractional Factorial Experiments (FFE)+ Multiobjective Genetic Algorithm (MOGA)	-Pareto frontier

REF.	PROBLEM IDENTIFICATION			ANALYSIS (MODELS M1-M4)*		SYNTHESIS (MODELS M5-M6)	
	VARIABLES {**x**}	CONSTRAINTS {**g**}	OBJECTIVES {**a**}	RESPONSE (M1-M3)*	FEASIBILITY (M4)	OBJECTIVE DM (M5-M6)	SUBJECTIVE DM (M6)
Jarmai et al., (2006)	- Stiffened and unstiffened shell scantlings	- Structural Constraints (Buckling, yield) - Manufacturing constraints	- min shell production cost	Analytical formulas	Analytical formulas	- Leap-frog optimization (LFOPC) - Dynamic-Q - ETOPC algorithm - (PSO)	
Pelletier. and Vel (2006)	- Pressure vessel Laminate variables nv= 30 (for each laminae: fiber volume fractions, fiber orientations, and thicknesses)	- Structural Constraints (Tsai–Wu, stiffenes) - technological (min-max)	- max failure pressure - max hoop rigidity -min areal mass density	Analytical formulas	Analytical formulas	Modified NSGA II	-Pareto frontier
Li et al. (2007)	Areas of 72-bar spatial truss structure – nv=16	-Structural (stress and displacements limits) - technological (min-max)	- min weight	Finite element method (FEM)	From FEM	-Particle swarm opt. (PSO) -Passive congregation PSO (PSOPC) -heuristic PSO (HPSO)	
Hansen and Horst (2008)	Blended wing body topology (n = 6), geometry (n = 4) and scantling variables (n=110)	- technological (min-max) -Structural constraints (bifurcation buckling, Von Mieses Stress for metalic, Tsai Hill for composite)	- min weight	FEM – MSC.Nastran	-FEM – Bifurcation Buckling -Analytical formulas	-Evolution Strategy (topo/geo opt) -NASTRAN Sol200 (SQP) (scantling opt)	
Kong et al (2006)	Containership deck thicknesses (nv = 64)	- technological (min-max) - vibrations (free and forced)	min forced vibration response (rms value)	FEM – MSC.Nastran	FEM – MSC.Nastran	OPTSHIP (GA and R tabu)	
Sinha and Kaushik (2007)	Automotive structure thicknesses and material (nv=9)	- technological (min-max) -Structural constraints (reliablity based force and deflection)	- min. structural weight - min door velocity	-Nonlinear FEM LS-DYNA -Response surface model	- Analytical Formulas	- FORM -Generic Design Optimization Toolkit GDOT (NSGA II)	Pareto Frontier
Degertekin (2007)	Four-storey 84-member space frame areas (nv =10)	- technological (min-max) -Structural (stress and displacements limits)	- min weight	- Nonlinear Analytical Formulas	- Nonlinear Analytical Formulas	- Simulated Aneeling - Genetic Algorithms	
Filomeno (2008)	2D wing geometrical and size variables (nv ~ 50+ fluid variables (angle of attack and velocity)	- tehnological (min-max) -Structural (stress and displacemants limits - Lift and drag coefficient	- min weight	-FLUENT CFD Navier Stokes and Laplace - Response surface	-Analytical Timoshenko beam	Sequential quadratic programming (SQP) MDO (All-in-one)	

3. OSNOVE OPTIMIZACIJKOG PROBLEMA I OPTIMIZACIJSKE METODE

Projektiranje složenih tankostjenih konstrukcija predstavlja složen, nelinearan problem, koji je teško izraziti analitički, te je zbog toga većina optimizacijskih metoda numeričko-iterativnog tipa te zahtijeva uporabu računala. Optimizacijski postupak [10] uvodi sistematsku metodologiju za određivanje vrijednosti projektnih varijabli koji maksimiziraju ili minimiziraju zadanu mjeru kvalitete (projekta) i istovremeno zadovoljavaju projektna ograničenja. Projektni problem najčešće se definira izrazom:

$$x \in R^n : \begin{array}{l} \min \ f(x_i) \\ h_j(x_i) = 0 \quad , j = 1,...,n_h \, , \\ g_k(x_i) \geq 0 \quad , k = 1,...,n_g , \\ x_{il} \leq x_i \leq x_{iu} . \end{array} \tag{3.1}$$

gdje su:

x – projektne varijable

f – funkcija cilja

h – ograničenja jednakosti

g – ograničenja nejednakosti

U idućem potpoglavlju bit će detaljnije prikazana formulacija optimizacijskog problema na način kako je izvedena u sustavu OCTOPUS Designer te osnovni koncepti vezani za višeciljnu optimizaciju. Nakon toga slijedit će opis optimizacijskih metoda korištenih u ovom radu. Na kraju ovog poglavlja nalazi se opis indikatora kvalitete Pareto fronte.

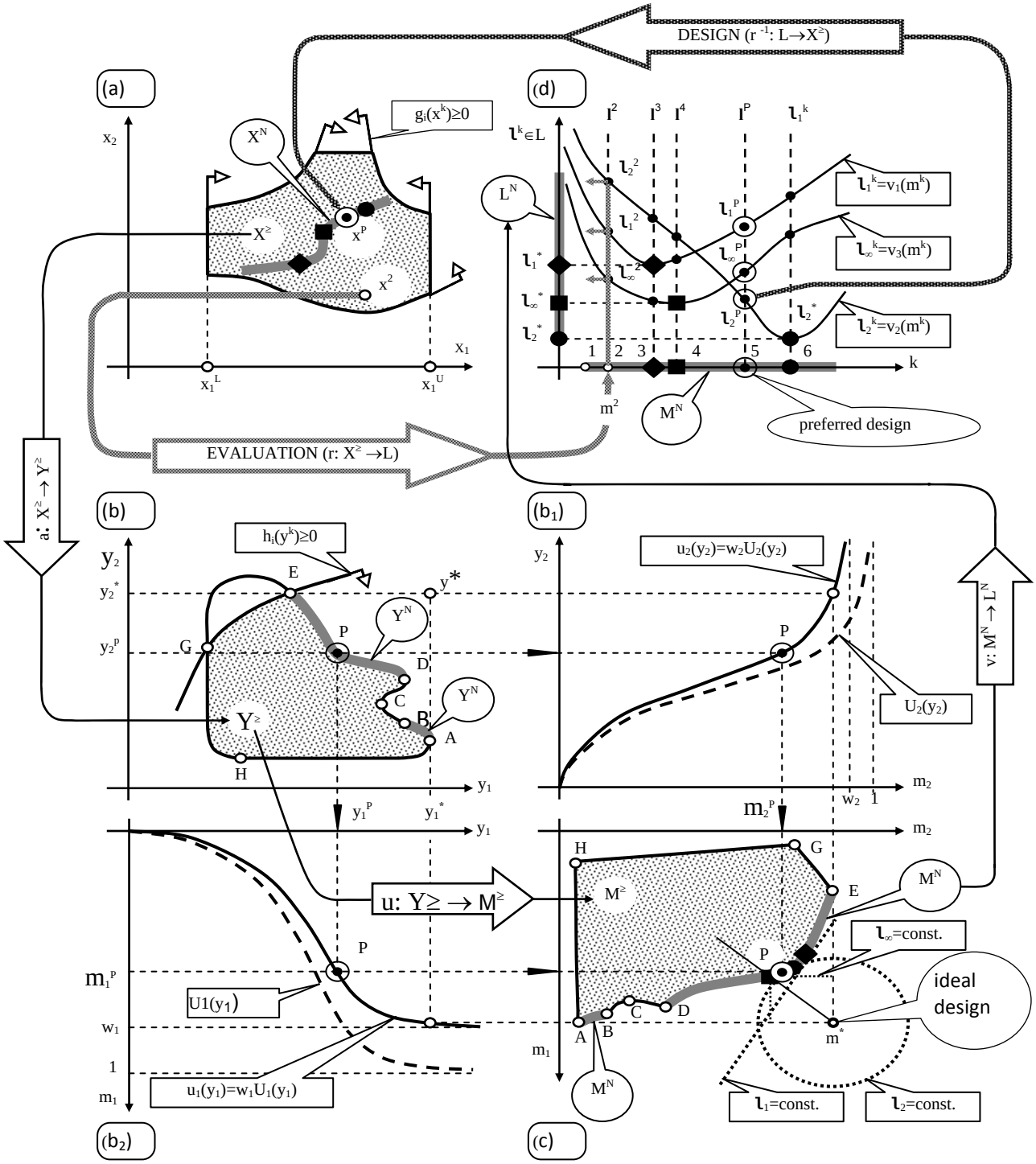
3.1. Definicija projektnog problema

Matematička definicija projektnog problema, prema [11], unutar OCTOPUS Designer okruženja vrši se unutar definicijskog modula (**Δ**). Modul **Δ** zahtjeva definiciju projektnih parametara, mjera kvalitete projekta (''izvrsnost'' - ključni indikatori performansi) te ograničenja.

Projektni parametri (deskriptori) $\mathbf{d} = \{\mathbf{X}, \bar{\mathbf{x}}\}$ uključuju projektne varijable **x**, za koje je odgovoran projektant i taktičke/tehničke konstante $\bar{\mathbf{x}}$ koje su redovito definirane unutar projektnog problema (tzv. *profil zadatka*)

Mjere kvalitete projekta definirane su pomoću skupa funkcija preslikavanja (*mappings*), a neke od tipičnih kriterijskih funkcija: cijena, masa, sigurnost itd.

Potpuni opis prostora i transformacija korištenih unutar OCTOPUS-a dani su na Sl. 3.1 (a-d). da bi se dobio bolji vizualni uvid u susretane pojmove/koncepte kod realnih DSP formulacija.



Sl. 3.1. Osnovna formulacija potpore procesu odlučivanja (DSP) [11].

3.1.1. Projektni prostor

Projektni prostor $\mathbf{X}$ - Sl. 3.1-*a.* je prostor razapet između projektnih varijabli x_i, $i = 1,\ldots,nv$. Kod strukturnih problema projektne varijable $\mathbf{x}$ su obično strukturni elementi definirani unutar osnovnog (**Φ**) fizičkog modula, zajedno sa svojim projektnim granicama x_i^U i x_i^L.

Svaki projekt *k* je predstavljen s točkom $\mathbf{x}^k = \{x_i\}$ (tj. $\mathbf{x}^2$ ili $\mathbf{x}^P$) unutar projektnog prostora $\mathbf{X}$. Projekti unutar pod-prostora (podskupa ukoliko je *k* konačan broj) podobnih rješenja $\mathbf{X}^{\geq}$ moraju zadovoljavati dani skup funkcija ograničenja $g_i(\mathbf{x}) \geq 0$, tehnoloških ili funkcionalnih zahtjeva te ostalih ograničenja. Potrebno je primijetiti da prostor $\mathbf{X}^{\geq}$ može biti konveksan (linija koja spaja dva projekta leži unutar $\mathbf{X}^{\geq}$) ili ne konveksan kako je prikazano na Sl. 3.1-a. Isto tako može se dogoditi da prostor $\mathbf{X}^{\geq}$ bude višestruko povezan (prostor sadržava rupe npr. zbog izbjegavanja pojave rezonancije kod vibracijskih problema), nadalje možemo imati diskretizirane projektne varijable (broj kontejnera, broj ukrepa, standardni profili itd.). Takovi zahtjevi onda značajno utječu na definiciju optimizacijskog problema.

3.1.2. Atributni prostor

Atributni prostor $\mathbf{Y}$ - *vidi* Sl. 3.1-*b.* je razapet između projektnih atributa y_i, dok se funkcije $\mathbf{y}^k = \mathbf{a}(\mathbf{x}^k)$ ili $\mathbf{a}: \mathbf{X}^{\geq} \rightarrow \mathbf{Y}^{\geq}$ koriste za formiranje atributnog prostora $\mathbf{Y}^{\geq} = \{\mathbf{y}^k\}$. Za svako podobno rješenje $\mathbf{x}^k$ u $\mathbf{X}$ mjere kvalitete rješenja $\mathbf{y}^k = \{y_i\}$ definiraju odgovarajuću točku u $\mathbf{Y}$ prostoru. Atributne funkcije $\mathbf{a}(\mathbf{x})$ definirane su modulu kvalitete (**Ω**).

Potrebno je naglasiti da nekoliko rješenja u $\mathbf{X}$ prostoru mogu definirati jednu jedinu točku u $\mathbf{Y}$ prostoru (ista cijena, sigurnost itd.). Isto tako potrebno je napomenuti da vrijednosti x_i ili y_i nisu međusobno usporedive te imaju različite jedinice. Zbog toga $\mathbf{X}$ i $\mathbf{Y}$ nisu metrički prostori tj. nemoguće je izračunati udaljenosti između projekata te je upravo iz tih razloga usporedba jedino moguća za svaki x_i ili y_i zasebno.

Ukoliko je određen smjer poboljšanja kvalitete (tj. minimalna cijena, maksimalna sigurnost), atributi se transformiraju u ciljeve (atributi sa smjerom). Tada možemo odrediti i ''Idealno'' rješenje $\mathbf{y}^*$ koje ima koordinate s vrijednostima najbolje ostvarene kvalitete za svaki cilj.

3.1.3. Definicija koncepta nedominiranosti

Na (Sl. 3.1-a, b.) - pod-prostor $\mathbf{Y^N}$ sastavljen od nedominiranih / Pareto optimalnih / efikasnih projekata može se definirati tek onda kada se na projekte u $\mathbf{Y}^{\geq}$ prostoru primjene projektantove preferencije prema određenim aspektima kvalitete. Tada ti projekti (obično samo mali dio svih podobnih projekata) postaju fokus pažnje projektanta, pošto dominiraju nad ostalim podobnim projektima.

Sklonost / preferiranost je binarni odnos kojim se projektni atribut y_i preferira nad y_j. Takozvani ''bolji skup'' se može definirati u odnosu na dani projekt $\mathbf{y}^0$ ukoliko su svi atributi preferirani u odnosu na $\mathbf{y}^0$. Obrnuto se može reći da je ''lošiji skup'' onaj skup koji sadržava sve projekte lošije od $\mathbf{y}^0$ po svim atributima tj. projekti su dominirani od strane $\mathbf{y}^0$. Konačno skup nedominiranih projekata $\mathbf{Y^N}$ je definiran kao skup koji nema bolji skup od sebe, tj. nije dominiran od strane nijednog projekta. Alternativno se možemo izraziti da je projekt nedominiran ukoliko je bolji od svih drugih projekata u $\mathbf{Y}^{\geq}$ prostoru bar po jednom cilju. Projekti u $\mathbf{Y^N}$ prostoru imaju definirane projektne varijable u $\mathbf{X^N}$ prostoru (vidi Sl. 3.1-a.).

3.2. Optimizacijske metode

U nastavku je dan opis optimizacijskih metoda korištenih u ovom radu.

3.2.1. Sekvencijalno linearno programiranje (SLP)

Najjednostavniji način efikasnog rješenja linearnog problema je *simplex* metoda koju je tijekom II svjetskog rata razvio *Dantzig* [14]. Općenito primalni problem za linearni model glasi:

$$\min_{\mathbf{x}} W = \min_{\mathbf{x}} \mathbf{c}^{\mathsf{T}}\mathbf{x} \tag{3.2}$$

tako da

$$\mathbf{A}\ \mathbf{x} \geq \mathbf{b}\ ;\ \mathbf{x} \geq 0 \tag{3.3}$$

gdje je **A** ($m \cdot n$) - matrica ograničenja (m- broj ograničenja, n-broj varijabli). Reci matrice **A** su linearno nezavisni te se matrica može parcijalizirati po stupcima:

$$\mathbf{A} = [\mathbf{BC}] \tag{3.4}$$

gdje je :

B - ($m \cdot m$) – kvadratna matrica

C - ($m \cdot n$-m) – matrica ostataka

$$\mathbf{c} = \left\{ \mathbf{c}^B\ \mathbf{c}^C \right\} \tag{3.5}$$

Podjednako se mogu parcijalizirati vektori funkcija cilja c i varijabli x:

$$\mathbf{x} = \left\{ \mathbf{x}^B \; \mathbf{x}^C \right\} \tag{3.6}$$

Matrica **B** naziva se bazom, a vektor $\mathbf{x}^B$ vektorom bazičnih varijabli.

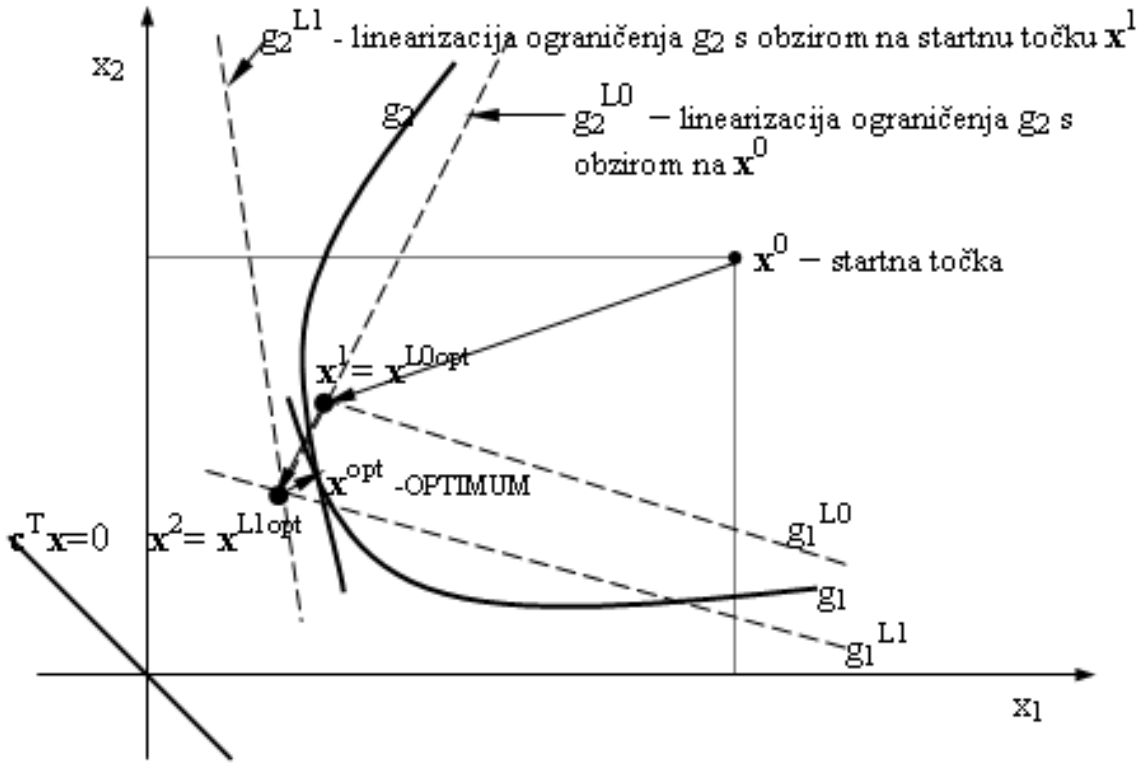
Baza je podobna ako je:

$$\mathbf{x}^B = \mathbf{B}^{-1}\mathbf{b} \geq 0 \tag{3.7}$$

jer u tom slučaju $\mathbf{x} = \left\{ \mathbf{x}^B, 0 \right\}$ zadovoljava sva ograničenja te se takav vektor naziva bazičnim rješenjem. Pretpostavimo li da imamo bazno rješenje u *simplex* algoritmu putujemo od jednog baznog rješenja do drugog zamjenom jedne bazne varijable jednom nebaznom tako da rješenje ostaje prihvatljivo, a vrijednost funkcije cilja *W* kontinuirano opada. Geometrijski prikazano, rješenja odgovaraju ekstremnim točkama (vrhovima) konveksnog poliedra, a algoritam rezultira u kretanju od vrha do vrha s padajućom vrijednošću funkcije cilja ***W.*** Spomenimo vrlo važnu transformaciju iz primarnog u dualni oblik i prednosti rješavanja problema u tom obliku. Za probleme tankostjenih konstrukcija gdje je broj ograničenja uvijek bitno veći od broja varijabli ($m > n$) prikladniji je dualni zapis i prikladno je izvršiti modifikacije standardnog algoritma, u tzv. algoritam dualne *simplex* metode [15]. Očito je da je dualna formulacija pogodnija od primarne zbog dimenzija matrice **B**, pošto je vrijeme stroja potrebno za inverziju matrice eksponencijalna funkcija dimenzija te se tu krije velika ušteda vremena.

Veliki broj strukturnih ograničenja i funkcija cilja nelinearne su funkcije Sl. 3.2. To znači da se stvarne vrijednosti funkcije cilja $f(\mathbf{x})$ i skup ograničenja $g(\mathbf{x})$ za neku točku neće podudarati s vrijednošću linearizirane funkcije $f^L(\mathbf{x})$ i $g^L(\mathbf{x})$ osim za točku $\mathbf{x}^0$ u kojoj je problem lineariziran. Optimum $\mathbf{x}^{L0opt}$ dobiven iz lineariziranog problema ne mora čak ni zadovoljiti ograničenja $g(\mathbf{x}) \geq 0$. Prirodno rješenje problema je sekvencijalna linearizacija funkcije cilja i ograničenja koja slijedi naše kretanje po projektnom prostoru. Rješenje lineariziranog problema u i-tom koraku $\mathbf{x}^{Liopt}$ za trenutni položaj $\mathbf{x}^i$ daje nam točku linearizacije $\mathbf{x}^{i+1}$ za sljedeći korak.

Ilustracija je dana na dvodimenzionalnom primjeru, Sl. 3.2. Za startnu točku $\mathbf{x}^0$ dobivamo linearizacije ograničenja $g_1^0(\mathbf{x})$ i $g_2^0(\mathbf{x})$ u obliku pravaca $g_1^{L0}(\mathbf{x})$ i $g_2^{L0}(\mathbf{x})$. Primijenivši dualnu revidiranu simplex metodu dobivamo onda točku $\mathbf{x}^{L0opt}$ za koju ponovo linearizirramo ograničenje dobivši tako $g_1^{L1}(\mathbf{x})$ i $g_2^{L1}(\mathbf{x})$ te optimum $\mathbf{x}^{L1opt}$, dok ne konvergiramo stvarnom optimumu $\mathbf{x}^{opt}$.



Sl. 3.2. Primjer sekvencijalne linearizacije ograničenja [15]

Projekt se smatra podobnim ako su ograničenja zadovoljena unutar propisane tolerancije. Ako nisu proces se nastavlja u novom podciklusu s linearizacijom za novu projektnu točku. Stacionarnost projektnih varijabli također se testira u odnosu na propisane tolerancije. Ako je ona ostvarena potrebno je provjeriti mjeru kvalitete (funkciju cilja). Ukoliko mjera kvalitete raste i unatoč tome što je došlo do stacionarnosti projektnih varijabli iteriranje u podciklusima se nastavlja, dok ne dođe do stacionarnosti. Postignuta rješenja s maksimalnom kvalitetom upotrebljavaju se za usporedbu u daljnjim podciklusima, a ukoliko dođe do stacionarnosti kvalitete postupak se zaustavlja. Detekcija oscilacija temelji se na naizmjeničnom postizavanju slične mjere kvalitete, ali za različite projektne točke [15]. Oscilacija, ako je uočena, rješava se metodom reduciranog pomaka gdje je pomak u projektnom prostoru reduciran postotkom proračunatog pomaka preko dijagonalne matrice **R**.

$$x_i^{red} = R_{ii}(x_{i+1} - x_i) + x_i \text{, gdje je } R_{ii} = 0.7 - 0.9 \text{ ; } R_{ij} = 0 \tag{3.8}$$

Metodom reduciranog pomaka usporava se optimizacijski postupak (povećavamo broj iteracija), ali ujedno onemogućavamo nagle skokove koji bi mogli rezultirati ispadanjem iz prostora podobnog projekta što je lako moguće uslijed provedene linearizacije i osjetljivosti problema. U ranim ciklusima algoritam obično sam ”izvlači” projekt iz zone oscilacija te je ovu strategiju potrebno primijeniti samo u završnim ciklusima.

3.2.2. Sekvencijalno kvadratično programiranje (SQP)

Od svoje popularizacije tijekom kasnih sedamdesetih godina prošlog stoljeća, Sekvencijalno kvadratično programiranje (SQP) jedna je od najuspješnijih metoda za rješavanje nelinearnih optimizacijskih problema s ograničenjima. Kao što je i slučaj s većinom optimizacijskih metoda, SQP nije samo jedna metoda već prije konceptualna metodologija iz koje su se razvili brojni algoritmi. Nelinearni problem koji se rješava može biti zadan standardnom formom:

$$x \in R^n : \begin{array}{l} \min f(x) \\ g_j(x) = 0, j = 1,...,m_e, \\ g_j(x) \geq 0, j = m_e + 1,...,m, \\ x_l \leq x \leq x_u. \end{array} \tag{3.9}$$

Dani izraz je generalan te uključuje specijalne slučajeve u kojima su ograničenja linearna, a funkcija cilja linearna ili kvadratična. No ipak najvažnija karakteristika SQP metode je rješavanje nelinearnih problema s nelinearnim ograničenjima i funkcijom cilja.

Osnovna ideja SQP-a je da se nelinearni problem modelira kao aproksimirani kvadratični problem oko trenutnog aproksimiranog rješenja $\mathbf{x}_k$ te da se rješenje tog potproblema iskoristi za dobivanje boljeg aproksimiranog rješenja $\mathbf{x}_{k+1}$. Ovaj proces se iterativno ponavlja, čime se dobiva sekvenca aproksimacija, dok proces ne konvergira do optimalnog rješenja odnosno dok se ne zadovolje kriteriji završetka aproksimacijske sekvence.

Jedna od važnih karakteristika SQP metode je da ne zahtjeva da niti početno rješenje, niti međurješenja budu podobna. To je velika prednost s obzirom da kod nelinearnih problema s uskim podobnim područjem može biti gotovo podjednako teško pronaći podobno rješenje kao i rješenje nelinearnog optimizacijskog problema. S obzirom da metoda obuhvaća sekvencijalno rješavanje kvadratičnih problema, potreban je brz i precizan algoritam za rješavanje kvadratičnog problema.

Za slučaj postojanja samo ograničenja jednakosti u kvadratičnom problemu, problem se reducira na rješavanje linearnog sustava jednadžbi, no ukoliko su prisutna i ograničenja jednakosti problem zahtjeva se sekvencijalno rješavanje sustava jednadžbi.

U praksi se mogu javiti brojni problemi s obzirom da određene pretpostavke pod kojima je metoda izvedena mogu biti samo uvjetno zadovoljene ili nezadovoljene, matrice mogu postati numerički singularne, skaliranje problema može biti neadekvatno itd. Stoga algoritmi za praktičnu primjenu moraju uključivati prilagodljive zaštite koje moraju osigurati izbjegavanje

takovih teškoća, odnosno zaustavljanje procesa i jasno izvještavanje o tipu uočenih problema. Detalji pojedinih metoda koje omogućuju izbjegavanje takovih poteškoća, odabir kontrolnih parametara metode te odabir kriterija konvergencije najčešće se kriju pod terminom implementacija, a imaju veliki utjecaj na uspješnost i efikasnost metode.

SQP spada u skupinu metoda koje garantiraju pronalaženje lokalnog rješenja optimizacijskog problema, ali ne i globalnog rješenja. Globalno rješenje se najčešće definira kao ono lokalno rješenje iz skupa svih podobnih lokalnih rješenja koje ima najbolju vrijednost funkcije cilja. Jedna od najčešće korištenih metoda za pronalaženje globalno optimalnog rješenja jest startanje procesa SQP sa više startnih projekata te se kao praktično globalno rješenje odabire ono koje konvergiralo do najbolje podobne vrijednosti funkcije cilja.

U nastavku će biti opisane pretpostavke i glavne korake SQP metode koja je implementirana u okviru programa NLPQLP [16].

3.2.2.1. NLPQLP implementacija SQP metode

NLPQLP [16] jest Fortranski programski kod za rješavanje problema nelinearnog programiranja u formi zadanoj prema (3.9). Pretpostavlja se da su funkcija cilja i ograničenja mogu izračunati s dovoljno velikom preciznošću te da su kontinuirano derivabilni. Osnovna ideja je generirati sekvencu problema kvadratičnog programiranja dobivenu kvadratičnom aproksimacijom Lagrangeove funkcije i linearizacijom ograničenja. Informacije drugog reda se osvježavaju kvazi-Newtonskom procedurom, a metoda je stabilizirana dodatnim linijskim pretraživanjem. Kao posebnosti NLPQLP metode navode se vrlo robusna implementacija te jednostavnost korištenja. Kontrolni parametri koje je potrebno namjestiti su samo maksimalni broj iteracija i željena tolerancija.

Metoda je integrirana u brojene programske sustave, npr.: ANSYS/POPT (CAD-FEM, Grafing), DesignXplorer (ANSYS, Canonsburg, MOOROPT (Marintek, Trondheim), iSIGHT (Enginious Software/Dassault), ModeFRONTIER (ESTECO, Trieste) itd. Popis korisnika metode uključuje AMD, BASF, Bell Labs, BMW, DLR, EADS, Eurocopter, General Electric, Lockheed Martin, NASA Langley, NASA Ames,USAF Research Lab, Wright R & D Center.

<u>Osnove algoritma:</u>

Kao što je već navedeno osnovna ideja je generirati sekvencu problema kvadratičnog programiranja dobivenu kvadratičnom aproksimacijom Lagrangeove funkcije i linearizacijom ograničenja. Stoga se definira kvadratična aproksimacija Lagrangeove funkcije L(**x**,**u**) te

aproksimacija Hesseove matrice od $\mathbf{L}(\mathbf{x}_k,\mathbf{u}_k)$ takozvanom kvazi-Newton matricom $\mathbf{B}_k$. Iz toga slijedi problem kvadratičnog programiranja:

$$\begin{aligned} & \min \frac{1}{2} d^T B_k d + \nabla f(x_k)^T d \\ d \in R^n : \quad & \nabla g_j(x_k)^T d + g_j(x_k) = 0, j = 1,\ldots,m_e, \\ & \nabla g_j(x_k)^T d + g_j(x_k) \geq 0, j = m_e+1,\ldots,m, \\ & x_l - x_k \leq d \leq x_u - x_k. \end{aligned} \tag{3.10}$$

Ako je d_k rješenje trenutnog problema, nova iteracija dobiva se izrazom:

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.11}$$

gdje je α_k (0,1] - odgovarajući korak.

Osvježavanje matrice $\mathbf{B}_k$ izvodi se standardnom kvazi Newtonovom BFGS (Broyden–Fletcher–Goldfarb–Shanno) metodom [17]. Za osvježavanje su potrebna samo razlika vektora $x_{k+1} - x_k$ i $\nabla L(x_{k+1}, u_k) - \nabla L(x_k, u_k)$. Pod određenim uvjetima može se garantirati da je matrica $\mathbf{B}_k$ pozitivno definitna.

3.2.3. Optimizacija rojem čestica (*Particle swarm optimization- PSO*)

Osnovna ideja PSO algoritma prezentirana u [18] proizlazi iz činjenice da na kretanje (let) svake pojedinačne čestice u cijeloj populaciji (roju) kroz projektni i atributni prostor utječu najbolji projekti (lokacije) nađeni do trenutne iteracije od strane čestice kao i cijelog roja.

Osnovni PSO algoritam možemo prikazati kroz 6 koraka (Sl. 3.3.)

KORAK 1: Inicijalizacija roja - nakon inicijalizacije roja čestice se obično nasumično rasporede po projektnom prostoru te se zatim svakoj čestici u odnosu na njezinu trenutnu lokaciju pridružuje mjera kvalitete koju zovemo dobrotom rješenja (*fitness*). Uobičajeni način za dobivanje početnih lokacija čestica dan je sljedećom jednadžbom:

$$^i x^j_{k=0} = x^j_{\min} + r \cdot (x^j_{\max} - x^j_{\min}) \tag{3.12}$$

x - projektna varijabla

$i = 1,\ldots N_p$ - razmatrana čestica

$j = 1,\ldots N_{DV}$ - trenutna projektna varijabla

r - slučajan broj između 0 i 1

k - trenutna iteracija

Početna brzina svake čestice se može postaviti na 0 za svaku projektnu varijablu, ali isto tako se može upotrijebiti sljedeći izraz

$$ {}^{i}v_{k=0}^{j} = \frac{x_{\min}^{j} + r \cdot (x_{\max}^{j} - x_{\min}^{j})}{\Delta t} \tag{3.13} $$

gdje je:

v - brzina čestice

Δt - inkrement vremena (u većini slučajeva koristi se 1)

KORAK 2: Izbor najboljih vlastitih i globalnih vodiča potrebnih za izračun brzine u KORAKU 3. Potrebno je da svaka čestica izabere svoju vlastitu najbolju lokaciju ($ {}^{i}p^{j} $) pronađenu do trenutne iteracije te isto tako odabere najbolju globalnu lokaciju ($ {}^{g}p^{j} $) pronađenu od strane cijelog roja do trenutne iteracije.

KORAK 3: Izračun nove brzine za svaku česticu u roju, prema izrazu:

$$ {}^{i}v_{k+1}^{j} = w\,{}^{i}v_{k}^{j} + c_1 \cdot r_1 \frac{\left({}^{i}p^{j} - {}^{i}x_{k}^{j}\right)}{\Delta t} + c_2 \cdot r_2 \frac{\left({}^{g}p_{k}^{j} - {}^{i}x_{k}^{j}\right)}{\Delta t} \tag{3.14} $$

$ {}^{i}p $ - najbolja lokacija pronađena od strane razmatrane čestice

$ {}^{g}p $ - najbolja (globalna) lokacija pronađena od strane roja

w - inercijski parametar

c_1 - kontrolni parametar komponente vlastite brzine

c_2 - kontrolni parametar komponente globalne brzine

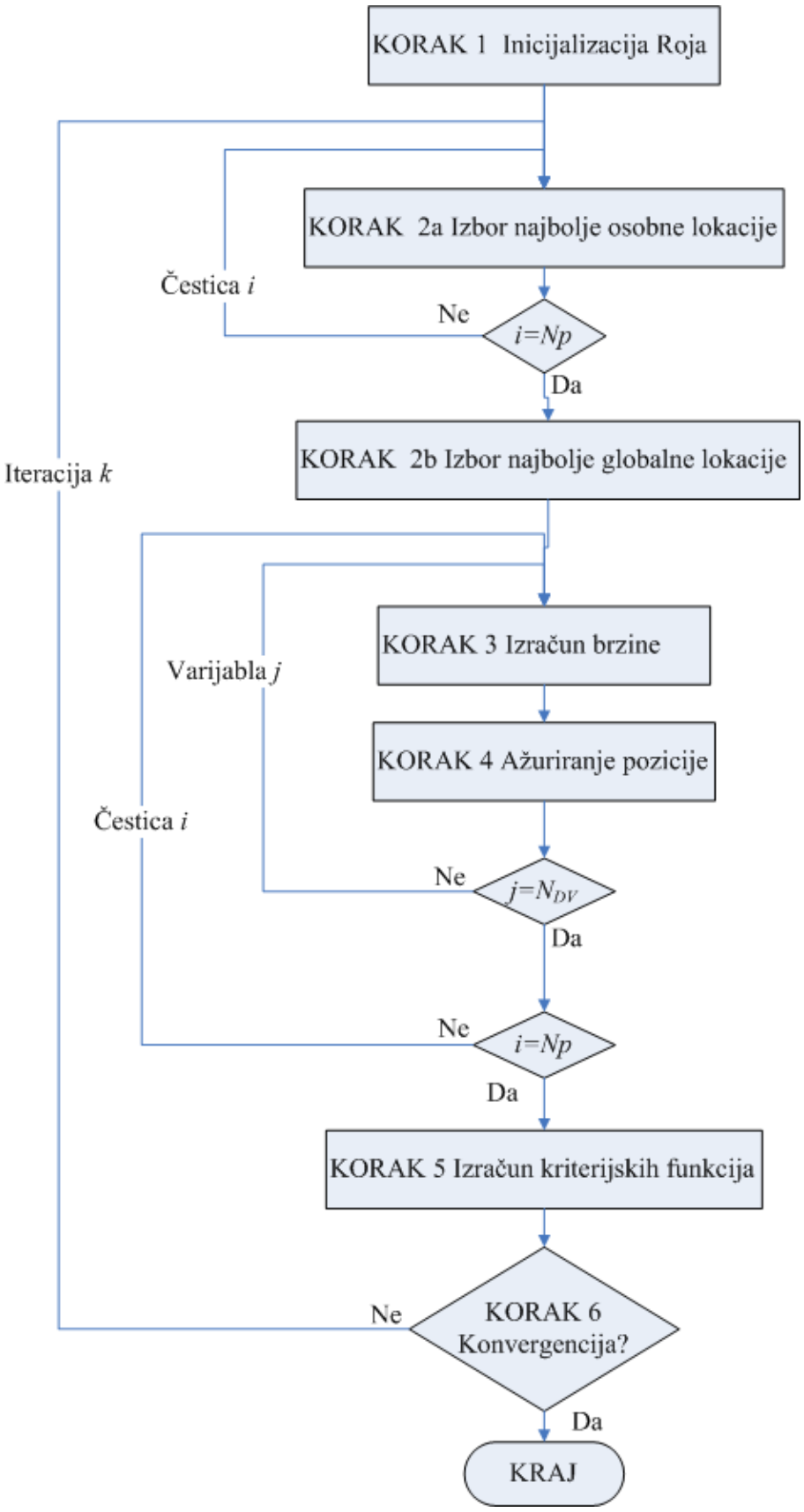
r_1, r_2 - slučajni brojevi između 0 i 1

Iz jednadžbe brzine možemo vidjeti da se brzina svake čestice sastoji od tri komponente:

$w^{i}v_{k}^{j}$ - inercijska komponenta

$c_1 \cdot r_1 \frac{\left({}^{i}p^{j} - {}^{i}x_{k}^{j}\right)}{\Delta t}$ - vlastita komponenta

$c_2 \cdot r_2 \frac{\left({}^{g}p_{k}^{j} - {}^{i}x_{k}^{j}\right)}{\Delta t}$ - globalna komponenta



Sl. 3.3. Dijagram toka osnovnog PSO algoritma

Kako se može vidjeti iz gore navedenih formula kretanjem čestice se upravlja pomoću: brzine iz prethodnog koraka, vlastitim znanjem čestice - vlastita najbolja lokacija i globalnim znanjem roja - najbolja lokacija pronađena od strane cijelog roja. Utjecaj pojedine komponente brzine može se podešavati preko kontrolnih parametara w, c_1, c_2. Uobičajene vrijednosti kontrolnih kreču se za w između 0.5 i 1.5 dok za c_1, c_2 između 1.5 i 2.5.

KORAK 4 Izračun novog položaja - pomoću brzine čestice dobivene iz prethodnog koraka, pronalazi se novi položaj čestice prema idućem izrazu:

$$ {}^{i}x_{k+1}^{j} = {}^{i}x_{k}^{j} + {}^{i}v_{k+1}^{j} \cdot \Delta t \tag{3.15}$$

KORAK 5 Izračun dobrote rješenja - za novi položaj (lokaciju) vrši se izračun dobrote za svaku česticu.

KORAK 6 Provjera konvergencije - ukoliko je dostignut zadani broj iteracija ili zadovoljen kriterij konvergencije algoritam se zaustavlja, inače se ponavljaju koraci 2 do 6 sve dok se ne zadovolji jedan od uvjeta (broj iteracija odnosno kriterij konvergencije).

Gore opisani algoritam je predviđen za optimizacijske probleme bez ograničenja, ali postoje brojna proširenja algoritma koja omogućuju rad s ograničenjima. Veliki broj proširenja za rad s ograničenjima možemo pronaći u raznim evolucijskim algoritmima (nekoliko primjera implementacije možemo pronaći u [19] i[20].

3.2.3.1. Prilagodba optimizacije rojem čestica za rad s višeciljnim problemima

Iako u svojoj osnovi PSO algoritam nije predviđen za rad s višeciljnim problemima, kroz proteklo desetljeće razvili su se brojna proširenja metode koja to omogućuju. U većini slučajeva upotrijebljene su i prilagođene već prije razvijene metode s područja višeciljnih genetskih algoritama ili višeciljnih evolucijskih strategija.

Kod optimizacije jednociljnog projektnog problema PSO optimizacijskim algoritmom, postoji samo jedno optimalno rješenja te je izbor globalnog i lokalnog vodiča za svaku česticu jednostavan i očit. Izbor je jednostavan pošto svaka čestica ima samo jedan lokalni i globalni vodič. Prelaskom na višeciljnu optimizaciju javlja se novi problem, a to je skup nedominiranih rješenja koji se generira zbog višeciljnosti problema. Upravo zbog toga kod višeciljne PSO optimizacije izbor globalnog i lokalnog vodiča postaje mnogostruko kompliciran, tj. svaka čestica sada ima skup nedominiranih rješenja iz kojeg mora odabrati svog globalnog i lokalnog vodiča. Unutar programskog sustava OCTOPUS Designer-a implementirana su tri MOPSO

algoritma koja svaki na svoj specifičan način vrše odabir vodiča iz skupa nedominiranih rješenja, a to su: *Hypercube*, *Sigma* i MOCLPSO [21].

MOPSO Hypercube

Metoda je predstavljena od strane [20], koju su razvili na spoznajama iz *Multiobjective Evolutionary Algorithm* (MOEA) područja. Osnovna ideja metode je stvaranje i održavanje arhive za spremanje nedominiranih lokacija pronađenih od strane čestica, gdje će zatim jedna od tih nedominiranih lokacija postati globalni vodič. Za izbor globalnog vodiča iz arhive upotrijebljena je metoda koju je dana u [22]. Ideja metode je podjela atributnog prostora na ekvidistantnu mrežu, dijeljenjem svakog atributa sa zadanim brojem podjela. Segmenti tako dobivene ekvidistantne mreže zovu se hiperkocke (*hypercubes*) te su unutar njih smještaju nedominirane lokacije iz arhive. Odabir globalnog vodiča iz generiranih hiperkocki objašnjen je u nastavku poglavlja.

Nakon inicijalizacije roja (KORAK 1 na Sl. 3.3.) ili KORAK-a 6 slijedi generiranje hiperkocki. Da bi se generirala ekvidistantna mreža tj. hiperkocke, potrebno je pronaći minimalnu i maksimalnu vrijednosti svakog atributa iz trenutnog skupa nedominiranih rješenja te tako dobiveni raspon vrijednosti po atributu podijeliti s unaprijed zadanom granulacijom.

Nakon generiranja ekvidistantne mreže, neke će od hiperkocki sadržavati nedominirane lokacije iz arhive (nedominirani projekti). Takvim se hiperkockama zatim dodjeljuje mjera dobrote, koja se računa prema izrazu:

$$dobrota = \frac{10}{NP} \tag{3.16}$$

NP - broj čestica unutar hiperkocke

Hiperkocke koje ne sadrže lokacije iz arhive bivaju izbrisane. Kod Hypercube metode sve čestice imaju zajedničkog globalnog vodiča koji se odabire iz skupa nedominiranih lokacija spremljenih u arhivi (KORAK 2b na Sl. 3.3.). Izbor globalnog vodiča *RouletteWheelSelection* metodom temelji se na mjeri dobrote hiperkocke dobivene jednadžbom (3.16), iz koje se vidi da gušće naseljene hiperkocke imaju manju vrijednost dobrote te prema tome imaju manju šansu da se iz njih odabere globalni vodič. Ukoliko u odabranoj hiperkocki ima više od jednog nedominiranog rješenja (što je gotovo uvijek slučaj), nasumično se odabire jedan od njih. Ovakvim načinom odabira vodiča osigurava se ujednačena popunjenost Pareto fronte pošto se daje veći značaj rjeđe naseljenim hiperkockama. Izbor lokalnog vodiča za svaku česticu (KORAK 2a na Sl. 3.3.) i svi ostali koraci izvode se jednako kao i u standardnom PSO. Procedura se

ponavlja za svaku iteraciju sve dok se ne zadovolji jedan od kriterija konvergencije (KORAK 6 na Sl. 3.3.)

Mjera dobrote hiperkocke ujedno služi i za reduciranje arhive, gdje hiperkocke s manjom vrijednošću dobrote imaju veću šansu (*RouletteWheelSelection*) da iz njih budu odabrane nedominirane lokacije koje će se izbaciti iz arhive. Pošto se izbacila jedna nedominirana lokacija iz arhive a time ujedno i iz hiperkocke, tada se za nju ponovo izvrši proračun mjere dobrote i pokrene *RouletteWheelSelection* metoda. Postupak se ponavlja dok se ne postigne željena veličina arhive.

MOPSO Sigma

Sigma metoda predstavljena je u radu [23], s osnovnom idejom stvaranja sigma vektora pomoću kojih bi se razriješio problem odabira globalnog vodiča iz skupa nedominiranih lokacija spremljenih u arhivi, čime se povećava selekcijski pritisak koji usmjerava algoritam prema obećavajućim dijelovima podobnog prostora.

Pošto je izvršena inicijalizacija roja KORAK 1. ili KORAK 6 Sl. 3.3, slijedi izračun sigma vektora za svaku česticu, odnosno njenu trenutnu lokaciju prema jednadžbi:

$$\sigma = \begin{bmatrix} f_1^2 - f_2^2 \\ : \\ f_i^2 - f_j^2 \end{bmatrix} \Big/ \sum_{i=1}^{NA} f_i^2 \tag{3.17}$$

f_i - iznos dobrote

NA - broj atributa

$i = 1.....NA$

$j = 1..... NA$

$i \neq j$

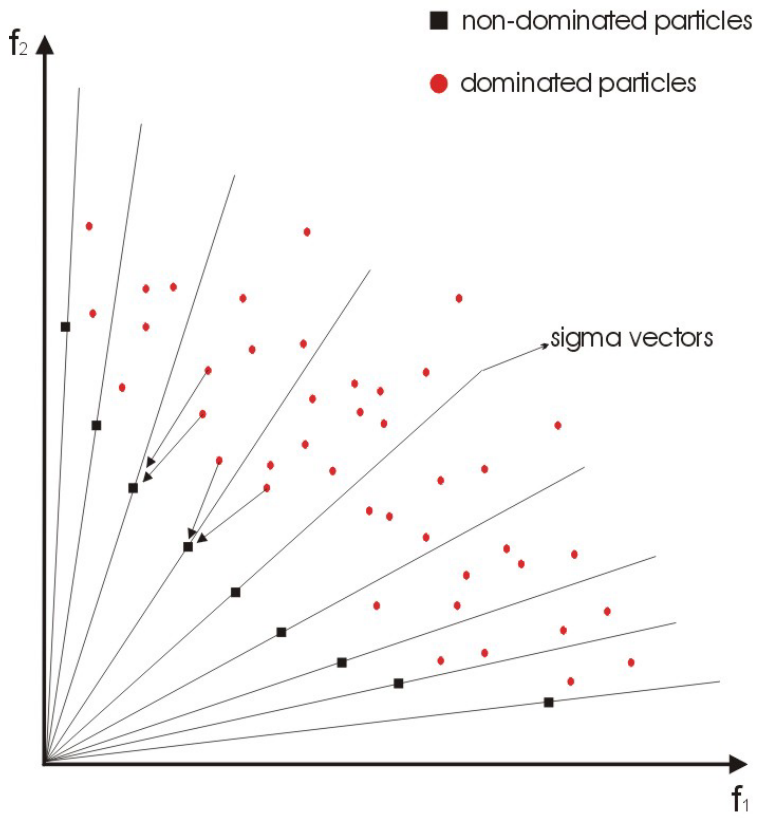
gdje je Sigma za jednociljni problem skalar dok za višeciljne probleme postaje vektor od $\frac{m}{2}$ elemenata (m je jednak broju atributa). Kod Sigma metode svaka čestica ima svog vlastitog globalnog vodiča, koji se odabire iz skupa nedominiranih lokacija spremljenih u arhivi (KORAK 2b na Sl. 3.3.). Izbor globalnog vodiča temelji se na računanju euklidskih udaljenosti između sigma vektora čestica iz trenutne iteracije, odnosno njene lokacije i sigma vektora nedominiranih lokacija iz arhivi, prema jednadžbi (3.18) - eksponent p = 2:

$$d = \left(\left| \sigma_i - \sigma_j \right|^p \right)^{\frac{1}{p}} \tag{3.18}$$

i = 1…….broj dominiranih čestica

j = 1……broj nedominiranih čestica

gdje je globalni vodič promatrane čestice najbliža nedominirana lokacija (vidi Sl. 3.4.). Izbor lokalnog vodiča za svaku česticu (KORAK 2a na Sl. 3.3.) i svi ostali koraci izvode se jednako kao i u standardnom PSO.



Sl. 3.4. Osnovni princip Sigma metode [21]

Procedura se ponavlja za svaku iteraciju sve dok se ne zadovolji jedan od kriterija - konvergencija KORAK 6.

MOPSO MOCLPSO

Metoda *Multiobjective Comprehensive Particle Swarm Optimization* MOCLPSO prikazana ne u [24]. Glavna karakteristika metoda je u tome što čestice uče ne samo od najboljeg globalnog i najboljeg lokalnog vodiča već i od najboljih lokalnih vodiča čestica, čime se povećava selekcijski pritisak koji usmjerava algoritam prema obećavajućim dijelovima podobnog prostora. Kao i kod Hypercube metode, MOCLPSO isto ima samo jednog globalnog vodiča za sve čestice, kojeg nasumično izabire iz skupa nedominiranih lokacija u arhivi. Taj se globalni vodič unutar MOCLPSO metode zove *elitna lokacija*.

Učenje čestice se određuje preko kontrolnih parametara definiranih od strane korisnika, pri čemu učenje čestice od *elitne lokacije* (globalnog vodiča) određuje faktor P_m dok se učenje

od lokalnog vodiča ili lokalnog vodiča ostalih čestica određuje faktorom P_c. Potrebno je naglasiti da iako čestica uči od *elitne lokacije*, to ne znači da za izračun brzine koristi sve vrijednosti projektnih varijabli, tj. uzima *m* vrijednosti projektnih varijabli. Faktor *m* računa se upotrebom faktora P_m prema jednadžbi:

$m = \left[P_m \cdot D\right]$ - uzeti broj projektnih varijabli (3.19)

D - broj dimenzija projektnog prostora

3.2.3.2. Prilagodba algoritma za rad s ograničenjima

Kao što je moguće vidjeti u osnovnom algoritmu optimizacije rojem čestica, algoritam inicijalno ne uključuje način rada s ograničenjima. Isto tako niti jedan od tri navedena načina proširenja metode ne objašnjavaju na koji način bi metode trebale raditi ukoliko se definiraju ograničenja. Zbog toga je navedene algoritme bilo neophodno modificirati kako bi se omogućilo da finalna Pareto rješenja budu podobna, odnosno da svako Pareto rješenje zadovoljava sva postavljena ograničenja.

Kod odabira najboljeg osobnog vodiča (najbolja vlastita lokacija) MOPSO optimizacijski algoritam koristi se sljedećim pravilima:

- obje lokacije podobne i nedominirane, izbor osobnog najboljeg osobnog vodiča vrši se slučajnim odabirom
- obje lokacije podobne i dominirane, izbor osobnog najboljeg osobnog vodiča vrši se slučajnim odabirom
- jedna lokacija je podobna/dominirana, a druga lokacija je podobna/nedominirana - odabire se podobna/nedominirana lokacija za osobnog vodiča
- jedna lokacija je podobna, dok je druga lokacija nepodobna - odabire se podobna lokacija za osobnog vodiča
- obje lokacije su nepodobne - svakoj od lokacija se dodjeljuje iznos dobrote ograničenja *cf* te se odabire ona lokacija s boljom dobrotom, prema izrazu:

$$cf_i = \sum_{j=1}^{NC} \frac{\left(cv_j\right)_i}{\max cv_j} \tag{3.20}$$

gdje je:

i - *i*-ta lokacija

j = 1….NC - broj ograničenja

cf - iznos dobrota ograničenja

cv_j - narušenost ograničenja

maxcv_j - maksimalna narušenost *j*-tog ograničenja s obzirom na narušenost ograničenja od svih lokacija.

Kao što se vidi izraz (3.20) vrednuje broj narušenih ograničenja kao i iznose narušenosti ograničenja. Također je bitno napomenuti da je normalizacija izvedena automatski bez potrebe za definiranjem normirajućih faktora prije optimizacije. S druge strane to je ujedno i nedostatak jer se normalizacija izvodi s trenutno postignutim maksimalnim prekršajima ograničenja što zahtjeva da se za svaku lokaciju u repozitoriju lokalnih i globalnih vodiča nakon svake iteracije iznova računaju dobrote ograničenja.

Odabir najboljeg globalnog vodiča zapravo je vrlo rijetko upravljan s izrazom za dobrotu ograničenja jer globalni vodič je skoro uvijek nedominirana podobna lokacija. Nepodobne lokacije mogu biti globalni vodiči samo u prvim iteracijama jako ograničenih problema, a tada se primjenjuje ekvivalentni skup odluka kao kod odabira lokalnih vodiča. Odabir globalnog vodiča zapravo je u najvećoj mjeri i dalje upravljan odabranom metodom za rad s više ciljeva.

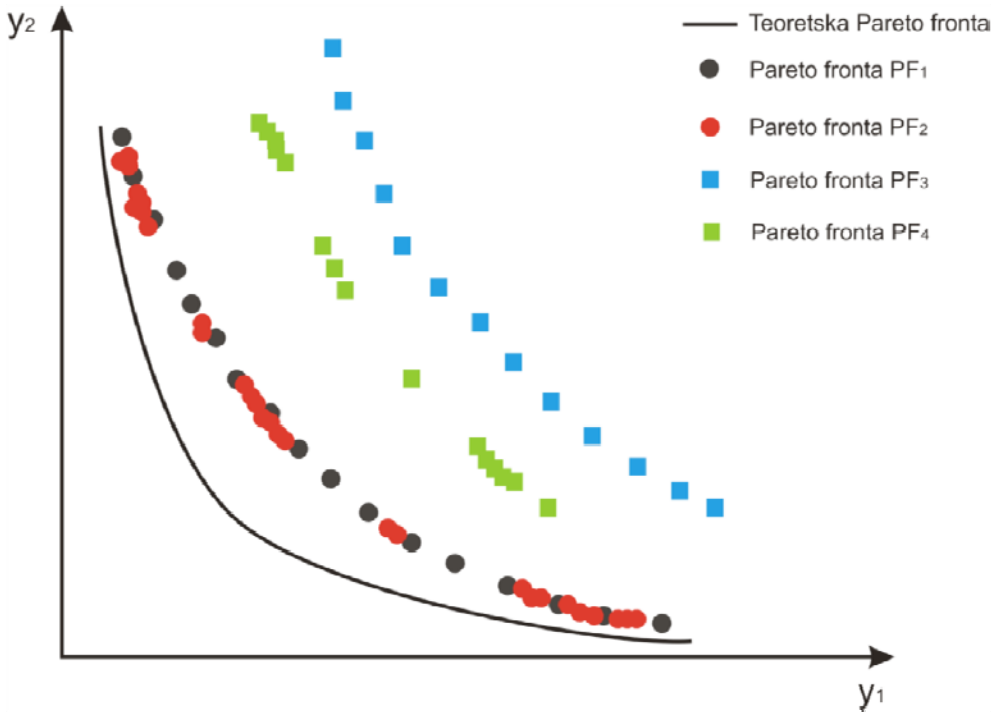
3.3. Mjerenje kvalitete Pareto fronte

Mjerenje kvalitete dobivene Pareto fronte ([25], [26), a time i optimizacijskih algoritama, obavlja se u kontroliranim uvjetima. To znači da je prije provođenja mjerenja definirana računalna oprema na kojoj će se provoditi svi eksperimenti. Isto tako, utvrđena je standardna rezolucija (broj poziva modela), koju je potrebno držati konstantnom, unutar jednog ciklusa mjerenja. Potreba za kontroliranim uvjetima proizlazi iz želje za dosljednom kvantitativnom usporedbom rezultata MOEA (*Multi Objective Evolutionary Algorithm*) algoritama. Usporedba se može promatrati između rezultata dobivenih mjerenim algoritmom, ili između dobivenih rezultata i prije definiranog *benchmarka*. U slučaju ispitivanja MOEA algoritama *benchmark* je, tzv. optimalna Pareto fronta, PF_{true}. Idealno bi bilo da postoji teoretska Pareto fronta (Pareto fronta kontinuiranog matematičkog modela). Nažalost, u velikom broju slučajeva to nije slučaj, pa se kod realnih problema koristi najbolja dobivena Pareto Fronta nekim optimizacijskim algoritmom.

Kriteriji na osnovu kojih se uspoređuju optimizacijski algoritmi, a koji su dobiveni mjerenjem u kontroliranim uvjetima, mogu se svrstati u dvije kategorije, uspješnost i djelotvornost. Pod uspješnošću podrazumijeva se potreban računalni rad za dobivanje rješenja, tj. CPU (*central processing unit*) vrijeme, broj iteracija/evaluacija, upotreba računalnih resursa,

skalabilnost (daje informaciju o veličini projektnog problema koji algoritam može riješiti) i lakoća korištenja (potreban napor da bi se svladala upotreba optimizacijskog algoritma, *user friendliness*). Dok djelotvornost podrazumijeva što bolje pretražen projektni prostor, a to znači ravnomjernije popunjenu Pareto frontu i što bolju aproksimaciju teoretske Pareto fronte.

Da bi se lakše predočio kompleksan kriterij, kao što je djelotvornost, na Sl. 3.5 prikazano je nekoliko Pareto fronti od kojih je jedna teoretska Pareto fronta. Usporedbom PF_4 i PF_3 potpuno je jasno da PF_4 dominira nad PF_3, ali isto tako je vidljivo da je PF_1 ravnomjernije popunjena od PF_4. PF_1 i PF_2 podjednako su blizu teoretske Pareto fronte, ali je PF_1 ipak bolja pošto je za razliku od PF_2 jednoliko popunjena. Cilj svakog optimizacijskog algoritma je generiranje Pareto fronte PF_1 koja je: jednoliko popunjena, blizu teoretske Pareto fronte i u potpunosti razapeta između rubova atributnog prostora.



Sl. 3.5. Kvaliteta Pareto fronte

Način na koji se postiže ravnomjerno popunjena Pareto fronta, što bolja aproksimacija Pareto fronte, itd., je generiranje jakog selekcijskog pritiska. Kod algoritama kao što je MOGA, selekcijski pritisak se generira pomoću operatora dodjele dobrote i operatorima selekcije.

Isto tako nije zanemariv ni aspekt efikasnosti optimizacijskog algoritma, posebno upotreba računalnih resursa, kao što su memorijski prostor i procesorsko opterećenje. Optimizacijski algoritam u svakoj iteraciji generira znatan broj novih nedominiranih rješenja koje je potrebno spremiti u memoriju računala. Pošto je memorijski prostor računala fizički

limitiran, potrebno je ograničiti broj spremljenih nedominiranih rješenja. Isto tako, veliki broj nedominiranih rješenja uvelike opterećuje CPU računala, pošto se svako rješenje dobiveno u trenutnoj iteraciji mora usporediti s postojećim nedominiranim rješenjima. Upravo zbog toga svaki optimizacijski algoritam na svoj specifičan način u svakoj iteraciji određuje koja nedominirana rješenja čuva, a koja odbacuje.

3.3.1. Indikatori kvalitete ostvarene Pareto fronte

MOEA indikatori predstavljaju vrijednost dobivenu u odnosu na broj podobnih (odnosno, nedominiranih) rješenja, za blizinu Pareto fronti ili nekom drugom skupu projekata. Također, MOEA indikatori mogu biti realne vrijednosti koje označavaju neku od karakteristika PF_{true} - raširenost, ravnomjernu popunjenost, itd. Potrebno je uočiti da su neke od mjera jednostavne za izračunavanje i razumijevanje, dok je kod ostalih indikatora kvalitete potrebna normalizacija atributa (atributi imaju različite mjerne jedinice), kako bi se generirale usporedne vrijednosti.

Kako bi se bolje razumio koncept nedominiranosti, a s njim i indikatori kvalitete Pareto fronte, u Tbl. 3.1 prema [25], ilustrirani su odnosi između rješenja i između skupova rješenja.

Iz simbola definiranima u Tbl. 3.1 i iz Sl. 3.6 a), moguće je uspostaviti sljedeće odnose između rješenja:

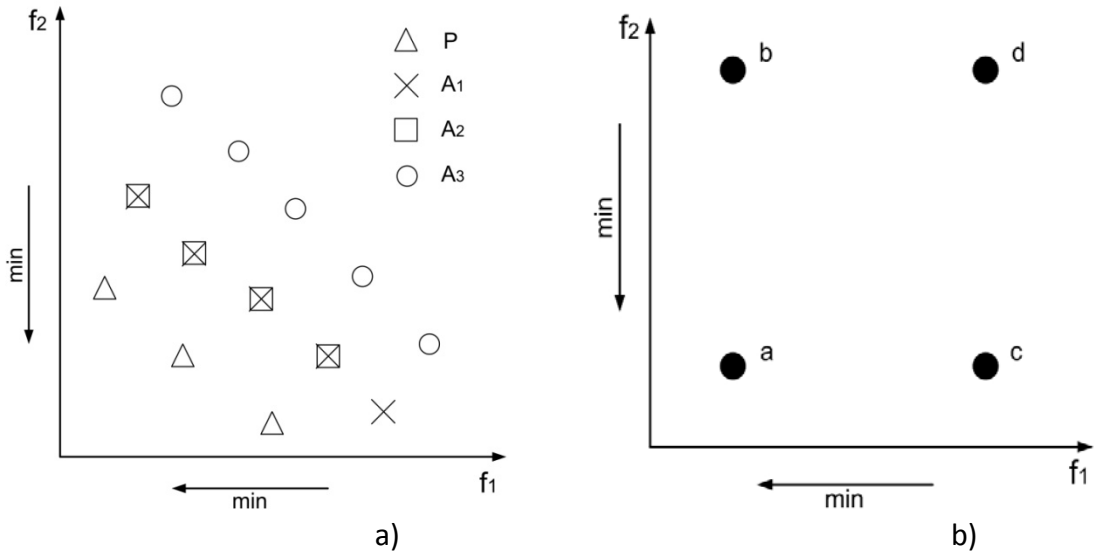
$a \succ b$, $a \succ c$, $a \succ d$, $b \succ d$, $c \succ d$, $a \succ\succ d$, $a \succeq a$, $a \succeq b$, $a \succeq c$, $a \succeq d$, $b \succeq b$, $b \succeq d$, $c \succeq c$, $c \succeq d$, $d \succeq d$, $b \| c$.

Isto tako, prema simbolima definiranima u Tbl. 3.1 Sl. 3.6 b), slijede odnosi dominiranosti između skupa rješenja za rezultate algoritama A_1, A_2 i A_3 imajući PF_{true} od *P*:

$A_1 \succ A_3$, $A_2 \succ A_3$, $A_1 \succ\succ A_3$, $A_1 \succeq A_1$, $A_1 \succeq A_2$, $A_1 \succeq A_3$, $A_2 \succeq A_2$, $A_3 \succeq A_3$, $A_1 \rhd A_2$, $A_1 \rhd A_3$, $A_2 \rhd A_3$.

Tbl. 3.1 Odnos dominacije rješenja i skupa rješenja

Odnos	Rješenja	
Jaka dominiranost	$f(x^1) \succ\succ f(x^2)$	$\forall_{i \in F}$,($f(x^1)$ je bolje od $f(x^2)$)
Dominiranost	$f(x^1) \succ f(x^2)$	$f(x^1)$ nije lošiji od $f(x^2)$ u svim ciljevima i bolji je u jednom cilju
Slaba dominiranost	$f(x^1) \succeq f(x^2)$	$f(x^1)$ nije lošiji od $f(x^2)$ u svim ciljevima
Neusporediv	$f(x^1) \| \| f(x^2)$	niti $f(x^1)$ slabo dominira $f(x^2)$ niti $f(x^2)$ slabo dominira $f(x^1)$
Indiferentan	$f(x^1) \sim f(x^2)$	$f(x^1)$ ima istu vrijednost $f(x^2)$ u svakom cilju
	Skup rješenja	
Jaka dominiranost	$A \succ\succ B$	svaki $x^2 \in B$ je jako dominiran barem jednim $x^1 \in A$
Dominiranost	$A \succ B$	svaki $x^2 \in B$ je dominiran barem jednim $x^1 \in A$
Bolji	$A \rhd B$	svaki $x^2 \in B$ je slabo dominiran barem jednim $x^1 \in A$ $A \neq B$
Slaba dominiranost	$A \succeq B$	svaki $x^2 \in B$ je slabo dominiran barem jednim $x^1 \in A$
Neusporediv	$A \| \| B$	niti A slabo dominira B ni B slabo dominira A
Indiferentan	$A \sim B$	A slabo dominira B i B slabo dominira A



Sl. 3.6. Grafički prikaz odnosa dominiranosti kod minimuma za rješenja i skupove rješenja

3.3.1.1. Vrijeme trajanja optimizacijskog procesa (*CalcTime*)

Kako mu ime govori, ovo je jednostavan indikator kvalitete koji računa ukupno vrijeme optimizacijskog procesa.

3.3.1.2. Broj poziva analitičkog modela (*NumModelCalls*)

Indikator kvalitete koji pokazuje broj poziva analitičkog modela.

3.3.1.3. Broj podobnih rješenja (*NumFeasible*)

Indikator kvalitete koji daje broj podobnih rješenja tijekom cijelog postupka optimizacije.

3.3.1.4. Ukupni broj nedominiranih rješenja (ONVG)

Indikator kvalitete koja mjeri ukupan broj nedominiranih rješenja. U literaturi se ovaj indikator još i naziva ONVG (*Overall Nondominated Vector Generation Ratio*), a računa definiran je kroz izraz:

$$ONVG = |PF_{known}| \tag{3.21}$$

3.3.1.5. Najbolji projekt

Najbolji projekt (*Best Objective*) vraća najbolji projekt po zadanom cilju.

$$bestObj = \min(|\max(Y_i)|) \tag{3.22}$$

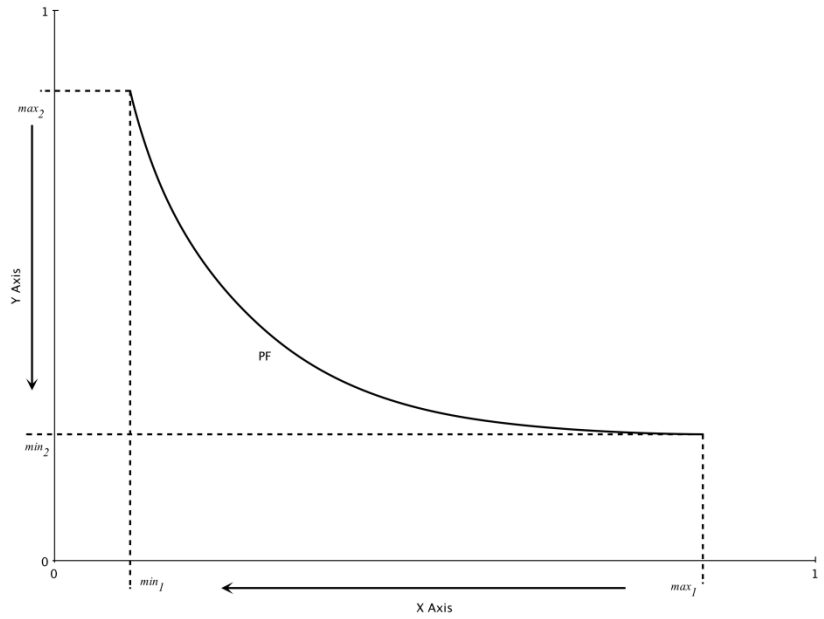
i=1, n - broj atributa

3.3.1.6. Normalizirana raširenost Pareto fronte

Ovaj indikator kvalitete (*Pareto Front Normalized Span*) mjeri raširenost Pareto fronte (Sl. 3.7), mjerenjem a definiran je izrazom:

$$pfSpan = \frac{\sum_{i=1}^{n} |\max_i - \min_i|}{n} \tag{3.23}$$

gdje su:
n - broj atributa
$\max_i$ - najveća pronađena vrijednosti atributa
$\min_i$ - najmanja pronađena vrijednosti atributa

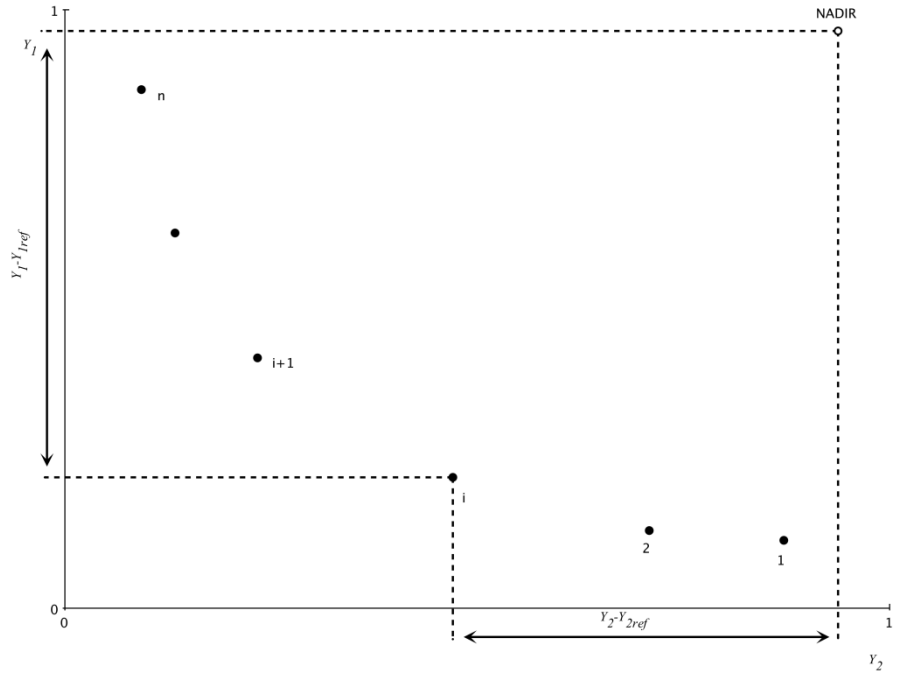


Sl. 3.7. Ilustracija indikatora raširenost Pareto fronte

3.3.1.7. Vanjski promjer

Vanjski promjer indikator kvalitete (Sl. 3.8) mjeri udaljenost trenutnog rješenja od NADIR rješenja (najgore rješenje, suprotno od utopije). Ovim indikatorom moguće je vidjeti pomicanje Pareto fronte kroz iteracije algoritma i njeno približavanje svom maksimumu.

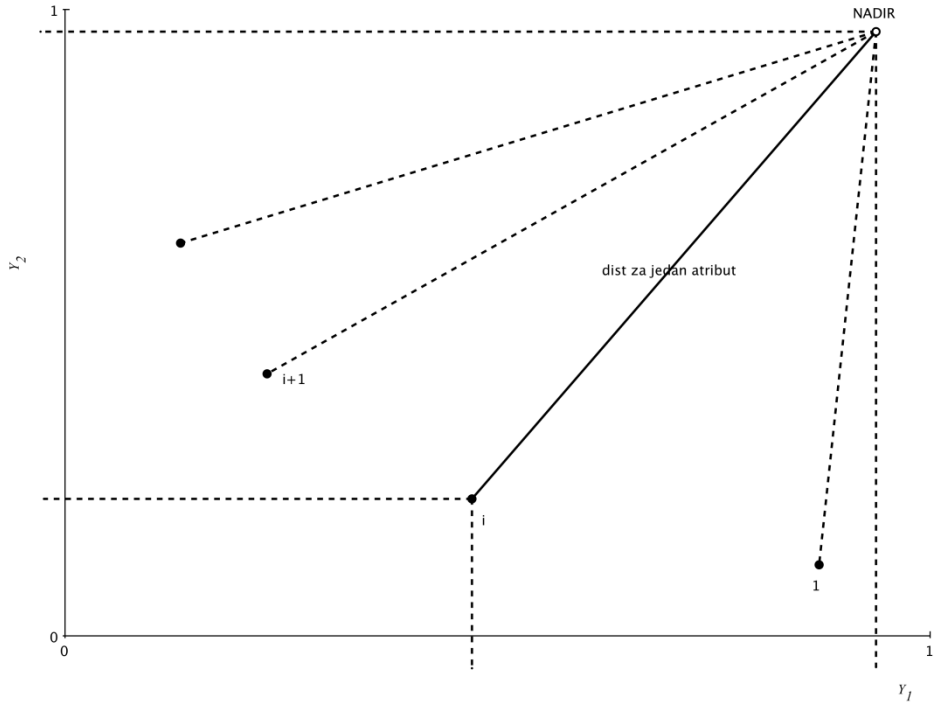
$$\max Dist = \max\left(\left|Y_i - Y_{refi}\right|\right);\ \ i = 1.....n \tag{3.24}$$



Sl. 3.8. Ilustracija indikatora vanjskog promjera Pareto fronte

3.3.1.8. Srednja NADIR udaljenost

Srednja NADIR udaljenost (*Average NADIR Distance*) je indikator kvalitete koji pokazuje ispupčenost Pareto fronte, odnosno srednju vrijednost udaljenosti od NADIR-a. Udaljenost pojedinog atributa od NADIR-a mjeri se Euklidskom metrikom



Sl. 3.9. Ilustracija indikatora srednje udaljenosti od NADIR-a

$$dist = \frac{\sum_{j=1}^{m} \sqrt{\sum_{i=1}^{n} \left(Y_i - Y_{iREF}\right)^2}}{m} \tag{3.25}$$

gdje su
j=1…..m - broj nedominiranih rješenja
i=1…..n - broj ciljeva

3.3.1.9. Omjer vanjskog promjera

Ovaj indikator kvalitete (*Outer Diameter Ratio*) mjeri odnos dvaju prije navedenih indikatora kvalitete:

$$NODR = \frac{\text{Srednja udaljenost od NADIR} - \text{a}}{\text{Vanjski promjer}} \tag{3.26}$$

3.3.1.10. Raspršenost (S)

Indikator kvalitete koji daje mjeru popunjenosti PF_{known} (Pareto fronte koja se mjeri). Izražen je kao srednja udaljenost između susjednih rješenja:

$$S = \sqrt{\frac{1}{\left|PF_{known}\right| - 1} \sum_{i=1}^{\left|PF_{known}\right|} (\bar{d} - d_i)^2} \tag{3.27}$$

$d_i = \min\left(\left|f_1^i(x) - f_1^j(x)\right| + \left|f_2^i(x) - f_2^j(x)\right|\right) \qquad i, j = 1.....n$

n = broj rješenja u PF_{known}

$\bar{d}$ = srednja vrijednost d_i

Ukoliko je S=0, to znači da su sva rješenja međusobno jednako udaljena. Ova mjera ne zahtijeva od korisnika poznavanje PF_{true}, ali se pretpostavlja da je MOEA konvergirao prema PF_{true}, prije izračuna indikatora.

3.3.1.11. Najveća greška Pareto Fronte (ME)

ME (*Maximum Pareto Front error*) je indikator kvalitete koji mjeri udaljenost dobivene Pareto fronte od referentne Pareto Fronte (Sl. 3.10). Ukoliko se za referentnu Pareto frontu izabere PF_{true}, ovaj indikator kvalitete mjeri koliko je udaljena PF_{known} od referentne PF_{true}. Odnosno, mjeri najveću minimalnu udaljenost između svakog rješenja u PF_{known} i odgovarajućeg najbližeg rješenja u PF_{true} .Izražava se izrazom:

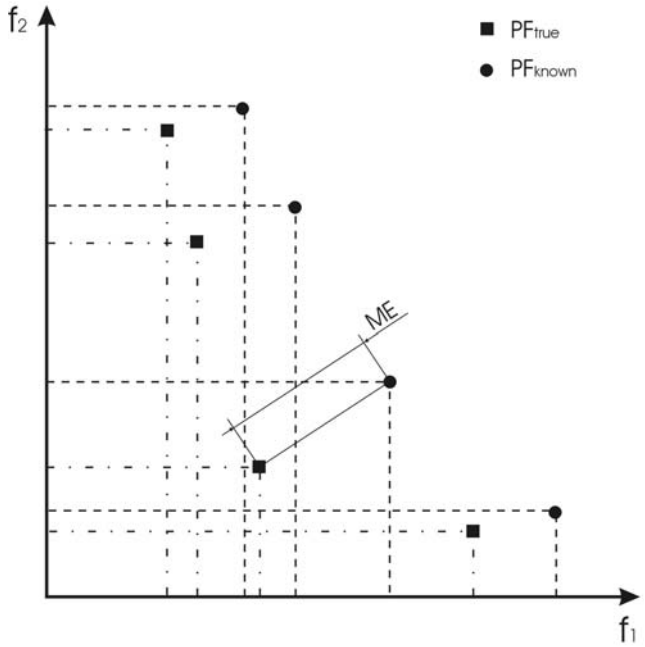
$$ME = \max_j \left\{ \left\{ \min_i \left(\sum \left| f_k^i(x) - f_k^j(x) \right|^p \right)^{\frac{1}{p}} \right\} \right\} \tag{3.28}$$

$i = 1.....\left|PF_{known}\right|$

$j = 1.....\left|PF_{true}\right|$

p – u većini slučajeva = 2 (standardna Euklidska udaljenost)

Rezultat ME = 0 označava $PF_{known} \subseteq PF_{true}$. Bilo koji drugi rezultat znači da barem jedno rješenje od PF_{known} nije u PF_{true}.



Sl. 3.10. Ilustracija indikatora najveće greške Pareto fronte

3.3.1.12. Epsilon indikator (I_ε)

Epsilon indikator kvantitativno definira za koliko je jedan skup rješenja lošiji od drugog, u odnosu na sve ciljeve. Indikator $I_\varepsilon(A,B)$ je jednak minimalnom faktoru ε, koji definira da postoji barem jedno rješenje u skupu *A*, koje nije lošije za iznos faktora ε po svim ciljevima za bilo koje rješenje u skupu *B*, tj. *A* slabo dominira *B*. Epsilon indikator je definiran kao:

$$I_{\varepsilon+}(A,B) = \inf_{e\in R}\left\{\forall z^2 \in B \exists z^1 \in A : z^1 \succeq_{\varepsilon+} z^2\right\} \tag{3.29}$$

gdje je $z^1 \succeq_\varepsilon z^2$, ako i samo ako

$$\forall 1 \le i \le n : z_i^1 \le \varepsilon \cdot z_i^2 \tag{3.30}$$

U praksi, vrijednost ε se može izračunati kao

$$I_\varepsilon(A,B) = \max_{z^2\in B}\min_{z^1\in A}\max_{1\le i\le n}\frac{z_i^1}{z_i^2} \tag{3.31}$$

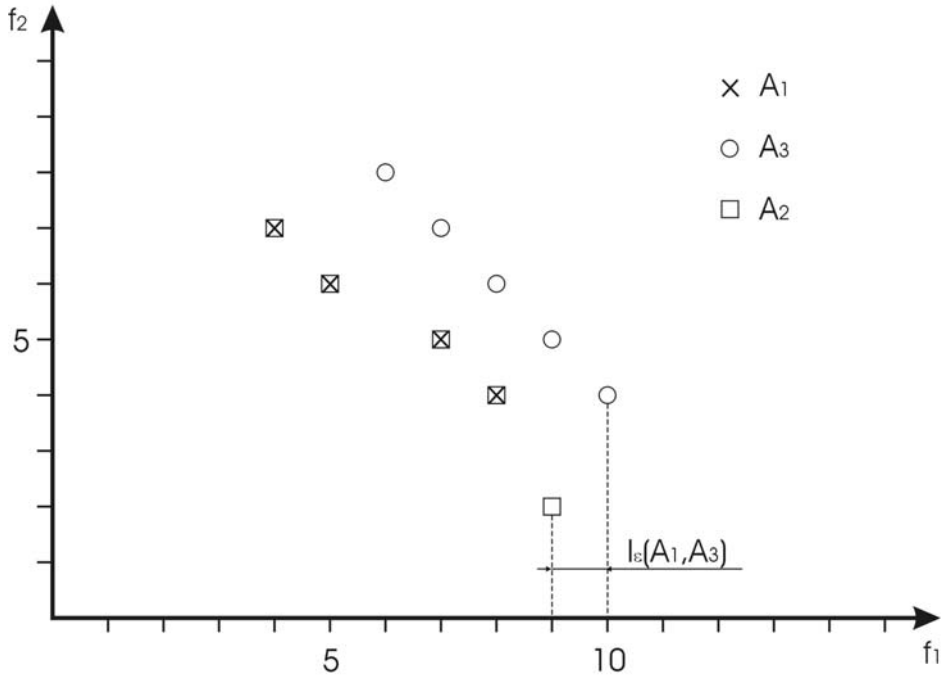
Analogijom, aditivni ε-indikator može se definirati kao,

$$I_{\varepsilon+}(A,B) = \inf_{e\in R}\left\{\forall z^2 \in B \exists z^1 \in A : z^1 \succeq_{\varepsilon+} z^2\right\} \tag{3.32}$$

gdje je $z^1 \succeq_{\varepsilon+} z^2$, ako i samo ako

$$\forall 1 \le i \le n : z_i^1 \le \varepsilon + z_i^2 \tag{3.33}$$

Na primjer, na Sl. 3.11 $I_\varepsilon(A_2, A_1) = 1$, $I_\varepsilon(A_2, A_3) = \frac{9}{10}$. Kod jednociljnih problema, $I_\varepsilon(A,B)$ je odnos dvije vrijednosti cilja, koje predstavljaju *A* i *B*.



Sl. 3.11. Ilustracija Epsilon indikatora

4. SUROGAT MODELIRANJE

U složenim inženjerskim sustavima metode analize mogu biti vremenski i računalno vrlo zahtjevne. U zrakoplovstvu i brodogradnji takvi proračuni mogu uključivati nelinearnu analizu strukture konačnim elementima, rješavanje nelaminarnih i nestacionarnih problema računalne mehanike fluida te proračune koji uključuju interakciju elastičnog tijela i fluida kroz koji se giba. Njihovo direktno korištenje unutar optimizacijskog procesa često nije moguće usprkos stalnom napretku računala, jer optimizacija zahtjeva više stotina ili čak tisuća analiza. Takvi problemi se mogu pokušati riješiti pojednostavljenim, često i linearnim, metodama niže razine točnosti, no u zadnje vrijeme se sve više koristi aproksimacijsko odnosno surogat modeliranje složenih metoda analize.

Sustavan prikaz raznih tehnika surogat-modeliranja, poput polinomske regresije (RS), kriginga, radijalnih baznih funkcija (RBF), neuronskih mreža, zajedno s tehnikama planiranja računalnih pokusa koje su neophodne pri generiranju surogata, dan je npr. u [27] te [28] i [29].

4.1. Uvod

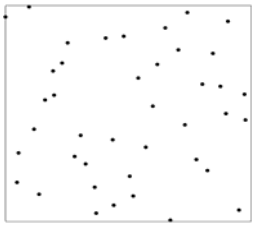
Surogat modeliranje zasniva se na korištenju statističkih metoda za izradu aproksimacije odnosno surogata skupih računalnih proračunskih modela. Osnovni koraci surogat modeliranja dani su na Sl. 4.1.

Planiranje računalnih pokusa obuhvaća odabir ili oblikovanje modela računalnih pokusa koji će dovoljno dobro opisati promatranu pojavu, uz što manji utrošak računalnih resursa. Pri tome je, u kontekstu primjene metode planiranja računalnih pokusa za surogat modeliranje, vrlo bitno koja je namjena surogat modela kao i koji surogat model se planira generirati. Više o planiranju računalnih pokusa u potpoglavlju 4.2.

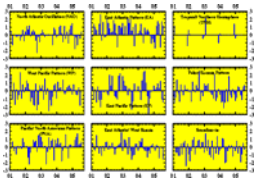
Izvođenje računalnih simulacija se izvodi prema odabranom planu pokusa iz prethodnog koraka, a rezultati se spremaju u neki oblik baze podataka na temelju koje može biti generiran surogat model.

Generiranje surogat modela obuhvaća generiranje odabranog surogat modela određene odzivne veličine na temelju podataka prikupljenih izvođenjem računalnih simulacija. Pri tome se pripadajućom metodologijom minimizira odstupanje između izvršenih računalnih pokusa i generiranog surogat modela. Detaljnije o surogat modelima i metodama minimizacije odstupanja od izvršenih pokusa u potpoglavlju 4.3.

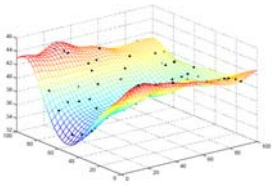
- planiranje računalnih pokusa



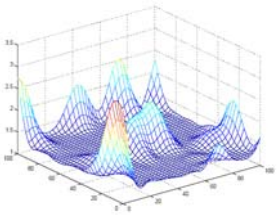
- izvođenje računalnih simulacija



- generiranje surogat modela



- validacija surogat modela



Sl. 4.1. Osnovni koraci surogat modeliranja

Validacija surogat modela omogućuje procjenu točnosti odnosno pouzdanosti surogat modela. Više o određivanju pouzdanosti surogat modela u potpoglavlju 4.3.4.

4.2. Metode planiranja računalnih pokusa

Plan pokusa predstavlja pokuse koje je potrebno izvesti pri čemu se kontrolni faktori (parametri, varijable) postavljaju na, tim planom, definirane vrijednosti. Drugim riječima to je plan pokusa u projektnom prostoru, to jest prostoru koji razapinju projektne varijable. Klasični planovi pokusa potječu iz teorije planiranja pokusa gdje je osnovni cilj izvođenje fizičkih eksperimenata. Kao što je već pojašnjeno u prethodnom potpoglavlju, fokus tih metoda je planiranje fizičkih pokusa za koje slučajna greška ima minimalan utjecaj na usvajanje ili odbacivanje postavljene hipoteze. Kako je kod računalnih pokusa slučajna pogreška jednaka nuli, jasno je da postoji mogućnost da pojedine metode uzorkovanje nisu pogodne za računalne pokuse, iako se može primijetiti se u mnogim istraživanjima ta činjenica zanemaruje. U nastavku će biti dan pregled najpopularnijih tehnika prema [30], [31] te [32]. Više o metodama planiranja pokusa može se pronaći u [33]. U [34] prikazana je metodologija za višekriterijsko adaptivno oblikovanje planiranja pokusa koja omogućuje odlučivanje o odabiru plana pokusa

uzimajući u obzir trošak i vrijeme potrebno za eksperimentiranje uz standardne kriterije kvalitete plana pokusa. Zbog malog broja radova iz ovog područja na hrvatskom jeziku, navedeni rad je uvažano spomenuti i zbog korištene terminologije koja je sukladna terminologiji što se koristi u nastavi sveučilišnih studija na Fakultetu strojarstva i brodogradnje.

Planovi pokusa se najčešće prikazuju matrično tako da redci (n) predstavljaju pojedine pokuse, dok su u stupcima vrijednosti pojedinih faktora (k). U nastavku će biti dani neke od najpopularnijih metoda za generiranje surogat modela. Pri tome će se dati primjeri kako bi izgledao plan pokusa za generiranje punog kvadratnog polinomskog surogat modela sa tri faktora. Za takav model minimalni broj pokusa za određivanje svih nepoznatih koeficijenata jest 10.

Prije opisa planova pokusa, važno je spomenuti njihove dvije važne karakteristike: ortogonalnost i rotabilnost.

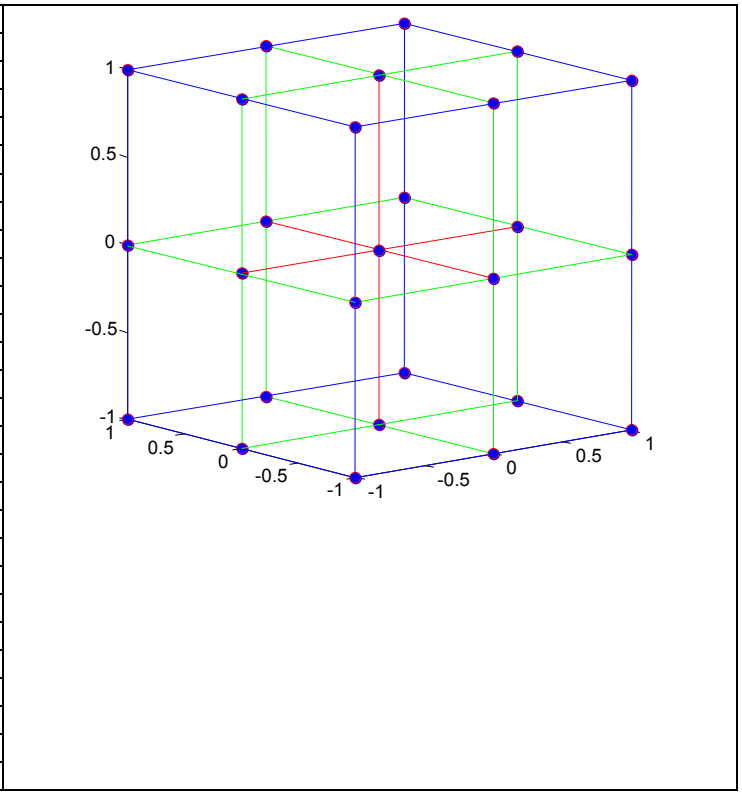
Ortogonalni plan pokusa je onaj plan pokusa kod kojeg su glavni efekti nezavisni, tj. nisu združeni. Ako s **X** označimo matricu pokusa, ortogonalan plan pokusa prvog reda jest onaj kojem je umnožak $\mathbf{X}^T\mathbf{X}$ dijagonalna matrica. Ako je matrica $\mathbf{X}^T\mathbf{X}$ dijagonalna, tada proizlazi da su stupci u matrici nezavisni i ortogonalni.

Rotabilan plan pokusa jest onaj kod kojega je vrijednost funkcije procijenjene varijance jednaka u svim točkama koje su jednako udaljene od centra plana pokusa. Ovo svojstvo je definirano iz potrebe dobivanja stabilnosti varijance kako bi se u eksperimentalnom prostoru dobila ujednačena kvaliteta kvadratnog polinomskog surogat modela. Trebalo bi naglasiti da je rotabilnost pokusa ili barem ortogonalnost moguće dosta jednostavno postići bez narušavanja ostalih važnih karakteristika pokusa. Više o ovim svojstvima moguće je pronaći u npr. [33] ili [34].

4.2.1. Puni faktorski plan pokusa

Puni faktorski plan pokusa (full factorial design) osnovni je faktorski plan pokusa. Broj pokusa je određen produktom broja nivoa svakog faktora. Najčešće su faktori na dva ili tri nivoa. Faktori s dva nivoa omogućuju određivanje glavnih efekata i interakcija, dok faktori na tri nivoa omogućuju određivanje glavnih efekata, interakcija i kvadratičnih efekata. S obzirom da se broj pokusa eksponencijalno povećava s povećanjem broja faktora, ovakav način uzorkovanja vrlo brzo dovodi do neprihvatljivog broja pokusa. Na Sl. 4.2 prikazan je puni faktorski plan uzoraka s tri faktora na 3 nivoa. Kako bi se lakše vizualizirao plan uzoraka rubni nivoi su spojeni plavim linijama, centralni bočni sa zelenim i centralni središnji s crvenim.

Pokus	F1	F2	F2
1	-1	-1	-1
2	0	-1	-1
3	1	-1	-1
4	-1	0	-1
5	0	0	-1
6	1	0	-1
7	-1	1	-1
8	0	1	-1
9	1	1	-1
10	-1	-1	0
11	0	-1	0
12	1	-1	0
13	-1	0	0
14	0	0	0
15	1	0	0
16	-1	1	0
17	0	1	0
18	1	1	0
19	-1	-1	1
20	0	-1	1
21	1	-1	1
22	-1	0	1
23	0	0	1
24	1	0	1
25	-1	1	1
26	0	1	1
27	1	1	1



Sl. 4.2. Puni faktorski plan uzoraka za tri faktora na tri nivoa

4.2.2. Djelomični faktorski planovi pokusa

Djelomični faktorski planovi pokusa (*fractional factorial design*) koriste se za veći broj parametara odnosno za skupe pokuse. Kod njih je ukupni broj pokusa samo dio punog faktorskog plana pokusa što podrazumijeva gubitak određenih informacija.

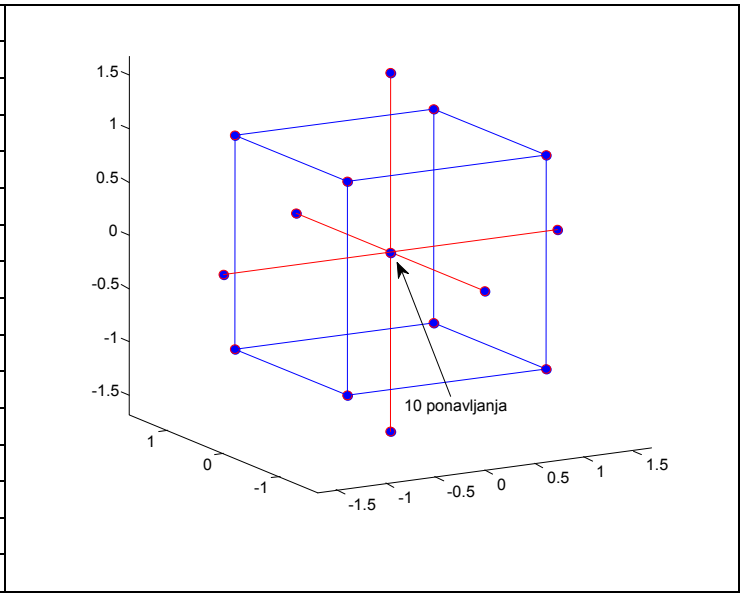
4.2.2.1. Centralni kompozitni plan pokusa

Za procjenu kvadratičnih efekata vrlo često se koriste centralni kompozitni plan pokusa (*central composite design*) koji je potpuno rotatabilan. Ovo je jedan od planova uzorkovanja pokusa namijenjen izvođenju fizičkih eksperimenata što se može vidjeti iz toga što se centralni pokus ponavlja više puta radi procjene slučajne greške. Stoga ga je za računalne pokuse moguće reducirati odnosno izvesti samo jedan centralni pokus. Naravno, ukoliko se plan uzorkovanja koristi u okviru analize koja uključuje neki oblik neizvjesnosti (uncertainty) podataka i dalje je potrebno koristiti izvorni oblik.

Na Sl. 4.3 prikazan je najčešće korišteni tzv. opisani centralni kompozitni plan uzoraka dobiven pomoću MATLAB-ove metode `ccdesign` ([35) s tri faktora na 5 nivoa. Kao što je napomenuto, pokusi 15-24 su s istom vrijednosti faktora. Vidljivo je da nivoi faktora nisu jednako udaljeni odnosno da su pokusi 9-14 imaju vrijednosti 1.6818 umjesto standardnih 1.5 a

svi planirani uzorci su jednako udaljeni od centra (0,0,0) te se u ovom slučaju svi planirani pokusi nalaze na oplošju sfere polumjera 1.6168.

Pokus	F1	F2	F2
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	1	-1	-1
6	1	-1	1
7	1	1	-1
8	1	1	1
9	-1.6818	0	0
10	1.6818	0	0
11	0	-1.6818	0
12	0	1.6818	0
13	0	0	-1.6818
14	0	0	1.6818
15-24	0	0	0

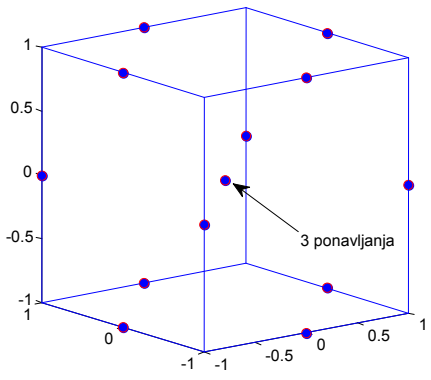


Sl. 4.3. Centralni kompozitni plan uzoraka

4.2.2.2. Box – Behnken plan pokusa

Kao i centralni kompozitni plan uzorkovanja ovaj plan uzorkovanja služi za generiranje punog kvadratnog polinomskog surogat modela. Također je namijenjen prvenstveno za fizičke pokuse te se centralni uzorak ponavlja nekoliko puta, no ipak zahtijeva nešto manje pokusa od centralnog kompozitnog plana. Na Sl. 4.4 prikazan je Box – Behnken plan uzoraka dobiven pomoću MATLAB-ove metode `bbdesign` s tri faktora na 3 nivoa. Kao što je napomenuto pokusi 13-15 su s istom vrijednosti faktora. Vidljivo je da su pokusi koncentrirani na središnjoj razini, te da niti jedan uzorak nema sva tri faktora na rubnim nivoima.

Pokus	F1	F2	F2
1	-1	-1	0
2	-1	1	0
3	1	-1	0
4	1	1	0
5	-1	0	-1
6	-1	0	1
7	1	0	-1
8	1	0	1
9	0	-1	-1
10	0	-1	1
11	0	1	-1
12	0	1	1
13-15	0	0	0



Sl. 4.4. Box – Behnken plan uzoraka

4.2.2.3. Ortogonalna polja (Taguchijevi planovi pokusa)

Ortogonalna polja (orthogonal arrays) djelomičan su faktorski plan pokusa. U literaturi se često nalaze i pod imenom Taguchijeva ortogonalna polja. Kao što im ime govori ovaj plan pokusa je ortogonalan, a još jedna od bitnih karakteristika im je da značajno reduciraju ukupni broj pokusa u odnosu na puni faktorski plan. Prvenstveno su namijenjeni za analizu varijance glavnih efekata bez interakcija no ponekad se koriste i za izradu linearnih ili čisto kvadratičnih polinomskih surogat modela. Često se koriste i za analizu robusnosti Taguchijevom metodom [36]. Jedna od najvećih biblioteka ortogonalnih polja dostupnih na webu nalazi se na internet adresi http://www2.research.att.com/~njas/oadir/. U Tbl. 4.1 prikazane su najčešće korištena ortogonalna polja. Pri tome je u prvom retku zapisan broj faktora, a u prvom stupcu broj pokusa. Ispod imena svakog ortogonalnog polja zapisano je koliko faktori imaju razina. Tako npr. ortogonalno polje L9 definira 9 pokusa za 4 faktora na 3 razine.

Tbl. 4.1 Najčešće korištena ortogonalna polja

n\k	**3**	**4**	**5**	**6**	**7**	**10**	**11**	**12**	**13**	**15**	**21**	**22**	**26**	**27**	**31**	**40**	**63**
4	L4 2																
8					L8 2												
9		L9 3															
12							L12 2										
16			L16b 4							L16 2							
18					L18 3,6												
25				L25 5													
27									L27 3,2								
32						L32b 4,2									L32 2		
36												L36 3		L36b 3,2			
50								L50 5,2									
54													L54 3,2				
64											L64b 2						L64 4
81																L81 3	

4.2.3. Optimalni planovi pokusa

Ne postoji uvijek sloboda izbora nivoa faktora prema prethodno opisanim planovima pokusa. Često izbor nivoa faktora može biti ograničen ili različit za pojedine čimbenike (planovi s mješovitim brojem razina). Primjerice za neke faktore se zna da su potrebni samo linearni članovi, dok za druge trebaju i kvadratni i interakcijski članovi polinomskog surogat modela. Za takve sustave vrlo je praktično generirati optimalni plan pokusa za dane uvijete upotrebom iterativnog algoritma Danas, komercijalni statistički programski paketi sadrže mogućnost generiranja takvih planova, koji iz ukupnog skupa kandidata (vrlo često je to puni faktorski plan pokusa) odaberu optimalni plan pokusa. U usporedbi s klasičnim planovima pokusa, matrice optimalno generiranih planova najčešće nisu ortogonalne i procijene utjecaja mogu biti korelirane, međutim predstavljaju optimum prema odabranim kriterijima.

Prema [34], kada se govori o kriterijima po kojima se oblikuje ili odabire plan pokusa najčešće se misli na abecedne (*alphabetic, lexicographic*) kriterije optimalnosti. Najčešće korišteni kriteriji optimalnosti su oni koji su bazirani na vrijednosti matrice $\mathbf{X}^T\mathbf{X}$, a to su D kriterij i A-kriterij. U drugu skupinu kriterija optimalnosti moguće je svrstati one kriterije koji su bazirani na vrijednosti funkcije procijenjene varijance. Najčešće upotrebljavani kriteriji druge skupine su G-kriterij i Q-kriterij. D-optimal plan pokusa je u literaturi, s područja izrade surogat modela računalnih simulacija, najraširenij te će ukratko biti opisan u nastavku.

<u>D-optimal</u>

Za plan pokusa se kaže da je D-optimalan ako mu se minimizira determinanta recipročne vrijednosti produkta matrice $\mathbf{X}^T\mathbf{X}$, odnosno:

$$\left|\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right| \tag{4.1}$$

Stoga mjera relativne efikasnosti plana pokusa 1 u odnosu na plan pokusa 2 prema D-kriteriju može biti zapisana kao:

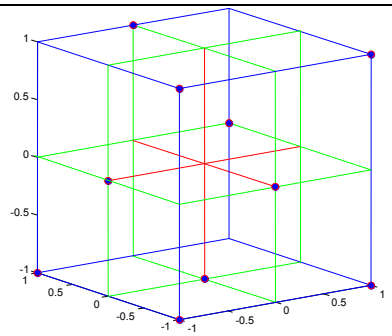
$$D_e = \left(\left|\frac{\left(\mathbf{X}_1^T\mathbf{X}_1\right)^{-1}}{\left(\mathbf{X}_2^T\mathbf{X}_2\right)^{-1}}\right|\right)^{\frac{1}{n}} \tag{4.2}$$

gdje je n broj faktora.

Na Sl. 4.4 prikazan je D-optimalni (odnosno lokalno D-optimalni) plan pokusa dobiven pomoću MATLAB-ove funkcije `cordexch` s tri faktora na 3 nivoa. Pri tome funkcija broj nivoa određuje automatski temeljem zadanog željenog surogat modela, dok je moguće odabrati

željeni broj eksperimenata. Naravno, broj eksperimenata mora biti dovoljan za određivanje nepoznatih koeficijenata surogat modela.

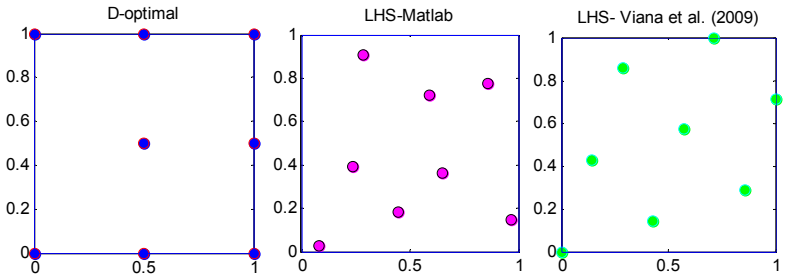
Pokus	F1	F2	F2
1	-1	0	0
2	-1	-1	1
3	-1	1	-1
4	0	1	1
5	0	-1	0
6	1	-1	1
7	1	-1	-1
8	-1	-1	-1
9	1	1	0
10	0	0	-1



Sl. 4.5. D- optimalni plan uzoraka s tri faktora za puni kvadratni polinomski surogat model

4.2.4. Popunjavajuće metode uzorkovanja

Kao što je navedeno u preglednim člancima [30] i [31] mnogi znanstvenici se zalažu za upotrebu popunjavajućih planova pokusa (*space filling designs*) deterministički određenih računalnih eksperimenata. Blokiranje, ponavljanje i randomizacija pokusa su potpuno nebitni kod računalnih pokusa. Za interpolirajuće surogat modele poput Kriginga i RBF-a popunjavajuće metode daju značajno bolju globalnu točnost u okviru zadane domene zbog toga što klasični planovi pokusa čak i kad se izuzme blokiranje, ponavljanje i randomizacija pokusa imaju tendenciju uzimanja rubnih nivoa koji su najčešće na rubu promatrane domene. Na Sl. 4.6 prikazana je razlika između klasičnog D-optimalnog plana pokusa s 2 faktora i 8 pokusa s dva tipa popunjavanja metodom latinskih hiperkocki (LHS-Latin hypercube sampling *ili LHD* -Latin hypercube design). U nastavku će biti rečeno nešto više o najčešće korištenoj popunjavajućoj metodi uzorkovanja – LHS-u.



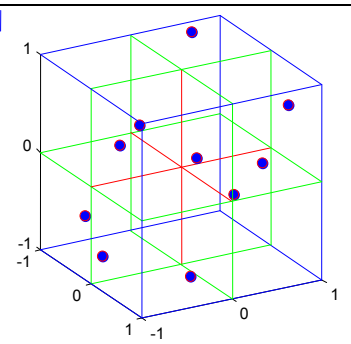
Sl. 4.6. Usporedba klasičnog D- optimalnog plan pokusa s dva popunjavajća LHS plana pokusa

4.2.4.1. Plan pokusa latinskim hiperkockama (LHS)

Prema [37], plan latinskim hiperkockama nudi brojne pozitivne karakteristike kao što su: ortogonalnost matrice pokusa, slobodan odabir broja dimenzija kao i broja pokusa te neovisnost o tipu surogat modela koji će biti generiran. LHS s n uzoraka i k faktora konstruira se

tako da se svaki od k faktora podijeli na n jednakih nivoa čime se generira hiperkocki a u samo n hiperkocki određenom procedurom generira se jedan uzorak. Uzorke je moguće generirati koristeći generator slučajnih brojeva no time se ne garantira uniformnost odabranih uzoraka u zadanoj domeni. Postoje brojne metode, od kojih su neke na bazi kombinatorijske optimizacije, koje odabiru uzorke najčešće pokušavajući maksimizirati radijus najmanje hipersfere koju je moguće generirati a da ne obujmi niti jedan uzorak. Jedan od vremenski vrlo učinkovitih, a sa stajališta uniformnosti vrlo uspješnih algoritama jest metoda s translatornom propagacijom (*LHTPA- Latin hypercube via translational propagation algorithm*) objašnjen u [37]. Na Sl. 4.6 može se vidjeti razlika između LHS plana pokusa generiranog Matlabovom funkcijom `lhsdesign` i plana pokusa dobivenog pomoću Matlab programskog koda preuzetog iz navedenog članka. Na Sl. 4.7 prikazan je LHTPA plan pokusa s tri faktora dobiven prema [37].

Pokus	F1	F2	F2
1	-0.5556	-0.5556	-1
2	0.7778	-0.3333	-0.7778
3	-0.3333	0.7778	-0.5556
4	-0.1111	-1	-0.3333
5	-1	-0.1111	-0.1111
6	0.1111	0.1111	0.1111
7	1	0.3333	0.3333
8	0.3333	1	0.5556
9	0.5556	-0.7778	0.7778
10	-0.7778	0.5556	1



Sl. 4.7. LHTPA- plan uzoraka s tri faktora za puni kvadratni polinomski surogat model

4.3. Surogat modeli

Vrijednost neke odzivne veličine dobivene računalnom simulacijom može se izraziti kao

$$y(\mathbf{x}) = \hat{y}(\mathbf{x}) + \varepsilon \tag{4.3}$$

gdje su: x_i - nezavisni faktori,

ε – greška modela,

$\hat{y}(\mathbf{x})$ - surogat model odzivne veličine.

Kod deterministčkih računalnih simulacija, ε jest samo greška aproksimacije i ne uključuje grešku mjerenja odnosno slučajnu grešku.

4.3.1. Odzivne površine - polinomski surogat modeli

Odzivne površine (*RS - response surfaces*) ili polinomski surogat modeli jedni su od najčešće korištenih surogat modela pri modeliranju deterministčkih računalnih simulacija. Razlog tome je i što je metodologija primjene polinomskih surogat modela pri analizi fizičkih pokusa detaljno razvijena i korištena u izuzetno širokom spektru znanstvenih područja. U

literaturi se često nalaze i pod nazivom metoda ili metodologija odzivnih površina (*RSM* - response surface methodology). Metodologija odzivne površine zapravo jest sažetak matematičkih i statističkih metoda kojima se modeliraju i analiziraju utjecaji nekoliko faktora (nezavisne varijable) na promatrani odziv [34]. Iz te definicije je vidljivo da RSM obuhvaća mnogo više od samog generiranja polinomskog surogat modela. U nekim radovima pojam RSM se zapravo koristi kao termin za surogat modeliranje općenito.

Najjednostavniji polinomski surogat model neke veličine *y* jest linearni, a može se izraziti kao:

$$\hat{y}_{RS-L}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i \tag{4.4}$$

gdje su: β_0, $\beta_{i=1,\ldots k}$ - nepoznati koeficijenti polinoma,

x_i - nezavisni faktori.

Matrični oblik ovog modela može se zapisati kao:

$$\hat{y} = \mathbf{B}\boldsymbol{\beta} \tag{4.5}$$

Gdje su matrice **B** i $\boldsymbol{\beta}^T$ za linearni model:

$$\mathbf{B} = \begin{bmatrix} 1 & x_1 \ldots & x_i \ldots & x_k \end{bmatrix} \tag{4.6}$$

$$\boldsymbol{\beta}^T = \begin{Bmatrix} \beta_0 & \beta_1 \ldots & \beta_i \ldots & \beta_k \end{Bmatrix} \tag{4.7}$$

Na sličan način može se izraziti i često korišteni kvadratni surogat model:

$$\hat{y}_{RS-kvad}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} x_i x_j \tag{4.8}$$

$$\mathbf{B} = [1 \quad x_1 \ldots \quad x_i \ldots \quad x_k \quad x_{11} \quad x_{12} \ldots \quad x_{ij} \ldots \quad x_{kk}] \tag{4.9}$$

$$\boldsymbol{\beta}^T = \begin{Bmatrix} 1 & \beta_1 \ldots & \beta_i \ldots & \beta_k & \beta_{11} & \beta_{12} \ldots & \beta_{ij} \ldots & \beta_{kk} \end{Bmatrix} \tag{4.10}$$

Iako su ovo najčešće korišteni polinomski modeli, na sličan način kako su izraženi linearni i kvadratni model, moguće je izraziti općeniti model u kojem umjesto nezavisnih članova polinoma x_i ili $x_i x_j$ egzistiraju takozvane bazne funkcije B(**x**). Prema tome općeniti izraz za model odzivnih površina glasi:

$$\hat{y}_{RS}(\mathbf{x}) = \sum_{i=1}^{b} \beta_i B_i \tag{4.11}$$

gdje je *b* broj baznih funkcija.

U matričnom obliku općeniti izraz za model odzivnih površina glasi:

$$\hat{y}_{RS} = \mathbf{B}\boldsymbol{\beta} \tag{4.12}$$

Gdje su matrice **B** i $\boldsymbol{\beta}^{\mathrm{T}}$:

$$\mathbf{B} = \left[B_1(\mathbf{x})\ldots \quad B_i(\mathbf{x})\ldots \quad B_b(\mathbf{x})\right] \tag{4.13}$$

$$\boldsymbol{\beta}^T = \left[\beta_1\ldots \quad \beta_i\ldots \quad \beta_b\right] \tag{4.14}$$

Naravno, kako bi se mogao generirati surogat model odzivnih površina potrebno je odrediti nepoznate koeficijente koji se nalaze u vektoru **β**. Ukoliko je prema odabranom planu pokusa izvršeno *n* računalnih simulacije i vrijednosti odzivne veličine spremljene u vektor $\mathbf{y}_{1\text{-}n}$, a vrijednosti nezavisnih faktora u matricu $\mathbf{x}_{1\text{-}n}$ moguće je prvo izračunati vrijednosti svih baznih funkcija $B_{i1\text{-}n}$ te je u tom slučaju regresijska matrica $\mathbf{B}_{1\text{-}n}$:

$$\mathbf{B}_{1-n} = \begin{bmatrix} B_1(\mathbf{x}_1)\ldots & B_i(\mathbf{x}_1)\ldots & B_b(\mathbf{x}_1) \\ B_1(\mathbf{x}_2)\ldots & B_i(\mathbf{x}_2)\ldots & B_b(\mathbf{x}_2) \\ B_1(\mathbf{x}_n)\ldots & B_i(\mathbf{x}_n)\ldots & B_b(\mathbf{x}_n) \end{bmatrix} \tag{4.15}$$

Ukupni raspisani matrični oblik može se zapisati kao:

$$\mathbf{y}_{1-n} = \begin{Bmatrix} y_1(\mathbf{x}_1) \\ y_j(\mathbf{x}_2) \\ y_n(\mathbf{x}_n) \end{Bmatrix} = \begin{bmatrix} B_1(\mathbf{x}_1)\ldots & B_i(\mathbf{x}_1)\ldots & B_b(\mathbf{x}_1) \\ B_1(\mathbf{x}_j)\ldots & B_i(\mathbf{x}_j)\ldots & B_b(\mathbf{x}_j) \\ B_1(\mathbf{x}_n)\ldots & B_i(\mathbf{x}_n)\ldots & B_b(\mathbf{x}_n) \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_i \\ \beta_b \end{Bmatrix} + \varepsilon = \mathbf{B}_{1-n}\boldsymbol{\beta} + \varepsilon \tag{4.16}$$

Nepoznati koeficijenti se najčešće računaju tako da se minimizira suma kvadratna razlike vektora poznatih odziva i umnoška regresijske matrice s vektorom koeficijenata:

$$\min\left(\left\|\mathbf{y}_{1-n} - B_{1-n}\boldsymbol{\beta}\right\|^2\right) \tag{4.17}$$

što je poznati matematički problem minimiziranja sume kvadrata odstupanja te se izraz (4.17) može zapisati kao:

$$\boldsymbol{\beta} = \left(\mathbf{B}_{1-n}^T\mathbf{B}_{1-n}\right)^{-1}\mathbf{B}_{1-n}^T\mathbf{y}_{1-n} \tag{4.18}$$

Uvjet za korištenje izraza (4.18) jest da matrica umnoška $\mathbf{B}_{1-n}^T\mathbf{B}_{1-n}$ nije singularna.

Kao što je vidljivo iz izraza (4.13) i (4.14) surogat model odzivnim površinama se lako proširuje baznim funkcijama odgovarajućeg reda, no korištenje polinoma višeg reda vrlo brzo povećava broj nepoznatih koeficijenata koje je potrebno odrediti.

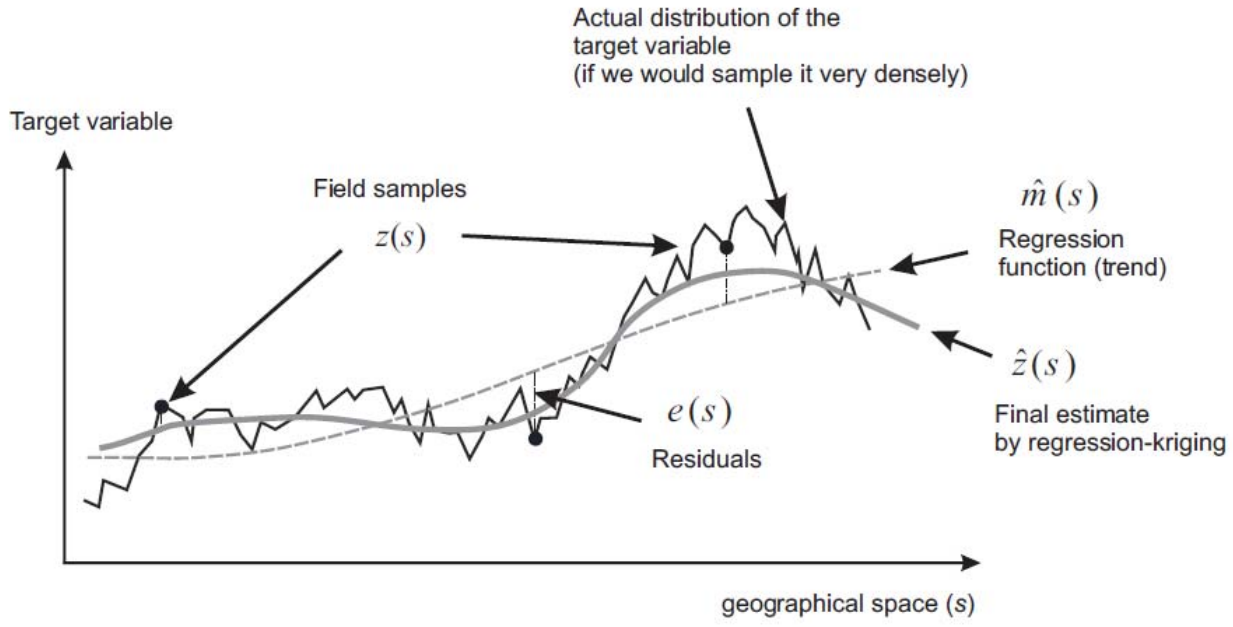
Još jedan od problema je što polinomi koji se tradicionalno koriste za polinomsku regresiju nisu ortogonalni što znači da mogu biti kolinearni. Umjesto klasičnih polinomskih

baznih funkcija mogu se koristiti bazne funkcije kojima je garantirana ortogonalnost u određenom intervalu. Tako se na primjer često koriste Legendrovi polinomi koji su ortogonalni u intervalu [0,1] (vidi npr. [27]).

4.3.2. Kriging

Kriging je geostatistička interpolacijska metoda [38] imenovana prema južnoafričkom inženjeru D, G. Krigeu koji ju je predložio i upotrijebio u okviru svog magistarskog rada [39] za procjenu koncentracije rudače u ležištima zlata. Primjena geostatističkih metoda danas je izašla daleko izvan okvira geoloških prostornih procesa i primjenjuje se u širokom spektru znanstvenih disciplina gdje je potrebno obrađivati prostorne podatke.

Kriging je sistematično uveden u modeliranje računalnih pokusa radovima [28] i [29]. Od tada je Kriging postao jedna od najčešće korištenih metoda surogat modeliranja računalnih simulacija te je moguće pronaći brojne radove s raznim unaprjeđenjima i primjenama. Neke od tih unaprjeđenja i primjena moguće je pronaći u preglednom radu [40]. Od podvrsta kriginga najčešće se koristi obični, jednostavni, i trend kriging (kriging s trendom, ranije nazivan univerzalni kriging). Postoje još i slične tj. izvedene metode kao kokriging, disjunktivni, indikatorski kriging, višestruko-indikatorski, itd.



Sl. 4.8. Modeliranje slučajnog procesa krigingom [29]

Opća jednadžba kriginga, poznata kao univerzalni kriging model, sastoji se od dva dijela, determinističkog i stohastičkog te se može se izraziti na sljedeći način:

$$\hat{y}_{KRG}(\mathbf{x}) = \mu(\mathbf{x}) + e(\mathbf{x}) \tag{4.19}$$

odnosno,

$$\hat{y}_{KRG}(\mathbf{x}) = \sum_{i=1}^{b} \beta_i B_i(\mathbf{x}) + z(\mathbf{x}) \tag{4.20}$$

Prvi, deterministički dio, modelira kretanje srednje vrijednosti slučajnog procesa i naziva se još i trend, dok drugi, stohastički dio, modelira ostatak (residuals) kao stacionarni slučajni Gaussijanski proces sa očekivanom (srednjom) vrijednošću nula, varijancom σ^2 i funkcijom kovarijance C.

Procjena stohastičkog dijela procesa $z(\mathbf{x})$ dobiva se umnoškom težinskih faktora $\boldsymbol{\lambda}$ i poznatih vrijednosti $\mathbf{z}_{1\text{-}n}$ u n kontrolnih točaka $\mathbf{x}_{1\text{-}n}$ dobivenim računalnim simulacijama

$$z(\mathbf{x}) = \boldsymbol{\lambda}(\mathbf{x}_{1-n} - \mathbf{x}) \cdot \mathbf{z}_{1-n}(\mathbf{x}_{1-n}) \tag{4.21}$$

Kao što je vidljivo iz izraza (4.21) težinski faktori ovise o udaljenosti promatrane lokacije definirane vektorom $\mathbf{x}$ od matrice lokacija $\mathbf{x}_{1\text{-}n}$ s poznatim odzivom. Iz ovoga je vidljivo da je jedina nepoznanica vektor težinskih faktora. U nastavku će biti opisani glavni izrazi, za tri najčešće vrste kriginga, koji određuju vektor nepoznatih težinskih faktora za lokaciju u kojoj želimo procijeniti odziv.

4.3.2.1. Jednostavni kriging

Srednja vrijednost je poznata konstanta, a krajnje jednadžbe glase:

$$\boldsymbol{\lambda}_{1-n}(\mathbf{x}_{1-n}, \mathbf{x}) = \boldsymbol{\Gamma}^{-1}(\mathbf{x}_{1-n}) \cdot \boldsymbol{\gamma}(\mathbf{x}_{1-n}, \mathbf{x}) \tag{4.22}$$

Gdje je matrica $\boldsymbol{\Gamma}$:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma(\mathbf{x}_1 - \mathbf{x}_1) & \gamma(\mathbf{x}_1 - \mathbf{x}_2) & \ldots & \gamma(\mathbf{x}_1 - \mathbf{x}_n) \\ \gamma(\mathbf{x}_2 - \mathbf{x}_1) & \gamma(\mathbf{x}_2 - \mathbf{x}_2) & \ldots & \gamma(\mathbf{x}_2 - \mathbf{x}_n) \\ . & . & . & . \\ \gamma(\mathbf{x}_n - \mathbf{x}_1) & \gamma(\mathbf{x}_n - \mathbf{x}_2) & \ldots & \gamma(\mathbf{x}_n - \mathbf{x}_n) \end{bmatrix} \tag{4.23}$$

i gdje je vektor $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma} = \begin{Bmatrix} \gamma(\mathbf{x}_1 - \mathbf{x}) & \gamma(\mathbf{x}_2 - \mathbf{x}) & \ldots & \gamma(\mathbf{x}_n - \mathbf{x}) \end{Bmatrix} \tag{4.24}$$

Iz izraza (4.22) do (4.24) vidljivo je se kriging surogat model zapravo sastoji od matrica $\boldsymbol{\Gamma}^{-1}$ i vektora $\boldsymbol{\gamma}$. Pri tome je priprema matrice $\boldsymbol{\Gamma}^{-1}$ računalno zahtjevna, no potrebno ju je pripremiti

samo jednom za jedan set lokacija $\mathbf{X}_{1\text{-}n}$. S druge strane vektor $\boldsymbol{\gamma}$ je vrlo jednostavan za pripremu no potrebno ga je pripremiti za svaku lokaciju za koju se želi procijeniti odzivna veličina.

Pri tome ne treba zaboraviti da je jedan od računalno najzahtjevnijih zadataka odrediti parametre modela procjene variograma $\gamma(\mathbf{x}_i\text{-}\mathbf{x}_j)$.

4.3.2.2. Obični kriging

Srednja vrijednost je nepoznata konstanta, a za izračun težinskih koeficijenata koristi se također izraz (4.22) s tim da se matrica $\boldsymbol{\Gamma}$ i vektor $\boldsymbol{\gamma}$ pripremaju malo drugačije:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma(\mathbf{x}_1-\mathbf{x}_1) & \gamma(\mathbf{x}_1-\mathbf{x}_2) & \dots & \gamma(\mathbf{x}_1-\mathbf{x}_n) & 1 \\ \gamma(\mathbf{x}_2-\mathbf{x}_1) & \gamma(\mathbf{x}_2-\mathbf{x}_2) & \dots & \gamma(\mathbf{x}_2-\mathbf{x}_n) & 1 \\ . & . & . & . & 1 \\ \gamma(\mathbf{x}_n-\mathbf{x}_1) & \gamma(\mathbf{x}_n-\mathbf{x}_2) & \dots & \gamma(\mathbf{x}_n-\mathbf{x}_n) & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \tag{4.25}$$

$$\boldsymbol{\gamma} = \begin{Bmatrix} \gamma(\mathbf{x}_1-\mathbf{x}) & \gamma(\mathbf{x}_2-\mathbf{x}) & \dots & \gamma(\mathbf{x}_n-\mathbf{x}) & 1 \end{Bmatrix} \tag{4.26}$$

Dodatni (zadnji) član vektora težinskih faktora $\boldsymbol{\lambda}$ jest upravo nepoznata srednja vrijednost procesa.

4.3.2.3. Trend kriging ili univerzalni kriging

Srednja vrijednost je nepoznata kombinacija poznatih baznih funkcija:

$$\mu(\mathbf{x}) = \sum_{i=1}^{b} \beta_i B_i(\mathbf{x}) \tag{4.27}$$

S obzirom da je u ovom slučaju potrebno odrediti nepoznate koeficijente vektora $\boldsymbol{\beta}$ vezanih uz odabrane bazne funkcije $\mathbf{B}$, matrica $\boldsymbol{\Gamma}$ i vektor $\boldsymbol{\gamma}$ uključuju te članove:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_J & \mathbf{B}_{1-n} \\ \mathbf{B}_{1-n}^T & \mathbf{0} \end{bmatrix} \tag{4.28}$$

$$\boldsymbol{\gamma}^T = \begin{Bmatrix} \boldsymbol{\gamma}_J^T & \mathbf{m}^T \end{Bmatrix} \tag{4.29}$$

Gdje je $\boldsymbol{\Gamma}_j = \boldsymbol{\Gamma}$ jednostavnog kriginga definirana izrazom (4.23), dok je $\boldsymbol{\gamma}_j = \boldsymbol{\gamma}$ jednostavnog kriginga definiran izrazom (4.24). Isto tako regresijska matrica $\mathbf{B}_{1\text{-}n}$ jest upravo matrica definirana za model odzivnih površina izrazom (4.15), dok je pripadajući vektor $\mathbf{m}$ vektor lagrangeovih množitelja**.**

Pri tome treba imati na umu da se model variograma $\gamma(\mathbf{x}_i\text{-}\mathbf{x}_j)$. treba računati za vrijednosti od kojih je oduzet trend. S obzirom da to znači da u matrici $\boldsymbol{\Gamma}_j$ egzistiraju nepoznanice

očigledno je da se direktnim korištenjem izraza (4.22) ne može dobiti vektor nepoznatih težinskih koeficijenata.

$$\boldsymbol{\lambda}^T_{trend}\left(\mathbf{x}_{1-n},\mathbf{x}\right)=\left\{\boldsymbol{\lambda}^T_{1-n}\left(\mathbf{x}_{1-n},\mathbf{x}\right)\quad \mathbf{m}^T\left(\mathbf{x}_{1-n}\right)\right\} \tag{4.30}$$

Najjednostavniji način je da se prvo odrede nepoznati koeficijenti **β** na isti način kao kod odzivnih površina tj. korištenjem izraza (4.18). Tada se može izračunati $\boldsymbol{\Gamma}_j$, a nakon toga i vektor nepoznatih težinskih koeficijenata. Takav postupak se u literaturi najčešće naziva običnim algoritmom najmanjih kvadrata (OLS – ordinary least squares). Problem ovog postupka je što nije garantirano da dio izračunat odzivnim površinama odnosno polinomskom regresijom opisuje samo srednju vrijednost, što znači da nije niti garantirano da težinski faktori obuhvaćaju samo ostatke (*residuals*). U literaturi se mogu pronaći metode poput generalizirane metode najmanjih kvadrata odstupanja (*FGLS – Feasible Generalized Least Squares*), npr. [41], koje generiraju točniji surogat model.

4.3.3. Radijalne bazne funkcije

Radijalne bazne funkcije (*RBF*-radial basis functions) jest metoda surogat modeliranja s egzaktnom interpolacijom podataka u multivarijabilnom prostoru [42]. Generalni izraz za aproksimaciju odzivne veličine y glasi:

$$\hat{y}_{RBF}\left(\mathbf{x}\right)=\sum_{i=1}^{b}\beta_i B_i\left(\mathbf{x}\right)+\sum_{j=1}^{n}\lambda_j\left(\mathbf{x}\right)\phi_j\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right) \tag{4.31}$$

ili u matričnom zapisu:

$$\hat{y}_{RBF}=\mathbf{B}\boldsymbol{\beta}+\boldsymbol{\phi}\boldsymbol{\lambda} \tag{4.32}$$

Gdje su matrice **X** i vektor **β** definirane izrazima (4.13) i (4.14) dok su matrica **ϕ** i vektor **λ** definirani sljedećim izrazima:

$$\boldsymbol{\phi}=\left[\phi_1\left(\left\|\mathbf{x}-\mathbf{x}_1\right\|\right)\ldots\quad\phi_j\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)\ldots\quad\phi_n\left(\left\|\mathbf{x}-\mathbf{x}_n\right\|\right)\right] \tag{4.33}$$

$$\boldsymbol{\lambda}^T=\left\{\lambda_1\ \ldots\quad\lambda_j\ldots\quad\lambda_n\right\} \tag{4.34}$$

U prethodnim izrazima sa simbolom ϕ_j označene su radijalne bazne funkcije s centrima u pripadajućoj izračunatoj lokaciji $\mathbf{x}_j$. Svaka bazna funkcija definira ovisnost o Euklidskoj udaljenosti od centra bazne funkcije $\left\|\mathbf{x}-\mathbf{x}_j\right\|$. Sljedećih nekoliko izraza definira neke od najčešće korištenih baznih funkcija:

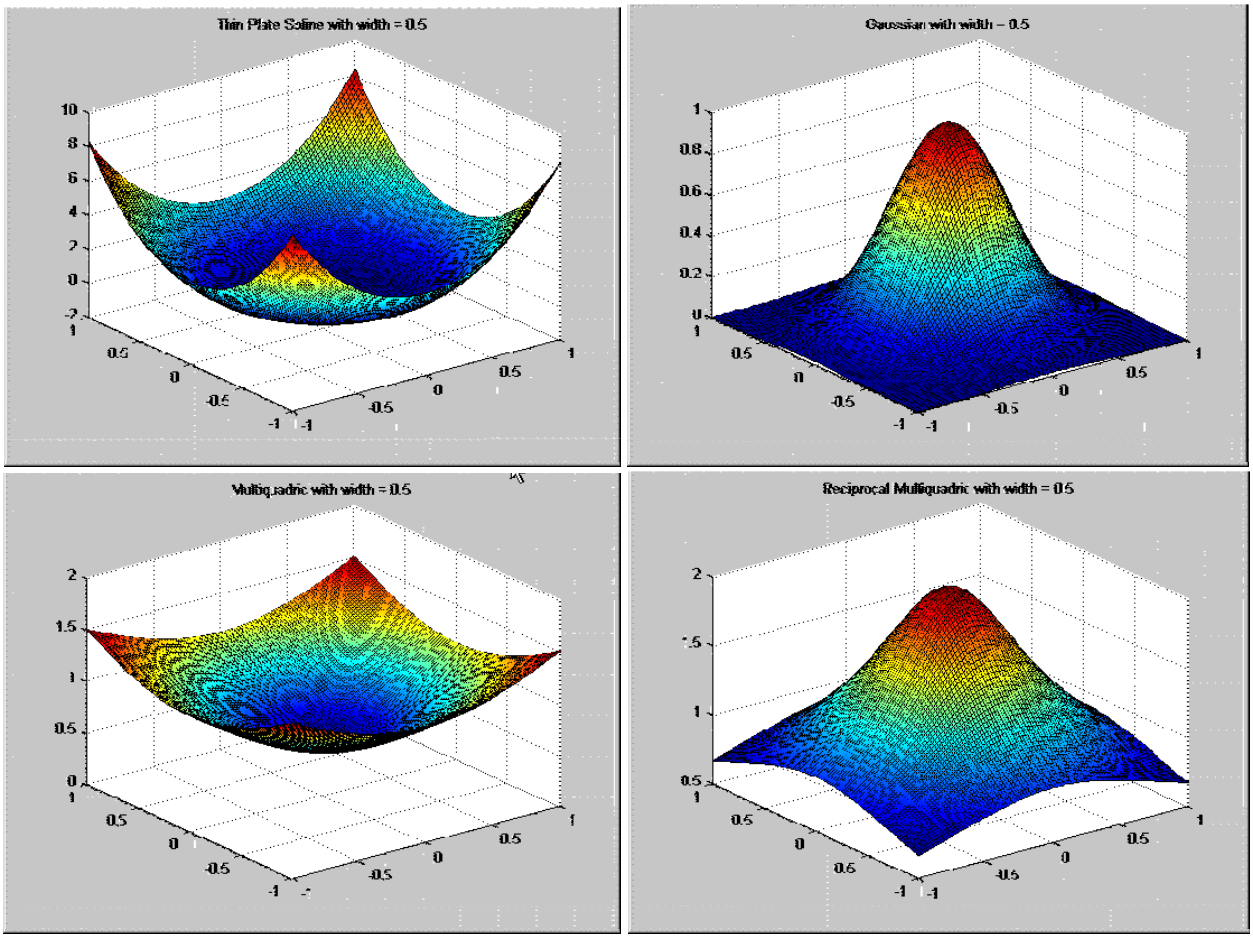
$$\phi_j\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=\left\|\mathbf{x}-\mathbf{x}_j\right\|^2\log\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right) \qquad \textit{thin plate spline} \qquad (4.35)$$

$$\phi_j\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=e^{\left(-\theta\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)^2} \qquad \text{Gaussova} \qquad (4.36)$$

$$\phi_j\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=\sqrt{\left\|\mathbf{x}-\mathbf{x}_j\right\|^2+\theta^2} \qquad \text{multikvadratična} \qquad (4.37)$$

$$\phi_j\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=\frac{1}{\sqrt{\left\|\mathbf{x}-\mathbf{x}_j\right\|^2+\theta^2}} \qquad \text{inverzna multikvadratična} \qquad (4.38)$$

Sve bazne funkcije imaju uveden kontrolni parametar θ kojim je moguće u određenoj mjeri kontrolirati zonu utjecaja, no zapravo nema jako veliki utjecaj na točnost surogat modela te se u brojnim referencama uopće ne koristi. Više o baznim funkcijama moguće je pronaći npr. u [43]. Na Sl. 4.9 prikazane su gornjim izrazima definirane bazne funkcije preuzete iz [35].



Sl. 4.9. Neke od najčešće korištenih baznih funkcija za RBF surogat modele [35]

Nepoznati koeficijenti **λ** i **β** dobivaju se rješavanjem sljedećeg linearnog sustava jednadžbi:

$$\begin{bmatrix}\mathbf{\Phi}_0 & \mathbf{B}_{1-n}\\ \mathbf{B}_{1-n}^T & \mathbf{0}\end{bmatrix}\cdot\begin{Bmatrix}\boldsymbol{\lambda}\\ \boldsymbol{\beta}\end{Bmatrix}=\begin{Bmatrix}\mathbf{y}_{1-n}\\ 0\end{Bmatrix} \qquad (4.39)$$

Gdje je matrica $\mathbf{B}_{1\text{-}n}$ definirana izrazom (4.15), dok je matrica $\boldsymbol{\phi}$ definirana kao:

$$\boldsymbol{\Phi}_0 = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ . & . & . & . \\ \phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_n - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix} \tag{4.40}$$

U konkretnim implementacijama nepoznati vektor $\boldsymbol{\beta}$ računa se sljedećim izrazom

$$\boldsymbol{\beta} = \left(\mathbf{B}_{1-n}^T \boldsymbol{\Phi}_0^{-1} \mathbf{B}_{1-n}\right)^{-1} \mathbf{B}_{1-n}^T \boldsymbol{\Phi}_0^{-1} \mathbf{y}_{1-n} \tag{4.41}$$

Dok se vektor nepoznatih težinskih koeficijenata $\boldsymbol{\lambda}$ računa pomoću izraza:

$$\boldsymbol{\lambda} = \boldsymbol{\Phi}_0^{-1}\left(\mathbf{y}_{1-n} - \mathbf{B}_{1-n}\boldsymbol{\beta}\right) \tag{4.42}$$

Pritom je vidljivo da se oba vektora nepoznatih koeficijenata mogu izračunati na temelju poznatih vrijednosti odzivnih veličine $\mathbf{y}_{1\text{-}n}$ i vrijednosti nezavisnih faktora $\mathbf{x}_{1\text{-}n}$. što znači da ostaju nepromijenjeni za predikciju bilo koje nepoznate lokacije **x**. Za predikciju odziva na lokaciji x potrebno je izračunati pripadajuće vrijednosti vektora **B** i $\boldsymbol{\phi}$ te korištenjem izraza (4.32) izračunati aproksimaciju odzivne veličine. Za bazne funkcije najčešće se uzimaju polinomi niskog reda odnosno linearna funkcija.

Kao i kod kriginga umjesto izraza (4.32) koji uključuje postojanje određenog trenda vrlo često koriste se i pojednostavljeni modeli u kojima se pretpostavlja da trend ne postoji ili da je trend nepoznata konstanta.

U slučaju pretpostavke da se trend ne modelira, izraz (4.32) se pojednostavljuje na:

$$\hat{y}_{RBF} = \boldsymbol{\phi}\boldsymbol{\lambda} \tag{4.43}$$

dok se nepoznati težinski koeficijenti računaju pomoću izraza:

$$\boldsymbol{\lambda} = \left(\boldsymbol{\Phi}_0^T \boldsymbol{\Phi}_0\right)^{-1} \boldsymbol{\Phi}_0^T \mathbf{y}_{1-n} \tag{4.44}$$

Za slučaj kad se trend modelira kao nepoznata konstanta izraz (4.32) se pojednostavljuje na:

$$\hat{y}_{RBF} = \boldsymbol{\phi}\boldsymbol{\lambda} + \alpha \tag{4.45}$$

Nepoznati koeficijenti dobivaju se rješavanjem sljedećeg linearnog sustava jednadžbi:

$$\begin{bmatrix} \boldsymbol{\Phi}_0 & \mathbf{e} \\ \mathbf{e}^T & \mathbf{0} \end{bmatrix} \cdot \begin{Bmatrix} \boldsymbol{\lambda} \\ \alpha \end{Bmatrix} = \begin{Bmatrix} \mathbf{y}_{1-n} \\ \mathbf{0} \end{Bmatrix} \tag{4.46}$$

gdje je e vektor jedinica veličine n:

$$\mathbf{e}^T = \{1 \quad 1 \dots \quad 1_n\} \tag{4.47}$$

Te se nepoznati težinski koeficijenti rješavaju izrazom:

$$\boldsymbol{\lambda} = \left(\boldsymbol{\Phi}_0^T \boldsymbol{\Phi}_0\right)^{-1} \boldsymbol{\Phi}_0^T \left(\mathbf{y}_{1-n} - \alpha\right) \tag{4.48}$$

dok je α upravo srednja vrijednost vektora poznatih odzivnih veličina $\mathbf{y}_{1\text{-}n}$.

Potrebno je napomenuti odabir RBF modela, bez trenda, s konstantnim trendom ili s polinomskim trendom, treba biti vođen karakteristikama originalnog procesa. Upotreba punog RBF modela s trendom ne mora značiti i točniju aproksimaciju. Ponekad upotreba modela s trendom može značajno narušiti preciznost surogata ako nije odabrana odzivna površina koja može dobro aproksimirati trend. Više o RBF surogat modelu moguće je pronaći npr. u [44] ili [43].

4.3.4. Validacija surogat modela

Validacija surogat modela obavlja se izračunavanjem greške ili nekog drugog mjerila točnosti, tako da se predikcija generiranog surogat modela uspoređuje sa rezultatima stvarnih simulacija.

Dobivena greška surogata na uzorku za validaciju je rezultat kojim se tumači uspješnost ili neuspješnost surogat modela te njegova korisnost u primjeni za predviđanje na budućim podacima. Srednja kvadratna greška (MSE -mean square error) definirana je sljedećim izrazom:

$$e_{MSE} = \int \left[\hat{y}(\mathbf{x}) - y(\mathbf{x})\right]^2 d\mathbf{x} \tag{4.49}$$

odnosno kad se testiranje vrši na konačnom broju kontrolnih uzoraka *n*:

$$e_{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left[\hat{y}_i(\mathbf{x}_i) - y_i(\mathbf{x}_i)\right]^2 \tag{4.50}$$

gdje je $\hat{y}$ predviđanje surogat modela, dok je *y* stvarni izlaz, a n je broj uzorku na kojima se testira točnost surogat modela.

U literaturi se za mjerenje greške najčešće koristi drugi korijen srednje kvadratne greške (RMS -root mean square error), prema formuli:

$$e_{RMS} = \sqrt{\int \left[\hat{y}(\mathbf{x}) - y(\mathbf{x})\right]^2 d\mathbf{x}} \tag{4.51}$$

odnosno kad se testiranje vrši na konačnom broju kontrolnih točaka *n*:

$$e_{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left[\hat{y}_i(\mathbf{x}_i) - y_i(\mathbf{x}_i)\right]^2} \tag{4.52}$$

U literaturi je moguće pronaći bezdimenzijske forme RMS greške koje daju očiglednu prednost zbog brže i jasnije percepcije veličine greške. Normaliziranu srednju kvadratne greške (NRMS – normalized root mean square error) moguće je izračunati pomoću izraza:

$$e_{NRMS} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\hat{y}_i(\mathbf{x}_i) - y_i(\mathbf{x}_i)\right]^2}}{\max(y_i) - \min(y_i)} \tag{4.53}$$

4.3.5. Analiza varijance polinomskih surogat modela

Pri radu sa surogat modelima jedan od prvih koraka jest odabir kontrolnih parametara u odnosu na koje će se generirati surogat model određene odzivne veličine. Pri tome treba imati na umu da je pri radu s deterministčkim računalnim simulacijama, za razliku od klasičnih eksperimenata, obično poznato koji su utjecajni parametri. Ipak utjecaj nekih parametara se može zanemariti te time pojednostavniti surogat model.

Za testiranje hipoteze o utjecaju pojedinog parametra na predviđanje polinomskog surogat modela može se koristiti postupak opisan u [33]. Procedura je bazirana na provjeri hipoteze o značaju koeficijenata vezanih uz određeni parametar nakon izbacivanja tih parametara iz surogat modela. Prije procedure bit će opisan postupak određivanja signifikantnosti potpunog surogat modela.

4.3.5.1. Provjera signifikantnosti polinomskog surogat modela

Provjera signifikantnosti polinomskog surogat modela izvodi se testiranjem sljedećih dviju hipoteza:

$$\begin{aligned} &H_0 : \beta_0 = \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\ &H_1 : \beta_i \neq 0 \text{ za barem jedan } i \end{aligned} \tag{4.54}$$

Odbacivanje hipoteze H0 implicira da barem jedan parametar surogat modela značajno utječe na aproksimaciju odzivne veličine. Ukupna suma kvadrata SS_{yy} sastoji se od zbroja sume kvadrata regresije SS_R i sume kvadrata greške modela SS_E:

$$SS_{yy} = SS_R + SS_E \tag{4.55}$$

gdje je:

$$SS_{yy} = \sum_{i=1}^{n} y_i^2 - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n} \tag{4.56}$$

$$SS_E = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 \tag{4.57}$$

$$SS_R = \sum_{i=1}^{n} \left(\hat{y}_i - \bar{y}\right)^2 \tag{4.58}$$

Srednje sume kvadrata regresije MS_R i greške MS_E mogu se izračunati izrazima:

$$MS_R = \frac{SS_R}{k-1} \tag{4.59}$$

$$MS_E = \frac{SS_E}{n-k} \tag{4.60}$$

gdje je *n* broj eksperimenata kojima je generiran polinomski surogat, dok je *k* broj koeficijenata polinomskog surogata uključujući konstantni član.

Uz poznate MS_R MS_E može se izračunati F_0:

$$F_0 = \frac{MS_R}{MS_E} \tag{4.61}$$

te na temelju izračunate vrijednosti odbaciti hipotezu H_0 ukoliko je zadovoljen izraz:

$$F_0 > F_{\alpha,k-1,n-k} \tag{4.62}$$

gdje je α željeni interval pouzdanosti.

4.3.5.2. Provjera signifikantnosti kontrolnih parametara polinomskog surogata

Prema postupku opisanom u potpoglavlju 15-7 [33] može se utvrditi signifikantnost članova koji su vezani za određeni koeficijent β_i polinomske regresije. Malom modifikacijom te procedure moguće je utvrditi signifikantnost određenog parametra izbacivanjem svih članova β_{i-r} vezanih za taj parametar.

U ovom slučaju testiraju se hipoteze:

$$\begin{aligned} &H_0 : \beta_i = \beta_{i+1} \cdots = \beta_{r-1} = \beta_r = 0 \\ &H_1 : \beta_{i-r} \neq 0 \,\text{za barem jedan}\, i \end{aligned} \tag{4.63}$$

Kao i kod provjere signifikantnosti surogata, tako se i u ovom slučaju može odbaciti hipoteza H_0 ukoliko je zadovoljen izraz (4.62) no u ovom slučaju F_0 se računa izrazom:

$$F_0 = \frac{MS_{R(\beta-\beta_{red})}}{MS_E} \tag{4.64}$$

gdje je:

$$MS_{R(\beta-\beta_{red})} = \frac{SS_{R\beta} - SS_{R\beta red}}{k - k_{red}} \tag{4.65}$$

$SS_{R\beta}$ je suma kvadrata ukupnog surogat modela i računa se izrazom:

$$SS_{R\beta} = \sum_{i=1}^{n} \left[\hat{y}_i\left(\beta\right)\right]^2 \tag{4.66}$$

$SS_{R\beta red}$ je suma kvadrata reduciranog surogat modela kojem su maknuti članovi vezani za testirani parametar:

$$SS_{R\beta red} = \sum_{i=1}^{n} \left[\hat{y}_i\left(\beta_{red}\right)\right]^2 \tag{4.67}$$

s tim da je *k* broj koeficijenata ukupnog surogat modela, dok je k_{red} broj koeficijenata reduciranog surogat modela.

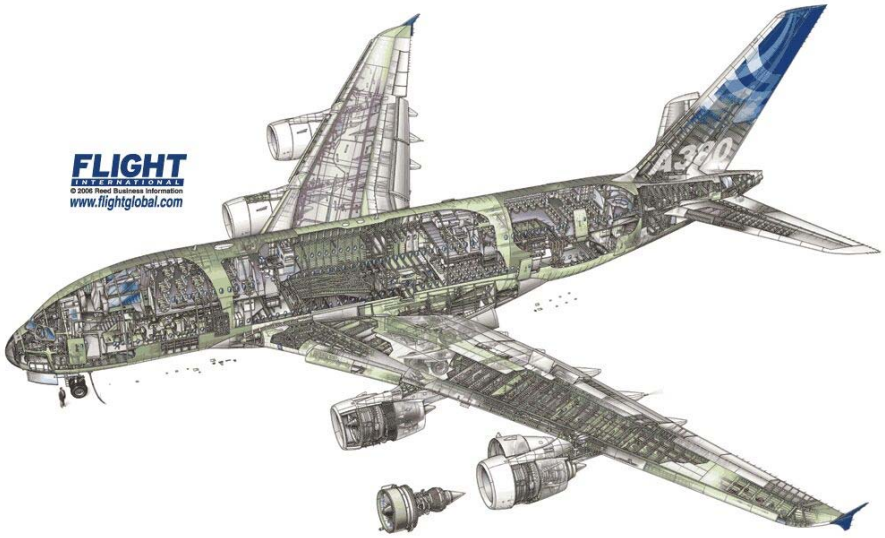
Za analizu utjecaja pojedinih parametara na određenu odzivnu veličinu najčešće se koristi analiza varijance (ANOVA). No za njeno korištenje neophodno je imati plan pokusa na kojem se određene vrijednosti pojedinih parametara ponavljaju. Stoga nije moguće provesti ovu analizu ukoliko su pokusi generirani nekom od popunjavajućih metoda (npr. LHS). Postupak opisan u ovom potpoglavlju nije ograničen tipom pokusa, a omogućuje donošenje odluke o zadržavanju ili odbacivanju određenog parametra u surogat modelu što osim pojednostavljenja modela ponekad omogućuje i povećanje točnosti surogat modela.

4.4. Primjena surogat modeliranja u optimizaciji

Recenzija i usporedba regresijskih metoda spomenutih aproksimacijskih modela nalazi se u radu [45]. Prema istraživanju ni jedna metoda ne dominira prema svim kriterijima za sve tipove problema te je opći zaključak da je za efikasno korištenje neophodno odabrati onu metodu koja je, prema jednom ili više kriterija kvalitete, najprikladnija za rješavanje danog problema. Moguća nadogradnja jest korištenje kombinacije (ansambla) surogata (*ensemble of surrogates*) [46] iz koje je moguće koristiti bilo koji od algoritama. Takav pristup omogućuje korištenje najbolje ocijenjene metode ili ponderirane prosječne sume skupine surogata umjesto korištenja samo jednog surogat modela. Težinski faktori pridruženi svakom surogat-modelu određuju se na temelju izračunate pogreške surogat modela. Jedna od primjena surogat modeliranja na dekomponiranom problemu može se pronaći u [47] gdje je prikazana metoda za korištenje odzivne plohe pri koordinaciji lokalnog i globalnog optimizacijskog problema. Uz aproksimacijske modele, kao zamjena za direktni proračun, može se koristiti i nasljeđivanje dobrote rješenja [48]. Ta metoda potiče iz modificiranog genetskog algoritma kojemu se za izračun dobrote nove jedinke ne poziva proračun kompleksnog modula za analizu problema, već se dobrota nasljeđuje od roditelja.

5. DEKOMPOZICIJA I KOORDINACIJA PROJEKTNOG PROBLEMA

U današnjoj svakodnevnici moglo bi se reći da su složeni sustavi posvuda oko nas: prijevozna sredstva koja koristimo, zgrade u kojima živimo, kućanski aparati koje koristimo te naravno informatičko komunikacijski uređaji poput računala i mobilnih telefona koji su u sve široj uporabi. Složene konstrukcije kakve se mogu pronaći u zrakoplovstvu, brodogradnji (Sl. 5.1) te strojarstvu općenito, sastoje se raznih sustava i podsustava na više razina koje idu sve do elementarnog materijala. Kao i kod prethodno navedenih primjera iz svakodnevnice složene konstrukcije su sustavi u smislu da su sastavljeni od niza entiteta koji zajedno služe određenoj svrsi.



a) Generalni plan putničkog zrakoplova Airbus 380 [49]

b) Generalni plan putničkog broda [50]

Sl. 5.1.Primjeri složenih konstrukcija s područja zrakoplovstva i brodogradnje

Sustav je složen ako sadržava veliki broj međusobno zavisnih komponenti odnosno podsustava. Takvi sustavi često onemogućuju projektantima uvid u to kako projektna varijanta jedne komponente utječe na ostale komponente ili na sam sustav. Airbus A380 (Sl. 5.1 a)) je primjer jednog takvog sustava na kojem se samo krilo sastoji od (prema [51]):

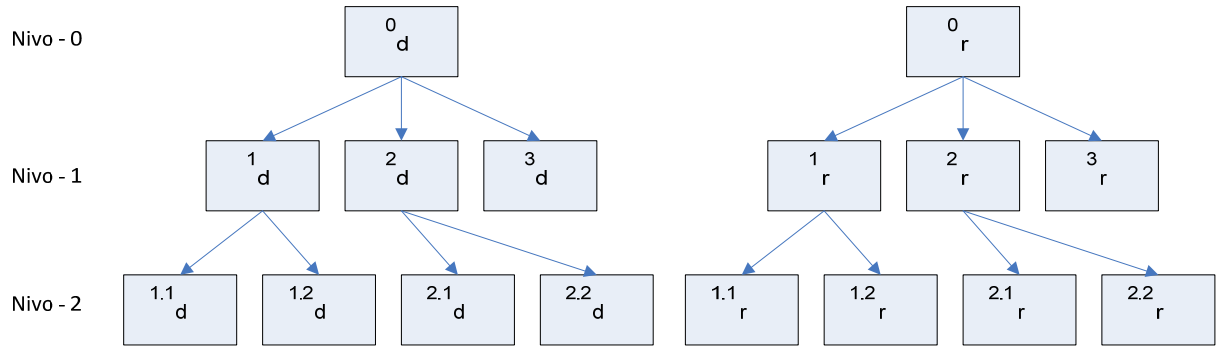
- „20 panela vanjske oplate od aluminijske slitine,
- 124 rebra, od čega je 76 metalnih, a 48 kompozitnih (ugljična vlakna),
- 6 ramenjača od aluminijske slitine,
- 314 uzdužnica od čega je 124 na gornjanci , a 190 na donjanci krila,
- 16 uređaja za poboljšanje uzgona na prednjoj ivici, 6 na stražnjoj te 22 kontrolne površine,
- 360,000 metara žice, hidrauličkih i pneumatskih cijevi za kontrolu predkrilaca, zakrilaca, krilaca i zračnih kočnica,
- 750,000 zakovica, vijaka i klinova.“

Uspješno projektiranje složenih inženjerskih sustava uključuje dekompoziciju sustava na određeni broj manjih podsustava odnosno potproblema sa svojim vlastitim ciljevima i ograničenjima. Međusobno povezivanje potproblema može se izvršiti na razne načine, ali se ponajviše koristi hijerarhijski princip gdje objekt na višem stupnju hijerarhije koordinira ili kontrolira objekt nižeg stupnja [52]. Posebno je važno razmotriti, da li je takav način koordinacije sposoban izvršavati svoju zadaću tako da rezultat optimizacije sustava bude zadovoljavajući u odnosu na ciljeve. Integrirani optimizacijski problem koji uključuje više potproblema s velikim brojem projektnih varijabli, nije moguće dekomponirati na samostalne potprobleme koji se međusobno nezavisno optimiziraju. Optimizacija jednog potproblema, koja ne uzima u obzir efekt interakcije s drugim potproblemima, može dovesti do snažne degradacije performansi cjelokupnog sustava tako da krajnji rezultat optimizacije može biti lošiji nego na početku. Cilj dekompozicije je uvođenje hijerarhijskog sustava koji dopušta dekompoziciju optimizacijskih problema na samostalne potprobleme gdje optimizacija potproblema vodi prema optimumu cjelokupnog sustava.

5.1. Opis korištene notacije za hijerarhijske sustave

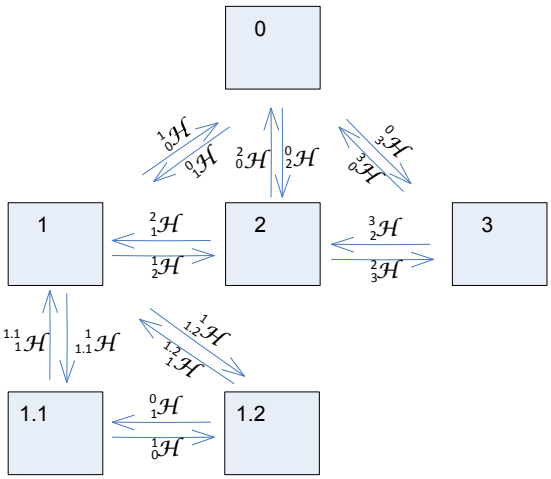
U radu je korištena notacija za hijerarhijske složene sustave s više razina (Sl. 5.1) definirana u radu [53], koja je donekle izmijenjena kako bi se usuglasila s ostalim dijelovima ovog rada. Prema toj notaciji globalni ili najgornji sustav se označava kao *Razina – 0*, a komponente sustava, poput deskriptora, dobivaju oznaku nula u gornjem lijevom uglu $^{0}\mathbf{d}$.

Jednu razinu niže, na Razini 1, nalaze se takozvana „ djeca“ razine 0 („roditelj“). Veličine na toj razini su numerirane gore lijevo od oznake veličine npr. $^{1}\mathbf{d}$. Na idućoj razini (Razina 2) nalaze se „ djeca“ Razine 1. Veličine na toj razini su numerirane gore lijevo od oznake veličine tako da prvo ide broj roditelja (s prethodne razine) nakon čega slijedi broj koji određuje pojedini podsustav u okviru sve djece koje imaju istog roditelja npr. $^{1.2}\mathbf{d}$. Na sličan način se mogu dodati i nove razine hijerarhije. Slično kao deskriptori sustava mogu se numerirati i odzivne veličine sustava odnosno odzivi (*responses*) kao što je prikazano na Sl. 5.1 b).



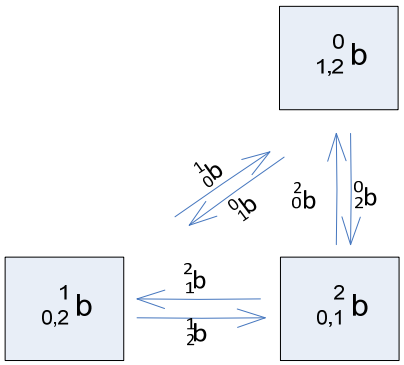
Sl. 5.2. Hijerarhijski složeni sustav s tri razine hijerarhije

Iako se na svakom sustavu mogu identificirati skupovi fizikalnih odziva **r**, ti odzivi su najčešće međusobno ovisni odnosno spregnuti s odzivima susjednih sustava (roditelj, braća i djeca). Svaka pojedina sprega mora se uzeti u obzir kao što je prikazano na Sl. 5.3. Zbog toga se uvodi je operator $\mathcal{H}$. Ovaj operator preslikava odziv jednog sustava na drugi. Gore lijevo od operatora označava se izvor informacije, odnosno iz kojeg sustava odziv, dok se dolje lijevo nalazi odredište informacije, odnosno u koji sustav ulazi npr. ${}_{1}^{0}\mathcal{H}$.



Sl. 5.3. Spreg (coupling) odziva sustava

U složenim sustavima postoji mogućnost isti deskriptori opisuju više različitih sustava te se nazivaju zajednički ili dijeljeni deskriptori *b*, a varijable koje su vezane na te deskriptore u pojedinim optimizacijskim potproblemima, zajedničke ili dijeljene varijable *z*. Zajedničke varijable (*z*) najčešće se razmatraju odvojeno od običnih varijabli (*x*). U potpunoj notaciji gornji lijevi indeks ($\overset{\cdot\cdot}{\mathbf{b}}$) označava sustav gdje je deskriptor dodijeljen, dok donji indeks ($\underset{\cdot\cdot}{\mathbf{b}}$) označava sustav koji dijeli zajednički deskriptor. Ukoliko više sustava dijeli zajednički deskriptor, svi se navode u donjem indeksu odvojeni zarezom.



Sl. 5.4. Zajednički deskriptori sustava

Kao što je već navedeno odzivi nekog sustava se operatorom $\mathcal{H}$ preslikavaju na drugi sustav. Kad je odziv preslikan s jednog sustava u drugi sustav, naziva se spregnuti odziv, označava oznakom ${}^{i}_{j}\mathbf{h}$ te je definiran izrazom:

$${}^{i}_{j}\mathbf{h} = {}^{i}_{j}\mathcal{H}\left({}^{i}\mathbf{r}\right) \tag{5.1}$$

Spregnuti odziv $\underset{\cdot\cdot}{\overset{\cdot\cdot}{\mathbf{h}}}$ može se tretirati i kao željeni odziv susjednih elemenata, odnosno umjesto da ga se računa preslikavanjem, njegova vrijednost se može odabrati tj. u definiranom optimizacijskom potproblemu može biti tretiran kao varijabla. Odgovarajući stvarni (fizikalni) odziv $\left({}^{i}\mathbf{r}\right)$ određuju se u pripadajućim sustavima.

Ograničenja konzistentnosti $\underset{\cdot\cdot}{\overset{\cdot\cdot}{\mathbf{c}}}$ uvode se kako bi se privremeno odvojili sustavi od svoje okoline što omogućuje proračun sustava bez interakcije sa susjednim elementima. Ograničenja konzistentnosti su matematički definirana izrazom:

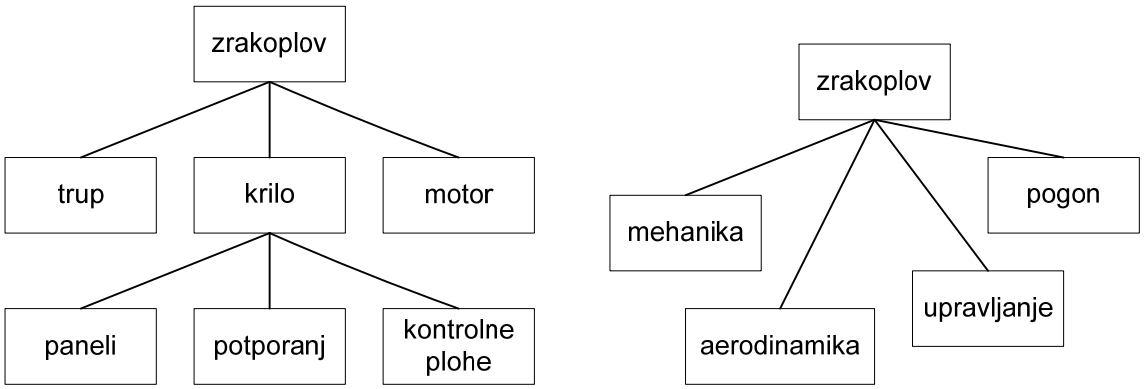
$${}^{i}_{j}\mathbf{c} = {}^{i}_{j}\mathcal{H}\left({}^{i}\mathbf{r}\right) - {}^{j}_{i}\mathbf{h}\,;\ i \neq j \tag{5.2}$$

$${}^{i}_{j}\mathbf{c}_z = {}^{i}_{j}\mathbf{z} - {}^{j}_{i}\mathbf{z}\,;\ i \neq j \tag{5.3}$$

Slično kao i kod operatora $\mathcal{H}$, gornji lijevi indeks $^{\cdot\cdot}\mathbf{c}$ označava izvor odziva ili zajedničkog deskriptora, dok donji lijevi indeks $_{\cdot\cdot}\mathbf{c}$ označava odredišni sustav.

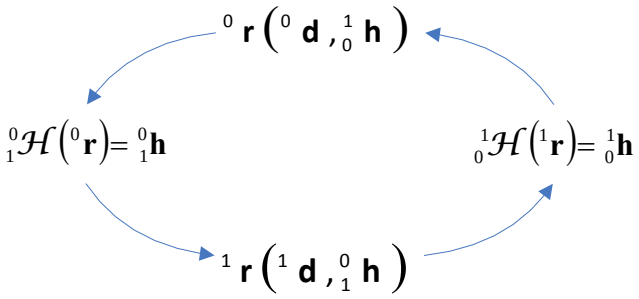
5.2. Dekompozicija problema

Cilj dekompozicije podjela sustava na više manjih podsustava ili disciplina. Metoda dekompozicije u literaturi usko je vezana uz multidisciplinarnu optimizaciju (*Multidisciplinary Design Optimization – MDO*) čiju primjenu većinom susrećemo u zrakoplovstvu (vidi npr. [3]). MDO zapravo je poseban slučaj dekompozicije projektnog problema u kojemu se dekompozicija vrši prema disciplini, npr. pri projektiranju krila zrakoplova problem se može dekomponirati na potprobleme aerodinamike, propulzije, strukture te performansi. Osim takve dekompozicije koristi se još i dekompozicija prema građevnim elementima/sustavima prema kojoj bi se problem projektiranja krila mogao hijerarhijski dekomponirati na potprobleme globalnih dimenzija krila, torzijske kutije te ukrepljenih panela.



Sl. 5.5.Primjeri dekompozicije problema projektiranja krila zrakoplova

U nastavku su opisane osnovne karakteristike dekompozicije prema [53] i [54]. Prema njima dekompozicija uključuje identificiranje slabih veza između spregnutih sustava, čime se omogućuje definiranje individualnih, ali spregnutih, optimizacijskih potproblema. Interakcija dvaju spregnutih sustava može se vidjeti na slici Sl. 5.6 gdje globalni sustav (razina 0) proizvodi fizikalni odziv $^{0}\mathbf{r}$ te se preslikava operatorom $_{1}^{0}\mathcal{H}$ u informaciju $_{1}^{0}\mathbf{h}$ koja je neophodna za računanje odziva sustava $^{1}\mathbf{r}$ na razini 1. Isto tako odziv elementa s razine 1 se operatorom $_{0}^{1}\mathcal{H}$ preslikava u informaciju koja je potrebna za računanje odziva globalnog sustava. Dekompozicijom se veze između spregnutih sustava privremeno brišu, što omogućuje da se oba sustava optimiziraju nezavisno.



Sl. 5.6.Spregnutost odziva dvaju sustava

5.2.1. Načini izvođenja dekompozicije

Dekompozicija se omogućuje uvođenjem ograničenja kompatibilnosti. Formulacije koje se koriste za održavanje konzistentnosti između sustava su:

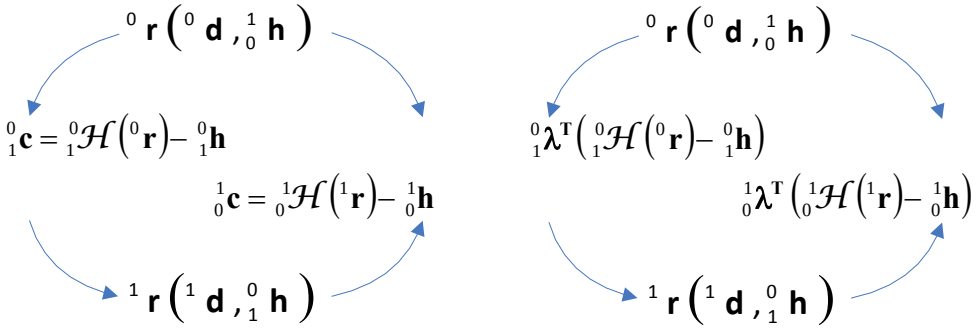
- jaka formulacija, izražena ograničenjima jednakosti,
- slaba tj. relaksirana formulacija.

Jaka formulacija matematički je izražena izrazom (5.4) , a ilustrirana je na Sl. 5.7 a)

$$\begin{aligned} {}^{0}_{1}\mathbf{c}&={}^{0}_{1}\mathcal{H}\left({}^{0}\mathbf{r}\right)-{}^{0}_{1}\mathbf{h}=0;\\ {}^{1}_{0}\mathbf{c}&={}^{1}_{0}\mathcal{H}\left({}^{1}\mathbf{r}\right)-{}^{1}_{0}\mathbf{h}=0 \end{aligned} \tag{5.4}$$

Slaba ili relaksirana formulacija (npr. pomoću Lagrangeovih množitelja) matematički je izražena izrazom (5.5), a ilustrirana je na Sl. 5.7 b)

$$\begin{aligned} {}^{0}_{1}\boldsymbol{\lambda}^{T}\,{}^{0}_{1}\mathbf{c}&={}^{0}_{1}\boldsymbol{\lambda}^{T}\left({}^{0}_{1}\mathcal{H}\left({}^{0}\mathbf{r}\right)-{}^{0}_{1}\mathbf{h}\right);\\ {}^{1}_{0}\boldsymbol{\lambda}^{T}\,{}^{1}_{0}\mathbf{c}&={}^{1}_{0}\boldsymbol{\lambda}^{T}\left({}^{1}_{0}\mathcal{H}\left({}^{1}\mathbf{r}\right)-{}^{1}_{0}\mathbf{h}\right) \end{aligned} \tag{5.5}$$



a) jaka formulacija (ograničenja jednakosti) *b) slaba formulacija (relaksirana)*

Sl. 5.7. Dekomponiranje sustava uvođenjem ograničenja konzistentnosti

Dekompozicija bazirana na relaksaciji ostvaruje se relaksiranjem ograničenja konzistentnosti. Najčešće se koriste sljedeće tri metode relaksacije:

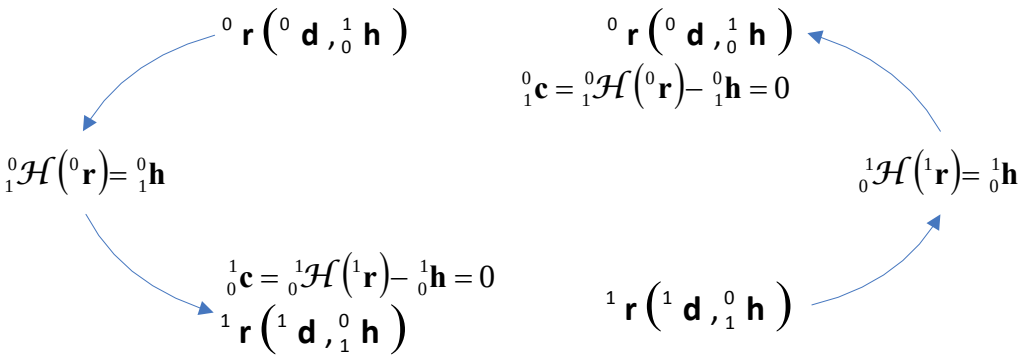
- relaksacija funkcijom kazne,
- Lagrangeova relaksacija,

- proširena Lagrangeova relaksacija.

  Prema načinu rasprezanja složenih sustava dekompozicija može biti:

  - hijerarhijska,
    - odozgo prema dole (*top-down*),
    - odozdo prema gore (*bottom-up*),
  - nehijerarhijska.

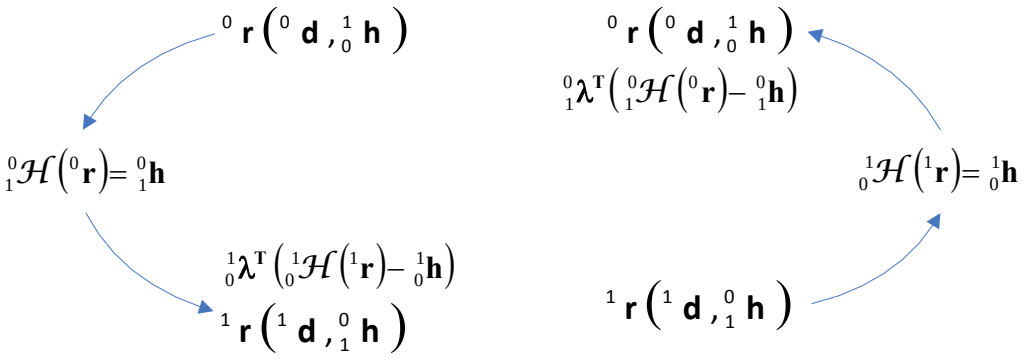
Hijerarhijska dekompozicija je rezultat identificiranja sustava u hijerarhiji koji diktiraju izlaz drugih (sebi podređenih) sustava. Na slici Sl. 5.8 a) prikazana je dekompozicija odozgo prema dolje s ograničenjima jednakosti. Globalni sustav (razina 0) propisuje potrebni odziv sustava s razine 1 ${}^{1}_{0}\mathbf{h}$, dok ograničenja konzistentnosti na razini 1 osiguravaju da se predviđeni odziv ostvari. Sl. 5.1 b) na sličan način prikazuje dekompoziciju odozdo prema gore.



a) odozgo prema dole *b) odozdo prema gore*

Sl. 5.8. Dekomponiranje hijerarhijskih sustava jakom formulacijom (ograničenja jednakosti)

Slučaj kad su ograničenja konzistentnosti relaksirana Lagrangeovim množiteljima prikazan je na Sl. 5.9. Ova relaksirana ograničenja se dodaju u cilj optimizacijskog potproblema sustava na način da se kršenjem ograničenja konzistentnosti pogoršava vrijednost funkcije cilja.



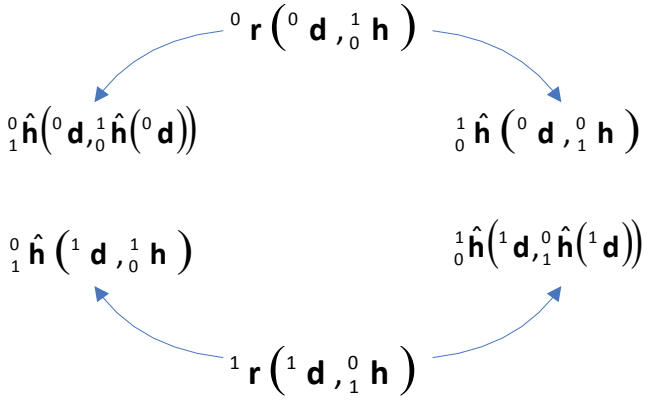
a) odozgo prema dole *b) odozdo prema gore*

Sl. 5.9. Dekomponiranje hijerarhijskih sustava slabom formulacijom (relaksacija ograničenja)

Za slučajeve prikazane na Sl. 5.8 i Sl. 5.9 spregnutost postoji samo u jednom smjeru, prema gore ili prema dolje ovisno o primijenjenoj formulaciji, a zamjenjuje se ograničenjima konzistentnosti. U suprotnom smjeru spregnutost je i dalje prisutna čime je jedan sustav postao „vođa“ a drugi „sljedbenik“.

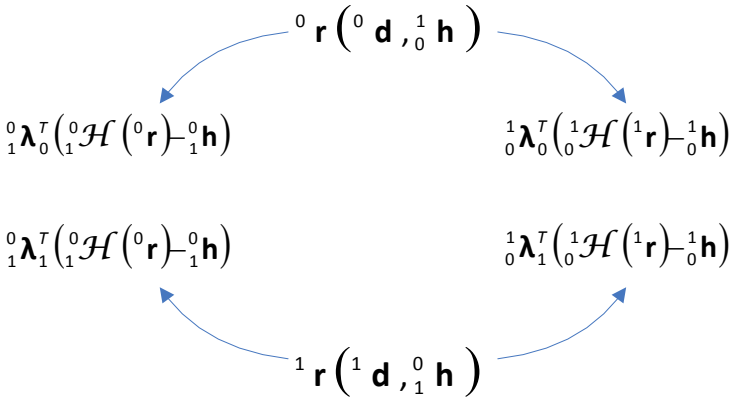
Nehijerarhijska dekompozicija uključuju zamjenu spregnutosti ograničenjima konzistentnosti u oba smjera. Kako bi se to omogućilo odzivi susjednih sustava se prikladnom metodom procjenjuju tj. aproksimiraju, a nakon pronalaska rješenja spregnuti parametri se koordiniraju te provjerava zadovoljenost ograničenja konzistentnosti.

Za slučaj osiguravanja konzistentnosti ograničenjima jednakosti kod nehijerarhijskih sustava neophodno je aproksimiranje odziva ${}_{j}^{i}\hat{\mathbf{h}}$ susjednih sustava surogat modelom (Sl. 5.10).



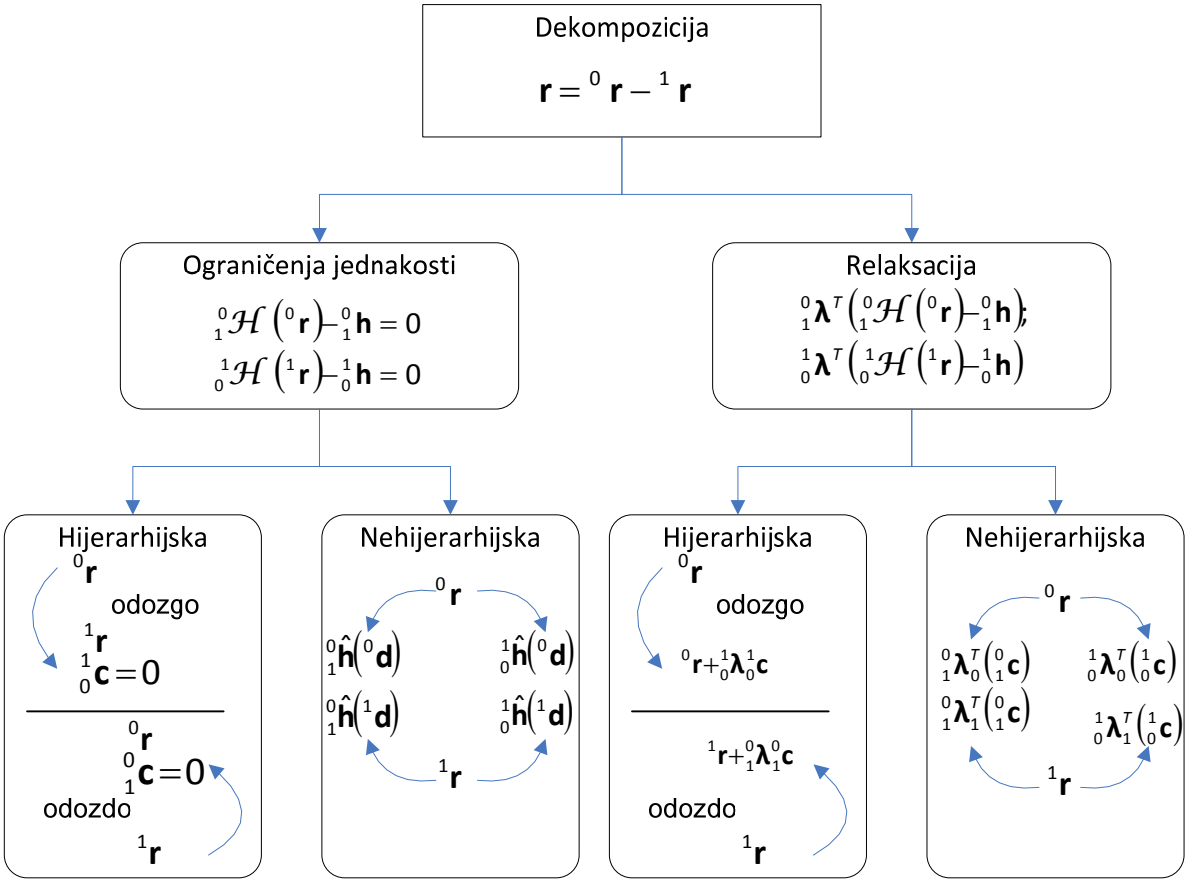
Sl. 5.10. Dekomponiranje nehijerarhijskih sustava jakom formulacijom (ograničenja jednakosti)

Nehijerarhijska dekompozicija s relaksacijom pomoću Lagrangeovih množitelja prikazana je na slici Sl. 5.11. Zahtjeva osvježavanje Lagrangeovih množitelja odabranom koordinacijskom tehnikom.



Sl. 5.11. Dekomponiranje nehijerarhijskih sustava slabom formulacijom (relaksacija ograničenja)

Na Sl. 5.12 dan je sažeti dijagramski prikaz osnovnih načina dekompozicije složenih sustava.



Sl. 5.12. Mogući pristupi za dekompoziciju složenih sustava

5.2.2. Klasifikacija interakcije sustava/podsustava prema jačini i količini spregnutosti

U prethodnom potpoglavlju navedeni su načini dekompozicije složenih spregnutih sustava, a u ovom poglavlju navest će se napomene vezane uz mjere spregnutosti sustava. Upravo formalizacija mjera spregnutosti te njihov utjecaj na uspješnost rješavanja problema primjenom dekompozicijsko-koordinacijskih metoda okarakterizirane su prema nedavnoj tribini eminentnih stručnjaka ovog područja [55] kao jedan od perspektivnih smjerova razvoja istraživanja.

Jačina spregnutosti CS (*coupling strength*) daje koliko jako odziv dekomponiranog sustava *i* utječe na odziv drugog njemu spregnutog sustava *j* što se prema korištenoj notaciji može napisati kao:

$$ {}^{i}_{j}CS = {}^{i}_{j}CS\left({}^{i}_{j}\mathbf{h}\right) = {}^{i}_{j}CS\left({}^{j}\mathbf{r}\left({}^{i}_{j}\mathbf{h}\right)\right) \tag{5.6}$$

Trenutno ne postoji univerzalno korištena mjera za jačinu spregnutosti, no najčešće se koristi parcijalna derivacija:

$$ {}^{i}_{j}CS = \frac{\partial\left({}^{j}\mathbf{r}\right)}{\partial\left({}^{i}_{j}\mathbf{h}\right)} \tag{5.7}$$

Izraz (5.7) nije skalar te i dalje nije pogodan za mjerenje i trenutno ne postoji konsenzus oko načina pretvaranja ovog izraza u skalar. Jedan od načina bi mogao biti primjenom tehnika koji se koriste u MADM-u tj. normalizacijom veličina ${}_j^i\mathbf{h}$ i ${}^j\mathbf{r}$ nakon čega bi slijedilo zbrajanje ili ponderirano zbrajanje. Ipak, najčešće mjerenje jačine spregnutosti praktično se provodi subjektivnom procjenom projektanta/istraživača na osnovu (5.7) ili čak spoznaje o razlici $\Delta\left({}^j\mathbf{r}\right)\text{i}\,\Delta\left({}_j^i\mathbf{h}\right)$ u dvije projektne točke.

Veličina vektora ${}_j^i\mathbf{h}$ ujedno daje i količinu podataka koji su spregnuti odnosno količinu spregnutosti CB (*coupling breadth*).

Korištenje ove dvije mjere može dati istraživačima-projektantima mogućnost procjene uspješnosti primjene dekompozicijsko-koordinacijskih tehnika čak i prije formalnog definiranja ukupnog dekomponiranog problema i korištenja odgovarajuće koordinacijske metode. Prikazivanjem graničnih slučajeva sa slabom i jakom spregnutošću te malom i velikom količinom spregnutosti može se formirati dijagram prikazan na Sl. 5.13 gdje su formirana četiri granična područja, P1 do P4, za koje se očekivana uspješnost primjene dekompozicijsko-koordinacijske tehnike može obrazložiti prema [55], [56] odnosno primjenom razumne inženjerske logike.

Sl. 5.13. Utjecaj mjera spregnutosti sustava na primjenu dekompozicijsko-koordinacijskih tehnika

Ishodište koordinatnog sustava korespondira sustavima koji su potpunosti raspregnuti i koji se mogu u potpunosti nezavisno optimizirati. U području P1 gdje su i jačina i količina

spregnutosti mali sustavi se mogu dekomponirati te relativno jednostavnim tehnikama koordinirano optimizirati.

S druge strane područje P4 je karakterizirano s velikom jačinom i količinom spregnutosti što bi u slučaju dekompozicije dovelo do visoko-dimenzionalnih optimizacijskih problema u kojima bi spregnute varijable mogle biti dominantne i po broju i po značaju u odnosu na lokalne varijable optimizacijskih problema podsustava. Stoga je takve sustave najčešće bolje optimizirati zajedno kao jedan sustav odnosno koristeći AiO (MDF) pristup.

Područje P2 okarakterizirano je sa malom količinom jako spregnutih veza. Primjena dekompozicijsko koordinacijskih metoda u ovom području daje najbolje rezultate i potrebno ih je primjenjivati kad god je to moguće.

Područje P3 okarakterizirano je sa velikom količinom slabo spregnutih veza. U ovom području može biti korisno zanemariti određene veze čija je jačina spregnutosti ispod određene razine.

5.3. Koordinacija

U prethodnom potpoglavlju predstavljene su metode za dekompoziciju analize ili optimizacije složenih sustava. S obzirom da su analize/optimizacije dekomponiranih sustava spregnute tijekom rješavanja je neophodna njihova koordinacija. Koordinacija ima zadaću kombiniranja međurezultata (*intermediate*) pojedinih podsustava sa svrhom pronalaska rješenja za cijeli složeni sustav.

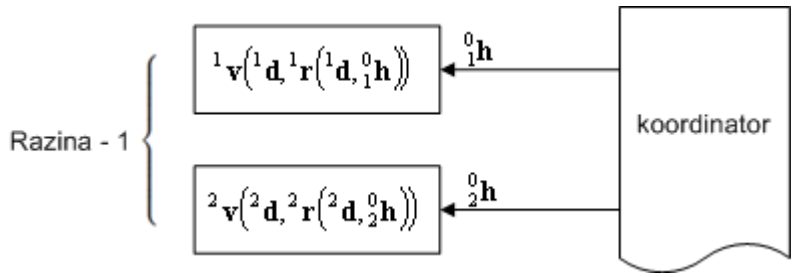
5.3.1. Tri tipa odluka pri koordinaciji

Koordinacija interakcije između podsustava uključuje odlučivanje jer spregnuti podaci, bilo odzivi ili varijable, ne mogu biti proizvoljno slani od sustava do njegovih susjeda. Stoga je neophodno donošenje odluka o tome koji podaci se šalju i kojim redom.

Prema [53] i [54], moguće je razlučiti tri tipa odluka:

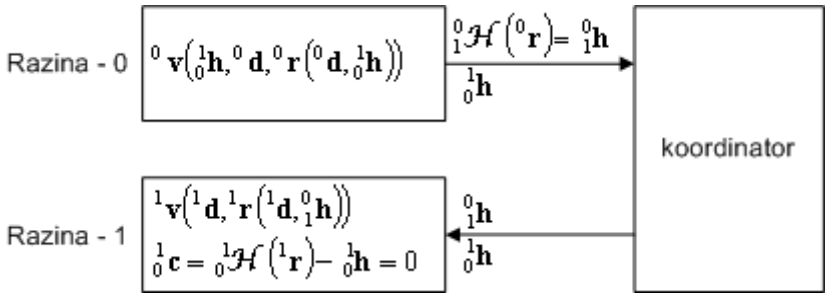
- odjeljive odluke,
- neodjeljive odluke,
- neodjeljive i spregnute odluke.

Odjeljive odluke se javljaju kod sustava kod kojih osvježavanje informacija nekog sustava ne ovisi o podacima koji se šalju njegovim susjedima (Sl. 5.14). Ovakav tip odluka je uobičajen kod sustava koji su na istoj razini, a nisu spregnuti. Stoga, nakon rješavanja sustava iz nadređene razine, osvježavanje informacija svih njemu podređenih sustava iz iduće razine je neovisno.



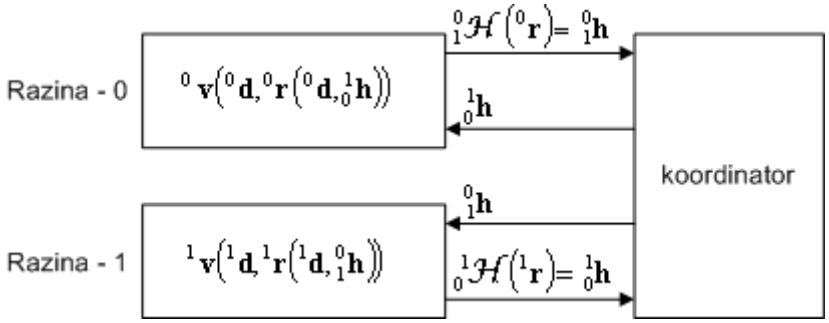
Sl. 5.14. Odjeljive odluke pri koordinaciji

Kod neodjeljivih odluka izlaz jednog sustava se šalje kao ulaz njegovim susjedima (Sl. 5.15). Ovakve odluke se donose sekvencijalno i rezultat su hijerarhijske dekompozicije odozgo ili odozdo. Kod ovog tipa sustava u koordinatoru se ne izvode nikakve operacije, a nakon što se završi optimizacija globalnog sustava na razini 0 njegovo stanje je nepromjenjivo dok su sustavima na podrazini propisani odzivi koje moraju zadovoljiti. Više nezavisnih istraživanja (npr.[57],[58]) je ustvrdilo da ovakav način koordinacije ne uspijeva konvergirati prema realnom rješenju ukoliko se ograničenja konzistentnosti tj. propisani odzivi ili zavisni deskriptori ne mogu zadovoljiti u podrazinama, bilo zbog gornjih i donji granica varijabli ili propisanih ograničenja. Kako bi se osiguralo zadovoljenje svih ograničenja na podrazinama, informacija o ograničenjima podređenih sustava se mora na neki način prenijeti u nadređeni odnosno globalni sustav.



Sl. 5.15. Neodjeljive odluke pri koordinaciji

Treći tip odluka su neodjeljive i spregnute odluke kod kojih je odziv jednog sustava ulaz njegovim susjedima i obrnuto (Sl. 5.16). Odluke se donose sekvencijalno kroz iteracije ili istovremeno (*concurrently*).



Sl. 5.16. Neodjeljive i spregnute odluke pri koordinaciji

Neodjeljive odluke koje zahtijevaju dovođenje dodatne informacije iz podređenog sustava u globalni sustav radi osiguranja zadovoljenja ograničenja u podređenim sustavima te neodjeljive i spregnute odluke pripadaju skupini dvorazinskih koordinacijskih metoda (*bi-level coordination methods*) koje će biti detaljnije opisane u poglavlju 5.4. Jednorazinske koordinacijske metode (*single-level coordination methods*) uključuju slanje modelskih podataka između sustava. Ukoliko je potrebna dodatna obrada modelskih podataka, generiranje surogat modela, analiza senzitivnosti ili evaluacija parametara relaksacije, potrebna je nova, druga razina u kojoj se obavljaju ti proračuni. Ukoliko se složeni sustav dekomponira na više razina, njegovu koordinaciju je moguće izvesti tako da se koordinacija sekvencijalno fokusira na spreg između dvije razine odnosno para podsustava te se po njihovom rješavanju spuštaju (ili dižu) za jednu razinu gdje se ponovno rješava koordinacijski problem. Drugi način rješavanja sustava s više od dvije razine odnosno sustava s neodjeljivim i spregnutim odlukama gdje je potrebna koordinacija više od dva podsustava, je višerazinskom koordinacijom (*multi-level coordination*). Višerazinska koordinacija višerazinskih sustava se praksi nešto rjeđe primjenjuje jer zahtjeva rješavanje problema velikog broja jednadžbi koje nastaju dodavanjem ograničenja konzistentnosti svih susjednih podsustava u optimizacijski problem te je najčešće jednostavnije problem riješiti dvorazinskim koordinacijskim tehnikama dio po dio. S obzirom da će se u ovom radu primjenjivati samo dekompozicija na dvije razine, u nastavku će se dati pregled dvorazinskih koordinacijskih metoda koje će biti primijenjene i za generiranje Pareto fronte pri projektiranju složenih tankostjenih konstrukcija.

5.3.2. Načini izvođenja koordinacije

Jedna od osnovnih podjela koordinacijskih metoda je prema zadovoljavanju ograničenja konzistentnosti (vidi [52]ili [53]) prema kojoj koje se mogu definirati dvije različite formulacije:

- koordinacija bazirana na ograničenjima jednakosti tj. koordinacija bazirana na modelu (*model-based coordination*),
- koordinacija bazirana na relaksaciji tj. koordinacija orijentirana prema cilju (*goal based coordination*)

Obje formulacije su direktna posljedica načina izvođenja dekompozicije (poglavlje 5.2.1).

Kod koordinacije bazirane na modelu, optimizacijski problem za sustav dekomponiran odozgo definiran je na sljedeći način:

$$
\begin{aligned}
&\min_{{}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}} \ {}^0y\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\\
\text{razina 0:}\quad &t.d. \quad {}^0\mathbf{g}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\le 0\\
&\qquad\quad {}^0\mathbf{g}_e\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)=0\\
&\qquad\quad {}^0\underline{\mathbf{x}}\le{}^0\mathbf{x}\le{}^{0\,-}\mathbf{x};\ {}^0_i\underline{\mathbf{z}}\le{}^0_i\mathbf{z}\le{}^{0\,-}_i\mathbf{z};\ {}^i_0\underline{\mathbf{h}}\le{}^i_0\mathbf{h}\le{}^{i\,-}_0\mathbf{h}
\end{aligned}
\tag{5.8}
$$

$$
\begin{aligned}
&\min_{{}^i\mathbf{x},{}^i_0\mathbf{z}} \ {}^iy\left({}^i\mathbf{x},{}^i_0\mathbf{z},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_0\mathbf{z},{}^0_i\mathbf{h}\right)\right)\\
&t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{z},{}^0_i\mathbf{h}\right)\right)\le 0\\
&\qquad\quad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{z},{}^0_i\mathbf{h}\right)\right)=0\\
\text{razina 1:}\quad &\qquad\quad {}^i\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)=0\\
&\qquad\quad {}^i\mathbf{g}_{{}^0_icz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)=0\\
&\qquad\quad {}^i\underline{\mathbf{x}}\le{}^i\mathbf{x}\le{}^{i\,-}\mathbf{x}\ \ {}^0_i\underline{\mathbf{z}}\le{}^0_i\mathbf{z}\le{}^{0\,-}_i\mathbf{z}\\
&gdje \quad {}^0_i\mathcal{H}\left({}^0\mathbf{r}\right)={}^0_i\mathbf{h}
\end{aligned}
\tag{5.9}
$$

Kod koordinacije orijentirane prema cilju, optimizacijski problem za sustav dekomponiran odozgo definiran je izrazom:

$$
\begin{aligned}
&\min_{{}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}} \ {}^0y\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\\
\text{razina 0:}\quad &t.d. \quad {}^0\mathbf{g}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\le 0\\
&\qquad\quad {}^0\mathbf{g}_e\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)=0\\
&\qquad\quad {}^0\underline{\mathbf{x}}\le{}^0\mathbf{x}\le{}^{0\,-}\mathbf{x};\ {}^0_i\underline{\mathbf{z}}\le{}^0_i\mathbf{z}\le{}^{0\,-}_i\mathbf{z};\ {}^i_0\underline{\mathbf{h}}\le{}^i_0\mathbf{h}\le{}^{i\,-}_0\mathbf{h}
\end{aligned}
\tag{5.10}
$$

$$
\begin{aligned}
&\min_{{}^i\mathbf{x},{}^i_0\mathbf{z}} \ {}^iy\left({}^i\mathbf{x},{}^i_0\mathbf{z},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_0\mathbf{z},{}^0_i\mathbf{h}\right)\right)+\sum{}^i\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r}\right)\right)+\sum{}^i\mathbf{g}_{{}^0_icz}\left({}^0_i\mathbf{c}_z\left({}^i_0\mathbf{z}\right)\right)\\
&t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{z},{}^0_i\mathbf{h}\right)\right)\le 0\\
&\qquad\quad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{z},{}^0_i\mathbf{h}\right)\right)=0\\
\text{razina 1:}\quad &\qquad\quad {}^i\underline{\mathbf{x}}\le{}^i\mathbf{x}\le{}^{i\,-}\mathbf{x};\ {}^0_i\underline{\mathbf{z}}\le{}^0_i\mathbf{z}\le{}^{0\,-}_i\mathbf{z}\\
&gdje \quad {}^i\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r}\right)\right)={}^i_0\boldsymbol{\lambda}^T\left({}^i_0\mathcal{H}\left({}^i\mathbf{r}\right)-{}^i_0\mathbf{h}\right)\\
&\qquad\quad {}^i\mathbf{g}_{{}^0_icz}\left({}^0_i\mathbf{c}_z\left({}^i_0\mathbf{z}\right)\right)={}^i_0\boldsymbol{\lambda}^T\left({}^0_i\mathbf{z}-{}^i_0\mathbf{z}\right)
\end{aligned}
\tag{5.11}
$$

Može se primijetiti da su optimizacijski problemi na globalnoj razini identični kod oba načina koordinacije, dok je na razini 1 kod koordinacije orijentirane prema cilju, cilj optimizacije modificiran s relaksiranim ograničenjima konzistentnosti. Kao što je već navedeno u prethodnom potpoglavlju, hijerarhijski dekomponirani problem ne uspijeva konvergirati prema realnom rješenju ukoliko se ograničenja konzistentnosti tj. propisani odzivi ili zavisni deskriptori ne mogu zadovoljiti u podrazinama, bilo zbog gornjih i donji granica varijabli ili propisanih ograničenja. Kako bi se osiguralo zadovoljenje svih ograničenja na podrazinama, informacija o ograničenjima sustava na podrazini se mora na neki način prenijeti u nadređeni odnosno globalni sustav. U poglavlju 5.4 bit će objašnjeno kako taj problem rješavaju neke od češće korištenih koordinacijskih metoda.

U [59] dane su klasifikacije metoda za dekompoziciju i koordinaciju koje uključuju i nešto preciznije svrstavanje metoda. Prema autorima podjela je prema relaksaciji ograničenja koja može biti: bez relaksacije i s relaksacijom projektnih ograničenja ili ograničenja konzistentnosti. Ograničenja koja su relaksirana i čije je zadovoljavanje zahtijevano samo na kraju sekvence koordinacijske metode se još nazivaju otvorena, a ograničenja čije je zadovoljavanje imperativ u svakoj iteraciji sekvence koordinacijske metode nazivaju se zatvorena. Pri tome se posebno razmatraju projektna ograničenja pojedinog optimizacijskog problema i ograničenja konzistentnosti. Stoga metode mogu imati otvorena/zatvorena projektna ograničenja (g, g_e) i/ili imati otvorena/zatvorena ograničenja konzistentnosti (g_c).

U istom radu autori naglašavaju važnost i podijele metoda prema redoslijedu izvršavanja pojedinih optimizacijskih problema te ih dijele i na ugniježđene (*nested*) i naizmjenične (*alternating*).

5.4. Koordinacijske metode

U ovom potpoglavlju dane su osnovne formulacije većinom dvorazinskih koordinacijskih metoda, koje prema svojim karakteristikama i dosadašnjim primjenama izgledaju obećavajuće glede primjene za projektiranje složenih tankostjenih konstrukcija.

Za rješavanje višedisciplinarnih spregnutih optimizacijskih problema (MDO – Multidisciplinary Design Optimization) još uvijek se često koriste tradicionalne jednorazinske višedisciplinarne metode kao što su MDF (*Multi-Disciplinary Feasible),* IDF (*Individual Disciplinary Feasible*) te AAO (*All-at-Once)*. Osim toga ove metode često se koriste i kao referenca za usporedbu s modernim višerazinskim metodama. Stoga je i dalje važno poznavanje osnovnih ideja i karakteristika ovih metoda, te će se na samom početku prvo dati osnove ovih metoda.

Nakon tradicionalnih MDO metoda navedene su metode koordinacije bazirane na ograničenjima jednakosti, iza kojih slijede metode bazirane na relaksaciji. Formulacije metoda i veći dio opisa preuzete su iz [54] gdje je se mogu pronaći i još neke metode koje nisu prikazane u ovom radu.

Sve metode bazirane na modelu, kao i neke metode koordinacije orijentirane prema cilju koriste neki oblik aproksimacije odnosno surogat modeliranja veličina iz spregnutih sustava. Aproksimirane veličine su označene krovićem iznad oznake veličine, npr. $^{i}\hat{\mathbf{g}}$ je aproksimacija ograničenja nejednakosti sustava *i*. Ovisno o metodi aproksimirane veličine mogu biti: vrijednosti funkcije cilja, vrijednosti ograničenja, vrijednosti margine ograničenja

(udaljenosti od nepodobnog područja), spregnuti odzivi susjednih sustava. Korišteni surogat modeli mogu biti lokalni koji se generiraju kroz svaku iteraciju usklađivanja razina te globalni surogat modeli koji se generiraju jednom, a tijekom optimizacije se eventualno osvježavaju. Najčešći lokalni surogat modeli su zapravo linearni modeli određenih veličina konstruirani za trenutno stanje složenog sustava. Više o aproksimacijskim metodama, koje se koriste pri projektiranju složenih sustava, u poglavlju 4.

5.4.1. Tradicionalne jednorazinske MDO metode

Pod jednorazinskom formulacijom podrazumijeva se da je čitav problem projektiranja zadanog složenog sustava formuliran u okviru jednog optimizacijskog potproblemom te da ga rješava jedna optimizacijska metoda tj. jedan optimizator.

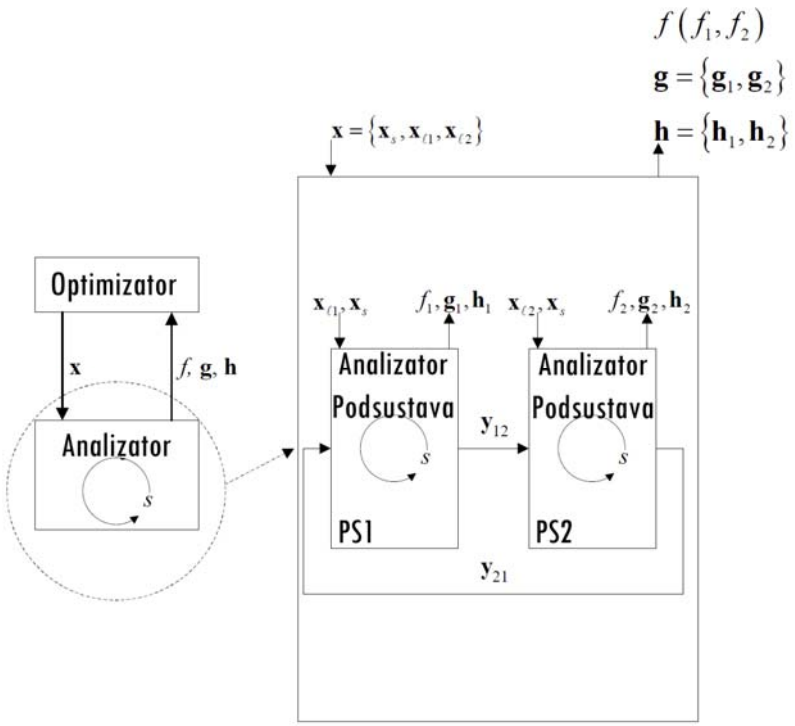
Tri osnovne jednorazinske MDO su:

1. MDF - *Multi-Disciplinary Feasible*, poznata još i kao "*Nested Analysis And Design*" (NAND), „*Single-NAND-NAND*" (SNN), „*All-in-One*" (AIO), i „*One-at-a-Time*".
2. IDF - *Individual Disciplinary Feasible*, poznata još i kao "*Simultaneous Analysis And Design*" (SAND) ili *Single-SAND-NAND*.
3. AAO - All-at-Once, poznata još i kao "*Single*-SAND-SAND", ili ponekad samo kao SAND. Ponekad se izraz AAO koristi kao oznaka za AIO ali u ovom slučaju to su različiti pojmovi.

U nastavku je dan opis jednorazinskih metoda prema radu [60].

5.4.1.1. Multi-Disciplinary Feasible

MDF formulacija je prikazana na Sl. 5.17. Primjenjuje se jedan optimizator na razini sustava i ako se sagleda sa razine optimizacije MDF se ne razlikuje od standardnog problema optimalnog projekta. Analizator sustava koordinira analizatore svih podsustava. Optimizator daje analizatoru sustav vektora projektne varijable x, a analizator vraća odgovarajuće povratne funkcije f, g i h.



Sl. 5.17. MDF arhitektura [60]

Kako bi se neka strategija formulacije mogla klasificirati kao MDF ona mora za svaku iteraciju izvršiti kompletnu sistemsku analizu sustava. Analiza je ugniježđena (*nested*) unutar projektiranja (otuda i naziv NAND). Optimizator je odgovoran za pronalazak optimalnog rješenja x*, dok je analizator sustava odgovoran samo za pronalazak skupa konzistentnih spregnutih varijabli y.

Definicija MDF problema je dana jednadžbom (5.12). MDF po prirodi nema hijerarhiju. U isključivo računalnom kontekstu ovakva struktura je povoljna ako su podsustavi vrlo slabo spregnuti (brza konvergencija analize) i ako izračun podsustava ne traži velike računalne resurse. U kontekstu organizacije, MDF omogućuje nastavak korištenja već postojećih alata za analizu bez ikakvih modifikacija. Ako je organizacijski definirano da se prvo izvodi sistemska analiza prije donošenja neke projektne odluke onda se može reći da je MDF prirodno podoban.

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{x}_\ell, \mathbf{x}_s]} \quad & f(\mathbf{x}) \\ \text{uz} \quad & \mathbf{g}(\mathbf{x}) = [\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_s}] \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_s}] = \mathbf{0} \end{aligned} \tag{5.12}$$

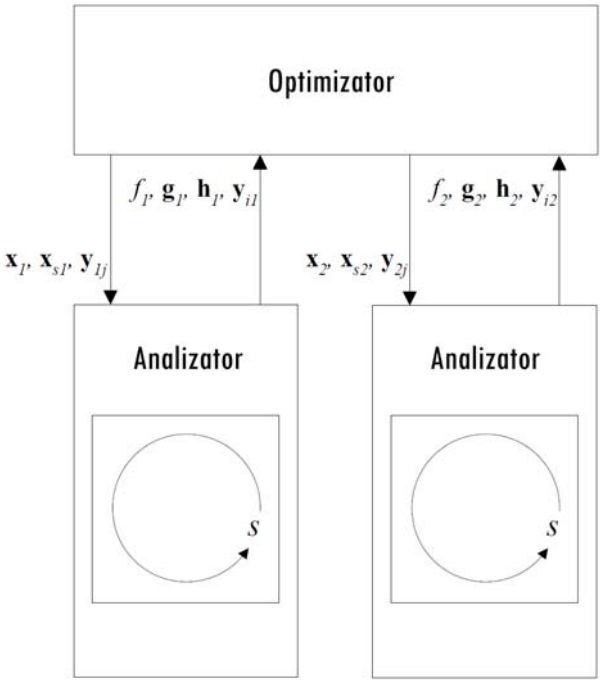
Naravno, potrebno je i u potpunosti razumjeti nedostatke MDF formulacije. Ovisi o učinkovitosti i robusnosti analize. Ako analiza ne konvergira i za samo jedno projektnu točku, optimizacija može biti nasilno prekinuta. MDF se ne može paralelizirati što rezultira značajnim produljenjem ciklusa projektiranja. Konačno, nužno je izvršiti više kompletnih analiza kako bi se

došlo do konačne razlike, za eventualnu primjenu algoritma optimizacije zasnovanom na gradijentima. Često se MDF okarakterizira kao „*brute force*“ pristup. Stoga su ovi nedostatci poticaj za razvoj drugih metoda.

5.4.1.2. Individual Disciplinary Feasible

Ovim pristupom se rješava dio ograničenja MDF formulacije. Kao i kod MDF formulacije, koristi se zasebna analiza za svaki podsustav, dok se jedna optimizacijska metoda primjenjuje na razini sustava.

Ključna razlika je u tome što optimizator koordinira interakciju između podsustava. Ovime ostvarujemo mogućnost paralelizacije, poboljšavaju se svojstva konvergencije i teži se boljim rješenjima projekta ukoliko postoji više rješenja analize. Osim toga, IDF ima puno veću robusnost od MDF formulacije.



Sl. 5.18 IDF arhitektura [60]

Arhitektura IDF metode je prikazana na Sl. 5.18. Optimizator odabire vrijednosti projektnih i spregnutih varijabli. Razlika između analize i projektiranja je nejasna jer se odvijaju istovremeno. Optimizator sustava dobavlja sve potrebne ulazne podatke za sve podsustave. Zbog toga, pojedini podsustavi ne moraju čekati izlazne podatke analize iz nekog drugog podsustava već se mogu analizirati paralelno. Kao i kod MDF metode, projektne odluke su centralizirane dok je analiza distribuirana.

Kako bi se očuvala konzistentnost sustava pri konvergenciji, uvode se dodatna ograničenja koja osiguravaju da su spregnute varijable izračunate u podsustavu jednake onima koje daje optimizator. Ako se ovaj proces prekine, prijelazno rješenje može biti nekonzistentno i/ili nepodobno. Kod MDF formulacije prekid daje konzistentno ali nepodobno rješenje.

Ova dodatna rješenja služe i kako bi se prekinule bilo kakve nehijerarhijske veze u sustavu čime se omogućava da se problem riješi na hijerarhijski način. Proizvoljan nehijerarhijski sustav možemo pomoću dodatnih ograničenja prikazati kao hijerarhijski čime se omogućuje primjena metoda namijenjenih za isključivo hijerarhijske sustave.

Naziv metode dolazi od činjenice da svaki podsustav zadovoljava svoje vlastite jednadžbe pri svakoj iteraciji dok je sustav konzistentan samo pri konvergenciji. IDF je više centraliziran od MDF metode i dimenzija optimizacijskog problema je veća. Mapa IDF formulacije odgovara organizaciji sa jednim upraviteljem projekta koji donosi sve projektne odluke i usuglašava grupe za analizu.

Jednadžba (5.13) daje zapis IDF formulacije. Slična je MDF formulaciji uz razliku varijable odluke uključuju projektne varijable x i spregnute varijable y kao i dodatak pomoćnih ograničenja $h_{aux}(x,y)$ koja osiguravaju konzistentnost sustava.

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{x}_\ell,\mathbf{x}_s],\mathbf{y}} \quad & f(\mathbf{x},\mathbf{y}) \\ \text{uz} \quad & \mathbf{g}(\mathbf{x},\mathbf{y}) = [\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_s}] \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x},\mathbf{y}) = [\mathbf{h_1},\mathbf{h_2},\ldots,\mathbf{h_s}] = \mathbf{0} \\ & \mathbf{h_{aux}}(\mathbf{x},\mathbf{y}) = \mathbf{y}(\mathbf{x},\mathbf{y}) - \mathbf{y} = \mathbf{0} \end{aligned} \tag{5.13}$$

U organizacijskom i računalnom kontekstu paralelnost IDF metode ima prednost nad sekvencijalnom prirodom MDF pristupa. Ako su dostupni paralelni alati analize, kao što su paralelni procesori ili više skupina za analizu, IDF može znatno skratiti vrijeme procesa projektiranja. Ako je dopuštena visoka razina centralizacije onda IDF može poslužiti kao idealna projektna strategija.

5.4.1.3. All-at-Once

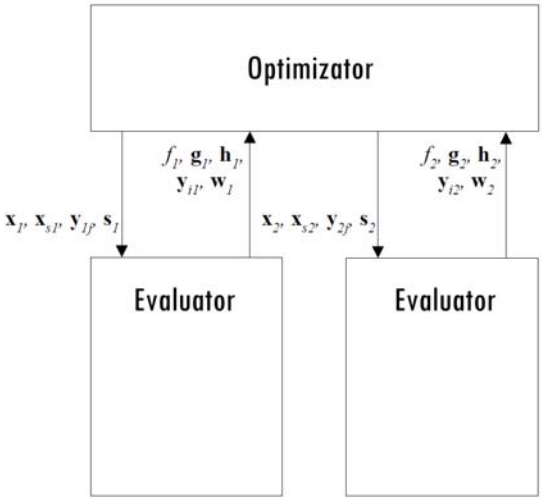
AAO ima vrlo centralizirani pristup. Umjesto da koristi neku rutinu analize za ocjenu svakog podsustava, koriste se evaluatori koji samo računaju ostatke (reziduale) jednadžbi podsustava. Optimizacija sustava sada ima tri kompleta varijabli:

1. Originalne projektne varijable **x**
2. Spregnute varijable **y**
3. Varijable stanja **s** (kao što su polja brzina, naprezanja itd.)

AAO centralizira i projektiranje i analizu, ali distribuira evaluaciju jednadžbi. Ovo može rezultirati visokom učinkovitošću, ali je teško mapirati na neku organizacijsku strukturu ili simulacijski alat zbog centralizacije i specifične strukture.

Arhitektura AAO je prikazana na Sl. 5.19. Vektor varijable stanja je podijeljen na varijable stanja za svaki podsustav ($\mathbf{s}=[\mathbf{s_1}, \mathbf{s_2},..., \mathbf{s_N}]$) i reziduali za svaki podsustav w_i se predaju optimizacijskoj rutini uz ostale bitne vrijednosti.

Formulacija AAO pristupa je dana jednadžbom (5.14). Slična je IDF formulaciji ali uključuje dodatno pomoćno ograničenje koje osigurava nulti rezidual kod konvergencije problema, a varijable uključuju **x**, **y** i **s**. Ovaj pristup se naziva „Sve odjednom“ (*All-At-Once*) zbog toga što se projektiranje, sistemska analiza i analiza podsustava odvijaju istovremeno.



Sl. 5.19 AAO arhitektura [60]

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{x}_\ell,\mathbf{x}_\mathbf{s}],\mathbf{y},\mathbf{s}} \quad & f(\mathbf{x},\mathbf{y}) \\ \text{uz} \quad & \mathbf{g}(\mathbf{x},\mathbf{y},\mathbf{s}) = [\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_s}] \le \mathbf{0} \\ & \mathbf{h}(\mathbf{x},\mathbf{y},\mathbf{s}) = [\mathbf{h_1},\mathbf{h_2},\ldots,\mathbf{h_s}] = \mathbf{0}. \\ & \mathbf{h_{aux}}(\mathbf{x},\mathbf{y},\mathbf{s}) = \left\{ \begin{array}{c} \mathbf{y}(\mathbf{x},\mathbf{y},\mathbf{s}) - \mathbf{y} \\ \mathbf{w}(\mathbf{x},\mathbf{y},\mathbf{s}) \end{array} \right\} = \mathbf{0}. \end{aligned} \tag{5.14}$$

5.4.2. Optimizacija linearnom dekompozicijom

Optimizacija linearnom dekompozicijom [61] spada u grupu metoda koje rješavaju hijerarhijski dekomponirani problem korištenjem formulacije prema izrazima (5.8) i (5.9).

Kako bi se osiguralo zadovoljavanje ograničenja konzistentnosti na podrazini, ova metoda koristi totalnu derivaciju po spregnutim odzivima sustava s nivoa 1 $\mathbf{D}_{\mathbf{i}_0\mathbf{h}}\left({}^{i}\mathbf{g}\right)$ svih

ograničenja s nivoa 1 za generiranje linearnog surogat modela svih ograničenja koji se dodaje u globalni optimizacijski problem na razini 0.

$$\mathbf{D}_{{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{g}\right)=\frac{\partial}{\partial\,{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{g}\right)+\left(\frac{\partial}{\partial\,{}^{i}\mathbf{x}}\left({}^{i}\mathbf{g}\right)\right)^{T}\cdot\mathbf{D}_{{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{x}\right) \tag{5.15}$$

Formulacija optimizacijskog problema, kod kojeg nema zavisnih varijabli već samo spregnuti odzivi, nalazi se u izrazima (5.16) do (5.18):

$$\begin{aligned}
&\min_{{}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h}}\ {}^{0}y\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h}\right)\right)\\
&t.d.\quad {}^{0}\mathbf{g}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h}\right)\right)\le 0\\
&\qquad {}^{0}\mathbf{g}_{e}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h}\right)\right)=0\\
\text{razina 0:}\qquad &{}^{i}\hat{\mathbf{g}}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)\le 0\\
&{}^{i}\hat{\mathbf{g}}_{e}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)=0\\
&{}^{i}\hat{\bar{\mathbf{x}}}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)\le 0\\
&{}^{i}\hat{\underline{\mathbf{x}}}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)\le 0\\
&{}^{0}\underline{\mathbf{x}}\le{}^{0}\mathbf{x}\le{}^{0}\bar{\mathbf{x}};\ {}_{0}^{i}\underline{\mathbf{h}}\le{}_{0}^{i}\mathbf{h}\le{}_{0}^{i}\bar{\mathbf{h}}
\end{aligned} \tag{5.16}$$

gdje je

$$\begin{aligned}
&{}^{i}\hat{\mathbf{g}}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)={}^{i}\mathbf{g}+\mathbf{D}_{{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{g}\right)\Delta_{0}^{i}\mathbf{h}+\mathbf{D}_{{}_{i}^{0}\mathbf{h}}\left({}^{i}\mathbf{g}\right)\left({}_{i}^{0}\mathcal{H}\left({}^{0}\mathbf{r}\right)-{}_{i}^{0}\mathbf{h}\right)\\
&{}^{i}\hat{\mathbf{g}}_{e}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)={}^{i}\mathbf{g}_{e}+\mathbf{D}_{{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{g}_{e}\right)\Delta_{0}^{i}\mathbf{h}+\mathbf{D}_{{}_{i}^{0}\mathbf{h}}\left({}^{i}\mathbf{g}_{e}\right)\left({}_{i}^{0}\mathcal{H}\left({}^{0}\mathbf{r}\right)-{}_{i}^{0}\mathbf{h}\right)\\
&{}^{i}\hat{\bar{\mathbf{x}}}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)=-{}^{i}\bar{\mathbf{x}}+{}^{i}\mathbf{x}+\mathbf{D}_{{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{x}\right)\Delta_{0}^{i}\mathbf{h}+\mathbf{D}_{{}_{i}^{0}\mathbf{h}}\left({}^{i}\mathbf{x}\right)\left({}_{i}^{0}\mathcal{H}\left({}^{0}\mathbf{r}\right)-{}_{i}^{0}\mathbf{h}\right)\\
&{}^{i}\hat{\underline{\mathbf{x}}}\left({}^{0}\mathbf{x},{}_{0}^{i}\mathbf{h},{}^{0}\mathbf{r}\right)={}^{i}\underline{\mathbf{x}}-{}^{i}\mathbf{x}-\mathbf{D}_{{}_{0}^{i}\mathbf{h}}\left({}^{i}\mathbf{x}\right)\Delta_{0}^{i}\mathbf{h}-\mathbf{D}_{{}_{i}^{0}\mathbf{h}}\left({}^{i}\mathbf{x}\right)\left({}_{i}^{0}\mathcal{H}\left({}^{0}\mathbf{r}\right)-{}_{i}^{0}\mathbf{h}\right)
\end{aligned} \tag{5.17}$$

$$\begin{aligned}
&\min_{{}^{i}\mathbf{x},{}_{0}^{i}\mathbf{z}}\ {}^{i}y\left({}^{i}\mathbf{x},{}_{0}^{i}\mathbf{z},{}^{i}\mathbf{r}\left({}^{i}\mathbf{x},{}_{0}^{i}\mathbf{z},{}_{i}^{0}\mathbf{h}\right)\right)\\
&t.d.\quad {}^{i}\mathbf{g}\left({}^{i}\mathbf{x},{}^{i}\mathbf{r}\left({}^{i}\mathbf{x},{}_{i}^{0}\mathbf{z},{}_{i}^{0}\mathbf{h}\right)\right)\le 0\\
&\qquad {}^{i}\mathbf{g}_{e}\left({}^{i}\mathbf{x},{}^{i}\mathbf{r}\left({}^{i}\mathbf{x},{}_{i}^{0}\mathbf{z},{}_{i}^{0}\mathbf{h}\right)\right)=0\\
\text{razina 1:}\qquad &{}^{i}\mathbf{g}_{{}_{0}^{i}c}\left({}_{0}^{i}\mathbf{c}\left({}^{i}\mathbf{r},{}_{0}^{i}\mathbf{h}\right)\right)=0\\
&{}^{i}\mathbf{g}_{{}_{i}^{0}cz}\left({}_{i}^{0}\mathbf{c}_{z}\left({}_{i}^{0}\mathbf{z},{}_{0}^{i}\mathbf{z}\right)\right)=0\\
&{}^{i}\underline{\mathbf{x}}\le{}^{i}\mathbf{x}\le{}^{i}\bar{\mathbf{x}}\ {}_{i}^{0}\underline{\mathbf{z}}\le{}_{i}^{0}\mathbf{z}\le{}_{i}^{0}\bar{\mathbf{z}}\\
&gdje\quad {}_{i}^{0}\mathcal{H}\left({}^{0}\mathbf{r}\right)={}_{i}^{0}\mathbf{h}
\end{aligned} \tag{5.18}$$

Vidljivo je da je u globalni optimizacijski problem dodana značajna količina lineariziranih ograničenja definiranih izrazom (5.17) svih sustava iz podrazine. Jasno je da je zbog linearizacije ograničenja razine 1 rezultat optimizacije razine 0 točan samo u blizini točke oko koje je linearizacija izvršena te je postupak potrebno sekvencijalno ponavljati kako bi se konvergiralo prema optimumu ukupnog sustava. Za smanjenje broja dodatnih linearnih ograničenja [62]

preporučuje korištenje ovojnice ograničenja tzv. KS funkcijom kojom su sva ograničenja iz podrazine kombinirana u jednu funkciju koja je potom linearizirana.

5.4.3. Linearizirana višerazinska dekompozicija

Linearizirana višerazinska dekompozicija (*Linearized multi-level optimization*) [58] rješava glavni nedostatak uočen kod metode predstavljene u prethodnom poglavlju, a to su nelinearna ograničenja konzistentnosti u optimizacijskom problemu na razini 1.

$$\text{razina 0:}\quad \begin{aligned} &\min_{{}^0\mathbf{x}} \; {}^0y\left({}^0\mathbf{x},{}^0\mathbf{r}\left({}^0\mathbf{x}\right)\right) \\ &t.d. \quad {}^0\mathbf{g}\left({}^0\mathbf{x},{}^0\mathbf{r}\left({}^0\mathbf{x}\right)\right)\le 0 \\ &\qquad {}^0\mathbf{g}_e\left({}^0\mathbf{x},{}^0\mathbf{r}\left({}^0\mathbf{x}\right)\right)= 0 \\ &\qquad {}^i\hat{\mathbf{g}}\left({}^0\mathbf{x},{}^0\mathbf{r}\right)\le 0 \\ &\qquad {}^i\hat{\mathbf{g}}_e\left({}^0\mathbf{x},{}^0\mathbf{r}\right)= 0 \\ &\qquad {}^0\underline{\mathbf{x}}\le {}^0\mathbf{x}\le {}^0\overline{\mathbf{x}} \end{aligned} \tag{5.19}$$

$$\text{gdje je}\quad \begin{aligned} {}^i\hat{\mathbf{g}}\left({}^0\mathbf{x},{}^0\mathbf{r}\right) &= {}^i\mathbf{g}+\mathbf{D}_{{}_0^i\mathbf{h}}\left({}^i\mathbf{g}\right)\left({}_i^0\mathcal{H}\left({}^0\mathbf{r}\right)-{}_i^0\mathbf{h}\right) \\ {}^i\hat{\mathbf{g}}_e\left({}^0\mathbf{x},{}^0\mathbf{r}\right) &= {}^i\mathbf{g}_e+\mathbf{D}_{{}_0^i\mathbf{h}}\left({}^i\mathbf{g}_e\right)\left({}_i^0\mathcal{H}\left({}^0\mathbf{r}\right)-{}_i^0\mathbf{h}\right) \end{aligned} \tag{5.20}$$

$$\text{razina 1:}\quad \begin{aligned} &\min_{{}^i\mathbf{x}} \; {}^iy\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x}\right)\right) \\ &t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x}\right)\right)\le 0 \\ &\qquad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x}\right)\right)= 0 \\ &\qquad {}^0\hat{\mathbf{g}}\left({}^1\mathbf{x},{}^1\mathbf{r}\right)\le 0 \\ &\qquad {}^0\hat{\mathbf{g}}_e\left({}^1\mathbf{x},{}^1\mathbf{r}\right)= 0 \\ &\qquad {}^i\underline{\mathbf{x}}\le {}^i\mathbf{x}\le {}^i\overline{\mathbf{x}} \end{aligned} \tag{5.21}$$

$$\text{gdje je}\quad \begin{aligned} {}^0\hat{\mathbf{g}}\left({}^0\mathbf{x},{}^0\mathbf{r}\right) &= {}^0\mathbf{g}+\mathbf{D}_{{}_0^i\mathbf{h}}\left({}^0\mathbf{g}\right)\left({}_0^i\mathcal{H}\left({}^i\mathbf{r}\right)-{}_0^i\mathbf{h}\right) \\ {}^0\hat{\mathbf{g}}_e\left({}^0\mathbf{x},{}^0\mathbf{r}\right) &= {}^0\mathbf{g}_e+\mathbf{D}_{{}_0^i\mathbf{h}}\left({}^0\mathbf{g}_e\right)\left({}_0^i\mathcal{H}\left({}^i\mathbf{r}\right)-{}_0^i\mathbf{h}\right) \end{aligned} \tag{5.22}$$

Odmah se može primijetiti da se na razini 0 spregnuti odzivi sustava iz razine 1 više ne tretiraju kao varijabla te zbog toga nije potrebno računati totalnu derivaciju $\mathbf{D}_{{}_0^i\mathbf{h}}\left({}^i\mathbf{g}\right)$ već samo $\mathbf{D}_{{}_i^0\mathbf{h}}\left({}^i\mathbf{g}\right)$. Takav pristup bitno pojednostavljuje proračun, pogotovo kad se derivacije računaju numerički.

Značajna razlika formulacije optimizacijskih problema na razini 1 u odnosu na optimizaciju linearnom dekompozicijom je što su nelinearna ograničenja jednakosti (ograničenja konzistentnosti) zamijenjena s linearnim surogat modelom ograničenja iz globalnog sustava koji također ovisi samo o derivativima $\mathbf{D}_{{}_0^i\mathbf{h}}\left({}^0\mathbf{g}\right)$.

Autori naglašavaju da je metoda pokazala da linearne aproksimacije u ovom pristupu omogućavaju značajno veće korake promjene varijabli između iteracija od metode is prethodnog potpoglavlja, čime se značajno reducira i vrijeme optimizacije. Nedostatak je naravno što se konzistentnost ne osigurava na matematički precizan način.

5.4.4. Kvazi-odvojiva dekompozicija

Kvazi-odvojiva dekompozicija (*Quasi-separable Decomposition*) ([63] i [64]), također spada u grupu metoda koje rješavaju hijerarhijski dekomponirani problem korištenjem formulacije prema izrazima (5.8) i (5.9). Kod ove metode koordinacije u optimizacijske probleme podsustava ne razini 1 uvodi se tzv. *slack* varijabla koja mjeri minimalnu marginu ograničenja tj. minimalnu udaljenost trenutne vrijednosti najgoreg ograničenja od granice zadovoljenosti. Cilj optimizacije podsustava na razini 1 je maksimizacija minimalne margine ograničenja. Formulacija optimizacijskog problema dana je izrazima (5.23) do (5.25):

$$\text{razina 0:}\quad \begin{aligned} &\min_{{}^0\mathbf{x},{}^i_0\mathbf{h}} \; {}^0y\left({}^0\mathbf{x},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^i_0\mathbf{h}\right)\right) \\ &t.d. \quad {}^0\mathbf{g}\left({}^0\mathbf{x},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^i_0\mathbf{h}\right)\right)\le 0 \\ &\qquad {}^0\mathbf{g}_e\left({}^0\mathbf{x},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^i_0\mathbf{h}\right)\right)=0 \\ &\qquad {}^i\hat{y}\left({}^0\mathbf{x},{}^i_0\mathbf{h},{}^0\mathbf{r}\right)\le 0 \\ &\qquad {}^0\underline{\mathbf{x}}\le{}^0\mathbf{x}\le{}^0\overline{\mathbf{x}};\;\; {}^i_0\underline{\mathbf{h}}\le{}^i_0\mathbf{h}\le{}^i_0\overline{\mathbf{h}} \end{aligned} \tag{5.23}$$

$$\text{gdje je}\quad {}^i\hat{y}\left({}^0\mathbf{x},{}^i_0\mathbf{h},{}^0\mathbf{r}\right)={}^i\hat{y}_G\left({}^i\mu\left({}^i_0\mathbf{h},{}^0\mathbf{r}\right)\right) \tag{5.24}$$

$$\text{razina 1:}\quad \begin{aligned} &\min_{{}^i\mathbf{x},{}^i\mu} \; {}^iy={}^i\mu \\ &t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{h}\right)\right)\le{}^i\mu \\ &\qquad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{h}\right)\right)=0 \\ &\qquad {}^i\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)={}^i_0\mathcal{H}\left({}^i\mathbf{r}\right)-{}^i_0\mathbf{h}=0 \\ &\qquad \underline{\mathbf{x}}\le{}^i\mathbf{x}\le{}^i\overline{\mathbf{x}} \end{aligned} \tag{5.25}$$

Problem razine 1 se zapravo rješava za određeni raspon spregnutih odziva ${}^0_i\mathcal{H}\left({}^0\mathbf{r}\right)={}^0_i\mathbf{h}$ i ${}^i_0\mathbf{h}$ čime se dobiva skup optimalnih rješenja ${}^i\mu^{opt}\left({}^i_0\mathbf{h},{}^0_i\mathcal{H}\left({}^0\mathbf{r}\right)\right)$ na osnovu kojih se generira globalni surogat model (5.24). U oba gore navedena rada korištena je metoda odzivnih ploha, koja dobro globalno aproksimira samo u slučaju relativne monotonosti margine ograničenja. Ukoliko taj uvjet nije ispunjen autori preporučuju generiranje više surogata koji pokrivaju različita područja domene. Taj surogat model se koristi pri optimizaciji na globalnoj razini (5.23), za osiguravanje zadovoljavanja ograničenja sustava na podrazini. Može se primijetiti da surogat model, slično kao i KS funkcija, reducira broj ograničenja na jedno po podsustavu iz podrazine, no za razliku od KS funkcije polinomski surogat model osigurava derivabilnost u svim točkama promatrane domene.

5.4.5. Združena optimizacija

Združena optimizacija (*Collaborative optimization – CO*) ([65], [66]) spada u grupu metoda koja osiguravaju konzistentnost relaksacijom ograničenja. Sama formulacija optimizacijskog problema združene optimizacije za sustave dekomponirane odozgo dana je izrazima (5.26) do(5.29), zapisana prema [67], jer je u izvornim radovima nejasno formulirana.

$$\text{razina 0:}\quad \begin{aligned} &\min_{{}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}} \ {}^0y\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\\ &t.d. \quad {}^0\mathbf{g}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\le 0\\ &\qquad {}^0\mathbf{g}_e\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)=0\\ &\qquad {}^0\mathbf{g}_{{}^i_0 c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r}\right)\right)=0\\ &\qquad {}^0\mathbf{g}_{{}^0_i cz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)=0\\ &\qquad {}^0\underline{\mathbf{x}}\le{}^0\mathbf{x}\le{}^0\overline{\mathbf{x}};\ {}^0_i\underline{\mathbf{z}}\le{}^0_i\mathbf{z}\le{}^0_i\overline{\mathbf{z}};\ {}^i_0\underline{\mathbf{h}}\le{}^i_0\mathbf{h}\le{}^i_0\overline{\mathbf{h}} \end{aligned} \tag{5.26}$$

$$\text{gdje je}\quad \begin{aligned} &{}^0\mathbf{g}_{{}^i_0 c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)=\sum\left({}^i_0\mathcal{H}\left({}^i\mathbf{r}\right)-{}^i_0\mathbf{h}\right)^2\\ &{}^0\mathbf{g}_{{}^0_i cz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)=\sum\left({}^0_i\mathbf{z}-{}^i_0\mathbf{z}\right)^2 \end{aligned} \tag{5.27}$$

$$\text{razina 1:}\quad \begin{aligned} &\min_{{}^i\mathbf{x},{}^i_0\mathbf{z}} \ {}^iy\left({}^i\mathbf{x},{}^i_0\mathbf{z},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_0\mathbf{z},{}^0_i\mathbf{h}\right)\right)={}^ig_{{}^i_0 c}+{}^ig_{{}^0_i cz}\\ &t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{z},{}^0_i\mathbf{h}\right)\right)\le 0\\ &\qquad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^0_i\mathbf{z},{}^0_i\mathbf{h}\right)\right)=0\\ &\qquad {}^i\underline{\mathbf{x}}\le{}^i\mathbf{x}\le{}^i\overline{\mathbf{x}}\ \ {}^0_i\underline{\mathbf{z}}\le{}^0_i\mathbf{z}\le{}^0_i\overline{\mathbf{z}} \end{aligned} \tag{5.28}$$

$$\text{gdje je}\quad \begin{aligned} &{}^i\mathbf{g}_{{}^i_0 c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)=\sum\left({}^i_0\mathcal{H}\left({}^i\mathbf{r}\right)-{}^i_0\mathbf{h}\right)^2\\ &{}^i\mathbf{g}_{{}^0_i cz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)=\sum\left({}^0_i\mathbf{z}-{}^i_0\mathbf{z}\right)^2 \end{aligned} \tag{5.29}$$

Potrebno je naglasiti da je metoda prvenstveno razvijena za nehijerahijske sustave u multidisciplinarnoj optimizaciji u kojima postoji spreg sustava na istom nivou, a globalni sustav je zadužen za njihovu koordinaciju te prikupljanje odziva svih podsustava koji su potrebni za definiranje funkcije cilja ukupnog problema. Formulacija globalnog sustava dana u izrazu (5.26) još se naziva i CO_2 formulacija jer ograničenja konzistentnosti predstavljaju sumu kvadrata odstupanja. Za navedenu formulaciju dokazano je u [67] da ne zadovoljava KKT uvjete optimalnosti što predstavlja veliki nedostatak ukoliko se primjenjuju optimizacijske metode bazirane na gradijentima (npr. SQP). Navedeni nedostatak je ispravljen s CO_1 formulacijom, u kojoj su ograničenja konzistentnosti bez kvadrata, no ta formulacija u određenim problemima izaziva probleme s konvergencijom. Osnovna prednost metode združene optimizacije je jednostavnost implementacije što objašnjava iznimnu popularnost metode u MDO-u. Metoda se može svrstati u metode koordinacije orijentirane prema cilju jer ograničenja konzistentnosti nisu prisutna u podsustavima već samo u glavnom/globalnom sustavu.

5.4.6. Analitičko propagiranje meta (ATC - Analytical Target Cascading)

Analitičko propagiranje meta – ATC [68] je hijerarhijska, višerazinska metoda za projektiranje složenih sustava. ATC formalizira proces propagiranja zadanih meta (*target*), odnosno zadanih spregnutih odziva i zavisnih varijabli, kroz sustave i podsustave. Kako bi se u potpunosti formulirala metoda potrebne su tri razine, no naravno metodu je moguće koristiti i na dvije razine. Pri tome nulta ili globalna razina te posljednja razina imaju različitu formulaciju problema od međurazina. Ukoliko se problem dekomponira na samo dvije razine to znači da su aktivne upravno globalna razina i zadnja razina. Formulacija prema [69] uključuje težinske faktore za pojedinu komponentu razlike mete i ostvarenih vrijednosti te je za sustave dekomponirane odozgo dana izrazima (5.30) do (5.34) :

razina 0:

$$
\begin{aligned}
&\min_{{}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},\boldsymbol{\varepsilon}_c,\boldsymbol{\varepsilon}_{cz}} \; {}^0y\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)+\sum {}^i\boldsymbol{\varepsilon}_c+\sum {}^i\boldsymbol{\varepsilon}_{cz} \\
&t.d. \quad {}^0\mathbf{g}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)\le 0 \\
&\qquad {}^0\mathbf{g}_e\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h},{}^0\mathbf{r}\left({}^0\mathbf{x},{}^0_i\mathbf{z},{}^i_0\mathbf{h}\right)\right)=0 \\
&\qquad {}^0\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r}\right)\right)\le {}^i\boldsymbol{\varepsilon}_c \\
&\qquad {}^0\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r}\right)\right)\le {}^i\boldsymbol{\varepsilon}_{cz} \\
&\qquad {}^0\underline{\mathbf{x}}\le {}^0\mathbf{x}\le {}^0\overline{\mathbf{x}};\; {}^0_i\underline{\mathbf{z}}\le {}^0_i\mathbf{z}\le {}^0_i\overline{\mathbf{z}};\; {}^i_0\underline{\mathbf{h}}\le {}^i_0\mathbf{h}\le {}^i_0\overline{\mathbf{h}};\; 0\le \boldsymbol{\varepsilon}_c\le \overline{\boldsymbol{\varepsilon}}_c;\; 0\le \boldsymbol{\varepsilon}_{cz}\le \overline{\boldsymbol{\varepsilon}}_{cz}
\end{aligned}
\tag{5.30}
$$

gdje je

$$
\begin{aligned}
&{}^0\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)={}^i\left\|{}^i_0\mathbf{s}_0\circ\left({}^i_0\mathcal{H}\left({}^i\mathbf{r}\right)-{}^i_0\mathbf{h}\right)\right\|_2^2 \\
&{}^0\mathbf{g}_{{}^0_icz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)={}^i\left\|{}^i_0\mathbf{s}_{z0}\circ\left({}^0_i\mathbf{z}-{}^i_0\mathbf{z}\right)\right\|_2^2
\end{aligned}
\tag{5.31}
$$

razina 1:

$$
\begin{aligned}
&\min_{{}^i\mathbf{x},{}^i_{0j}\mathbf{z},{}^{i,j}_i\mathbf{h},\boldsymbol{\varepsilon}_c,\boldsymbol{\varepsilon}_{cz}} \; \sum {}^0\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)+\sum {}^0\mathbf{g}_{{}^0_icz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)+\sum {}^{i.j}\boldsymbol{\varepsilon}_c+\sum {}^{i.j}\boldsymbol{\varepsilon}_{cz} \\
&t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{h},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{h}\right)\right)\le 0 \\
&\qquad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{h},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{h}\right)\right)=0 \\
&\qquad {}^i\mathbf{g}_{{}^i_{i.j}c}\left({}^i_{i.j}\mathbf{c}\left({}^{i.j}\mathbf{r},{}^{i.j}_i\mathbf{h}\right)\right)\le {}^{i.j}\boldsymbol{\varepsilon}_c \\
&\qquad {}^i\mathbf{g}_{{}^{i.j}_icz}\left({}^{i.j}_i\mathbf{c}_z\left({}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{z}\right)\right)\le {}^{i.j}\boldsymbol{\varepsilon}_{cz} \\
&\qquad {}^i\underline{\mathbf{x}}\le {}^i\mathbf{x}\le {}^i\overline{\mathbf{x}};\; {}^i_0\underline{\mathbf{z}}\le {}^i_0\mathbf{z}\le {}^i_0\overline{\mathbf{z}};\; {}^i_{i.j}\underline{\mathbf{z}}\le {}^i_{i.j}\mathbf{z}\le {}^i_{i.j}\overline{\mathbf{z}};\; {}^{i.j}_i\underline{\mathbf{h}}\le {}^{i.j}_i\mathbf{h}\le {}^{i.j}_i\overline{\mathbf{h}}; \\
&\qquad 0\le {}^{i.j}\boldsymbol{\varepsilon}_c\le {}^{i.j}\overline{\boldsymbol{\varepsilon}}_c;\; 0\le {}^{i.j}\boldsymbol{\varepsilon}_{cz}\le {}^{i.j}\overline{\boldsymbol{\varepsilon}}_{cz}
\end{aligned}
\tag{5.32}
$$

gdje je

$$
\begin{aligned}
&{}^i\mathbf{g}_{{}^{i.j}_ic}\left({}^{i.j}_i\mathbf{c}\left({}^{i.j}\mathbf{r},{}^{i.j}_i\mathbf{h}\right)\right)={}^{i.j}\left\|{}^{i.j}_i\mathbf{s}_i\circ\left({}^{i.j}_i\mathcal{H}\left({}^{i.j}\mathbf{r}\right)-{}^{i.j}_i\mathbf{h}\right)\right\|_2^2 \\
&{}^i\mathbf{g}_{{}^{i.j}_icz}\left({}^i_{i.j}\mathbf{c}_z\left({}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{z}\right)\right)={}^{i.j}\left\|{}^{i.j}_i\mathbf{s}_{zi}\circ\left({}^i_{i.j}\mathbf{z}-{}^{i.j}_i\mathbf{z}\right)\right\|_2^2
\end{aligned}
\tag{5.33}
$$

razina 2:

$$
\begin{aligned}
&\min_{{}^{i.j}_i\mathbf{x},{}^{i.j}_i\mathbf{z}} \; {}^ig_{{}^{i.j}_ic}\left({}^{i.j}_i\mathbf{c}\left({}^{i.j}\mathbf{r},{}^{i.j}_i\mathbf{h}\right)\right)+{}^ig_{{}^{i.j}_icz}\left({}^{i.j}_i\mathbf{c}_z\left({}^i_{i.j}\mathbf{z},{}^{i.j}_i\mathbf{z}\right)\right) \\
&t.d. \quad {}^{i.j}\mathbf{g}\left({}^{i.j}\mathbf{x},{}^{i.j}\mathbf{r}\left({}^{i.j}\mathbf{x},{}^i_{i.j}\mathbf{z},{}^i_{i.j}\mathbf{h}\right)\right)\le 0 \\
&\qquad {}^{i.j}\mathbf{g}_e\left({}^{i.j}\mathbf{x},{}^{i.j}\mathbf{r}\left({}^{i.j}\mathbf{x},{}^i_{i.j}\mathbf{z},{}^i_{i.j}\mathbf{h}\right)\right)=0 \\
&\qquad {}^{i.j}\underline{\mathbf{x}}\le {}^{i.j}\mathbf{x}\le {}^{i.j}\overline{\mathbf{x}}\;\; {}^{i.j}_i\underline{\mathbf{z}}\le {}^{i.j}_i\mathbf{z}\le {}^{i.j}_i\overline{\mathbf{z}}
\end{aligned}
\tag{5.34}
$$

Iz navedene formulacije vidljivo je da se zadovoljenje meta koje su postavljene promatranom sustavu od nadređenog sustava tretira kao funkcija cilja, dok se zadovoljenje meta koje postavlja promatrani sustav podsustavima tretira kao ograničenje koje je potrebno ostvariti s određenom tolerancijom $\boldsymbol{\varepsilon}_c$. Istovremeno se u funkciju cilja dodaju vrijednosti tolerancija koji su tim problemima ujedno i projektne varijable. S obzirom da zadnja razina nema podrazina, na njoj nije potrebno zadovoljiti ograničenja konzistentnosti, već samo što bolje zadovoljiti propisane mete od nadsustava, što se ostvaruje minimizacijom funkcije cilja. [69] navodi da s obzirom da se $\boldsymbol{\varepsilon}_c$ i $\boldsymbol{\varepsilon}_{cz}$ minimiziraju u funkciji cilja, ograničenja konzistentnosti podređenih sustava vezana na njih su aktivna te se prema tome mogu uključiti u funkciju cilja umjesto njih. U takvoj formulaciji tolerancije $\boldsymbol{\varepsilon}_c$ i $\boldsymbol{\varepsilon}_{cz}$ više nisu niti varijable već u potpunosti iščezavaju te je u tom slučaju formulacija za srednju razini (razinu 1) dana izrazom (5.35), dok za ostale razine vrijedi slična transformacija izraza:

$$\begin{aligned}
&\min_{{}^i\mathbf{x},{}^i_0\mathbf{z},{}^i_{i,j}\mathbf{z},{}^{i,j}_{\ i}\mathbf{h}} \sum {}^0\mathbf{g}_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)+\sum {}^0\mathbf{g}_{{}^0_icz}\left({}^0_i\mathbf{c}_z\left({}^0_i\mathbf{z},{}^i_0\mathbf{z}\right)\right)+\sum {}^i\mathbf{g}_{{}^i_{i,j}c}\left({}^i_{i,j}\mathbf{c}\left({}^{i.j}\mathbf{r},{}^{i,j}_{\ i}\mathbf{h}\right)\right)+\sum {}^i\mathbf{g}_{{}^{i.j}_{\ i}c}\left({}^{i.j}_{\ i}\mathbf{c}_z\left({}^i_{i.j}\mathbf{z},{}^{i.j}_{\ i}\mathbf{z}\right)\right)\\
&t.d. \quad {}^i\mathbf{g}\left({}^i\mathbf{x},{}^i_{i,j}\mathbf{z},{}^{i,j}_{\ i}\mathbf{h},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_{i,j}\mathbf{z},{}^{i,j}_{\ i}\mathbf{h}\right)\right)\le 0\\
&\qquad {}^i\mathbf{g}_e\left({}^i\mathbf{x},{}^i_{i,j}\mathbf{z},{}^{i,j}_{\ i}\mathbf{h},{}^i\mathbf{r}\left({}^i\mathbf{x},{}^i_{i,j}\mathbf{z},{}^{i,j}_{\ i}\mathbf{h}\right)\right)=0\\
&\qquad {}^i\underline{\mathbf{x}}\le {}^i\mathbf{x}\le {}^i\overline{\mathbf{x}};\ {}^i_0\underline{\mathbf{z}}\le {}^i_0\mathbf{z}\le {}^i_0\overline{\mathbf{z}};\ {}^i_{i,j}\underline{\mathbf{z}}\le {}^i_{i,j}\mathbf{z}\le {}^i_{i,j}\overline{\mathbf{z}};\ {}^{i,j}_{\ i}\underline{\mathbf{h}}\le {}^{i,j}_{\ i}\mathbf{h}\le {}^{i,j}_{\ i}\overline{\mathbf{h}};
\end{aligned} \tag{5.35}$$

Prema radu [69], težinski faktori ${}^i_0\mathbf{s}_0$ i ${}^i_0\mathbf{s}_i$ su jednaki za svaku komponentu, no u nekim radovima (npr. [70]) težinski faktori su različiti na različitim razinama, čime je prema autorima rada, ubrzana konvergencija. U [69] matematički je dokazana konvergencija metode pod pretpostavkom konveksnosti problema. Promatranjem izraza (5.30) i (5.31) može se uočiti da ako ${}^i_0\mathbf{s}_0 \to \infty$ funkcija cilja je ima beskonačnu kaznu čak i za najmanje kršenje ograničenja konzistentnosti te je rješenje ATC metode jednako AIO rješenju.

Osim kažnjavanja funkcije cilja prema izrazu (5.31) u literaturi su prisutne još neke formulacije. U [71] predlaže se korištenje Lagrangeovih množitelja. U tom slučaju ekvivalent izrazu (5.31) glasi:

$${}^0g_{{}^i_0c}\left({}^i_0\mathbf{c}\left({}^i\mathbf{r},{}^i_0\mathbf{h}\right)\right)={}^i_0\boldsymbol{\lambda}^T\left({}^i_0\mathcal{H}\left({}^i\mathbf{r}\right)-{}^i_0\mathbf{h}\right) \tag{5.36}$$

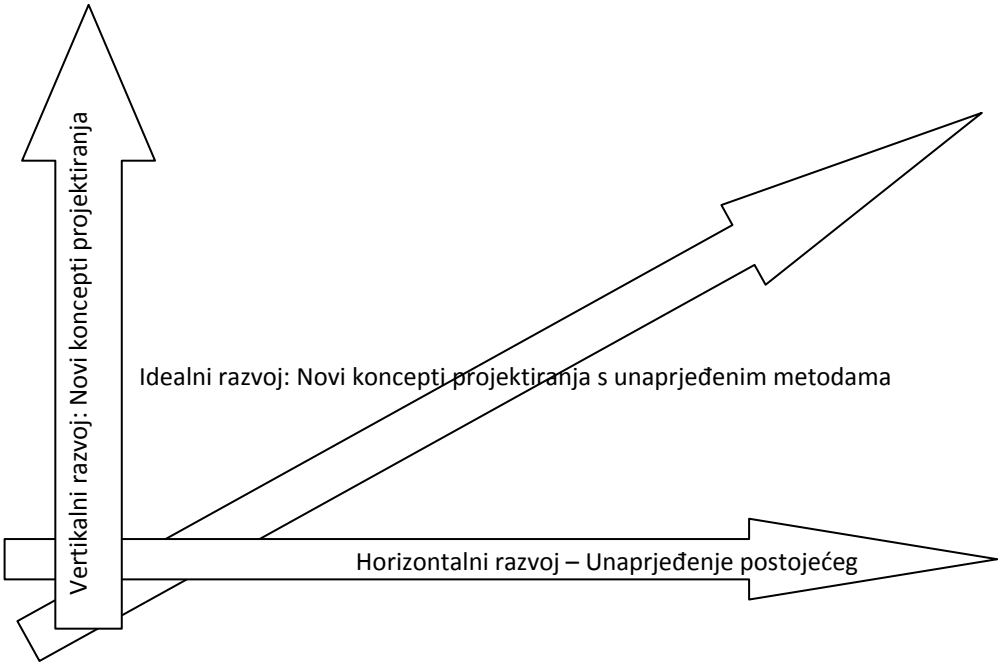
U zadnje vrijeme u literaturi je najčešće prisutna Lagrangeovim množiteljima augmentirana formulacija ATC metode [72], koja zapravo kombinira izraze (5.31) i (5.36) a omogučuje bržu konvergencije u odnosu na originalnu ATC formulaciju.

5.5. Perspektivni smjerovi razvoja optimizacije bazirane na dekompoziciji odnosno multidisciplinarne optimizacije

Kako bi se jasnije odredio značaj primjene i unapređenja dekompozicijsko-koordinacijskih metoda za projektiranje složenih sustava u ovom potpoglavlju biti će prikazani potencijalno perspektivni smjerovi razvoja prema mišljenju nekih od vodećih stručnjaka koji se bave ovim područjem. Prema jednom od zadnjih preglednih članaka s područja optimizacije složenih sustava bazirane na dekompoziciji [55], identificirana su dva osnovna smjera razvoja: horizontalni i vertikalni.

Horizontalni smjer razvoja obuhvaća razvoj i unapređenje metoda sinteze (optimizacijskih metoda, surogat modela, dekompozicijsko koordinacijskih metoda) kako bi se omogućilo povećanje dimenzionalnosti problema te učinkovitije generiranje rješenja.

Vertikalni smjer identificira dostignuća koja su konceptualno i kvantitativno potpuno nova tj. zaokrete u samom pristupu projektiranju složenih sustava (npr. optimizacija cijele porodice određenog proizvoda s ciljem minimizacije vremena za ukupni povratak investicije)



Sl. 5.20. Horizontalni i vertikalni razvoj optimizacije bazirane na dekompoziciji

Očekuje se da se ova dva okomita smjera razvoja simbiotski potpomažu te omoguće postizanja još kvalitetnijih složenih sustava u budućnosti čime bi se potaknula primjena takvih rješenja u svakodnevnoj industrijskoj praksi. U radu [55] kao dva najperspektivnija smjera horizontalnog razvoja navedeni su: 1)formalna klasifikacija spregnutih problema sa stajališta jačine i količine spregnutosti, 2) prilagođavanje metoda trendovima masivnog paralelnog računarstva.

6. UNAPRJEĐENJE PROCESA PROJEKTIRANJA SLOŽENIH TANKOSTJENIH KONSTRUKCIJA

S obzirom na računalnu zahtjevnost višekriterijskog projektiranja složene tankostjene konstrukcije, neophodno je smanjiti broj proračuna odziva složene tankostjene konstrukcije, odnosno omogučiti paralelno izvršavanje pojedinih simulacija unutar procesa projektiranja. Višekriterijska optimizacija cjelovitog nemodificiranog optimizacijskog problema (*All-in-One problem*) gotovo je nemoguća bez znatnih računalnih resursa i velikog utroška vremena. U ovom poglavlju prikazani su predloženi načini poboljšanja kvalitete i efikasnosti višekriterijske optimizacije tankostjene konstrukcije definiranjem projektnih sekvenci u okviru razvijenog računalnog okruženja za potporu pri višekriterijskom projektiranju. Implementacija istraženih metoda u okruženje za potporu projektiranju OCTOPUS Designer, nalazi se u poglavlju 7, dok je primjena istraženih metoda na primjerima dana u poglavlju 8.

6.1. Diskusija o kriterijima kvalitete Pareto fronte realnih složenih tankostjenih konstrukcija

U poglavlju 3.3.1 dani su indikatori kvalitete koji se koriste za mjerenje kvalitete Pareto fronte. Pri usvajanju mjera kvalitete za realne probleme iz inženjerske prakse bitno je imati na umu da u većini slučajeva projektanti znaju u kojem području atributnog prostora bi se trebala nalaziti moguća rješenja. Također projektantima nije od interesa da se istražuju dijelovi atributnog prostora za koje znaju da im projekti neće biti konkurentni po nekom od projektnih atributa. Stoga je vrlo važno ograničiti atributni prostor kako bi se izbjeglo nepotrebno trošenje vremenskih i računalnih resursa. Iz toga proizlazi da se dio subjektivnih odluka projektanata treba donijeti prije pokretanja faze generiranja nedominiranih riješena. Vezano uz odabir kriterija kvalitete to znači da kriterije koji mjere ukupnu rastegnutost (npr. Normalizirana raširenost Pareto fronte ili Vanjski promjer) Pareto fronte ima smisla koristiti tek nakon primjene tih ograničenja.

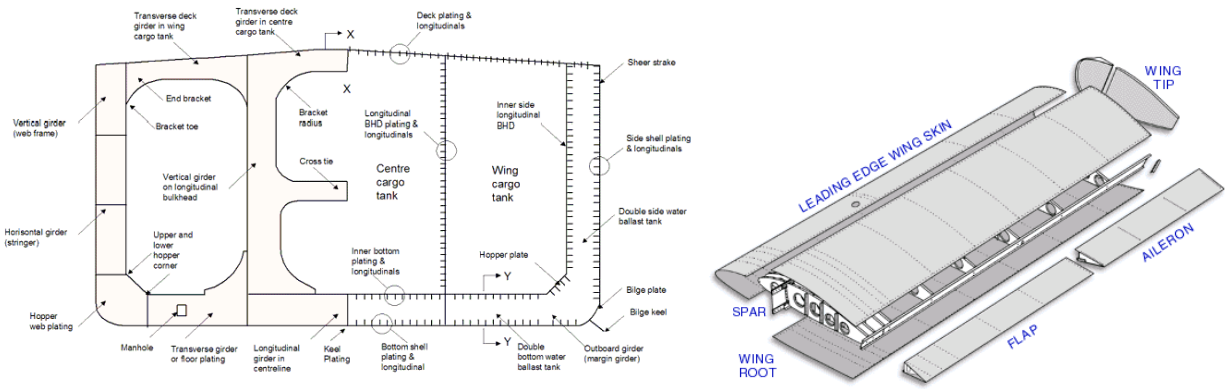
S obzirom da je projektiranje u inženjerskoj praksi vremenski značajno ograničen proces, važno je optimizacijom generirati određeni broj dobrih projekata u zadanom vremenskom intervalu kako bi projektanti pravodobno mogli donijeti kvalitetne odluke. Stoga su vrlo važne i jednostavne mjere kvalitete koje mjere ukupno vrijeme procesa generiranja rješenja, broj poziva određenog modula za analizu, broj eksperimenata potrebnih za generiranje nekog surogat modela te vrijeme generiranja surogat modela.

U ovom radu fokus je na primjeni metoda koji omogućuju paralelno izvršavanje dijelova na optimizacijske procedure na višeprocesorskim računalima, odnosno klasterima računala, jer

se time omogućuje skraćivanje ukupnog vremena optimizacije iako je zbirno gledano utrošak računalnih resursa često značajno veći nego kad bi se cijela optimizacija odvijala sekvencijalno. S tim u svezi ostvarene rezultate potrebno komentirati s obzirom na kako korištene tako i potencijalne mogućnosti korištenja na višeprocesorskim računalima ili klasterima računala.

6.2. Predložena dekompozicija i rješavanje problema projektiranja složene tankostjene konstrukcije

Kod složenih tankostjenih konstrukcija često je vrlo jednostavno identificirati funkcionalne komponente konstrukcije koji zajedno daju traženu funkciju ukupne konstrukcije. Na Sl. 6.1 vidljive su funkcionalne cjeline brodske konstrukcije glavnog rebra tankera i zrakoplovne konstrukcije krila malog zrakoplova. Kako bi dekompozicija uopće bila moguća, model analize mora omogućiti pristup ulaznim i izlaznim veličinama tih funkcionalnih dijelova odnosno njihovim komponentama. Kao što je navedeno u poglavlju 2 najčešći pristup analizi konstrukcije u konceptualnoj fazi projektiranja metalnih tankostjenih konstrukcija je da se prikladnim linearnim MKE modelom izračunaju pomaci i naprezanja konstrukcije koji se zatim uvrštavaju u adekvatne kriterije podobnosti promatranog dijela konstrukcije. Pri tome su funkcionalne komponenta međusobno spregnute odzivom, dok je proračun podobnosti funkcionalnih komponenti raspregnut.



Sl. 6.1. Funkcionalne cjeline nekih složenih tankostjenih konstrukcija

6.2.1. Dekompozicija problema

Projektirani dio konstrukcije može se podijeliti na strukturne podsustave *i*= 1 … *Ns* iste razine koji predstavljaju funkcionalne komponente konstrukcije. Pri tome su neki odzivi podsustava funkcije ne samo deskriptora svog podsustava ${}^{i}\mathbf{d}$ već i deskriptora drugih podsustava ${}^{j}\mathbf{d}$ kao što je dano izrazom (6.1)

$${}^{i}\mathbf{r}={}^{i}\mathbf{r}\left({}^{i}\mathbf{d},{}^{j}\mathbf{d}\right);\ i\neq j \tag{6.1}$$

Ukupni odziv pojedinog strukturnog podsustava sastoji se od strukturnih odziva **ρ** (pomaci i naprezanja), kriterija podobnosti **α** i mjera kvalitete konstrukcije **Ω**:

$${}^{i}\mathbf{r}^{T}=\left\{{}^{i}\boldsymbol{\rho},{}^{i}\boldsymbol{\alpha},{}^{i}\boldsymbol{\Omega}\right\}; \tag{6.2}$$

Pri tome su najčešće samo strukturni odzivi funkcija deskriptora ostalih podsustava, dok su kriteriji podobnosti i mjere kvalitete najčešće funkcija deskriptora i odziva promatranog podsustava:

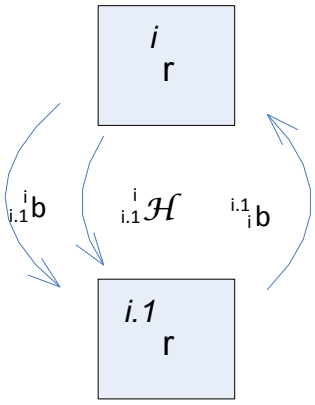
$${}^{i}\mathbf{r}=\left\{\begin{array}{l}{}^{i}\boldsymbol{\rho}={}^{i}\boldsymbol{\rho}\left({}^{i}\mathbf{d},{}^{j}\mathbf{d}\right)\\ {}^{i}\boldsymbol{\alpha}={}^{i}\boldsymbol{\alpha}\left({}^{i}\mathbf{d},{}^{i}\boldsymbol{\rho}\right)\\ {}^{i}\boldsymbol{\Omega}={}^{i}\boldsymbol{\Omega}\left({}^{i}\mathbf{d},{}^{i}\boldsymbol{\alpha}\right)\end{array}\right\} \tag{6.3}$$

Stoga je moguće izraz (6.1) pojednostaviti na:

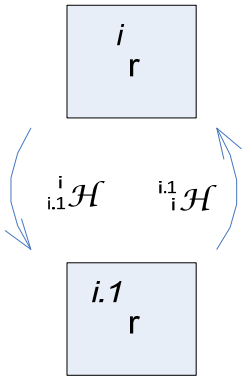
$${}^{i}\boldsymbol{\rho}={}^{i}\boldsymbol{\rho}\left({}^{i}\mathbf{d},{}^{j}\mathbf{d}\right);\ i\neq j \tag{6.4}$$

Iz (6.3) je vidljivo da funkcionalne komponente imaju dva proračunska modula ${}^{i}\boldsymbol{\alpha},{}^{i}\boldsymbol{\Omega}$ koji isključivo ovise o deskriptorima i odzivima svog sustava te ih je moguće tretirati kao podsustav sustava funkcionalne komponente strukture. Pri tome su sustavi spregnuti preko deskriptora ${}^{i}\mathbf{d}$ što znači da su to zajednički deskriptori te ih se prema notaciji iz potpoglavlja 5.1 označava sa ${}^{j}_{i.k}\mathbf{b}$ odnosno ${}^{i.k}_{i}\mathbf{b}$. S obzirom da je mogući broj podsustava *k* sustava *i* samo 1, umjesto k može se odmah pisati 1, tj. ${}^{j}_{i.1}\mathbf{b}$ i ${}^{i.1}_{i}\mathbf{b}$. Odzivi strukture ${}^{i}\boldsymbol{\rho}$. operatorom preslikavanja ${}^{i}_{i.1}\mathcal{H}$ preslikavaju se u veličine odziva strukture koje zahtijevaju **α**,**Ω** moduli na podsustavu *i.1*, što je shematski prikazano na Sl. 6.2. S obzirom da su podsustavu *i.1* to ujedno i jedini deskriptori, dekompozicijom na ovaj način bi podsustav *i.1* imao premalo slobode za zadovoljavanje ograničenja konzistentnosti. Kod jednostavnih tipova konstrukcije koje se sastoje samo od štapova ili greda uobičajeno je da se odzivi strukture ne računaju direktno preko deskriptora već preko veličina koje daju krutost promatranoj konstrukciji što je za slučaj štapa površina *A*, dok je za slučaj grede to moment tromosti presjeka oko neutralne osi *I*. Funkcionalne

komponente metalne tankostjene konstrukcije se sastoje od elemenata uzdužne i poprečne strukture koji primarno nose uzdužna odnosno poprečna opterećenja kao što je opisano u poglavlju 2. Pri tome je proračun pomaka prvenstveno ovisan o površini i momentu inercije uzdužnih elemenata funkcionalne komponente strukture te o momentu inercije poprečnih elemenata. Ukoliko je modulu za analizu pomaka moguće nametnuti te veličine određenim preslikavanjem, tada je moguća interakcija prema Sl. 6.3.



Sl. 6.2. Sprega disciplina funkcionalnih komponenti preko zavisnih deskriptora



Sl. 6.3. Sprega disciplina funkcionalnih komponenti preko operatora preslikavanja odziva

Tada vrijedi:

$$ {}^{i}\mathbf{r} = {}^{i}\mathbf{r}\left({}^{j}\mathbf{d}, {}^{i.1}_{i}\mathcal{H}\left({}^{i.1}\mathbf{r}\right)\right) = {}^{i}\mathbf{r}\left({}^{j}\mathbf{d}, {}^{i.1}_{i}\mathbf{h}\right),\ i \neq j; \tag{6.5} $$

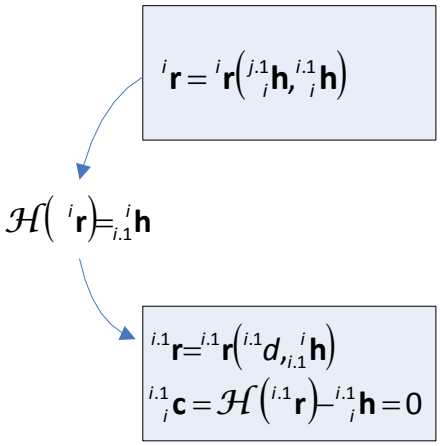
$$ {}^{i.1}\mathbf{r} = {}^{i.1}\mathbf{r}\left({}^{i.1}\mathbf{d}, {}^{i}_{i.1}\mathcal{H}\left({}^{i}\mathbf{r}\right)\right) = {}^{i.1}\mathbf{r}\left({}^{i.1}\mathbf{d}, {}^{i}_{i.1}\mathbf{h}\right), \tag{6.6} $$

te je za osiguranje konzistentnosti sustava *i* sa podsustavom *i*.1 potrebno zadovoljiti izraze:

$$ {}^{i.1}_{i}\mathbf{c} = \mathcal{H}\left({}^{i.1}\mathbf{r}\right) - {}^{i.1}_{i}\mathbf{h} = 0; \tag{6.7} $$

$$ {}^{i}_{i.1}\mathbf{c} = \mathcal{H}\left({}^{i}\mathbf{r}\right) - {}^{i}_{i.1}\mathbf{h} = 0; \tag{6.8} $$

Dekomponiranjem sustava *i* i *i.1* hijerarhijski odozgo jakom formulacijom, ograničenje konzistentnosti (6.8) je zadovoljeno eksplicitno, dok podsustav *i.1* mora zadovoljiti ograničenje (6.7). Shematski prikaz ove dekompozicije prikazan je na Sl. 6.4.



Sl. 6.4. Dekompozicija funkcionalne komponente strukture hijerarhijski odozgo

U izrazu (6.5) odziv sustava *i* još uvijek ovisi o deskriptorima ostalih funkcionalnih komponenti *j*, no s obzirom da je moguće napisati da i ostale funkcionalne komponente više ne ovise o deskriptorima već o odzivima svojih podsustava ${}^{j}\mathbf{r} = {}^{j}\mathbf{r}\left({}^{i}d, {}^{j.1}_{j}\mathcal{H}\left({}^{j.1}\mathbf{r}\right)\right)$, vrijede izrazi:

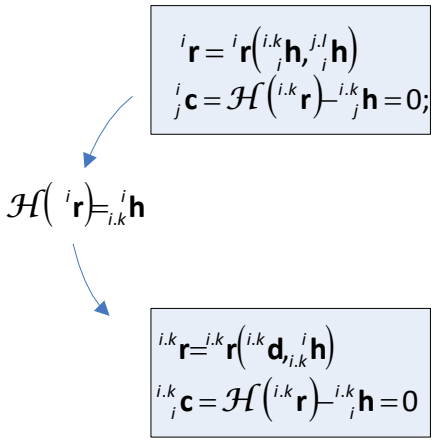
$${}^{i}\mathbf{r} = {}^{i}\mathbf{r}\left({}^{j.1}_{i}\mathbf{h}, {}^{i.1}_{i}\mathbf{h}\right),\ i \neq j = 1,2...n_s; \tag{6.9}$$

$${}^{i}_{j}\mathbf{c} = \mathcal{H}\left({}^{i.1}\mathbf{r}\right) - {}^{i.1}_{j}\mathbf{h} = 0;\ i \neq j = 1,2...n_s \tag{6.10}$$

Iz gornja dva izraza jasno se vidi da su strukturni odzivi svih funkcionalnih komponenti strukture spregnuti te da ih je moguće dekomponirati, no broj ograničenja konzistentnosti bi bio vrlo velik jer je svaki sustav spregnut sa svim ostalim sustavima. Kod nekih tipova konstrukcija bilo bi moguće identificirati jake i slabe veze između sustava te bi se broj ograničenja mogao smanjiti uz zanemarivanje ograničenja konzistentnosti slabo spregnutih sustava. Zbog toga je najčešće ipak efikasnije da se strukturni odzivi računaju u istom sustavu.

Iz gore navedenog slijedi da se dekompozicija strukturnog problema na funkcionalne cjeline zapravo svodi na dekomponiranje modula za proračun kriterija podobnosti **α** i mjera kvalitete konstrukcije **Ω** pojedinih funkcionalnih komponenti strukture, dok proračun odziva ostaje u zajedničkom sustavu. Ukoliko je moguće grupirati funkcionalne komponente na način da unutar grupe postoje jake veze, a između grupa slabe veze koje se mogu zanemariti ili manji broj jakih veza, onda se strukturni odzivi mogu dekomponirati nehijerarhijski. Ovakav način grupiranja je moguć recimo korištenjem OCTOPUS Analyzer-a [73], kojim je moguće računati strukturni odziv na više presjeka broda nezavisno, ili generiranjem surogat modela.

Na Sl. 6.5 prikazana je shema dekomponiranog modela za analizu složene tankostjene konstrukcije podjelom na funkcionalne komponente prema gore danoj diskusiji.



Sl. 6.5. Dekompozicija funkcionalne komponente strukture hijerarhijski odozgo

i=1,…*No; j*=1,…*No; i≠j;*

No – broj grupa odziva

k=1,..*Ns* grupe *i*

l=1,..*Ns* grupe *j*

Ns - broj podstruktura odnosno funkcionalnih komponenti strukture.

Pri tome je:

$$^{i}\mathbf{r} = {}^{i}\mathbf{r}\left({}^{i.k}_{i}\mathbf{h}, {}^{j.l}_{i}\mathbf{h}\right) = {}^{i}\boldsymbol{\rho}\left({}^{i.k}_{i}\mathbf{h}, {}^{j.l}_{i}\mathbf{h}\right) \tag{6.11}$$

$$^{i.k}\mathbf{r} = \left\{ \begin{array}{l} {}^{i.k}\boldsymbol{\Phi} = {}^{i.k}\boldsymbol{\Phi}\left({}^{i.k}\mathbf{d}\right) \\ {}^{i.k}\boldsymbol{\alpha} = {}^{i.k}\boldsymbol{\alpha}\left({}^{i.k}\mathbf{d}, {}^{i}\boldsymbol{\rho}\right) \\ {}^{i.k}\boldsymbol{\Omega} = {}^{i.k}\boldsymbol{\Omega}\left({}^{i.k}\mathbf{d}, {}^{i.k}\boldsymbol{\alpha}\right) \end{array} \right\} \tag{6.12}$$

$$^{i}_{i.k}\mathbf{h} = {}^{i}_{i.k}\mathcal{H}\left(\subseteq {}^{i}\mathbf{r}\right) = {}^{i}_{i.k}\mathcal{H}\left(\subseteq {}^{i}\boldsymbol{\rho}\right) \tag{6.13}$$

$$^{i.k}_{i}\mathbf{h} = {}^{i.k}_{i}\mathcal{H}\left(\subseteq {}^{i.k}\mathbf{r}\right) = {}^{i.k}_{i}\mathcal{H}\left({}^{i.k}\boldsymbol{\Phi}\right) \tag{6.14}$$

$$^{i.k}\boldsymbol{\Phi}^{T} = \left\{ {}^{i.k}A_{L}, {}^{i.k}I_{L}, {}^{i.k}I_{T}, {}^{i.k}A_{T} \right\} \tag{6.15}$$

Za slučaj kada postoji kad je i=1 odnosno kad postoji samo jedna grupa strukturnih odziva (svi odzivi povezani jakim vezama) izraz (6.11) uz pomoć (6.15) može se pisati jednostavnije:

$$^{i}\mathbf{r} = {}^{i}\mathbf{r}\left({}^{i.k}_{i}\mathbf{h}\right) = {}^{i}\boldsymbol{\rho}\left({}^{i.k}_{i}\mathbf{h}\right) = {}^{i}\boldsymbol{\rho}\left({}^{i.k}\boldsymbol{\Phi}^{T}\right) = {}^{i}\boldsymbol{\rho}\left({}^{i.k}A_{L}, {}^{i.k}I_{L}, {}^{i.k}I_{T}, {}^{i.k}A_{T}\right),\ i = 1,2...N_{s} \tag{6.16}$$

Pri tome su $^{i}\boldsymbol{\rho}\left({}^{i.k}A_{L}, {}^{i.k}I_{L}, {}^{i.k}I_{T}, {}^{i.k}A_{T}\right)$ odzivi na osnovu kojih je moguće izračunati $^{i.k}\boldsymbol{\alpha}\left({}^{i.k}\mathbf{d}, {}^{i}_{i.k}\mathbf{h}\right)$.

Ukupni set deskriptora konstrukcije se sastoji od topoloških, geometrijskih, materijalnih (kompoziti) i dimenzijskih deskriptora. Deskriptori dimenzija tankostjenih konstrukcija su najčešće i projektne varijable pri optimizaciji tankostjene konstrukcije u određenoj fazi osnivanja konstrukcije, dok su geometrijski i topološki deskriptori promjenjivi samo u konceptualnoj fazi projektiranja konstrukcije. Čak i ukoliko optimizacijski problem obuhvaća sva tri tipa deskriptora obično su deskriptori dimenzija nalaze u podoptimizacijskom problemu određene varijante topoloških i geometrijskih deskriptora. Stoga se uz izostavljanje članova koji su nepromjenjivi može zapisati:

$$ {}^{i}\boldsymbol{\rho} = {}^{i}\boldsymbol{\rho}\left({}^{i}\mathbf{d}^{S}, {}^{j}\mathbf{d}^{S}\right);\ i \neq j \tag{6.17} $$

6.2.2. Sekvenca rješavanja problema

Za odabir i primjenu prikladne sekvence rješavanja problema važno je znati koje su mogućnosti interakcije s modelom analize, kao i koje su dostupne optimizacijske metode te za određene slučajeve i metode surogat modeliranja. S tim u svezi je vrlo bitna priroda atributa odnosno funkcija cilja koji ukupnog sustava. Također je bitno da metoda koordinacije omogući efikasno iznalaženje ne samo jednog rješenja već skupa Pareto optimalnih rješenja.

Predložena sekvenca za rješavanje problema sastoji se od dvije glavne faze:

1) generiranja globalnih surogat modela odziva,
2) generiranja Pareto optimalnih rješenja.

6.2.2.1. FAZA I - Surogat model baziran na optimalnim smjerovima funkcionalnih komponenti

Cilj prve faze je generiranje globalnih surogat modela odziva prema izrazu (6.16), kojim se zamjenjuju pozivi skupog proračuna realnog modula za proračun strukturnog odziva modela. Način rada sa surogat modelima strukturnog odziva složene tankostjene konstrukcije dan je u nastavku u potpoglavlju 6.3

6.2.2.2. FAZA II – Generiranje Pareto optimalnih rješenja

U drugoj fazi generiraju se Pareto rješenja koja će se interaktivno prezentirati projektantu. Sama faza je generiranja rješenja je također interaktivna iako ju je moguće automatizirati propisivanjem kriterija napredovanja u iduće korake. Na Sl. 6.6 predstavljen je dijagram toka sekvence generiranja Pareto rješenja. U dijagramu je vidljivo da prvi koraci optimizacije koriste surogat modele odziva generirane u prethodnoj fazi. Dekomponirani

problem se rješava za određeni broj kombinacija težinskih faktora atributa projektnog problema.

Nakon što se riješi dekomponirani problem za definirane kombinacije težinskih faktora (prema potpoglavlju 6.2.2.3), ta rješenja mogu biti upotrjebljena kao inicijalni projekti višeciljnom heurističkom rješavaču (MOPSO, MOGA) koji rješava cjelokupni problem (AiO), a odzivi se i dalje računaju surogat modelima.

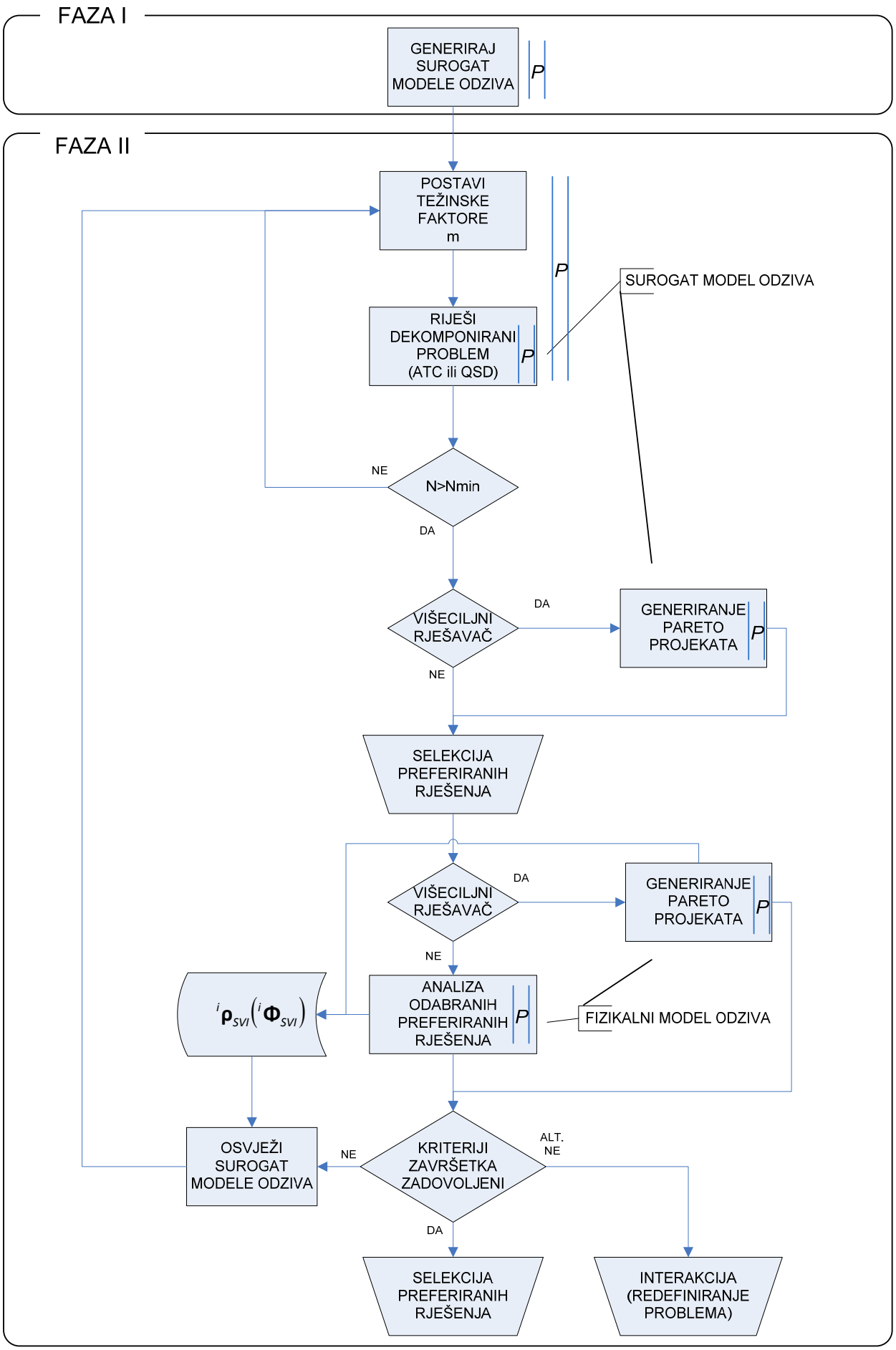
Nakon što je generiran dovoljni broj Pareto rješenja sa surogat modelima odziva, moguća je interakcija s projektantom pri čemu projektant odabire rješenja koja želi analizirati fizikalnim modelom odziva, odnosno koje želi ubaciti za inicijalne projekte višeciljnom optimizacijskom algoritmu koji ovaj put koristi odzive fizikalnog modela, a ne surogat modela. Ovaj korak može biti i u potpunosti automatiziran definiranjem kriterija odabira projekata (npr. najboljih *n* po odabranoj normi).

Ukoliko kriteriji završetka sekvence nisu zadovoljeni, surogat modeli se osvježavaju za sve novo izračunate odzive te sekvenca ponovno starta postavljanjem početnih težinskih faktora. Ukoliko kriteriji završetka pokažu da je postupak generirao loše odnosno izrazito nepodobne projekte postoji mogućnost da je problem loše definiran te je potrebna interakcija s projektantom te eventualna redefinicija problema.

Zadovoljenjem kriterija završetka sekvence, rješenja se interaktivno prezentiraju projektantu te dodatno obrađuju metodama višekriterijskog odlučivanja kako bi se projektantu omogućio što kvalitetniji odabir preferiranih rješenja.

Ovako definirana sekvenca rješavanja ima nekoliko mogućnosti paralelizacije koje su na dijagramu označene simbolom $|P|$. Osim integrirane paralelizacije koju relativno jednostavno mogu imati heuristički višeciljni rješavači i već navedene mogućnosti paralelnog izvođenja analize odziva tijekom generiranja surogat modela, postoje naredne mogućnosti paralelizacije:

- paralelno rješavanje dekomponiranog problema
- paralelno rješavanje sekvence za određenu kombinaciju težinskih faktora.



Sl. 6.6. Dijagram toka generiranja Pareto rješenja

6.2.2.3. Koordinacija problema

Kao što je navedeno u potpoglavlju 5.2.2, pri odabiru koordinacijske tehnike neophodno je procijeniti o jakost i količinu spregnutosti podsustava. Podstrukture tankostjene konstrukcije se prema podjeli prikazanoj na Sl. 5.6 sa prema jakosti i količini spregnutosti nalaze u području P3 koje je okarakterizirano je velikom količinom slabo spregnutih veza, odnosno granično s P4 koje je okarakterizirano s velikom količinom slabo spregnutih veza. Kao što je navedeno u potpoglavlju 5.2.2, u području P3 može biti korisno zanemariti određene veze čija je jačina spregnutosti ispod određene razine dok je u području P4 najčešće najefikasnije rješavanje originalnog AiO problema. Oba navedena pristupa zapravo izbjegavaju korištenje eksplicitnog koordinacijskog algoritma, koji kod ovog tipa problema može znatno smanjiti efikasnost zbog znatnog povećanja dimenzionalnosti optimizacijskih potproblema uvođenjem eksplicitnih ograničenja konzistentnosti i/ili zajedničkih varijabli. Stoga će značajnije spregnuti dijelovi strukture biti definirani u jedinstvenom optimizacijskom potproblemu, dok će slabe veze biti usklađivane kroz iteracijske cikluse globalnog surogat modela odziva i globalnog MKE proračuna odziva.

Vrlo sličan pristup koji se u praksi projektiranja tankostjenih brodskih konstrukcija pokazao vrlo efikasnim, nalazi se u prvotnoj (ili tzv. *Legacy*) verziji programskog sustava MAESTRO [74], [75]. Tim pristupom se optimizacija relativno malih podsustava (vojeva) s ciljem minimizacije mase odnosno kompozitne, višekriterijske funkcije, izvodi nezavisno, kao da spregnutost ne postoji (lokalna optimizacija), dok se određeni oblik koordinacije provodi linearnom MKE analizom globalnog sustava, za optimalno stanje deskriptora podsustava iz prethodne iteracije. Računalne simulacije koje se pozivaju u okviru optimizacije pojedinog potproblema (voja) uključuju izračun novog stanja izdržljivosti (*capability*) pojedine evaluacijske jedinice, dok se zahtijevano stanje naprezanja (*demand*) računa s pretpostavkom o nepromijenjenom stanju pomaka uz korekciju naprezanja s tzv. derivativima uzdužnih naprezanja. No s obzirom da se ti derivativi naprezanja računaju samo u odnosu na promatrani potproblem/voj njihov utjecaj na aproksimaciju stanja naprezanja je vrlo mali. Time se utjecaj promjene deskriptora pojedinog potproblema prenosi na ostale probleme preko promjene stanja pomaka. Iako ta verzija MAESTRO sustava nudi i formalnu koordinaciju elemenata primarne čvrstoće konstrukcije (tzv. *Dual level optimization*), u praksi se mnogo robusnijom i efikasnijom pokazala samostalna lokalna optimizacija. Treba imati na umu da za efikasnost ovog algoritma ključnu ulogu ima algoritam SLP-a (vidi [74])koji je visoko prilagođen problemu. U praksi se pokazalo da je najveći nedostatak te metode nedovoljno dobro propagiranje

informacija o odzivu elemenata poprečne čvrstoće. Globalni surogat modeli odziva, predloženi u ovom radu namijenjeni su upravo uklanjanju tih nedostataka.

6.3. Primjena surogat-modela za supstituiranje proračuna odziva modula za analizu konstrukcije

Kao što je navedeno u poglavlju 4, surogat modeliranje može se koristiti za zamjenu zahtjevnih proračunskih metoda tijekom projektiranja, odnosno optimizacije. Pri tome je broj parametara koji se koriste za generiranje surogat modela jedan od najvažnijih ograničavajućih faktora. Povećavanjem broja parametara nelinearno se povećava i broj pokusa potrebnih za treniranje surogat modela prihvatljive točnosti.

Kako bi se stekao utisak o dimenzionalnosti problema koji se nastoji riješiti, u nastavku diskusije koristit će se karakteristike primjera iz poglavlja 8.2, projektiranje tankostjene konstrukcije broda za prijevoz vozila i putnika – RoPax. Pri tome će se razmatrati najčešći pristup analizi konstrukcije u konceptualnoj fazi projektiranja metalnih tankostjenih konstrukcija kojim se prikladnim linearnim MKE modelom izračunaju pomaci i naprezanja konstrukcije koji se zatim uvrštavaju u adekvatne kriterije podobnosti promatranog dijela konstrukcije.

6.3.1. Odabir odzivnih veličina za surogat model

Surogat modeli koji se koriste pri optimizaciji najčešće direktno zamjenjuju funkciju cilja i/ili ograničenja. Za slučaj optimizacije modela RoPax broda u primjeru to bi značilo da je potrebno generirati ~10 000 surogat modela za svaki slučaj opterećenja s tim da je svaki model ovisan o ~200 parametara. Iako se za generiranje svih modela mogu koristiti isti pokusi, ovako veliki broj surogat modela ovisnih o velikom broju parametara sigurno ne obećava ubrzanje procesa projektiranja.

Temeljem diskusije dane u potpoglavlju 6.2.1, moguće je proračun podobnosti pojedinog evaluacijskog elementa – (*patch*) ${}^{i}\boldsymbol{\alpha}$ izraziti kao funkciju ovisnu o deskriptorima tog *patcha* te strukturnog odziva na tom *patchu* ${}^{i}\boldsymbol{\rho}$.

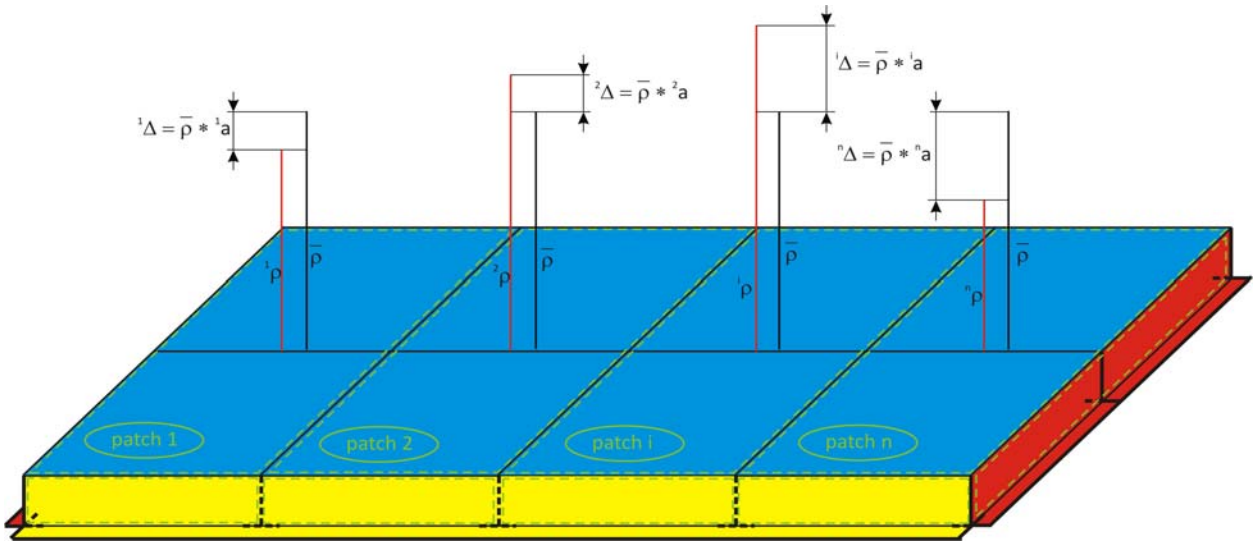
$$ {}^{i}\boldsymbol{\alpha} = {}^{i}\boldsymbol{\alpha}\left({}^{i}\mathbf{d}, {}^{i}\boldsymbol{\rho}\left({}^{i}\mathbf{d}, {}^{j}\mathbf{d}\right)\right) \tag{6.18} $$

Komponenta podobnosti koja ovisi o lokalnim deskriptorima *patcha* sama je izdržljivost (*capability*) i računalno je najčešće znatno jednostavnija za izračun od strukturnog odziva koji je najčešće funkcija deskriptora svih deskriptora modela analize. Stoga umjesto direktnog generiranja surogata kriterija podobnosti može se generirati surogate strukturnog odziva, čijim

korištenjem bi se mogli izračunati aproksimirani kriteriji podobnosti. No s obzirom da je broj strukturnih odziva potrebnih za izračun skupine kriterija podobnosti na nekom *patchu* tek neznatno manji od broja tih kriterija, ovom transformacijom nisu moguće dovoljne uštede na broju surogat modela. U slučaju RoPax-a to bi rezultiralo smanjenjem broja surogata na ~7000 za svaki slučaj opterećenja.

Idući korak u smanjenju broja surogat modela jest baziran na dekompoziciji strukture na funkcionalne cjeline. Ideja je da se, umjesto surogat modela strukturnih odziva potrebnih za izračun kriterija podobnosti, koriste surogat modeli srednjeg stanja strukturnog odziva na određenoj funkcionalnoj cjelini. Funkcionalne cjeline razmatrane u ovom slučaju su promatrane sa stajališta funkcije nošenja opterećenja te će se u nastavku nazivati podstrukturama. Na taj način se u slučaju RoPax-a smanjuje broj surogata na ~100 za svaki slučaj opterećenja. Naravno, primjena ovih naprezanja na svim *patchevima* određene podstrukture unijet će značajnu grešku koja je proporcionalna sa samom veličinom podstrukture zbog promjene distribucije odziva po podstrukturi. Kako bi se ova greška reducirala, svaki *patch* bilježi odnos između razlike svog zadnjeg proračunatog naprezanja i zadnjeg izračunatog srednjeg naprezanja prema srednjem naprezanju:

$$ {}^{i}a = \frac{{}^{i}\boldsymbol{\rho} - {}^{j}\overline{\boldsymbol{\rho}}}{{}^{j}\overline{\boldsymbol{\rho}}} \tag{6.19} $$



Sl. 6.7. Uključivanje promjene distribucije odziva po podstrukturi

Aproksimirane vrijednosti strukturnog odziva na *patchu* dobivaju se izrazom:

$$ {}^{i}\hat{\boldsymbol{\rho}} = {}^{j}\hat{\overline{\boldsymbol{\rho}}} \cdot \left(1 + {}^{i}a\right) \tag{6.20} $$

Pri tome je pretpostavljeno da se promjena koeficijenta ${}^{i}a$ ne ovisi značajno o promjeni deskriptora dimenzija konstrukcijskih elemenata. Za strukturne modele u kojima to nije zadovoljeno, potrebno je u finalnim ciklusima optimizacije izvršiti ponovnu kalibraciju koeficijenata ${}^{i}a$ sa stvarnim stanjem odziva no u ovom slučaju u odnosu na aproksimirana srednja naprezanja.

$$ {}^{i}\hat{a} = \frac{{}^{i}\boldsymbol{\rho} - {}^{j}\hat{\boldsymbol{\rho}}}{{}^{j}\hat{\boldsymbol{\rho}}} \tag{6.21} $$

Time se surogati zapravo kalibriraju na trenutno stanje odziva te u lokalnom području oko točke kalibriranja daju značajno točnije rezultate.

6.3.2. Odabir parametara surogat modela

Kao što je već navedeno, broj parametara surogat najznačajnije utječe na složenost surogat modela. S obzirom na diskusiju danu u potpoglavlju 6.2.2.3, jasno je da neće biti moguće isključiti neki od deskriptora. Ipak značajnu redukciju je moguće ostvariti ukoliko se deskriptori grednih elemenata tretiraju kao jedan, kompozitni deskriptor. Pri tome je ključno da se omogući naizgled neobično ponašanje parametra surogat modela, a to je da se ostvarene vrijednosti pojedinih deskriptora grede bilježe iz samog modela analize, a ne direktno iz plana pokusa.

Time se postiže značajna ušteda jer su u konceptualnim i preliminarnim MKE modelima gredni elementi i makrolementi orebrenih ljuski, značajno zastupljeni. Tako se broj parametara na primjeru iz potpoglavlja 8.1 reduciran s 35 na 11, od čega je 8 tzv. kompozitnih parametara koji definiraju više deskriptora poprečnog presjeka grednih elemenata. U primjeru u potpoglavlju 8.2 s početnih 237, broj parametara je reduciran na 81.

Vrijednosti deskriptora koje se preslikavaju u model analize, za planirane pokuse, upravljane su izrazom (6.22) koji je zapravo isti kao i za standardne parametre s tim da se u ovom slučaju normalizirana vrijednost planiranog pokusa kolone dodijeljene određenom parametru, dodjeljuje za svakom od deskriptora kompozitnog parametra grede.

$$ d_{B_i}^{j} = d_{Bi\min} + \frac{u_{DOE}^{j}\left(d_B\right)}{d_{Bi\max} - d_{Bi\min}} \tag{6.22} $$

Gdje je

$u_{DOE}^{j}\left(d_B\right)$ - normalizirana vrijednost planiranog pokusa *j* za svaki deskriptor *I* grede *B*.

Za korektno generiranje surogat modela nužno je osigurati jednoliku zastupljenost određenog parametra po testiranoj domeni. Kod jednostavnih parametara čije vrijednosti direktno kontrolira odabrani plan pokusa to je automatski zadovoljeno. No kod kompozitnih parametara to je neophodno provjeriti. Također je važan odabir prijenosne funkcije izlaznih vrijednosti kojom se očitana vrijednost više deskriptora poprečnog presjeka pretvara u jednu vrijednost. U ovom radu testirano je nekoliko takvih funkcije:

relativna udaljenost $$f_D(\mathbf{d}_B) = \frac{\sum_{i=1}^{n_{dB}} \frac{d_{Bi} - d_{Bi\min}}{d_{Bi\max} - d_{Bi\min}}}{n_{dB}} \tag{6.23}$$

funkcija površine $$f_A(\mathbf{d}_B) = \frac{A_B - A_{B\min}}{A_{B\max} - A_{B\min}} \tag{6.24}$$

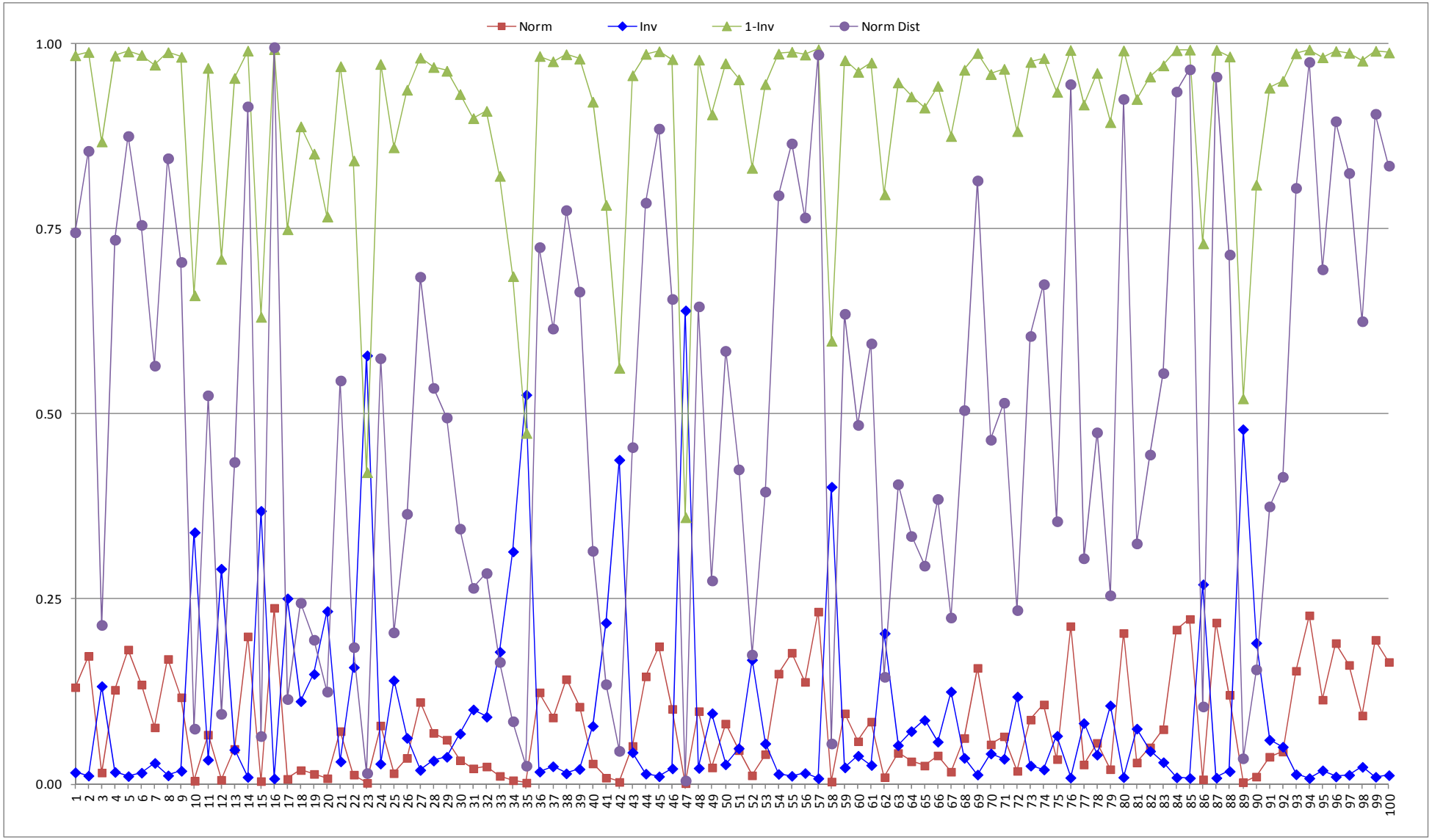
funkcija momenta inercije $$f_I(\mathbf{d}_B) = \frac{I_B - I_{B\min}}{I_{B\max} - I_{B\min}} \tag{6.25}$$

funkcija inverznog momenta inercije $$f_{I^{-1}}(\mathbf{d}_B) = \frac{(I_{B\min})^{-1} - (I_B)^{-1}}{(I_{B\min})^{-1} - (I_{B\max})^{-1}} \tag{6.26}$$

funkcija momenta otpora $$f_W(\mathbf{d}_B) = \frac{W_B - W_{B\min}}{W_{B\max} - W_{B\min}} \tag{6.27}$$

funkcija inverznog momenta otpora $$f_{W^{-1}}(\mathbf{d}_B) = \frac{(W_{B\min})^{-1} - (W_B)^{-1}}{(W_{B\min})^{-1} - (W_{B\max})^{-1}} \tag{6.28}$$

Potrebno je naglasiti da su ulazne vrijednosti deskriptora grede pri svakom od planiranih pokusa uvijek upravljane izrazom (6.22), bez obzira na odabir funkcije za preslikavanje izlaznih vrijednosti. Kao što je vidljivo, sve definirane prijenosne funkcije izlaznih vrijednosti transformiraju prikupljene vrijednosti deskriptora grede u interval [0,1]. Na Sl. 6.8 prikazano je kako su, za gredu pokrova pontona primjera iz potpoglavlja 8.1, distribuirane vrijednosti prijenosnih funkcija: normalizirane udaljenosti, momenta inercije i normaliziranog momenta inercije, za 100 pokusa generiranih u svrhu treniranja surogat modela. Na slici su vidljive i vrijednosti funkcije $\left(1 - f_{I^{-1}}\right)$ koja je zapravo korištena umjesto $f_{I^{-1}}$ kako bi smjer porasta prijenosne funkcije imao isti smjer kao i porast samih deskriptora. Vidljivo je da je prijenosna funkcija f_D jednoliko distribuirana po prostoru što je i bilo za očekivati s obzirom da se (6.23) može shvatiti i kao inverzno preslikavanje izraza (6.22). No isto tako je vidljivo da su vrijednosti ostalih prikazanih funkcija preslikavanja nejednoliko distribuirane po domeni. Razlog za to je što ostale funkcije ne ovise linearno o deskriptorima grede. Kako bi se surogat modelima ipak dostavile vrijednosti jednoliko distribuirane po domeni surogata ([-1, 1] za surogate integrirane u OCTPOPUS Designer) modificirana je normalizacija parametara surogata.



Sl. 6.8. Nekorigirane, momentom inercije normalizirane vrijednosti parametra koje opisuje jaki nosač palube pontona

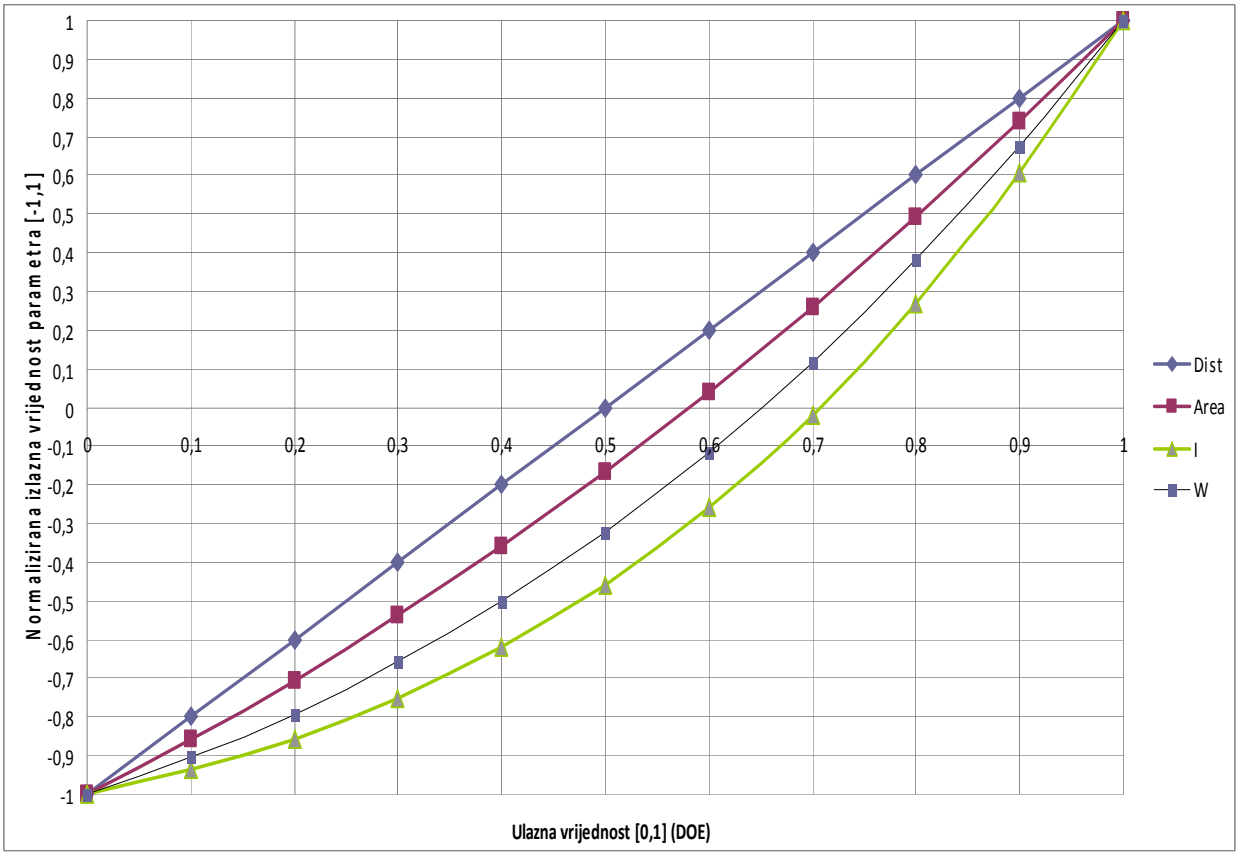
Na Sl. 6.9 prikazano je kako se za 10 jednoliko distribuiranih pokusa transformiraju izlazne vrijednosti za parametra grede f_D, f_A, f_W i f_I korištenjem standardne normalizacije za parametre surogat modela:

$$u_s(d)=\frac{2(d-d_{\min})}{d_{\max}-d_{\min}}-1; \quad [-1,1] \tag{6.29}$$

koja za slučaj rada s prijenosnim funkcijama izlaznih vrijednosti ima oblik:

$$u_s(f(d_B))=\frac{2(f(d_B)-f(d_{B_{\max}}))}{f(d_{B_{\max}})-f(d_{B_{\min}})}-1=2f(d_B)-1; \quad [-1,1] \tag{6.30}$$

Vidljivo je da je samo f_D održala jednoliku distribuciju, što bi rezultiralo degradacijom svojstava surogat modela pri korištenju drugih prijenosnih funkcija normaliziranih standardnom normalizacijom prema izrazu (6.30).

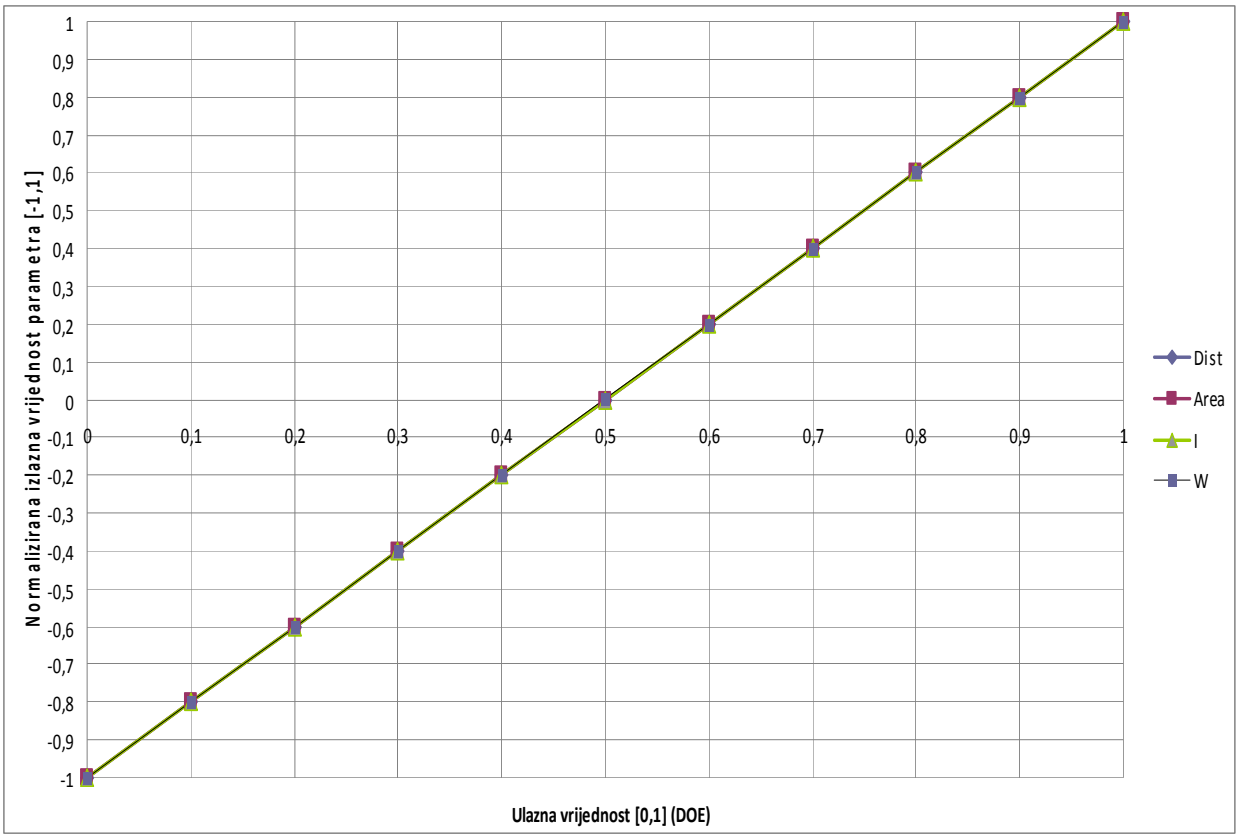


Sl. 6.9. Nekorigirana normalizacija složenog deskriptora grednog nosača

Korigiranje nejednolikosti distribucije je izvedeno korištenjem normalizacije s potencijama prema sljedećem izrazu

$$u_{sk}(f(d_B))=\frac{2([f(d_B)]^p-[f(d_{B_{\max}})]^p)}{[f(d_{B_{\max}})]^p-[f(d_{B_{\min}})]^p}-1=2[f(d_B)]^p-1; \quad [-1,1] \tag{6.31}$$

Korištenjem odgovarajuće potencije *p* postiže se jednolika distribucija za promatrane funkcije što je vidljivo na Sl. 6.10 za vrijednosti potencija *p* prema Tbl. 6.1.



Sl. 6.10. Korigirana normalizacija složenog deskriptora grednog nosača

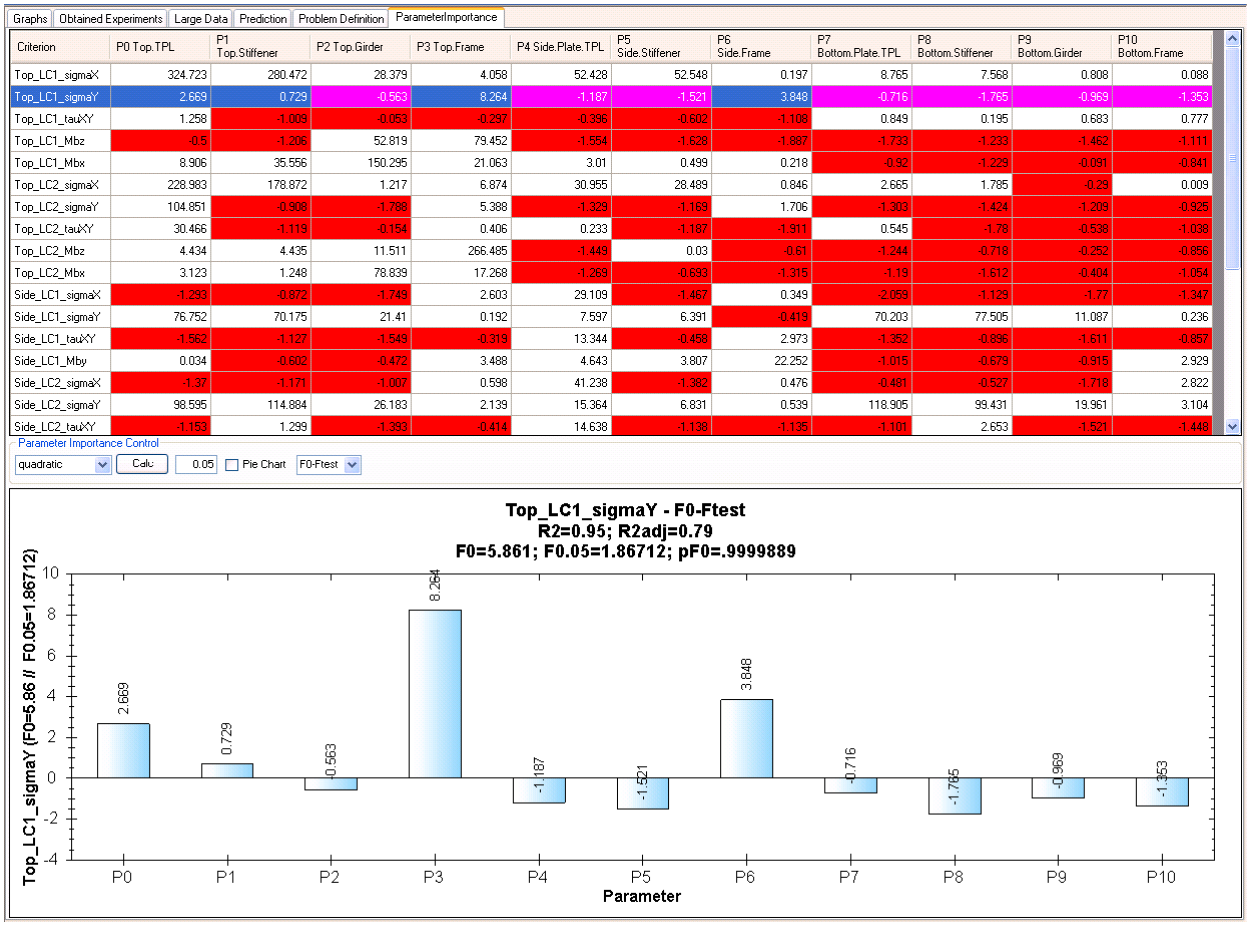
Tbl. 6.1 Odabrane potencije *p* za korekciju normalizacije prijenosnih funkcija deskriptora grede

Prijenosna funkcija deskriptora grede d_B	**Oznaka**	**Potencija *p***
relativna udaljenost	f_D	1
površina	f_A	2
moment otpora	f_W	3
inverzni moment otpora	$f_{W^{-1}}$	1/3
moment inercije	f_I	4
inverzni moment inercije	$f_{I^{-1}}$	1/4

6.3.3. Analiza utjecajnosti parametara surogat modela

Analiza utjecajnosti pojedinih parametara, prema postupku opisanom u potpoglavlju 4.3.5.2, u ovom radu je iskorištena za procjenu utjecaja izmjera modela jednostavnog pontona iz primjera danog u potpoglavlju 8.1, na stukturne odzive po podstrukturama. Na Sl. 6.11 prikazani su rezultati testiranja u okviru okruženja OCTOPUS Designer. U gornjem dijelu prikazanog sučelja, u recima su prikazane odzivne veličine, a u stupcima parametri čija utjecajnost se provjerava. Parametri koji su prema *F* testu procijenjeni neutjecajnima, označeni su crvenom bojom. Na donjem dijelu sučelja prikazan je stupčani dijagram koji prikazuje vrijednosti F_0-F_{test} za trenutno selektirani odziv (Top_LC1_SigmaY na Sl. 6.11), na osnovu kojeg se vrlo brzim vizualnim pregledom može vidjeti utjecaj pojedinog parametra na promatrani kriterij / odziv.

U nastavku su navedene dvije moguće primjene analize utjecajnosti parametara na surogat modele promatrane odzivne veličine.



Criterion	P0 Top.TPL	P1 Top.Stiffener	P2 Top.Girder	P3 Top.Frame	P4 Side.Plate.TPL	P5 Side.Stiffener	P6 Side.Frame	P7 Bottom.Plate.TPL	P8 Bottom.Stiffener	P9 Bottom.Girder	P10 Bottom.Frame
Top_LC1_sigmaX	324.723	280.472	28.379	4.058	52.428	52.548	0.197	8.765	7.568	0.808	0.088
Top_LC1_sigmaY	2.669	0.729	-0.563	8.264	-1.187	-1.521	3.848	-0.716	-1.765	-0.969	-1.353
Top_LC1_tauXY	1.258	-1.009	-0.053	-0.297	-0.396	-0.602	-1.108	0.849	0.195	0.683	0.777
Top_LC1_Mbz	-0.5	-1.206	52.819	79.452	-1.554	-1.628	-1.887	-1.733	-1.233	-1.462	-1.111
Top_LC1_Mbx	8.906	35.556	150.295	21.063	3.01	0.499	0.218	-0.92	-1.229	-0.091	-0.841
Top_LC2_sigmaX	228.983	178.872	1.217	6.874	30.955	28.489	0.846	2.665	1.785	-0.29	0.009
Top_LC2_sigmaY	104.851	-0.908	-1.788	5.388	-1.329	-1.169	1.706	-1.303	-1.424	-1.209	-0.925
Top_LC2_tauXY	30.466	-1.119	-0.154	0.406	0.233	-1.187	-1.911	0.545	-1.78	-0.538	-1.038
Top_LC2_Mbz	4.434	4.435	11.511	266.485	-1.449	0.03	-0.61	-1.244	-0.718	-0.252	-0.856
Top_LC2_Mbx	3.123	1.248	78.839	17.268	-1.269	-0.693	-1.315	-1.19	-1.612	-0.404	-1.054
Side_LC1_sigmaX	-1.293	-0.872	-1.749	2.603	29.109	-1.467	0.349	-2.059	-1.129	-1.77	-1.347
Side_LC1_sigmaY	76.752	70.175	21.41	0.192	7.597	6.391	-0.419	70.203	77.505	11.087	0.236
Side_LC1_tauXY	-1.562	-1.127	-1.549	-0.319	13.344	-0.458	2.973	-1.352	-0.896	-1.611	-0.857
Side_LC1_Mby	0.034	-0.602	-0.472	3.488	4.643	3.807	22.252	-1.015	-0.679	-0.915	2.929
Side_LC2_sigmaX	-1.37	-1.171	-1.007	0.598	41.238	-1.382	0.476	-0.481	-0.527	-1.718	2.822
Side_LC2_sigmaY	98.595	114.884	26.183	2.139	15.364	6.831	0.539	118.905	99.431	19.961	3.104
Side_LC2_tauXY	-1.153	1.299	-1.393	-0.414	14.638	-1.138	-1.135	-1.101	2.653	-1.521	-1.448

Sl. 6.11. Prikaz sučelja OCTOPUS Designera za provjeru utjecajnosti parametara korištenjem polinomskih surogat modela

6.3.3.1. Primjena analize utjecajnosti parametara surogat modela na poboljšanje točnosti surogat modela

Tijekom razrade literature u poglavlju4, kao i tijekom testiranja surogat modela, postao je jasan negativan utjecaj povećanja broja parametara koji služe za definiranje nekog surogat modela, na samu točnost surogat modela. Osim očiglednog povećanja broja potrebnih pokusa s povećanjem broja parametara, smanjuje se i točnost samog surogat modela što je naročito zamjetno kod Kriginga i RBF-a. Ove dvije metode zasnovane su na mjerenju normalizirane udaljenosti lokacija definiranih vrijednostima parametara za pojedini pokus (vidi izraze (4.23) i (4.25) za Kriging te (4.33) za RBF). Iz toga se da zaključiti da dodavanje novih parametara zamućuje doprinos preostalih parametara, što dovodi do hipoteze da će izbacivanje parametara, čija varijanca je iznad one propisane F testom, rezultirati poboljšanjem točnosti surogat modela. Preliminarna testiranja provedena u ovom radu idu u prilog postavljenoj hipotezi, no da bi se ona i dokazala potrebno je provesti opsežnije testove na više modela ali i s različitim zahtijevanim intervalima pouzdanosti α.

6.3.3.2. Primjena surogat-modela za procjenu jakosti i količine spregnutosti

Kao što je na navedeno u potpoglavlju 5.5, prema raspravi danoj u [55] jedan od najperspektivnijih smjerova razvoja optimizacije složenih sustava jest i formalna klasifikacija spregnutih problema sa stajališta jačine i količine spregnutosti. Tamo je i navedeno da se kao najčešća mjera, u praksi, koriste međusobne parcijalne derivacije spregnutih veličina. Jedan od očiglednih problema tog pristupa jest i to što se na taj način procjenjuje spregnutost samo u lokalnom području oko projektne točke za koje se te parcijalne derivacije računaju.

Alternativa tom načinu mjerenja spregnutosti mogla bi biti formalna analiza varijance parametara (vidi npr.[33]) čime bi se mogao procijeniti globalni utjecaj spregnutih veličina. Ipak, odmah se nameće osnovni problem takovog pristupa a to je što se tako mogu izračunati samo utjecaji parametara na izlazne veličine no ne i međusobni utjecaj odzivnih veličina. Razlog tomu je što je za provođenje analize varijance neophodno postaviti ulazne parametre na nekoliko fiksnih razina, što je neizvedivo ukoliko se želi testirati utjecaj jedne odzivne veličine na drugu.

Moguće rješenje tog problema mogla bi biti primjena metodologije opisane u potpoglavlju 4.3.5.2 kojom se provjerava utjecaj pojedinog parametra na varijancu surogat modela. S obzirom da taj pristup ne zahtjeva postavljanje parametara na fiksne nivoe, tretiranjem određene izlazne veličine kao parametra (npr. slično kao kod tretiranja kompozitnog parametra grede u potpoglavlju 6.3.2) mogla bi se provesti analiza utjecajnosti

neke odzivne veličine na drugu kojoj se aproksimira odziv. Naravno, odmah treba navesti da ukupna točnost polinomskog surogat modela, koji se koristi za aproksimacijsko modeliranje pojedine odzivne veličine, mora a biti zadovoljavajuća. To znači da ovaj postupak neće dati vjerodostojna rješenja u slučajevima kad nije moguće dobiti zadovoljavajuću točnost od polinomskih surogat modela, npr. zbog izrazite nelinearnosti problema.

6.4. Primjena hibridnih optimizacijskih metoda

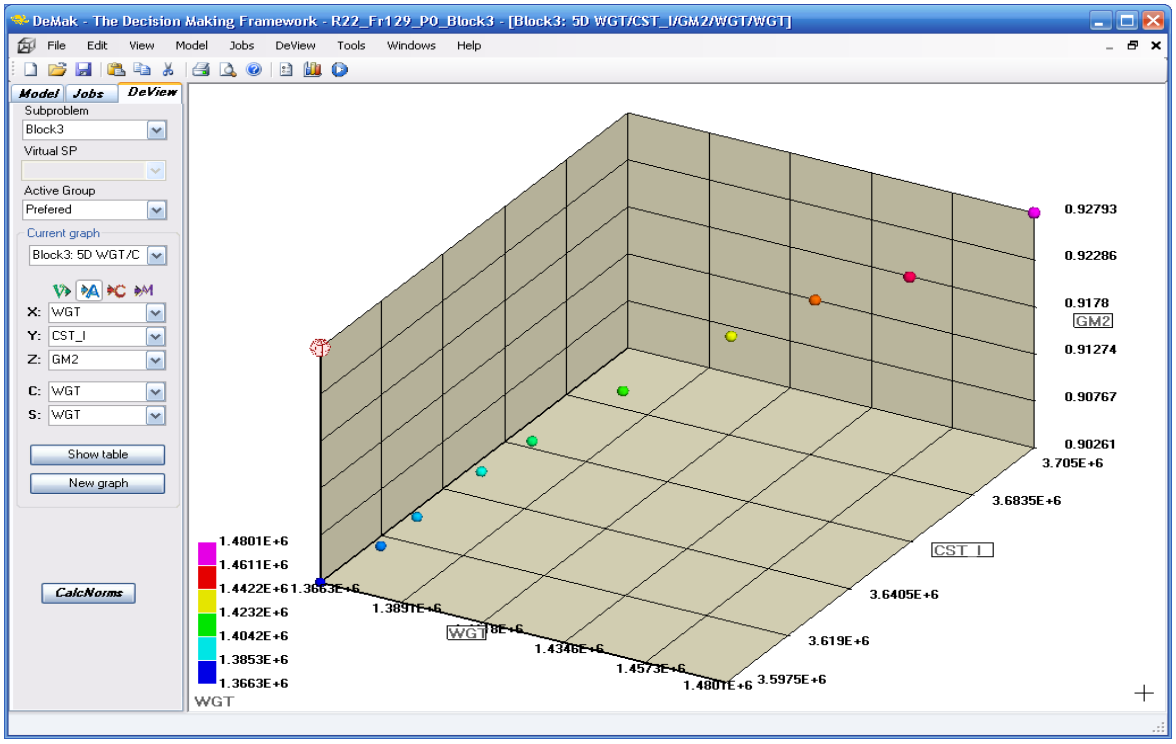
Uporaba hibridnih optimizacijskih metoda nastalih efikasnim korištenjem rezultata koji su generirani prethodnim ciklusom optimizacije za početnu populaciju ili inicijalne startne pozicije sljedeće optimizacijske metode omogućuje povećanje uspješnosti rješavanje određenog problema. Osnovna ideja je iskoristiti optimizacijsku metodu koja najviše odgovara pojedinom zadatku. Pri tome se u literaturi najčešće spominju dvije naoko dijametralno suprotne varijante:

1) korištenje globalnih optimizacijskih metoda (najčešće heurističkih, npr. PSO) u prvim fazama optimizacije nakon čega slijedi određeni broj optimizacija, lokalnim optimizacijskim metodama (najčešće metode bazirane na gradijentima, npr. SQP), startanih s najboljih globalnih rješenja,
2) korištenje lokalnih optimizacijskih metoda (najčešće metode bazirane na gradijentima, npr. SQP ili SLP), u prvim fazama optimizacije nakon čega slijedi optimizacija globalnom optimizacijskom metodom (najčešće heurističkom: GA, PSO odnosno MOGA, MOPSO) kojoj se uz dio slučajno generiranih početnih rješenja, u inicijalnu populaciju ubacuju prethodno dobivena rješenja.

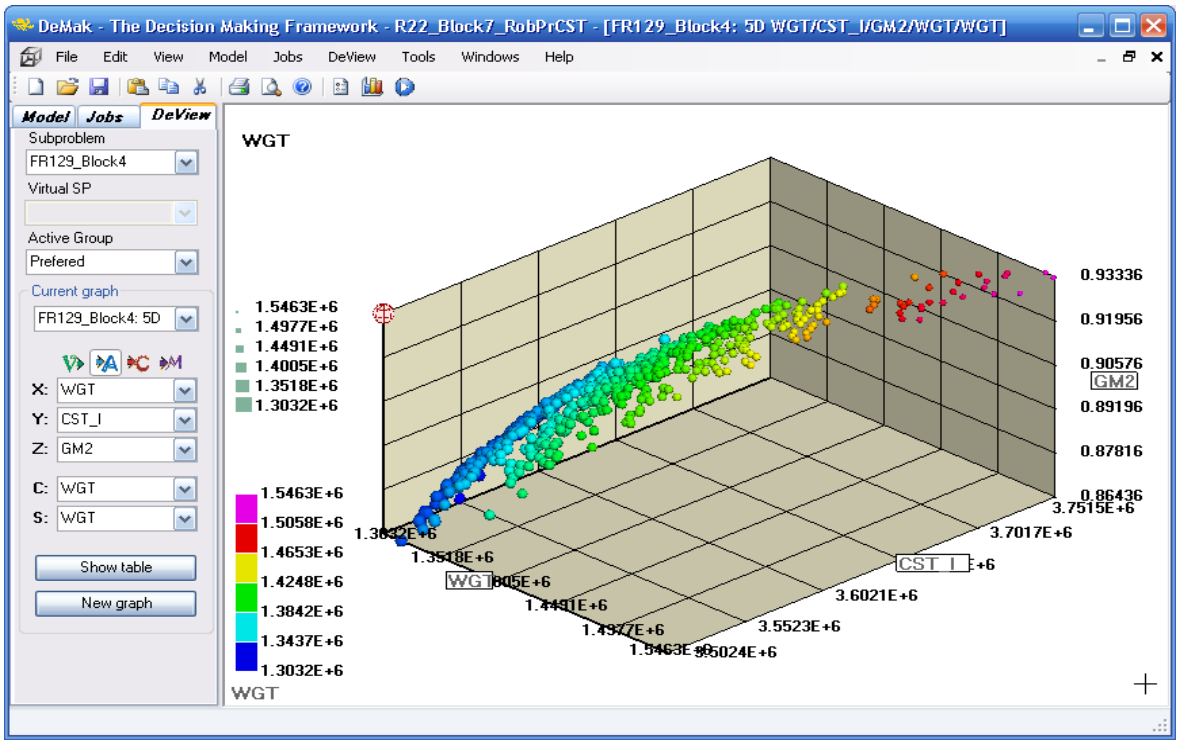
Prvi pristup se najčešće koristi u jednociljnoj optimizaciji nelinearnih problema s velikim brojem lokalnih optimuma. Opisanom procedurom se osigurava izbjegavanje lokalnih optimuma, što je osnovni nedostatak gradijentnih metoda te efikasno konvergiranje prema globalnom optimumu.

Drugim pristupom se ubrzava konvergencija heurističkih algoritama pri višeciljnoj optimizaciji gdje heuristički algoritmi poput MOGA i MOPSO prevladavaju. S obzirom da je njihov osnovni nedostatak potreba za velikim brojem iteracija za popunjavanje Pareto fronte, injektiranje nekoliko početnih rješenja s izrazito dobrim vrijednostima pojedinih ciljeva rezultira značajnim ubrzanjem procesa. Upravo ovaj pristup je korišten u okruženju za potporu pri projektiranju OCTOPUS Designer, pri optimizaciji u KORAKU 2 projektiranja RoPax broda EU FP6 projekta IMPROVE ([76], [78]). Na Sl. 6.12 mogu se vidjeti efikasna rješenja dobivena

optimizacijom problema SLP metodom za nekoliko odabranih odnosa važnosti pojedinih atributa koji sačinjavaju kompozitnu funkciju cilja, dok se na Sl. 6.13 mogu vidjeti finalna rješenja dobivena MOPSO algoritmom kojem je dio početne populacije popunjen rješenjima iz prethodne faze.



Sl. 6.12. Efikasna rješenja generirana SLP-om za različite omjere važnosti ciljeva [78]



Sl. 6.13. Pareto fronta generirana MOPSO algoritmom [78]

7. PROGRAMSKO OKRUŽENJE ZA POTPORU PRI PROJEKTIRANJU

Prije samog opisa arhitekture potrebno je napomenuti da se povezivanje alata za potporu pri projektiranju odnosno optimizaciju i alata za analizu promatranog problema može izvesti na dva bitno različita načina:

1) implementiranjem optimizacijskih metoda u postojeću arhitekturu programskog alata za analizu,
2) implementiranjem programskog alata za analizu promatranog problema unutar programa za potporu pri projektiranju odnosno optimizaciji.

Prvi način povezivanja omogućuje intuitivno zadavanje optimizacijskog modela jer su moduli za analizu zajedno sa svim svojim ulaznim i izlaznim podacima dostupni u svom izvornom obliku. Ovako se obično implementiraju jednostavnije optimizacijske metode, vrlo rijetko s mogućnostima dekompozicije ili generiranja surogat modela, jer se sve ove mogućnosti moraju zasebno integrirati unutar arhitekture programa za analizu projektnog problema što bitno komplicira integraciju. S obzirom da ovakav način integracije obično izvode stručnjaci za analizu promatranog problema, otežana je mogućnost nadogradnje metoda sinteze nakon što se originalni moduli za potporu pri projektiranju unaprijede.

Drugi način omogućuje primjenu svih metoda za potporu pri projektiranju, koje su razvijene unutar nekog programskog rješenja, na integriranim modulima za analizu. Time je naravno omogućena jednostavna nadogradnja modula za sintezu te direktno korištenje nadograđenih modula na postojećim integriranim programskim alatima za analizu problema. Kako bi se omogućila komunikacija s ulazno-izlaznim podacima modula za analizu te podatke je potrebno dostaviti modulima sinteze u općenito tj. generičkom obliku. Najčešće se to izvodi pomoću skupa metoda kojima se prenose najčešće nestrukturirani ulazno-izlazni podaci. To znači da se podaci nalaze se u obliku osnovnih tipova podataka (cjelobrojne ili realne varijable) odnosno polja osnovnih tipova podataka. S obzirom da se tako ne može prenijeti logička struktura modula za analizu, smanjena je intuitivnost definiranja optimizacijskog problema što je jedan od najvećih nedostataka ovog načina povezivanja.

Okruženje za potporu pri projektiranju, OCTOPUS Designer, od njegove osnovne definicije dane u [79], [80], [81], [82], [83] te [84], izvedeno je tako da omogući drugi način povezivanja, s tim da je u određenoj mjeri omogućen prijenos strukturiranih podataka kroz strukturirani generički model. U tablici Tbl. 7.1 može se vidjeti hijerarhijska struktura projektnog sustava OCTOPUS, u obliku u kojem je bio 2003. godine.

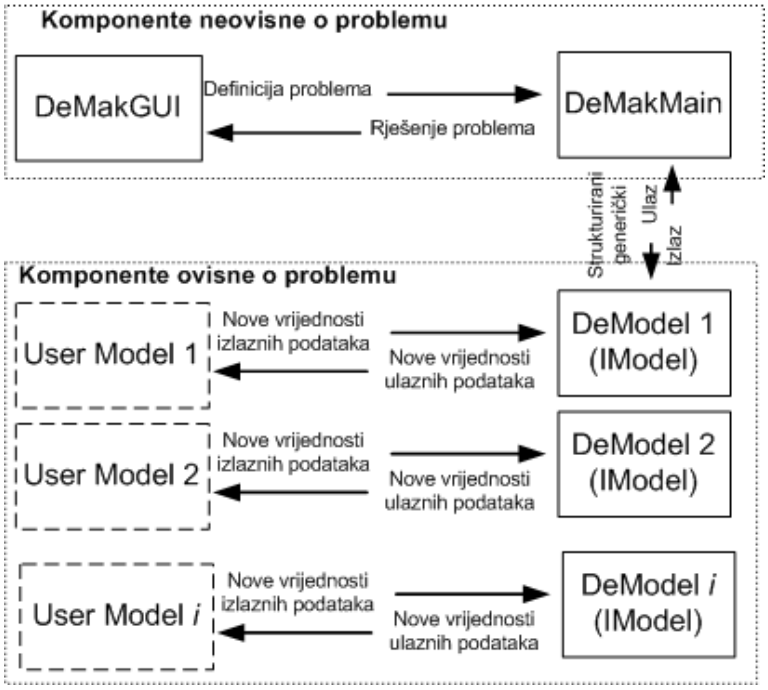
Tbl. 7.1 Hijerarhijska struktura projektnog sustava OCTOPUS [85]

Nivo	Programi i programske ljuske	Modul
7	Interaktivna ljuska za subjektivno odlučivanje u metričkom prostoru	SYNCHRO / DEMAK-S
6	Ljuska za automatizirano odlučivanje (optimizaciju)	PARALLEL
	(a) Program za adaptivno generiranje projekata u afinom prostoru – lokalna/globalna optimizacija	DEMAK-G
	(b) Program za globalnu optimalnu redistribuciju uzdužnog materijala u popr. Presjeku	CALMOP
	(c) Program za koordinaciju lokalne i globalne optimizacije	GAZ
5	Ljuska za određivanje statusa projektnih kriterija (projektna ograničenja i atributi), eliminaciju nepodobnih projekata i rješavanje konfliktnih situacija	CRITER
4	Programi i ljuske za proračun vrijednosti projektnih kriterija	
	(a) Proračun minimalnih dimenzija po Pravilima CRS	CRMIND
	(b) Program za proračun cijene i težine strukture	US0 *
	(c) Program za nelinearni proračun kolapsne izdržljivosti strukture	US1 *
	(d) Program za deterministički proračun strukturne sigurnosti	US2 *
	(e) Ljuska za AFOSM proračun pouzdanosti strukturnih elemenata, Ljuska za proračun pouzdanosti okvirnog rebra na nivou sustava.	(US3-c) US3-s
	(f) Program za proračun unutarnje raspodjele energije	(US4)
3	Biblioteke projektnih kriterija podobnosti	
	(a) Biblioteka kriterija granične nosivosti	EPAN1 *
	(b) Biblioteka kriterija granične uporabljivosti	EPAN2 *
	(c) Biblioteka kriterija po Pravilima CRS	EPAN3 *
	(d) Biblioteka tehnološko/proizvodnih ograničenja	ETEH
2	Programi za proračun odziva (deformacija i naprezanja) na 2-D FE modelima	
	(a) Program za proračun odziva uzdužnih elemenata poprečnog presjeka putem proširene teorije grede- UZDUŽNA I TORZIJSKA ČVRSTOĆA	LTOR *
	(b) Program za proračun odziva poprečne strukture - POPREČNA ČVRSTOĆA	TOKV *
	(c) Program za proračun odziva u superelementima – PODSTRUKTURE	(SUPEL)
	(d) Program za proračun 2-D struktura pod djelovanjem tlaka – PREGRADE	TROS
	(e) Program za proračun uzdužne čvrstoće broda	FIN
1	Ljuska za definiranje projektnih kombinacija opterećenja	
	(a) Proračun opterećenja na osnovi proračuna pomorstvenosti	DLOADS
	(b) Eksplicitno zadavanje komponenti opterećenja	EXLOAD *
	(c) Proračun komponenti opterećenja po Pravilima CRS	CRLOAD *
0	Interaktivna ljuska za (re)modeliranje strukture i vizualizaciju rezultata	CRINDAT*/ MM*/ MG *

U ovom radu okruženje je implementirano koristeći objektno orijentirano programiranje, što omogućuje stvaranje računalnog modela baziranog na realnim objektima. Cijeli sustav izveden je unutar Visual Studia 2008 [86] u .NET tehnologiji koristeći nekoliko programskih jezika (C#, FORTRAN i C++). Početak rada na ovoj, objektno orijentiranoj, verziji OCTOPUS Designera započeo je 2003. godine tehnologijskim projektom Ministarstva znanosti Republike Hrvatske, „Integrirani sustav za brodske konstrukcije“ [87].

Na Sl. 7.1 prikazane su komponente OCTOPUS DESIGNER-a i njihova međusobna interakcija. Važno je za primijetiti da su *DeMakGUI* i *DeMakMain* neovisni o definiranom projektnom modelu.

DeModel komponenta preslikava DeModel komponentu (npr. OCTOPUS Analyzer za strukturne probleme) u generički oblik, time definirajući generički model sa implementiranim konceptima sučelja (*interface*) koji će u nastavku biti objašnjeni Takvom programskom arhitekturom omogućuje se komunikacija između komponenti ovisnih o problemu i komponenti neovisnih o problemu.



Sl. 7.1. Komponente OCTOPUS DESIGNER

Važno je napomenuti da nije potrebno rekompajliranje komponenti neovisnih o problemu za povezivanje s novim implementacijama *DeModel* komponenti te da je moguća istovremena komunikacija s više *DeModel* komponenti čime se na razini okruženja za potporu pri projektiranju omogućuje integracija različitih alata za analizu (npr. alata za proračun aerodinamike ili hidrodinamike s alatima za analizu čvrstoće).

7.1. Tražene karakteristike okružja za potporu pri projektiranju

Dugotrajni razvoj arhitekture okruženja za potporu pri projektiranju (*Decision Support Problem*) od 1992 do sada, zasnovan je na preciznoj definiciji željenih karakteristika okruženja, gledano sa stajališta implementacije i korištenja okružja:

1) prilagodljivost modulima za analizu

   Jedna od najznačajnijih karakteristika sustava je mogućnost integriranja raznih modula za analizu problema (dosada su primijenjeni modeli za projektiranje brodova i zrakoplova, tankostjenih konstrukcija, održavanje brodova i sl.). Pri tome važno da se modeli analize mogu integrirati primjereno kompleksnosti strukture modela analize s time da se jednostavni modeli trebaju integrirati vrlo jednostavno i brzo putem jednog ili više analitičkih blokova. U ovome radu razmatrani su modeli sa složenom, hijerarhijskom strukturom koji se trebaju integrirati čuvajući što je moguće više od svoje logičke strukture te je s tom svrhom razvijen novi, generički strukturirani model (potpoglavlje 7.2.3).

2) omogućiti korištenje velikog skupa metoda sinteze

   Kao što je navedeno u prethodnim poglavljima, za rješavanje složenih problema projektiranje iz inženjerske prakse potrebno je omogućiti korištenje raznih metoda sinteze koje uključuju optimizacijske metode te planiranje pokusa. U ovom radu izrađen je tzv. surogat modeler (potpoglavlje 7.2.8) koji omogućuje integraciju različitih tipova surogat modela. Razvijen je i objektni model općenitog polinomski surogat model (potpoglavlje7.2.8.1) kojem je moguće dodavati različite tipove linearnih funkcija, a do sada su kreirani linearni, interakcijski, čisti kvadratni i puni kvadratni model. U surogat modeler su integrirani i prilagođeni Kriging i RBF surogat modeli razvijeni u diplomskom radu [88], te je razvijen općeniti polinomski surogat model.

3) omogućiti paralelno izvršavanja određenih sekvenci naredbi

   Paralelno procesiranje u prethodnoj inačici programa OCTOPUS primijenjeno je prvi puta u radu [80] na proračunskoj plohi s 40 procesora. S obzirom na karakteristike arhitekture današnjih računala, kod kojih se ukupna proračunska snaga više bitno ne povećava povišenjem radnog takta procesora, već povećanjem broja radnih jezgri, jedna od bitnih karakteristika jest mogućnost paralelnog izvršavanja određenih sekvenci naredbi. U okviru ovog rada razvijen je asinkroni rješavaš višestruko-ponavljajućih zadataka baziran na osnovnim bibliotekama .NET Remoting-a, koji

osim iskorištavanja višeprocesorskih resursa jednog računala omogućuje i povezivanje računala u male klastere (potpoglavlje 7.2.9).

4) omogućiti jednostavnu i kontinuiranu nadogradnju optimizacijskih metoda (hibridni rješavači).

   Već postojećim rješavačima integriranim u OCTOPUS Designer (MCS, ES, FFD, MOGA) u ovom radu dodani su novi. Tijekom ovog istraživanja izrađen je i bazni model za gradijentne optimizacijske algoritme koji izračunom numeričkih derivacija omogućuje rad s onim modelima analize koji nemaju mogućnost analitičkog izračuna derivacija funkcije cilja i ograničenja. Time je omogućena integracija NLPQLP optimizacijskog algoritam te razvoj novog SLP algoritma koji je baziran na vrlo efikasnom algoritmu implementiranom u MAESTRO [74]. U suradnji sa studentima, tijekom izrade seminarskog rada [21], razvijeni su višeciljni optimizacijski PSO algoritmi.

5) razviti potporu za interaktivno zadavanje projektnog problema

   Od ishodišnog interaktivnog modula ([79], [81]), kroz diplomski rad [89] te [87] razvijan je interaktivni model. U ovome radu omogućeno je zadavanje fleksibilne sekvence izvršavanja projektnog problema pomoću interaktivnog dijagrama toka čime je značajno olakšano razumijevanje složenih projektnih problema [90]. Također je omogućeno interaktivno zadavanje dekomponiranog problema u zasebnom modulu te intuitivno povezivanje koordinacijskih metoda s dekompozicijskim sustavima.

6) razviti potporu za interaktivnu grafičku prezentaciju i analizu rezultata.

   Iz osnovnih funkcionalnosti ishodišnog grafičkog/vizalizacijskog modula ([79], [81]), kroz diplomski rad [89] te [87]razvijen je 5D grafički model. U ovom radu ovaj modul je proširen s više grafičkih 2D prezentacija poput vizualizacije rješenja optimizacijskog problema, vizualizacije variograma, vizualizacije utjecaja pojedinih deskriptora na aproksimirane odzive.

7.2. Opis arhitekture

U nastavku je dan opis arhitekture razvijenog okruženja za potporu pri projektiranju pomoć UML (*Unified Modeling Language*) dijagrama [91] popraćen s tekstualnim opisom. Pri tome su u tekstu entiteti iz samog programskog koda označeni drugim fontom, a imena klasa i

koncepta sučelja dodatno označeni plavom bojom. Tako npr. `DeMakProblem` označava klasu DeMakProblem, a `DeMakProblem()` konstruktor te klase.

7.2.1. Unified Modeling Language (UML) dijagrami

UML (*Unified Modeling Language*) je unificirani jezik za grafički prikaz programskih rješenja, koji omogućuje programerima da skiciraju svoje ideje na standardan i lako razumljiv način. Upravo vizualna strana UML dijagrama omogućava da se ideje lako prezentiraju svim sudionicima, kako programerima tako i korisnicima.

UML dijagrami se sastoje od niza grafičkih elemenata čijim se kombiniranjem stvaraju vizualno razumljivi dijagrami. Svrha tih dijagrama je višestruki prikaz sustava, kojeg tada nazivamo modelom. UML model najlakše se može dočarati kao skalirani model zgrade zajedno s arhitektovim tumačenjem zgrade. Važno je za zapamtiti da UML model opisuje svrhu sustava, ali ne i njegovu implementaciju.

Pošto je UML jezik, tada mora imati skup pravila za kombiniranje građevnih elemenata. Pravila i elementi razlikuju se za svaki pojedini tip UML dijagrama. Postoje sljedeći tipovi dijagrama:

- dijagram klasa (*Class Diagram*)
- dijagram objekata (*Object Diagram*)
- dijagram koji opisuje ponašanje sistema onako kako ga vidi korisnik (*Use Case Diagram*)
- dijagram stanja (*State Diagram*)
- dijagram toka (*Sequence Diagram*)
- dijagram aktivnosti (*Activity Diagram*)
- komunikacijski dijagram (*Communication Diagram*)
- dijagram komponenti (*Component Diagram*)
- implementacijski dijagram (*Deployment Diagram*)

Od gore navedenih tipova UML dijagrama, za opis programskog sustava OCTOPUS DESIGNER, korišten je samo dijagram klasa koji je moguće dobiti automatski na temelju izvornog programskog koda u Visual Studio 2008 razvojnom okolišu [86]. Isto tako važno je napomenut da izrađeni dijagrami sami po sebi ne mogu u potpunosti objasniti način rada i implementiranja novih komponenti, ali mogu značajno ubrzati proces upoznavanja implementiranog koda.

Tipičan *Class Diagram* () prikazuje polja sa nazivima klasa - npr. *DeMakJob*, te njihovim tipom - npr. *Abstract Class*. U grafičkom prikazu klase mogu se prikazati - metode (*methods*), polja (*fields*) i svojstva (*properties*). Prilikom prikaza pojedine klase unutar *Class Diagram-a* popis objekata koje klasa sadržava ne mora biti potpun, tj. prikazuju se samo oni objekte koji su bitni za razumijevanje dok se svi ostali objekti mogu sakriti. Grafički prikaz nasljeđivanja i sadržavanja unutar dijagrama klasa specificiran je na sljedeći način:

- nasljeđivanje je prikazano linijom s bijelo ispunjenom strelicom (`OptimizationJob` klasa je izvedena iz klase `DeMakJob`)
- sadržavanje preko svojstava prikazano je smeđom linijom s dvostrukom strelicom ukoliko je svojstvo polje ili takozvana kolekcija objekata (*collection*), a za sve ostale tipove koristi se jednostruka strelica
- sadržavanje preko *field-a* prikazano je plavom linijom s dvostrukom strelicom ukoliko je *field* polje, a za sve ostale tipove koristi se jednostruka strelica
- apstraktne metode i svojstva klasa, kao i sve metode i svojstva koje propisuju koncepti sučelja, prikazani su u kurzivu

7.2.2. Osnovne klase okruženja za potporu pri projektiranju

Osnovne klase i koncepti sučelja okruženja za potporu pri projektiranju prikazani su na Sl. 7.2. Na vrhu Sl. 7.2 nalazi se klasa `DeMakProblem` koja sadrži listu poslova (`List<DeMakJob> DeMakJobs`) te upravljaju izvođenjem skupine modela analize (`List<IAnalysisModel> AnalysisModels`) pri rješavanju određenog problema projektiranja. Kao što se vidi na Sl. 7.2, definirane su dva različita tipa poslova koje se pri projektiranju mogu obavljati sa stajališta načina izvršavanja određenog posla*:* `OptimizationJob` i `DOEBasedJob`. Prvim tipom posla upravlja odabrani optimizacijski algoritam odnosno određena sekvenca rješavanja unutar koje se nalaze optimizacijski potproblemi. Drugi tip posla je okarakteriziran korištenjem metode planiranja pokusa (DOE). Za razliku od prvog tipa posla, kod njega se prije pokretanja sekvence posla zna točno koji proračuni će biti obavljeni. Svi tipovi poslova moraju implementirati apstraktnu metodu `Sequence()` kojom se pokreće odgovarajući radni zadatak.

Prikazani klase predstavljaju glavne dijelove OCTOPUS DESIGNER-a kroz koje će se objasniti osnove implementacije teorije, prikazane u prethodnim poglavljima, u objektno orijentirani programski kod.

Sl. 7.2. Osnovne klase i koncepti sučelja okruženja za potporu pri projektiranju

7.2.3. Generički model za analizu (*DeModel*)

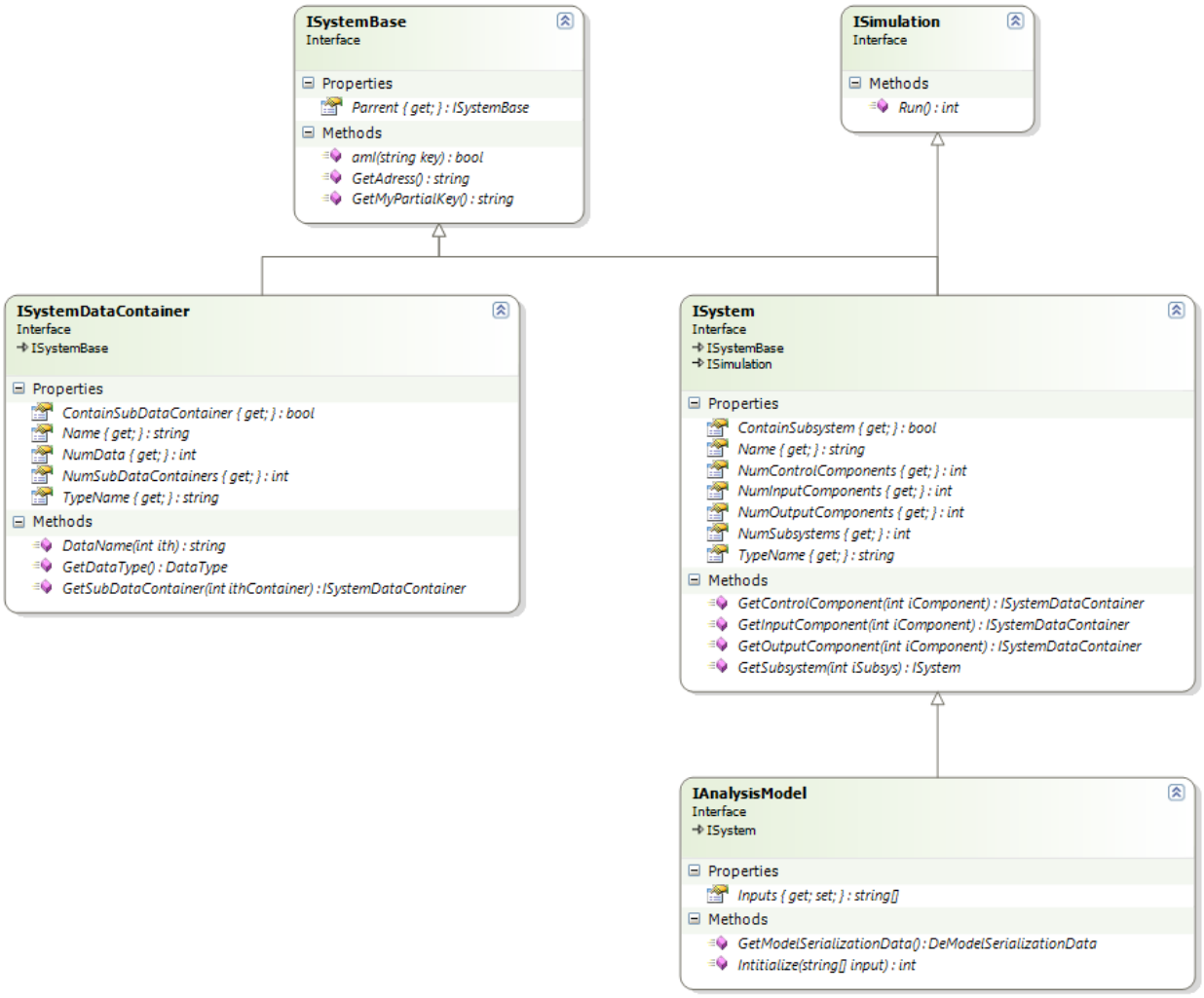
Kao što je već navedeno u zahtijevanim karakteristikama okruženja, jedan od glavnih zadataka je omogućiti komunikaciju odnosno integraciju s alatima za analizu problema. Pri tome je naglašeno da je potrebno omogućiti integraciju primjerenu složenosti strukture modela analize, odnosno da se jednostavni modeli mogu integrirati vrlo jednostavno i brzo, a da se

modeli sa složenom, često hijerarhijskom, strukturom mogu integrirati čuvajući što je moguće više od svoje logičke strukture.

Kako bi se omogućilo što jasnije prenošenje logičke strukture alata za analizu zamišljeno je da se model alata za analizu može sastojati od proizvoljnog broja sustava koji imaju ulazne podatke, izlazne podatke te druge podsustave, dok je generički model i sam preuzima ponašanje sustava. Kako bi se omogućila ovako logički jednostavno no implementacijski složeno ponašanje definiran su koncepti sučelja (*interface*) prema dijagramu na Sl. 7.3.

Dva osnovna koncepta sučelja na dijagramu su `ISystemDataContainer` i `ISystem`. `ISystemDataContainer` a definiraju osnovno ponašanje kontejnera podataka. Svaki kontejner podataka mora znati vratiti svoje ime (`Name`), ukupni broj podataka koje sadržava (`NumData`) te ime pojedinog podatka(`DataName(ith)`). `ISystemDataContainer` koncept sučelja omogućuje da kontejner podataka sadržava i podkontejnere podataka te na poziv svojstva `NumSubDataContainers` vraća broj podkontejnera, dok pozivom metode `GetSubDataContainer(ith)`vraća pojedini podkontejner.

`ISystem` koncept sučelja definira ponašanje sustava unutar modula analize. Pri tome svaki sustav može imati referencu na tri tipa podataka: ulazne, kontrolne i izlazne. Osim toga svaki sustav mora implementirati i ponašanje koncepta sučelja `ISimulation` kojim se omogućava nezavisno pokretanje simulacije pojedinog sustava.



Sl. 7.3. Osnovni koncepti sučelja generičkog modela za analizu

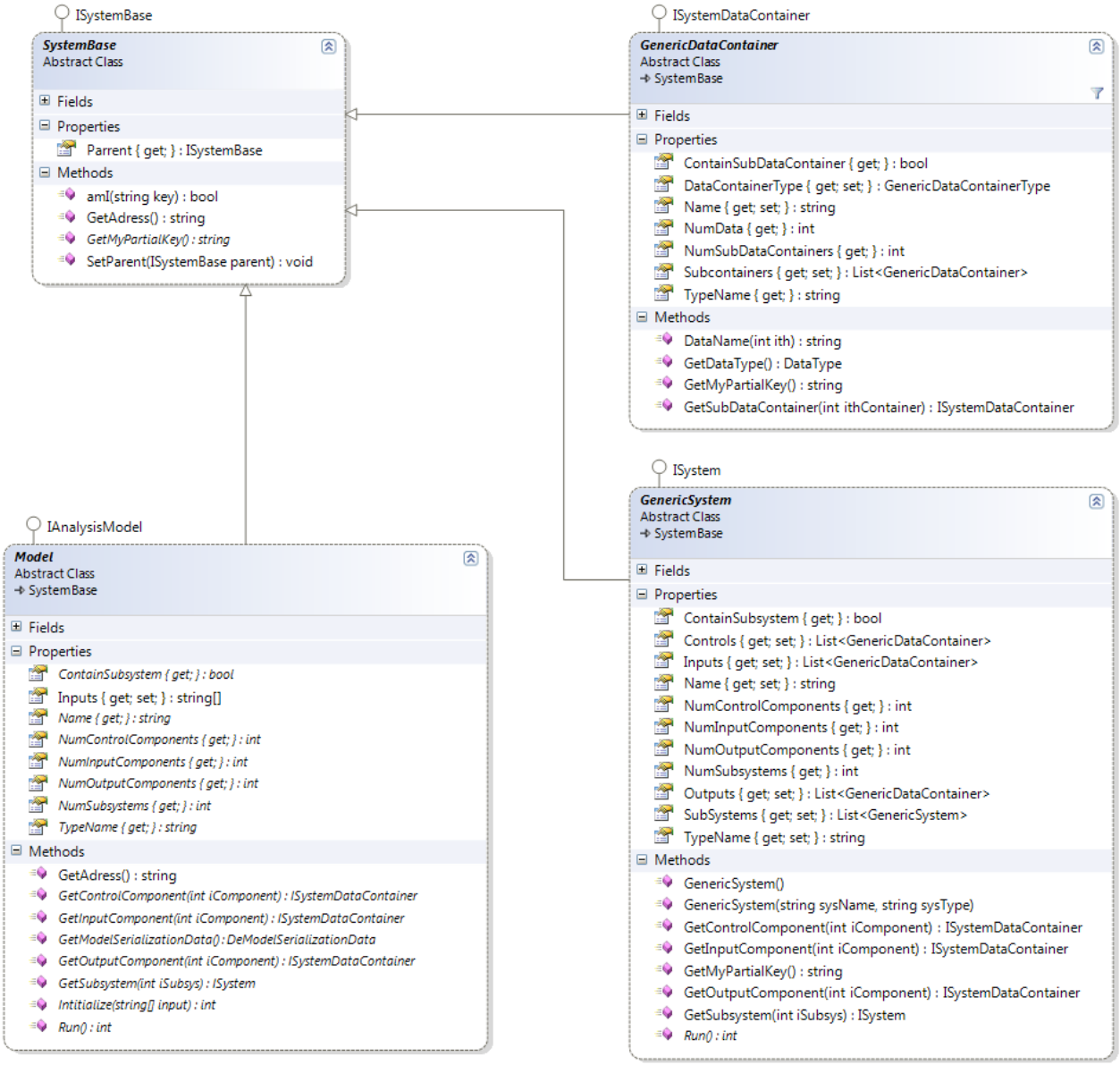
Oba osnovna koncepta sučelja imaju definirano bazično ponašanje konceptom sučelja `ISystemBase`. Kroz njega je definirano tko je roditelj (`Parent`) kojem objektu, odnosno gdje je definiran određeni objekt. Važno je napomenuti da je moguće da dva podsustava imaju reference na iste kontejnere ulaznih podataka, dok je roditelj može biti sustav koji je nadređen tim podsustavima ili čak sam model (npr. Maestro svojstva (*property*) su definirani na samom modelu, a svaki *StructuralSubsystem* kao podsustav specificira da ovisi o pojedinim svojstvima). Naravno, svaki `ISystemBase` objekt ima samo jednog roditelja.

`IAnalysisModel` uz već definirano ponašanje kroz `ISystem`, omogućuje spremanje podataka potrebnih da bi za ponovno pokretanje tj. Inicijalizaciju pojedinog modela analize prilikom pokretanja određenog problema projektiranja.

Važno je naglasiti da generički model definiran ovim konceptima sučelja omogućuje da se pojedini model za analizu implementira na različite načine, odnosno da se odabere onaj način za koji se smatra da najbolje odgovara uzevši u obzir prenošenje logičke strukture, ali i vrijeme integracije.

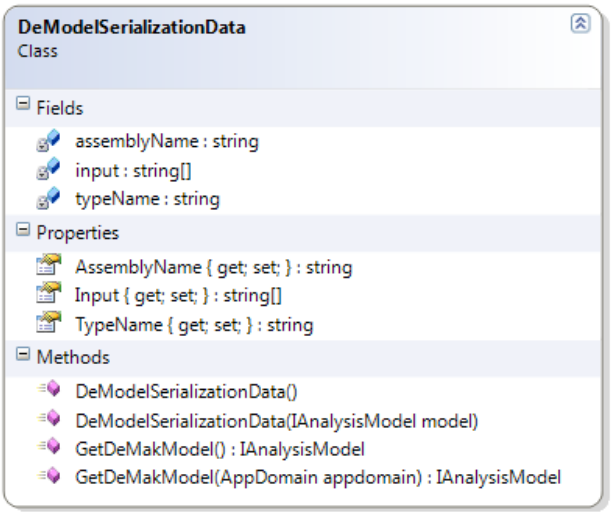
Drugi važan dio koncepta sučelja `ISystemBase` su metode za identifikaciju objekata u implementiranom modelu analize pri serijalizaciji odnosno deserijalizaciji podataka. To su metode `GetMyPartialKey()`, `GetAdress()` i `amI(key)`. `GetMyPartialKey()` metoda vraća niz znakova koji jedinstveno identificiraju `ISystemBase` instancu unutar svoga roditelja, dok `GetAdress()` vraća jedinstveni niz znakova koji jedinstveno identificira `ISystemBase` instancu unutar modela analize tj. instance klase koja implementira `IAnalysisModel`. Metoda `amI(key)` uspoređuje ulazni niz znakova s adresom objekta te vraća vrijednost `true` ukoliko se poklapaju.

Kako bi se omogućila jednostavnija integracija novih modela analize, osim definiranog koncepta sučelja, *DeModel* komponenta nudi i implementaciju gore opisanog sučelja unutar nekoliko apstraktnih klasa koje i dalje ostaju dovoljno općenite da se može integrirati bilo koji model analize (Sl. 7.4). Ukoliko korisnik na ovaj način odluči integrirati svoj model analize, određeni dio koncepta sučelja je već implementiran, dok je neophodno implementirati one metode koje su proglašene apstraktnima (na dijagramu prikazani kurzivom). Trenutno su svi integrirani alati za analizu implementirani korištenjem ovih osnovnih klasa, ali ne postoji nikakva zapreka da korisnik implementira svoj model korištenjem direktno koncepata sučelja.



Sl. 7.4. Generička implementacija osnovnih koncepata sučelja generičkog modela za analizu

Kao što je navedeno u uvodu ovog poglavlja jedna od osnovnih ideja je bila mogućnost dodavanja novih tipova modula za analize bez potrebe za rekompajliranjem ostatka okruženja tj. dinamički. Klasa zadužena za dinamičko povezivanje modela analize s programskim okruženjem je `DeModelSerializationData` i prikazana je na Sl. 7.5.



Sl. 7.5. Klasa zadužena za dinamičko povezivanje korištenih modula za analizu

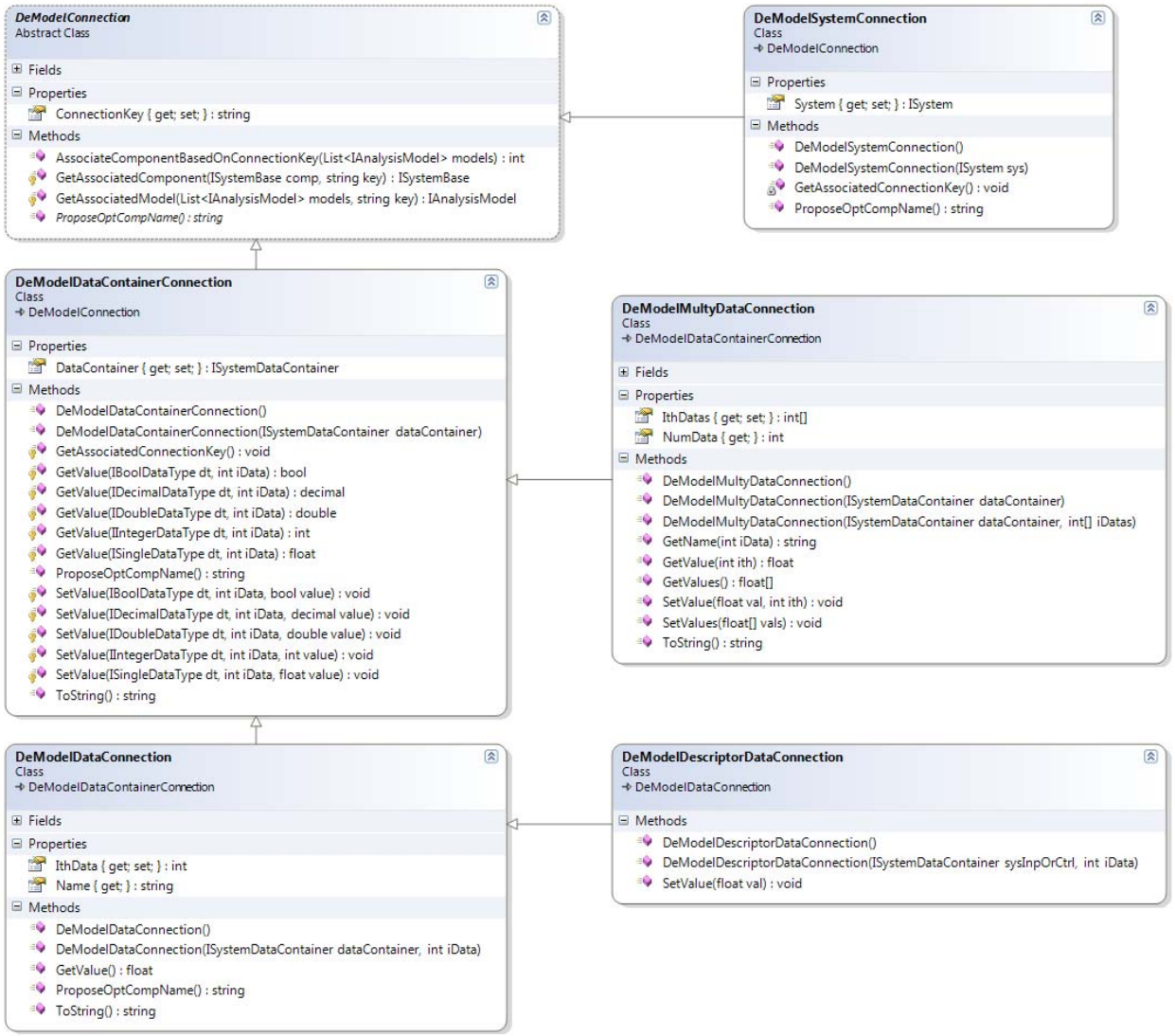
Microsoft .NET omogućava dinamičko kreiranje nove instance neke klase tako da se specificira ime klase i ime assemblija u kome se nalazi, metodom `System.Activator.CreateInstance(assemblyName, className)` i upravo ta metoda se koristi unutar `GetDeMakModel()` metode za instanciranje specificiranog modela. Pri tome je potrebno zadati svojstva: `AssemblyName`, `TypeName`, `Input[]`. `TypeName` jest ime klase koja implementira koncept sučelja `IAnalysisModel`, dok je `AssemblyName` ime assemblija u kome se klasa nalazi. `Input[]` je jednodimenzionalno polje znakova s podacima o konkretnom problemu koji se analizira. Npr. za slučaj projektiranja broda za prijevoz putnika i vozila korištenjem Maestro alata za analizu tankostjene konstrukcije potrebni podaci izgledaju:

```
<SerializationModels>
  <DeModelSerializationData>
    <Input>
      <string>RoPax22.xml</string>
    </Input>
    <AssemblyName>MaestroModel</AssemblyName>
    <TypeName>MaestroModel.MaestroModel</TypeName>
  </DeModelSerializationData>
</SerializationModels>
```

S obzirom da se ovo dinamičko instanciranje objekta modula za analizu odvija samo jednom pri otvaranju određenog DeMak problema, osnovni nedostatak ovakvog tipa instanciranja, sporije instanciranje, jest zanemariv.

Povezivanje optimizacijskih podataka koji se nalaze unutar *DeMakMain* komponente izvodi se pomoću klase `DeModelConnection` odnosno odgovarajućih izvedenih klasa ovisno o tome o kojem tipu optimizacijskog podatka je riječ (Sl. 7.6). `DeModelConnection` klasa omogućuje prenošenje informacije o trenutnom stanju određene komponente modela

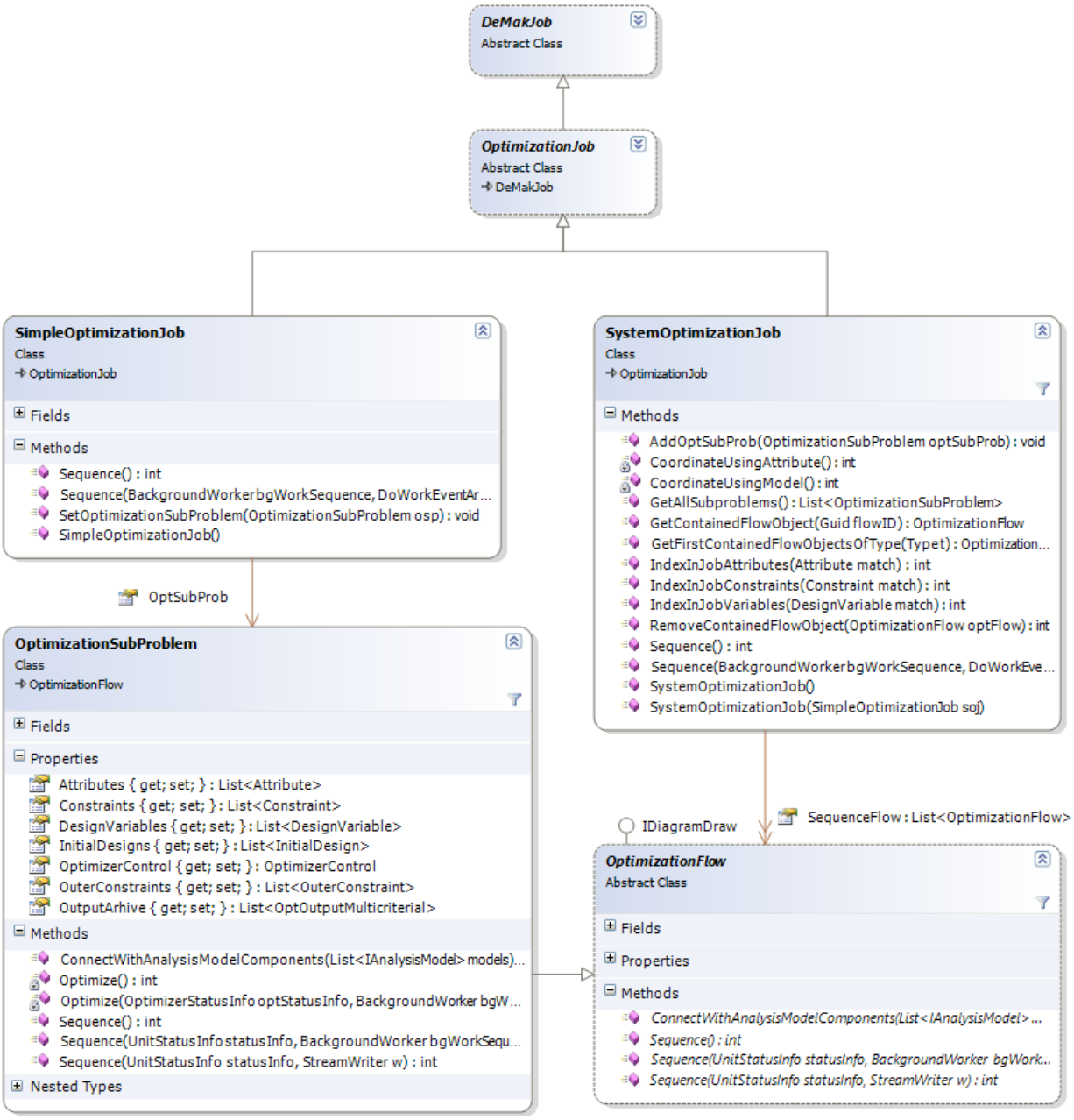
metodom `GetValue()`, dok je izmjena trenutnog stanja pojedine komponente omogućena metodom `SetValue(value)`. Druga važna funkcija ove klase je serijalizacija odnosno povezivanje pojedine komponente s modelom analize nakon deserijalizacije. Osnova za spajanje sa modelom je svojstvo `ConnectionKey` koje zapravo sadržava adresu spojene `ISystemBase` instance. Nakon deserijalizacije poziva se metoda `AssociateComponentBasedOnConnectionKey(List<IAnalysisModel> models)` koja pretražuje sve modele analize koji se nalaze u Listi modela te traži komponentu s adresom koja je jednaka svojstvu `ConnectionKey`. Ukoliko se komponenta ne pronađe, korisniku se javlja upozorenje s detaljnom informacijom koja optimizacijska komponenta nije spojena na odgovarajuću komponentu modela analize.



Sl. 7.6. Dijagram klasa zaduženih za povezivanje generičkog modela za analizu s optimizacijskim komponentama

7.2.4. Optimizacijski poslovi u okruženju za potporu projektiranju

Unutar apstraktne klase `OptimizationJob` definirana su osnovna svojstva i ponašanje metode koje jedan optimizacijski posao mora imati. Izvedene klase su `SimpleOptimizationJob` i `SystemOptimizationJob`. (Sl. 7.7)



Sl. 7.7. Klase za definiciju optimizacijskih zadataka (`OptimizationJob`)

`SimpleOptimizationJob` klasa se koristi za definiranje jednostavnih optimizacijskih zadataka i zapravo sadržava samo jednu instancu klase `OptimizationSubProblem`, kojom su zapravo definirane sve optimizacijske komponente problema zajedno s odabranim optimizacijskim algoritmom (više u potpoglavlju 7.2.4.1).

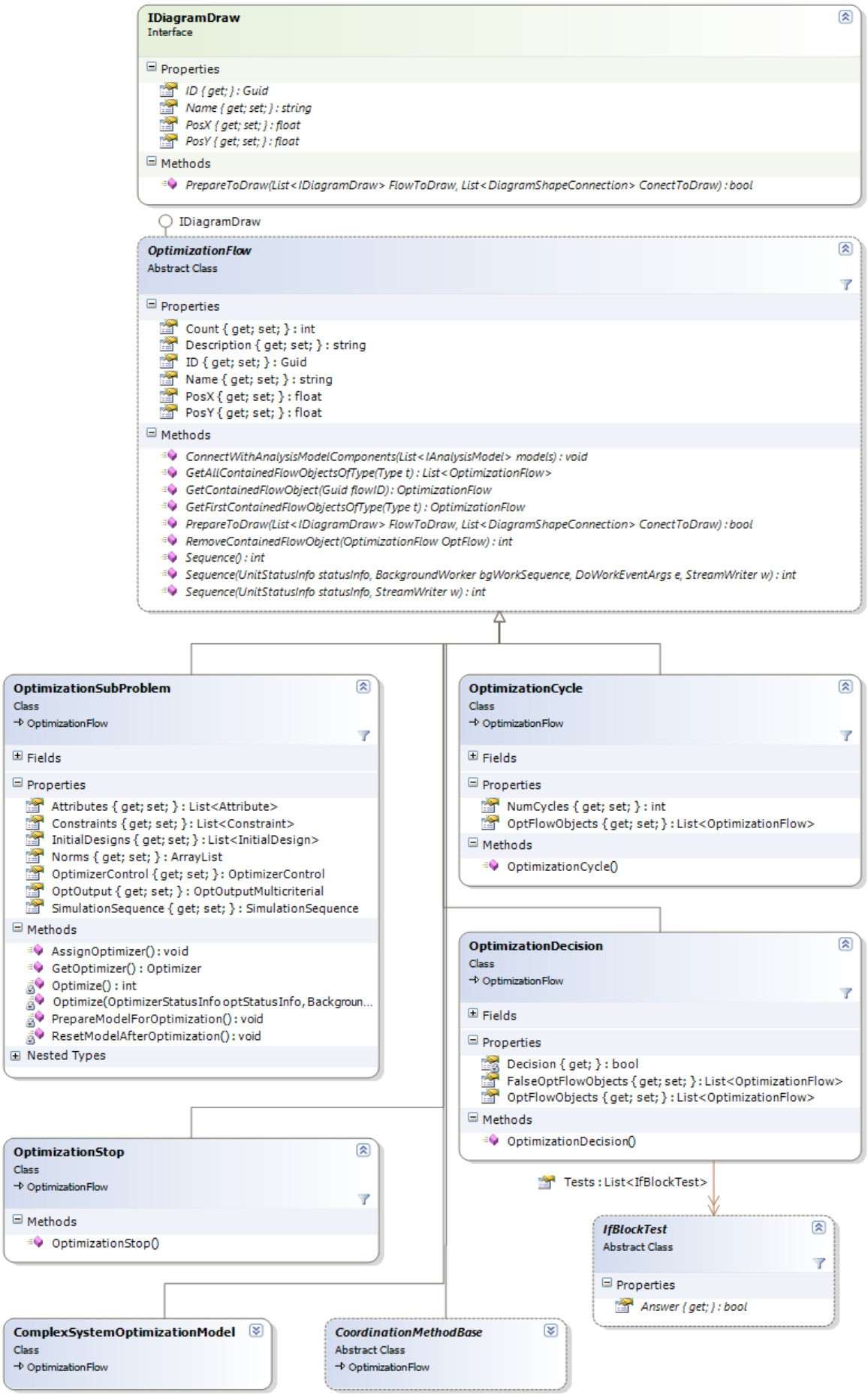
Klasa SystemOptimizationJob posjeduje sve potrebne metode i objekte za rješavanje realnih inženjerskih problema koji zahtijevaju dekompoziciju projektnog problema, hibridne optimizacijske algoritme te neke druge oblike napredne programske logike. Da bi to bilo moguće SystemOptimizationJob sadrži kolekciju instanci klase OptimizationFlow koje upravljaju sa procesom. OptimizationFlow klasa i pripadajuće izvedene klase detaljno su opisane u potpoglavlju 7.2.4.1, a na Sl. 7.7 su prikazana sa samo osnovnim svojstvima i metodama radi lakšeg razumijevanja njihova odnosa s OptimizationJob-om.

Metoda Sequence()u SystemOptimizationJob-u redom pokreće izvršavanje istoimenih metoda u sadržanim List<OptimizationFlow> SequenceFlow objektima.

7.2.4.1. Klase za detaljno i fleksibilno definirane optimizacijske sekvence

OptimizationFlow apstraktna bazna klasa svim klasama kojim se upravlja izvršavanjem određenog složenog optimizacijskog zadatka (Sl. 7.8). Uz samo izvođenje sekvence ova klasa osigurava osnovne informacije potrebne za grafičko predočavanje sekvence u interaktivnom dijagramu implementiranjem IDiagramDraw koncepta sučelja. Kao što je već navedeno izvođenje instance bilo koje klase izvedene iz OptimizationFlow bazne klase mora implementirati metodu Sequence()kojom se izvršava dio sekvence definiran tim objektom.

Kao što se vidi na Sl. 7.8 klase koje nasljeđuju OptimizationFlow baznu klasu su: OptimizationSubProblem, OptimizationCylcle, OptimizationDecision, OptimizationStop, ComplexSystemOptimizationModel i CoordinationMethodBase. Pri tome su zadnje dvije klase izdvojene zbog posebnosti njihova ponašanja. Upravo tim klasama je omogućena dekompozicija i koordinacija složenih sustava te će biti prikazane zasebno u potpoglavlju 7.2.5.



Sl. 7.8. Klase za detaljno i fleksibilno definiranje optimizacijske sekvence (`OptimizationFlow`)

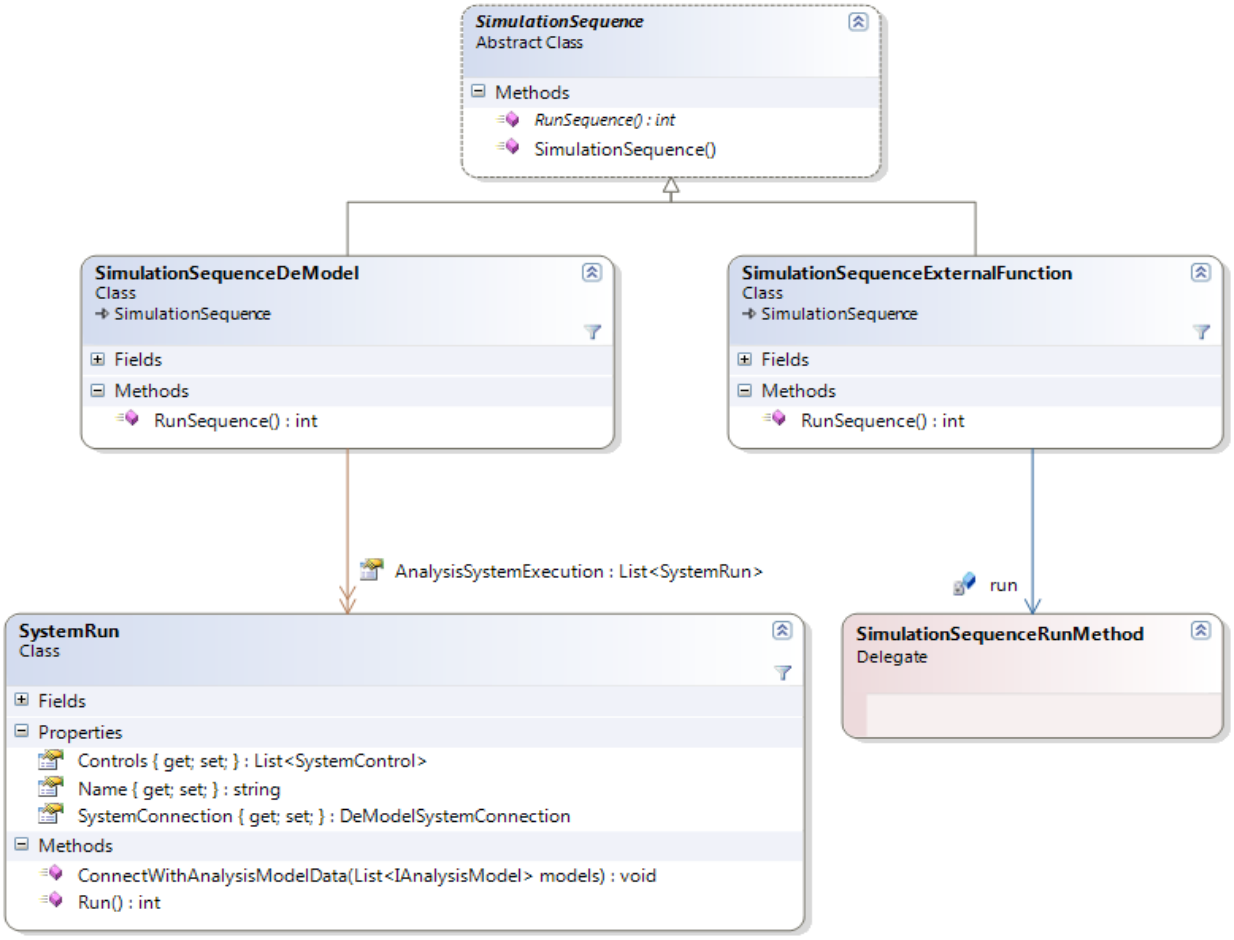
Klasa OptimizationSubProblem

Klasa `OptimizationSubProblem` služi za definiranje jednostavnog optimizacijskog problema i kao što je već ranije navedeno jednostavni optimizacijski zadaci mogu se riješiti korištenjem instance ove klase u okviru `SimpleOptimizationJob`-a. Ova klasa sadržava sve podatke koji su potrebni optimizacijskoj metodi tj. sadržava formulaciju optimizacijskog potproblema sa svim komponentama optimizacijskog problema, kontrolne podatke optimizacijskog algoritma, sučelje za pozivanje potrebnih proračuna u integriranom modulu za analizu nakon promjene projektnih varijabli te naravno i rezultirajuće projektna rješenja kao izlaz odabrane optimizacijske metode.

Liste koje sadržavaju komponente optimizacijskog problema su: `List<DesignVariable> DesignVariables`, `List<Attribute> Attributes`, `List<Constraint> Constraints`, `List<OuterConstraint> OuterConstraints`. Klase koje definiraju komponente optimizacijskog problema zasebno su objašnjene u potpoglavlju 7.2.6.

Osim komponenti optimizacijskog problema, ova klasa sadržava kontrolne podatke odabrane optimizacijske metode svojstvom `OptimizerControl OptimizerControl`. Optimizator (`Optimizer`) tj. implementirana optimizacijska metoda interno se instancira pri pokretanju metode `Sequence()` ili eksplicitno metodom `AssignOptimizer()`, na osnovu odabranih kontrolnih podataka.

Sučelje za pozivanje potrebnih proračuna u modelu analize nakon promjene projektnih varijabli dano je svojstvom `SimulationSequence` koje je tipa `SimulationSequence`. Osim za optimizacijske poslove ovo sučelje se koristi i za sve zadatke koje zahtijevaju određeni proračun modela analize te se koristi i kod poslova baziranih na planiranju pokusa.



Sl. 7.9. Klase za definiciju optimizacijskih zadataka (`OptimizationJob`)

Klasom `SimulationSequence` omogućeno je onoliko detaljna kontrola izvršavanja proračuna koliko to dopuštaju integrirani moduli za analizu projektiranog problema sa nezavisnim pozivanjem svojih podsustava kontroliranim pomoću podataka u `ControlDataContainer`-ima. Na Sl. 7.9 uz baznu klasu prikazane su klase `SimulationSequenceDeModel` te `SimulationSequenceExternalFunction`. Prvom klasom omogućuje se izvođenje jedne ili više `SystemRun` instanci od kojih svaka može imati vezu s jednom instancom `ISystem` koncepta sučelja. Time je omogućena analiza samo onih sustava koji se i koriste u nekom zadatku odvojeno od ostatka modela

Klasom `SimulationSequenceExternalFunction` omogućeno je definiranje simulacije koja je definirana vanjskom metodom tzv. delegatom (odgovara pokazivaču na funkciju u nekim drugim programskim jezicima) , čime je omogućeno da pozivanjem metode `RunSequence()` pozove složenija sekvenca proračuna od jednostavne analize sustava određenog modula analize.

Klasa OptimizationCycle

Iduća klasa koja nasljeđuje baznu klasu `OptimizatonFlow` jest `OptimizationCycle` (Sl. 7.8). Njome je omogućeno ponavljanje sekvence `OptimizationFlow` objekata koji se nalaze u listi `OptFlowObjects` onoliko puta koliko je specificirano svojstvom `NumCycles`.

Klasa OptimizationDecision

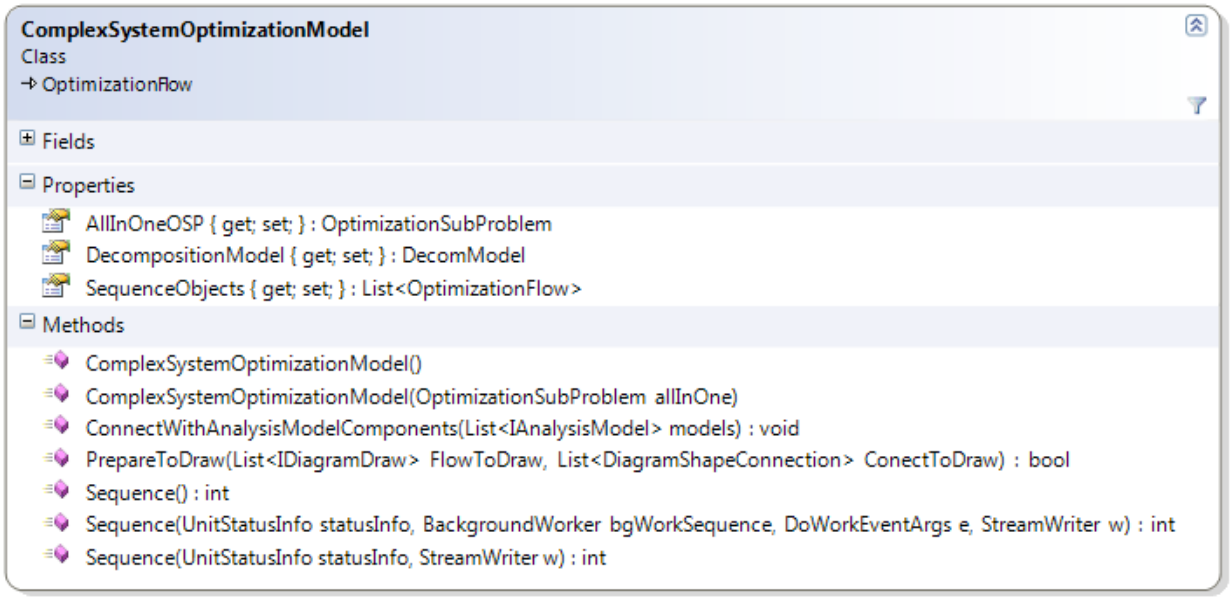
Ovom klasom je omogućeno donošenje odluka o tome koju grupu `SequenceFlow` objekata izvoditi ovisno o postavljenom zadovoljenju propisanog testa. Ako niz testova zadanih test istinit svojstvo izvodi se sekvenca objekata koji se nalaze u listi `OptFlowObjects`, dok se u suprotnom izvodi sekvenca objekata koji se nalaze u listi `FalseOptFlowObjects`.

Klasa OptimizationStop

Izvođenje sekvence `OptimizationFlow` objekata nekog `SystemOptimizationJob`-a završava se automatski nakon što se izvedu svi objekti u listi `SequenceFlow`, međutim ponekad je proces potrebno zaustaviti nakon zadovoljenja određenog uvjeta propisanog u `OptimizationDecision` objektu, što je omogućeno klasom `OptimizationStop`.

7.2.5. Klase za definiranje optimizacijskih problema složenih sustava korištenjem dekompozicije na podsustave

Kao što je navedeno ranije, klasa `ComplexSystemOptimizationModel` (Sl. 7.10) izdvojena je iz prikaza ostalih `OptimizationFlow` klasa zbog svoje važnosti i složenosti. Njome, odnosno ponašanjem koje daju sadržane klase omogućena je dekompozicija i koordinacija složenih optimizacijskih problema. Pri tome je zamišljeno da se projektiranje uvijek započinje definiranjem ukupnog problema (All In One), a da se dekompozicija koristi samo onda kad se procjeni da je potrebna, odnosno da će rješavanje projektnog problema na ovaj način biti efikasnije. Stoga konstruktor ove klase zahtjeva generirani ukupni optimizacijski problem te ga sprema kao svojstvo `OptimizationSubProblem` `AllInOneOSP`. Ukupni optimizacijski problem se nakon toga može dekomponirati, čime se dobiva `DecomModel` `DecompositionModel`.Za rješavanje dekomponiranog modela neophodno je odabrati odgovarajuću koordinacijsku metodu koja zajedno s drugim objektima u `List<OptimizationFlow>` `SequenceObjects` rješava dekomponirani model. U potpoglavlju 7.2.5.1 dan je detaljan opis dekomponiranog modela, dok je u potpoglavlju 7.2.5.2 dan opis koordinacijskih metoda za rješavanje dekomponiranog modela.



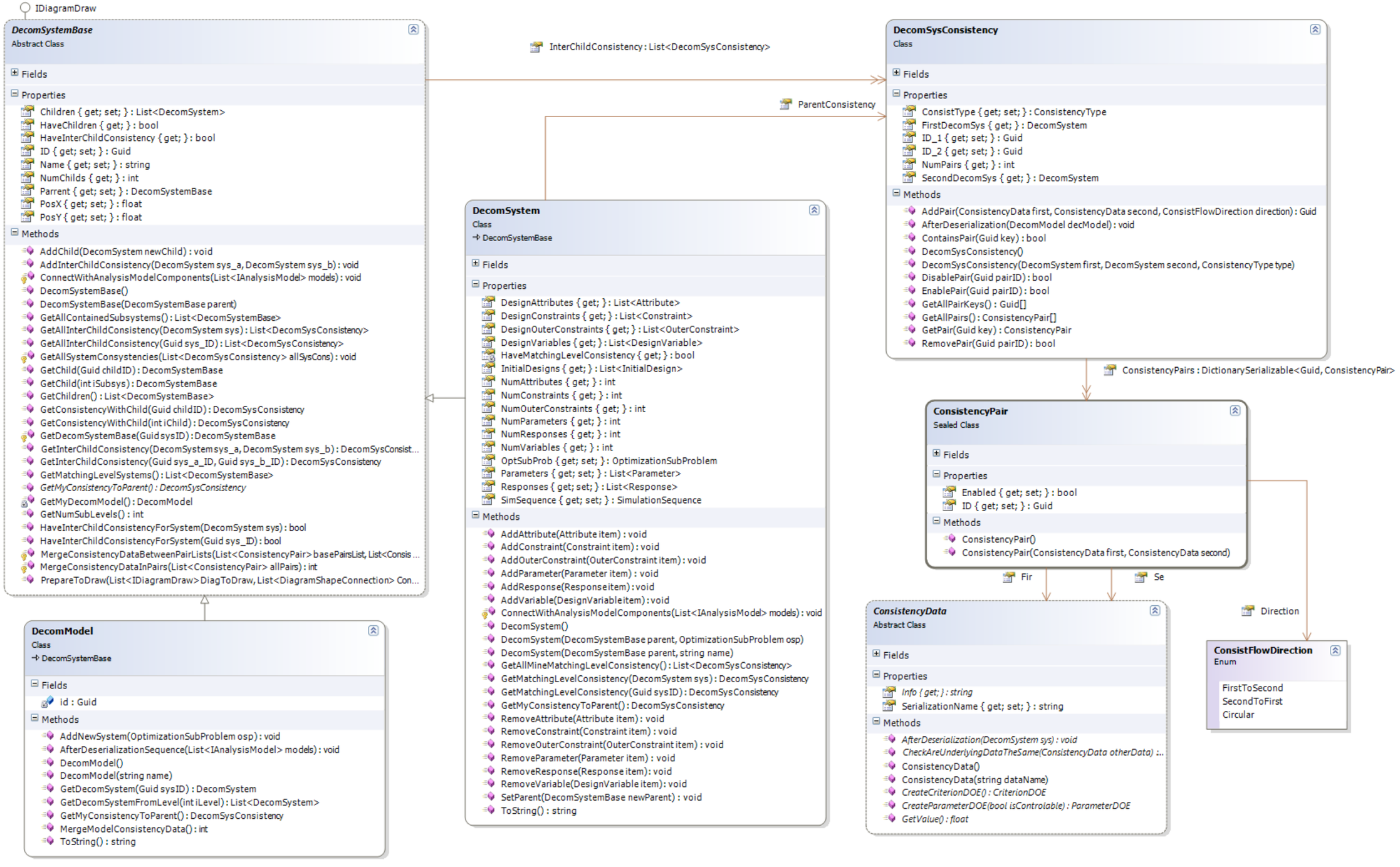
Sl. 7.10. Klasa `ComplexSystemOptimizationModel`

7.2.5.1. Klase za definiranje dekompozicijskog modela

Na Sl. 7.11 prikazane su klase koje omogućuju generiranje dekompozicijskog modela. Pri tome je okosnica svega klasa `DecomSystem` koja definira jedan dekomponirani sustav odnosno podsustav. Ukupni optimizacijski problem `AllInOneOSP`, koji je sadržan u `ComplexSystemOptimizationModel` klasi, može se automatski ili ručno dekomponirati na više dekompozicijskih podsustava. Automatska dekompozicija je omogućena specificiranjem `ISystem` tipa (objašnjen u potpoglavlju 7.2.3) prema kojem se iz `AllInOneOSP` izdvajaju dekompozicijski podsustavi koji sadrže optimizacijske komponente (varijable, ograničenja i atribute) vezane za dani tip.

Kao što je vidljivo na Sl. 7.11, klase `DecomSystem` i `DecomModel` nasljeđuju baznu klasu `DecomSystemBase` koja implementira zajednički dio ponašanja. `DecomModel` i `DecomSystem`, sadržavaju `List<DecomSystem> Children`, iz čega i proizlazi zajedničko ponašanje. Drugim riječima `DecomModel` sadržava jedan ili više dekompozicijskih sustava (`DecomSystem`) koji su na vrhu hijerarhije , nemaju roditelja koji je istovremeno i sam dekompozicijski sustav tj. njihov `DecomSystemBase Parrent` jest upravo instanca klase `DecomModel`.

Sa slike je vidljivo da su u dekompozicijskom sustavu definirane komponente optimizacijskog problema listama: `List<DesignVariable> DesignVariables`, `List<Attribute> DesignAttributes`, `List<Constraint> DesignConstraints`, `List<OuterConstraint> DesignOuterConstraints` slično kao i u klasi `OptimizationSubProblem`.



Sl. 7.11. Dijagram klasa dekompozicijskog modela

Osim optimizacijskih komponenti koje su preuzete iz izvornog AllInOneOSP-a, dekompozicijski sustav sadržava liste s još dva tipa podataka: List<Parameter> Parameters i List<Response> Responses. Dodatni tipovi podataka su potrebni za propisivanje spregnutosti dvaju dekompozicijskih sustav u slučaju da ti podaci nisu ujedno i komponente optimizacijskog problema. Pri tome tip Parameter definira ulazni podatak nekom dekompozicijskom sustav, dok tip Response definira izlazni podatak nekog dekompozicijskog sustava. Kao i u slučaju optimizacijskih komponenti, ta dva tipa podataka su za određenu implementaciju koncepta sučelja IAnalisysModel vezani klasom DeModelConnection.

S obzirom da dekompozicijski sustavi nisu nezavisni potrebno ih je na određeni način spregnuti što je i izvedeno pomoću klasa DecomSysConsistency i ConsistencyPair. DecomSysConsistency omogućuje propisivanje spregnutosti između dva dekomponirana sustava (FirstDecomSys i SecondDecomSys). Veza odnosno konzistentnost para podataka tih dvaju sustava definirana je klasom ConsistencyPair. Svaki par konzistentnosti sadržava po jedan podatak iz svakog od dekompozicijskih sustava (ConsistencyData First i ConsistencyData Second). Definiranjem određenog para konzistentnosti specificira se da ta dva podatka na kraju optimizacijskog procesa moraju imati istu vrijednost (teoretske osnove definirane u 5.1 i 5.2). Pri tome treba naglasiti da se dosad spomenutim klasama ne određuje kako će se taj podatak interpretirati tijekom koordinirane optimizacije sustava, već se samo definira koje veze postoje. S obzirom da koordinacijske metode na različite načine tretiraju spregove tj. konzistentnosti dvaju sustava, taj dio ponašanja je definiran u bloku gdje se opisuju klase koordinacijskog modela. Time je omogućena jednostavnija zamjena koordinacijskog modela u slučaju da prvo-odabrani koordinacijski model ne daje zadovoljavajuće rezultate. Isto tako je jednostavnije definiranje više koordinacijskih modela za potrebe njihove usporedbe u znanstvenoistraživačkom radu.

Svaki DecomSystem ima definiran DecomSysConsistency ParentConsistency kojim je definirana konzistentnost podsustava s njegovim nadređenim sustavom odnosno roditeljem. Dekompozicijski sustavi koji su na vrhu hijerarhije odnosno čiji je roditelj instanca klase DecomModel imaju taj podatak prazan odnosno null. Za razliku od veze s roditeljem, veza nekog dekompozicijskog podsustava s drugim podsustavom se definira samo u slučaju potrebe. Razlog za takvu izvedbu je što u hijerarhijskim sustavima uvijek postoje određene veze između roditelja i njegove djece (podsustava), dok veza između dvoje djece (podsustavi) često ne postoji odnosno ne definira se eksplicitno zbog slabe jakosti te veze. Kao što je to i prirodno, veza dva djeteta se zapravo nalazi na njihovom roditelju, a s obzirom da roditelj mogu biti i

klasa i ta veza je zapravo propisana klasom `DecomSystemBase` kroz `List<DecomSysConsistency> InterChildConsistency`. Pri tome je u slučaju nepostojanja veza između djece (podsustava) ta lista prazna.

Kao što je vidljivo na Sl. 7.11 klasama `DecomSystemBase` i `DecomSystem` propisano nekoliko metoda za dohvaćanje podataka o konzistentnosti dekompozicijskog sustava koje uglavnom služe koordinacijskom modelu.

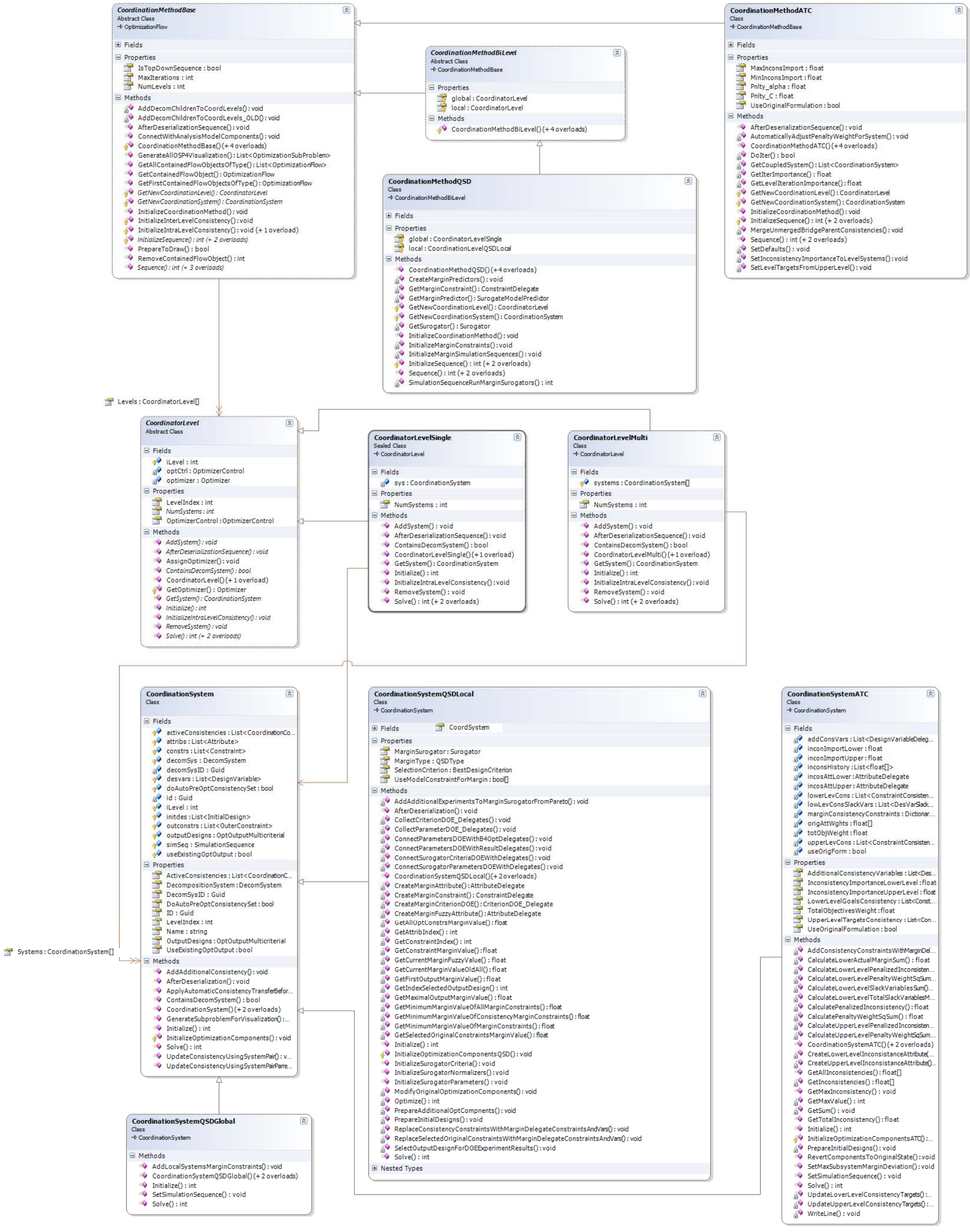
Jedna od bitnih karakteristika dekompozicijskog modela definiranog na ovaj način jest omogućeno definiranje dekompozicijskih sustava koji ne moraju odgovarati sustavima u modelu analize, odnosno dekompozicijski sustav može se sastojati od više sustava modela analize. Pod određenim uvjetima moguće je čak i da dva ili više dekompozicijskih sustava budu vezani za jedan sustav analize ukoliko je moguće pomoću kontrolnih parametara simulacije tog sustava analize definirati nezavisno izvođenje proračuna tih dvaju (ili) više apstraktno definiranih dekompozicijskih podsustava. Time je osigurano da i modeli analize koji nisu najučinkovitije pripremljeni za rad unutar okruženja za potporu pri projektiranju budu definirani kao složeni sustavi čime im je omogućeno efikasnije rješavanje dekompozicijsko koordinacijskim metodama. Ipak, uredno definirani model analize omogućuje lakše, a u nekim slučajevima i automatsko definiranje dekompozicijskog modela ili njegovog većeg dijela.

7.2.5.2. Klase za definiranje koordinacijskog modela

U prethodnom potpoglavlju dan je opis arhitekture dekompozicijskog modela kojim su definirane optimizacijske komponente te konzistentnosti između dekompozicijskih sustava/podsustava. Kako bi se taj model riješio potreban je koordinacijski model.

Arhitektura koordinacijskog modela je kreirana tako da omogući fleksibilno dodavanje bilo koje koordinacijske metode s tim da se ponašanje koje je zajedničko određenim skupinama metoda definira u baznim, odnosno apstraktnim klasama. Pri tome je teoretska podloga za razvijanje objektnog koordinacijskog modela dana u potpoglavlju 5.3. S obzirom da koordinacijski model zapravo rješava dekompozicijski model sam objektni model je uvelike povezan s prethodno opisanim dekompozicijskim modelom.

Klase za definiranje koordinacijskog modela prikazane su na Sl. 7.12. Bazna klasa za sve koordinacijske metode je `CoordinationMethodBase`. Kao što je bilo vidljivo na Sl. 7.8 ta klasa nasljeđuje klasu `OptimizationFlow` čime je i ona jedna od klasa koja rješava dio projektnog problema i to upravo onaj dio problema koji je definiran dekompozicijskim modelom, odnosno to je bazna klasa za metode koje rješavaju instancu `ComplexSystemOptimizationModel`-a.



Sl. 7.12. Dijagram klasa koordinacijskog modela

Važan dio za razumijevanje optimizacije složenih, a naročito hijerarhijskih sustava pa tako i koordinacijskih metoda je pojam koordinacijske razine koji je u arhitekturu koordinacijskog modela uvedene baznom klasom `CoordinatorLevel`. Tako svaka koordinacijska metoda sadržava jednu ili više razina definiranih svojstvom `CoordinatorLevel[] Levels`. Svaka od koordinacijskih razina sadrža jedan ili više koordinacijskih sustava `CoordinationSystem`. S obzirom da se koordinacijske razine s jednim i sa više od jednog sustava razlikuju u ponašanju, definirane su dvije izvedene klase `CoordinationLevelSingle` i `CoordinationLevelMulti`.

Klasa `CoordinationSystem` je u određenoj mjeri proširenje klase `DecomSystem` s konkretnim ponašanjem baziranim na temelju optimizacijskih komponenti sadržanog `DecompositionSystem`-a te definiranih konzistentnosti. Osnovna zadaća klase `CoordinationSystem`, odnosno klasa koje ju nasljeđuju, jest priprema optimizacijskog potproblema te izlaganje dobivenog rješenja koordinacijskoj metodi. Za pripremu modificiranog optimizacijskog problema potrebno je na odgovarajući način interpretirati zadane veze konzistentnosti. Tomu služi klasa `CoordinationConsistency` koja svojim metodama omogućuje kreiranje dodatnih optimizacijskih komponenti, s tim da koordinacijska metoda odlučuje koje komponente će biti generirane. S obzirom da pojedine optimizacijske komponente zahtijevaju i dodatne ulazne podatke (npr. definiranje gornje i donje granice varijabli) ti podaci se zadaju kroz ovu klasu. Pri tome `CoordinationConsistency` sadržava referencu na `ConsistencyData Data` podatak što je zapravo pripadajući podatak iz para konzistentnosti dekomponiranih sustava te je definirana dodatna optimizacijska komponenta vezana uz njega. Drugi podatak iz para konzistentnosti se koristi samo kako bi se saznala vrijednost s kojom je potrebno održavati konzistentnost i pristupa mu se pomoću svojstva `float Target` ukoliko je to potrebno, npr. ako koordinacijska metoda iz instance `CoordinationConsistency` klase generira ograničenje konzistentnosti (najčešće ograničenje jednakosti), onda `Target` predstavlja vrijednost koju `Data` treba imati na kraju optimizacije tog potproblema. Drugim riječima instanca klase `CoordinationConsistency` nema mogućnost modificiranja vrijednosti drugog podatka iz definiranog para konzistentnosti, već samo onog koji se nalazi u promatranom dekompozicijskom sustavu, ali omogućuje i dobivanje informacije o tome koliko su u trenutnom stanju podaci konzistentni odnosno nekonzistentni.

Za slučajeve kad klasa `CoordinationSystem` ne osigurava dovoljno specifično ponašanje koje traži pojedina koordinacijska metoda, istu je moguće naslijediti i proširiti njeno ponašanje odgovarajućim svojstvima i metodama. Tako se vidi da je implementirana koordinacijska

metoda kvazi-odvojiva dekompozicija (QSD, vidi potpoglavlje 5.4.4) zahtijevala dosta dodatnog ponašanja, ponajprije na lokalnom nivou zbog potrebe za generiranjem surogat modela optimalnih vrijednosti svakog koordinacijskog sustava.

Kako bi se osiguralo da svaka koordinacijska metoda instancira odgovarajuće instance `CoordinatorLevel` i `CoordinationSystem` klasa, `CoordinationMethodBase` klasa propisuje apstraktne metode koje generiraju instance tih klasa:

```
CoordinatorLevel GetNewCoordinationLevel(int iCoordLevel,int numSystems);
CoordinationSystem GetNewCoordinationSystem(DecomSystem decSys, CoordinatorLevel lev);
```

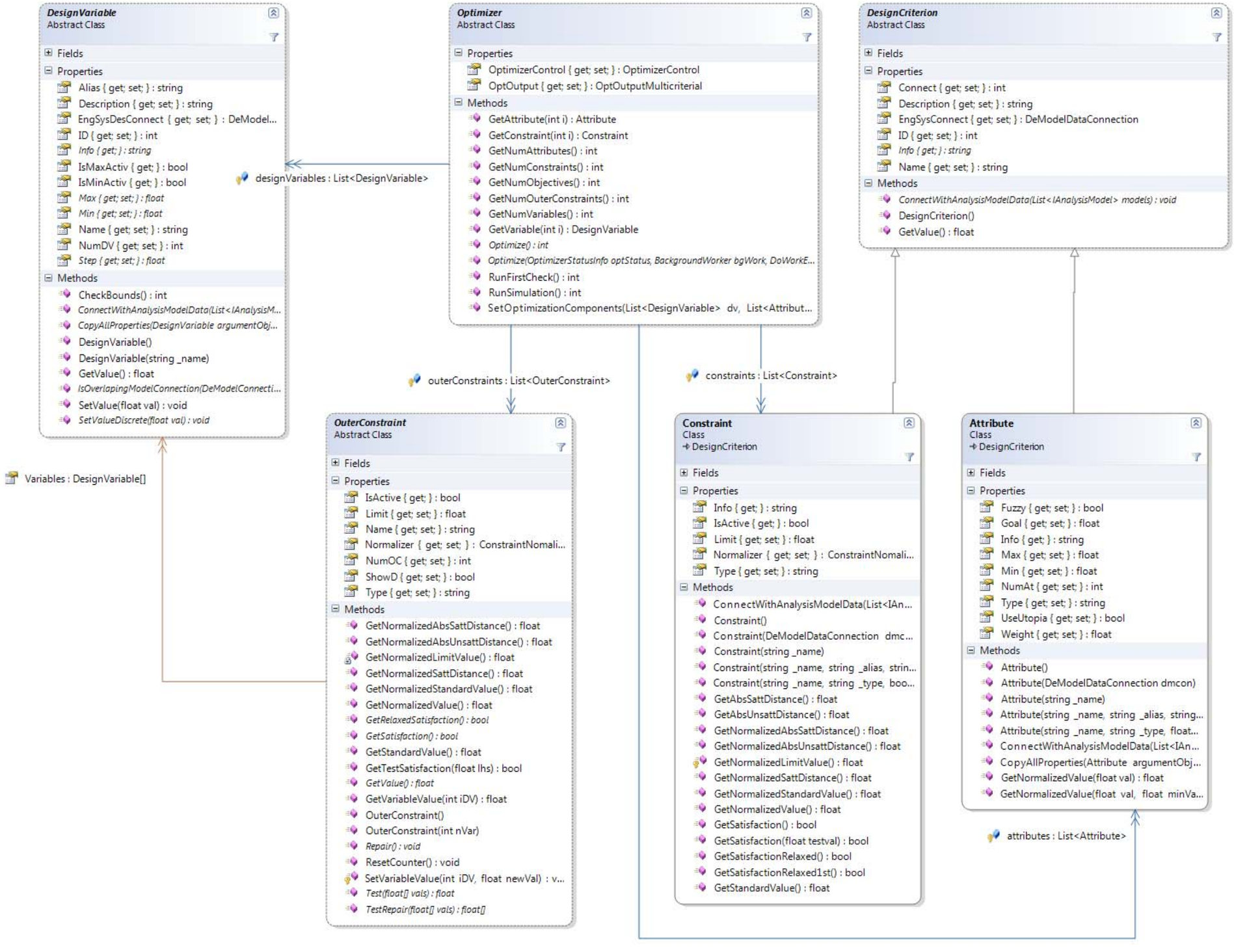
Time je omogućeno da izvedene klase instanciraju klase s potrebnim posebnostima, a da bazna klasa odrađuje zajednički dio ponašanja s baznim klasama `CoordinationMethodBase`, `CoordinatorLevel` i `CoordinationSystem`.

Klasama `CoordinationMethodATC` i `CoordinationSystemATC` implementirana ATC metoda za koordinaciju dekomponiranog problema (potpoglavlje 5.4.6) koja je odabrana prvenstveno zbog svoje općenitosti i mogućnosti dekompozicije na neograničeni broj nivoa. Kao što je i navedeno u potpoglavlju 5.4.6 metodi je, za razliku od nekih drugih koordinacijskih metoda baziranih na relaksaciji funkcije cilja, matematički, a ne samo eksperimentalno, dokazana konvergencija. U odnosu na metode bazirane na održavanju konzistentnosti ograničenjima jednakosti, ova metoda nudi jednostavniju integraciju u okruženje za potporu pri projektiranju. Naravno, kao i sve metode koje konzistentnost osiguravaju relaksacijom funkcije cilja, potreban je veći broj iteracija nego kod metoda koje konzistentnost među podsustavima održavaju ograničenjima jednakosti.

7.2.6. Klase za definiranje komponenti optimizacijskog problema

Klase koje su potrebne za identifikaciju komponenti optimizacijskog problema prikazane su na Sl. 7.13. Na slici su reference optimizacijskih komponente sadržane u apstraktnoj klasom `Optimizer` koja propisuje podatke na osnovu kojih konkretna izvedenica klase `Optimizer` odnosno konkretna optimizacijska metoda rješava optimizacijski problem. Pri tome je potrebno naglasiti da je optimizacijski problem kojeg rješava klasa `Optimizer` definiran i sadržan klasom `OptimizationSubProblem` ili u slučaju optimizacije složenih sustava klasom `CoordinationSystem` kao što je već ranije opisano.

Liste koje sadržavaju komponente optimizacijskog problema su: `List<DesignVariable> designVariables, List<Attribute> attributes, List<Constraint> constraints, List<OuterConstraint> outerConstraints.`



Sl. 7.13. Dijagram klasa za definiranje komponenti optimizacijskog problema

Apstraktna klasa `DesignVariable` definira osnovne metode i objekte za sve izvedene klase, tj. za sve tipove projektnih varijabli definiranih unutar OCTOPUS DESIGNER-a. Sučelje koje definira ova klasa omogućuje implementiranim optimizacijskim metodama da prikupe sve informacije o projektnim varijablama koje im trebaju za rad bez obzira kojeg je točno izvedenog tipa pojedina varijabla.

Klase `Constraint` i `Attribute` izvedeni su iz klase `DesignCriterion` unutar koje se definiraju osnovni parametri za definiciju atributa i ograničenja. `DesignCriterion` klasa sadržava `DeModelDataConnection` `EngSysConnect` preko je vezan za model analize. Klasom `Attribute` definirana su svojstva projektnih atributa, dok su klasom `Constraint` definirana svojstva projektnih ograničenja.

`OuterConstraint` je apstraktna klasa za sve tipove vanjskih ograničenja zadanih od strane projektanta te definira zajedničke metode i svojstva koje moraju implementirati sve izvedene klase. Za razliku od klase `Constraint` u ovoj klasi su ograničenja definirana eksplicitno u odnosu na projektne varijable te za provjeru njihove zadovoljenosti ne treba pozivati proračun određenog sustava modela analize već je dovoljno izračunati eksplicitno zadani izraz. Sam naziv „vanjska ograničenja“ proizlazi upravo iz činjenice da su definirana izvan modela analize.

Apstraktna klasa `Optimizer` definira sučelje koje svaka izvedena klasa (optimizacijska metoda) mora zadovoljiti, ali isto tako pruža svim izvedenim algoritmima svu podršku koja ima je potrebna za optimizaciju. Klasa `Optimizer` tako daje direktni pristup svim komponentama projektnog problema, omogućava promjenu vrijednosti deskriptora modela analize na temelju promjene projektnih varijabli, pozivanje proračuna modela analize te preuzimanje izračunatih vrijednosti atributa i ograničenja.

7.2.7. Poslovi bazirani na planiranju pokusa (DOE) u okruženju za potporu projektiranju

Poslovi bazirani na planiranju pokusa okarakterizirani su korištenjem metode planiranja pokusa (DOE). Kod ovog tipa posla je nakon odabira određenog modela pokusa poznato koje numeričke simulacije je potrebno obaviti.

Apstraktna klasa `DOE` definira osnovno ponašanje koje svaki plan pokusa mora zadovoljiti. Pri tome je osnovna metoda za to `InitializeExperimentsPlan()` kojom implementirani plan inicijalizira interne podatke tj. nivoe odabranog plana pokusa. Pozivom metode `Next()` odabire se idući pokus i postavljaju vrijednosti parametara definiranih klasom `ParameterDOE`. Pozivanjem instance klase `SimulationSequence` izvršava se određena simulacija te spremaju ostvareni rezultati za definirane kriterije (`CriterionDOE`).

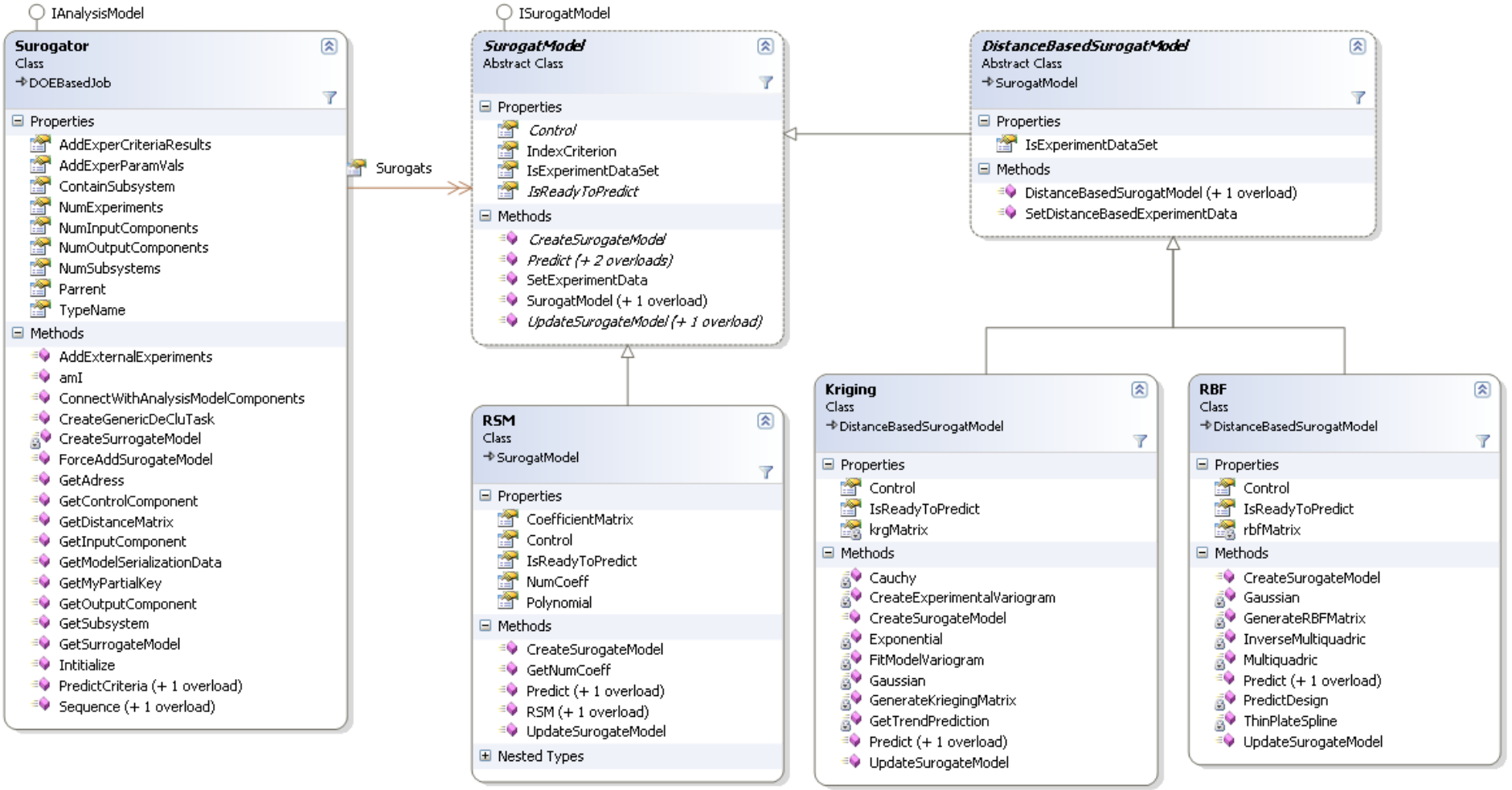
Sl. 7.14. Dijagram klasa za definiranje poslova baziranih na planiranju pokusa

Prvi implementirani ovakav tip posla jest `RobustnessJob` koji je zasnovan na komponenti za računanje robusnosti IC4RD (*IMPROVE Component For Robust Design*), razvijenoj u okviru EU FP6 projekta IMPROVE [5]. Ovim tipom posla omogućeno je projektiranje korištenjem informacija o robusnosti projektiranog problema i to izračunom omjera signala i šuma (*Signal/Noise –S/N*) ili informacije sustava. Primjena ovih mjera robusnosti kroz OCTOPUS Designer može se vidjeti u [36] za projektiranje tankostjene konstrukcije, dok je u [92] robusnost korištena kao jedan od kriterija pri konceptualnom projektiranju trgovačkih brodova.

7.2.8. Surogat-modeler (Surogator)

Surogat modeler (klasa `Surogator`) je `DOEJob` tip posla čiji je zadatak generiranje surogat modela odabranih kriterija te predviđanje vrijednosti tih kriterija na zadanim lokacijama (vidi Sl. 7.15). S obzirom je osnovna ideja surogat modeliranja, kao što je i prikazano u poglavlju 4, zamjena proračuna skupih modula analize očigledno je da ovaj tip posla mora imati slične mogućnosti i sučelje prema modulima sinteze kao i moduli analize. U klasi `Surogator` to je izvedeno implementiranjem koncepta sučelja `IAnalysisModel`. Na taj način je i formalizirano ponašanje surogat modelera kao modela analize. U Tbl. 7.2 nalazi se popis svih kontrolnih kodova surogator modelera s opisom funkcionalnosti.

Tipovi surogat modela koji su do sad uključeni u OCTOPUS Designer su `Kriging` (potpoglavlje 4.3.2) i `RBF` (potpoglavlje 4.3.3), pri čemu su ti moduli razvijeni u okviru diplomskog rada [88] te Polinomski surogat modeli (potpoglavlje 4.3.1) čija implementacija je detaljno opisana u idućem potpoglavlju.



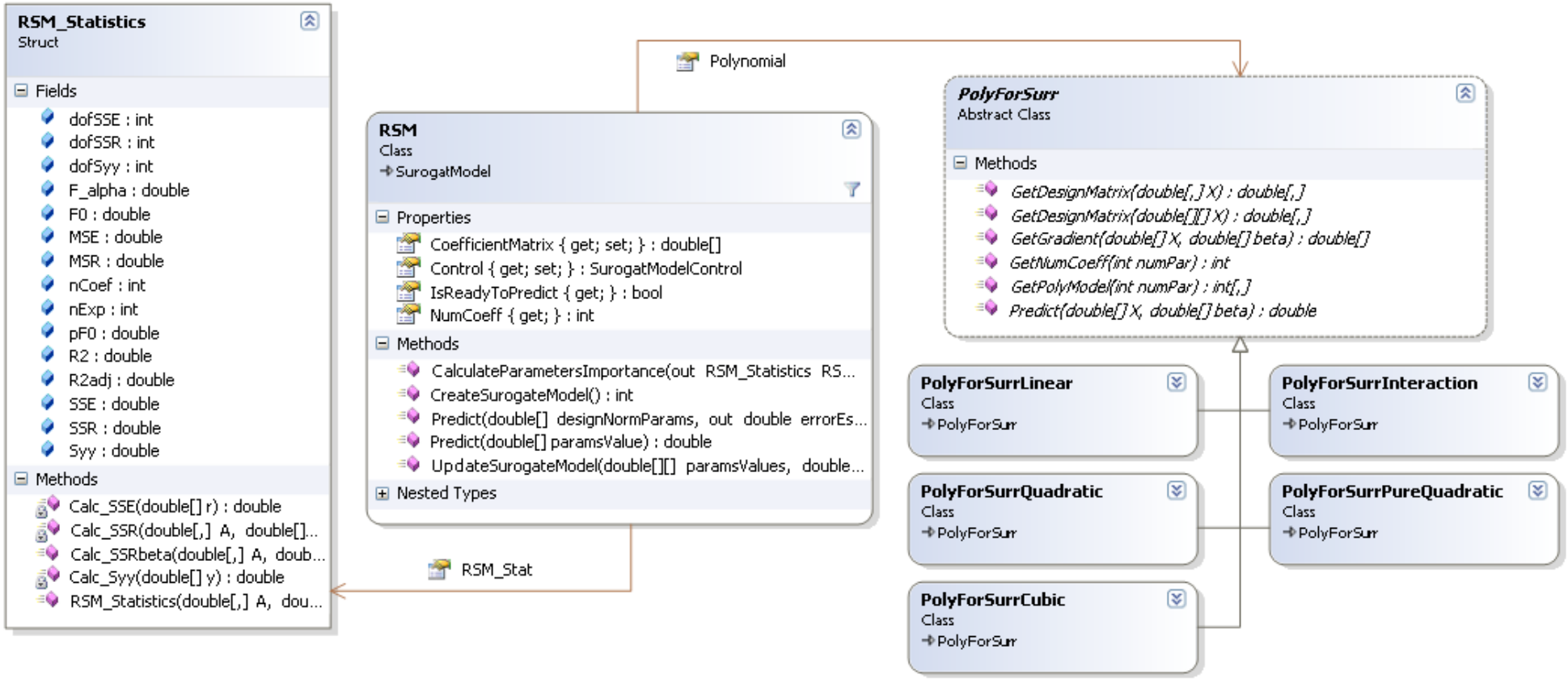
Sl. 7.15. Dijagram klasa za definiranje surogat modelera

Tbl. 7.2 Opis kontrolnih kodova surogatora (kad se koristi kao model analize)

Kod	Značenje	Komentar
0	• Izračun (predikcija) aproksimacije kriterija trenutnim surogat modelima za zadane vrijednosti kontrolnih parametara surogat modela	
1	• Izračun stvarne vrijednosti kriterija, koristeći simulacijsku sekvencu surogatora, za zadane vrijednosti kontrolnih parametara surogat modela. • Spremanje dobivenih rezultata u listu dodatnih eksperimenata	
2	• Osvježavanje tj. generiranje novih surogat modela na osnovu osnovnih i dodatno izvršenih pokusa. • Izračun (predikcija) aproksimacije kriterija trenutnim surogat modelima za zadane vrijednosti kontrolnih parametara surogat modela	5+0
3	• Izračun stvarne vrijednosti kriterija, koristeći simulacijsku sekvencu surogatora, za zadane vrijednosti kontrolnih parametara surogat modela. • Spremanje dobivenih rezultata u listu dodatnih eksperimenata • Osvježavanje tj. generiranje novih surogat modela na osnovu osnovnih i dodatno izvršenih pokusa.	1+5
4	• Izračun stvarne vrijednosti kriterija, koristeći simulacijsku sekvencu surogatora, za zadane vrijednosti kontrolnih parametara surogat modela. • Spremanje dobivenih rezultata u listu dodatnih eksperimenata • Osvježavanje tj. generiranje novih surogat modela na osnovu osnovnih i dodatno izvršenih pokusa. • Izračun (predikcija) aproksimacije kriterija trenutnim surogat modelima za zadane vrijednosti kontrolnih parametara surogat modela	3+0
5	• Osvježavanje tj. generiranje novih surogat modela na osnovu osnovnih i dodatno izvršenih pokusa.	
11	• Izračun (predikcija) aproksimacije kriterija trenutnim surogat modelima za trenutne vrijednosti modela analize na koji su povezani parametri surogatora	
12	• Izračun aproksimacije kriterija surogat modelima za trenutne vrijednosti modela analize na koji su povezani parametri surogatora • Preslikavanje aproksimiranih vrijednosti surogat modela na trenutne vrijednosti povezanih deskriptora modela analize (moguće samo za one kriterije koji su deskriptori modela analize)	11+

7.2.8.1. Općeniti polinomski surogat model

Kao što je i rečeno u potpoglavlju 4.3.1 polinomski surogat model je jedan od najčešće korištenih surogat modela. Bez obzira na odabrani polinom, postupak generiranja surogat modela se uvijek svodi na određivanje nepoznatih koeficijenata najmanjih kvadrata odstupanja. Na Sl. 7.16 prikazan je dijagram klasa korištenih za definiranje općenitog polinomskog surogat modela u OCTOPUS Designer-u. Pri tome se vidi da klasa `RSM` kroz svojstvo `Polynomial` sadržava instancu neke od klasa koje nasljeđuju apstraktnu klasu `PolyForSurr`. Osnovni zadatak nasljednika te klase je generiranje regresijske matrica $\mathbf{B}_{1\text{-}n}$: (4.15) metodom `GetDesignMatrix()`, te predviđanje vrijednosti aproksimiranog kriterija metodom `Predict()`.



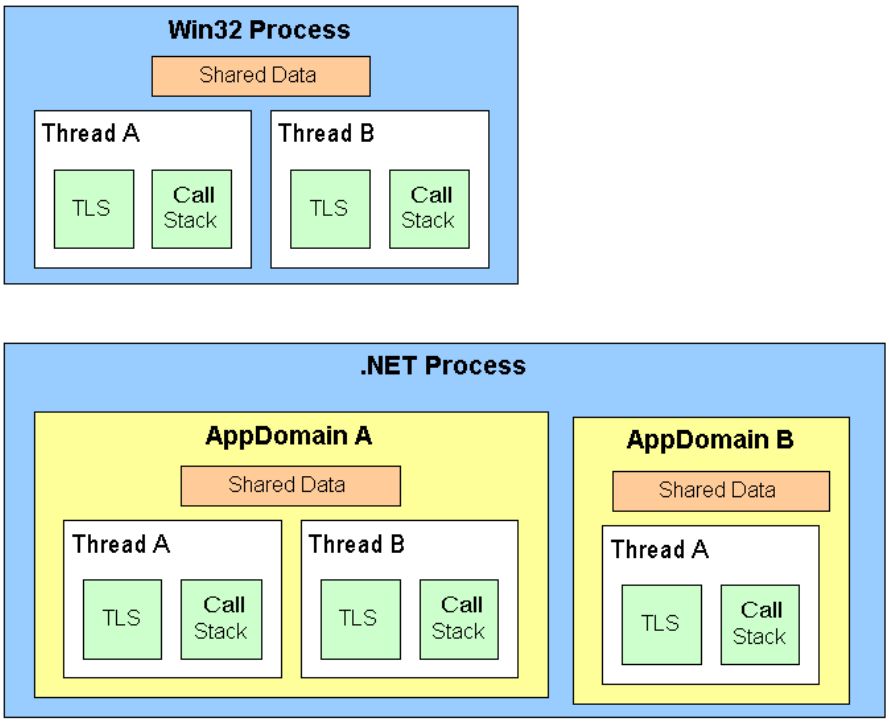
Sl. 7.16. Dijagram klasa za definiranje općenitog polinomskog surogat modela

Na ovaj način je omogućena vrlo jednostavna implementacija novog tipa polinoma, jer je dovoljno dodati novu klasu koja nasljeđuje apstraktnu klasu `PolyForSurr` i kroz propisane metode ostvariti svoje ponašanje. Na Sl. 7.16 vidi se tipovi polinoma koji su trenutno implementirani u OCTOPUS Designer, no ne postoji nikakva zapreka da se ne implementiraju drugi tipovi polinoma (npr. ortogonalni Legendrovi polinomi). Strukturom `RSM_Statistics` omogućena je dijagnostika generiranog polinoma, ali i analiza signifikantnosti parametara surogat modela (vidi potpoglavlje 4.3.5.2)

7.2.9. Asinkroni rješavač višestruko ponavljajućih zadataka – DeCluster

S obzirom na karakteristike arhitekture današnjih računala, kod kojih se ukupna proračunska snaga više bitno ne povećava povišenjem radnog takta procesora, već povećanjem broja radnih jezgri, jedna od bitnih karakteristika jest mogućnost paralelnog izvršavanja određenih sekvenci naredbi. U okviru okruženja za potporu pri projektiranju OCTOPUS Designer, to je ostvareno na trima razinama: 1) generiranjem nove dretve (*thread*), 2) generiranjem nove aplikacijske domene i 3) generiranjem .NET Remoting servera i klijenta. Prvim načinom omogućava se paralelno izvršavanje jednostavnih zadaća koje ne koriste globalne statičke podatke, dok druga dva načina omogućuju rad takovim aplikacijama.

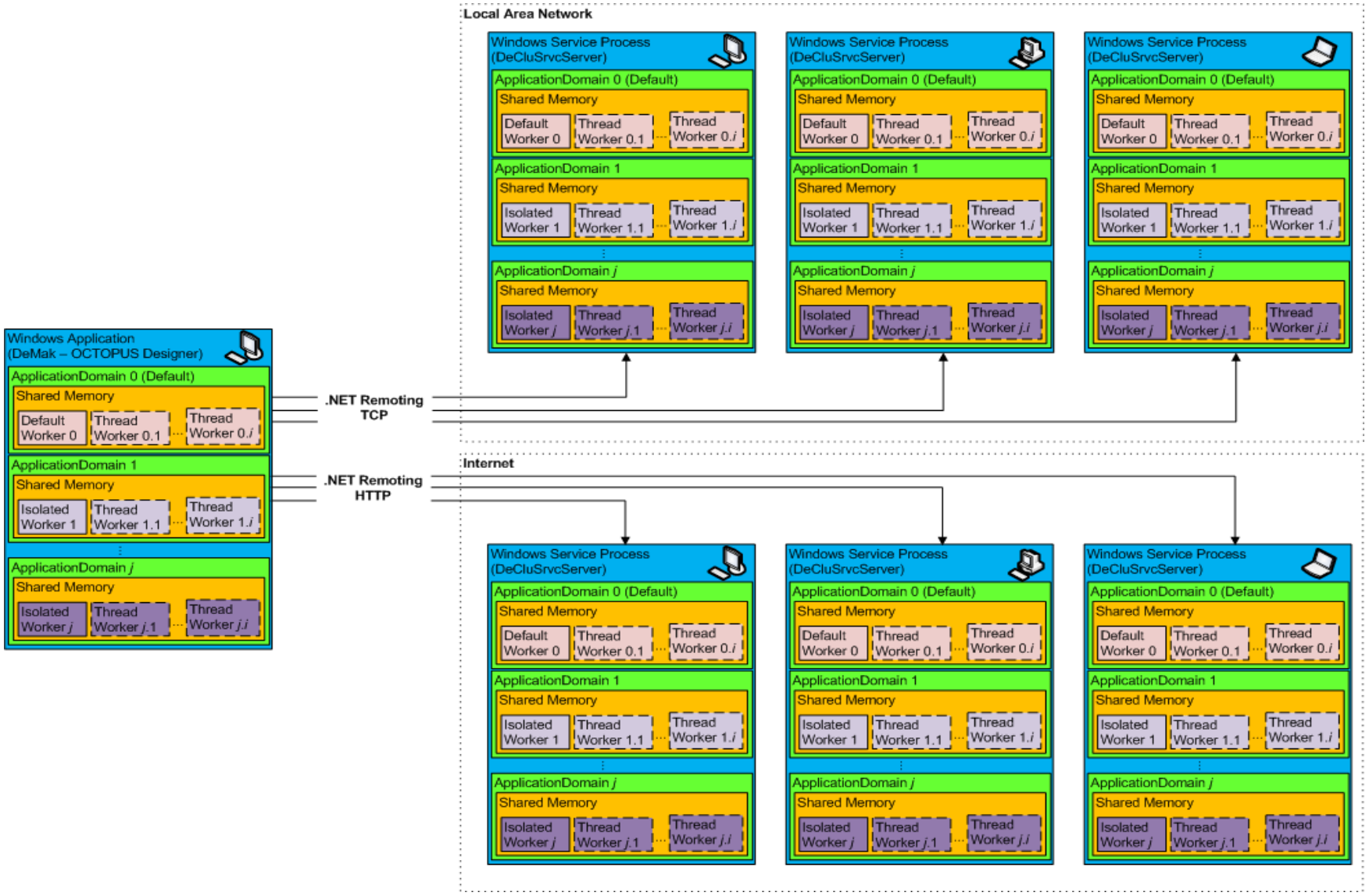
Kod standardnih Windows Win32 procesa (Sl. 7.17 a)) memorija koju dijele dretve je dijeljena, te je za slučaj kad određena zadaća koja koristi globalne statičke podatke zahtjeva neovisno izvršavanje, neophodno je te zadaće izolirati u nezavisne procese te omogućiti komunikaciju između procesa. U .NET procesu uveden je još i novi entitet koji se naziva aplikacijska domena (*application domain*) a koji omogućuje i odvajanje memorije u okviru istog procesa (Sl. 7.17 a)). Generiranjem nove aplikacijske domene omogućuje se rad s modulima analize koji koriste statičke globalne varijable u okviru istog procesa jer se time odvaja memorija instanci modela slično kao kod generiranja novog procesa.



Sl. 7.17. Organizacija memorije standardne Win32 aplikacije i .NET aplikacije [93]

Korištenjem .NET Remoting-a moguće je ostvariti komunikaciju više procesa na jednom ili više računala te osim izolacije memorije na jednom računalu omogućuje se i izvršavanje paralelnih zadaća na računalima koja su povezana preko lokalne mreže korištenjem TCP protokola. Korištenjem HTTP protokola moguće je povezati i računala koja se nalaze izvan lokalne mreže odnosno računala spojena na Internet (Sl. 7.18). Pri izradi ovog potprograma kao osnovna literatura je korišten [94] u kome su objašnjeni razni detalji vezani za rad s .NET Remoting bibliotekom.

U nastavku će biti dani dijagrami klasa `TaskManager` i `TaskWorker` klase preko kojih je moguće objasniti implementaciju DeCluster-a.

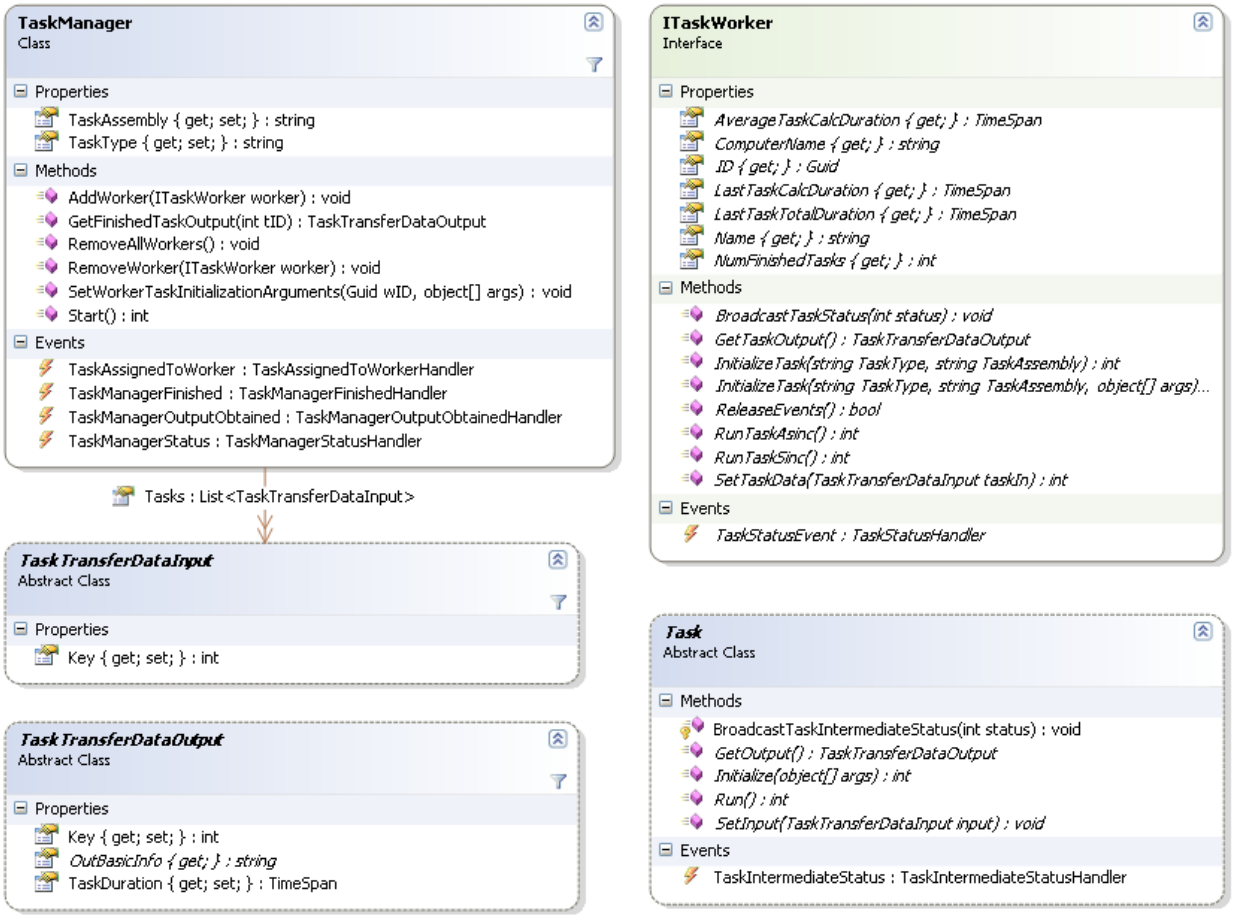


Sl. 7.18. Arhitektura DeCluster sustava

7.2.9.1. Dijagram klase za upravljanje rješavanja skupine ponavljajućih zadataka (TaskManager)

Osnovna zadaća *DeCluster* potprograma jest rješavanje višestruko ponavljajućih zadataka istog tipa, no s različitim ulaznim podacima, u paralelnoj, asinkronoj sekvenci. Takav tip zadatka jest npr. izvođenje računalnih simulacija definiranih planom pokusa za generiranje surogat modela. Asinkrono izvođenje zadataka, u odnosu na sinkrono, u ovakvom tipu klastera koji se sastoji od računala/servera različitih performansi, nudi bolje iskorištavanje računalnih kapaciteta jer sekvenca izvršavanja računala pojedinog servera u klasteru ne ovisi o ostalim serverima.

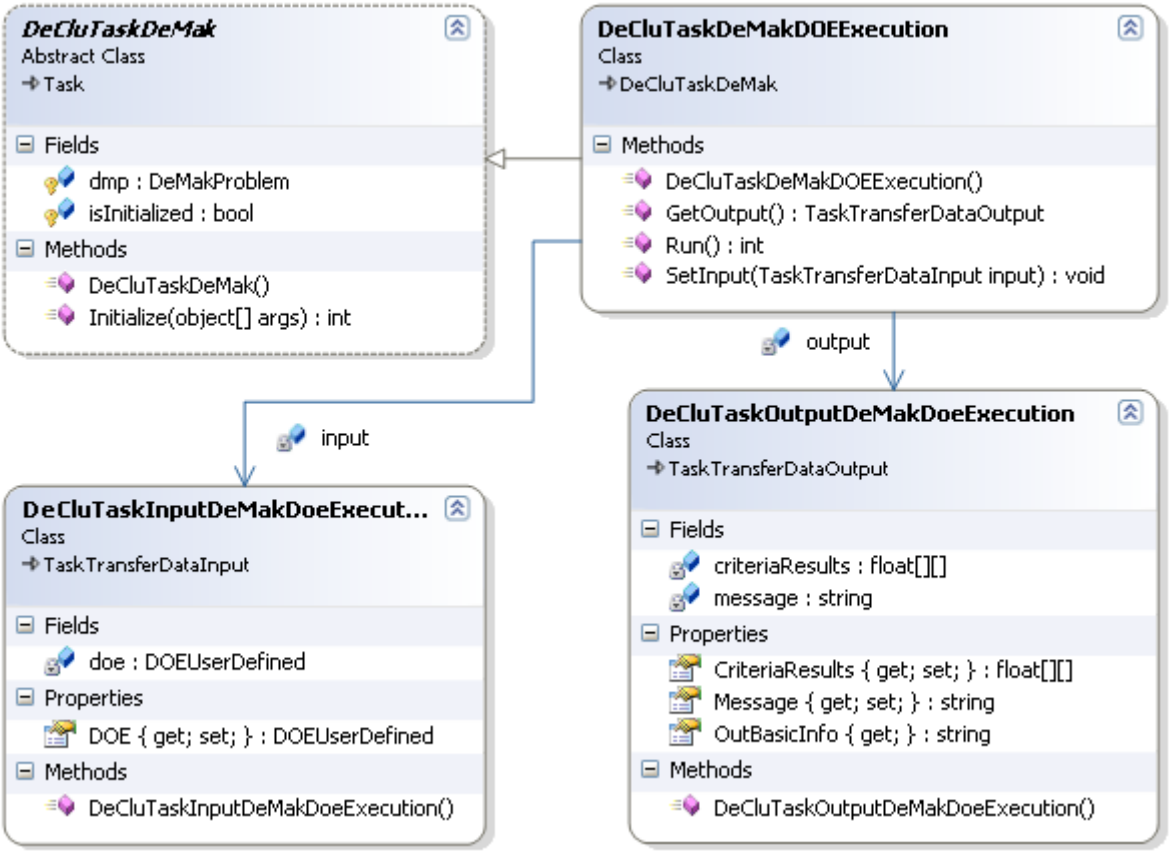
U DeCluster sustavu instanca klase `TaskManager` zadužena je za zadavanje ponavljajućeg tipa zadatka (`Task`) svim dostupnim rješavačima tj. instancama koncepta sučelja `ITaskWorker`. (vidi Sl. 7.19) Pri tome apstraktna klasa `TaskTrasnferDataInput` definira osnovno ponašanje ulaznih podataka određenom zadatku, dok `TaskTransferDataOutput` definira osnovno ponašanje izlaznih podataka. Više o klasama koje implementiraju koncept sučelja `ITaskWorker` u potpoglavlju 7.2.9.2.



Sl. 7.19. Dijagram klase za upravljanje rješavanja skupine ponavljajućih zadataka (TaskManager)

Svakoj instanci koncepta sučelja `ITaskWorker`, koja je dostupna `TaskManager`-u, dodjeli se i inicijalizira određeni tip zadatak (`Task`) metodom `InitializeTask()`. Nakon što su svim rješavačima inicijalizirani zadaci, TaskManager redom zadaje trenutni zadatak iz repozitorija `TaskTransferDataInput` i asinkrono ga pokreće. Nakon što rješavač obavi zadani zadatak, dojavljuje `TaskManager`-u svoj status završenosti uz pomoć *eventa* `TaskStatusEvent`. Nakon te dojave, TaskManager poziva metodu GetTaskOutput() tog rješavača i prenosi dobivena rješenja u repozitorij izvršenih zadataka. Ukoliko u repozitorij neizvršenih zadatak nije prazan, slobodnom rješavaču se dodjeljuje novi zadatak Procedura se ponavlja dok se ne riješe svi zadaci.

Na Sl. 7.20 vidljiva je implementacija osnovnih klasa za definiranje DeClu tipa zadatka za izvršavanje numeričkih simulacija definiranih pomoću plana pokusa. Tom implementacijom je omogućeno rješavanje pokusa za generiranje surogat modela strukturnih odziva za primjere u potpoglavljima 8.1 i 8.2.

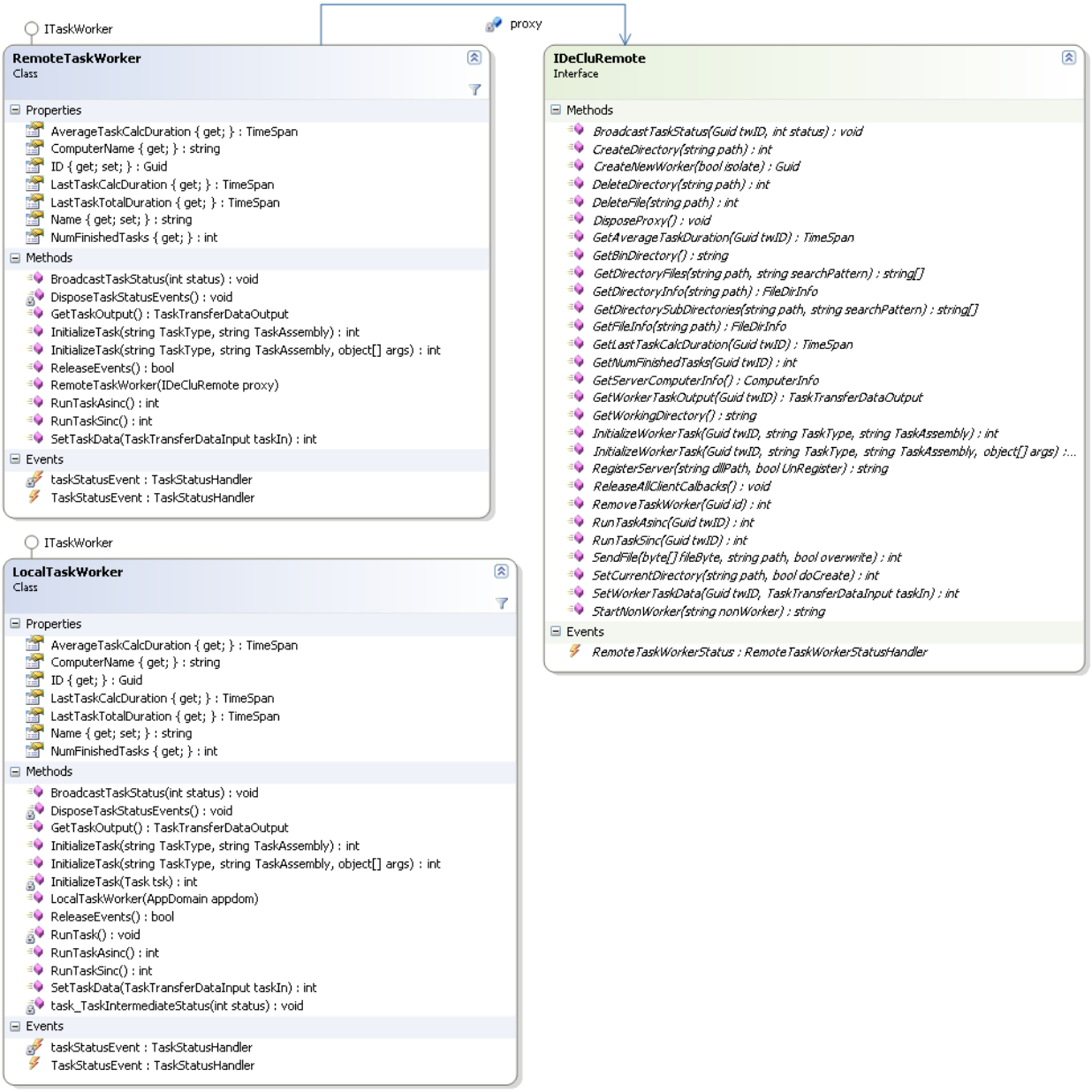


Sl. 7.20. Dijagram klasa za definiranje izvođenja DOE simulacija pomoću DeCluster sustava

7.2.9.2. Dijagram klasa rješavača pojedinog zadataka (TaskWorker)

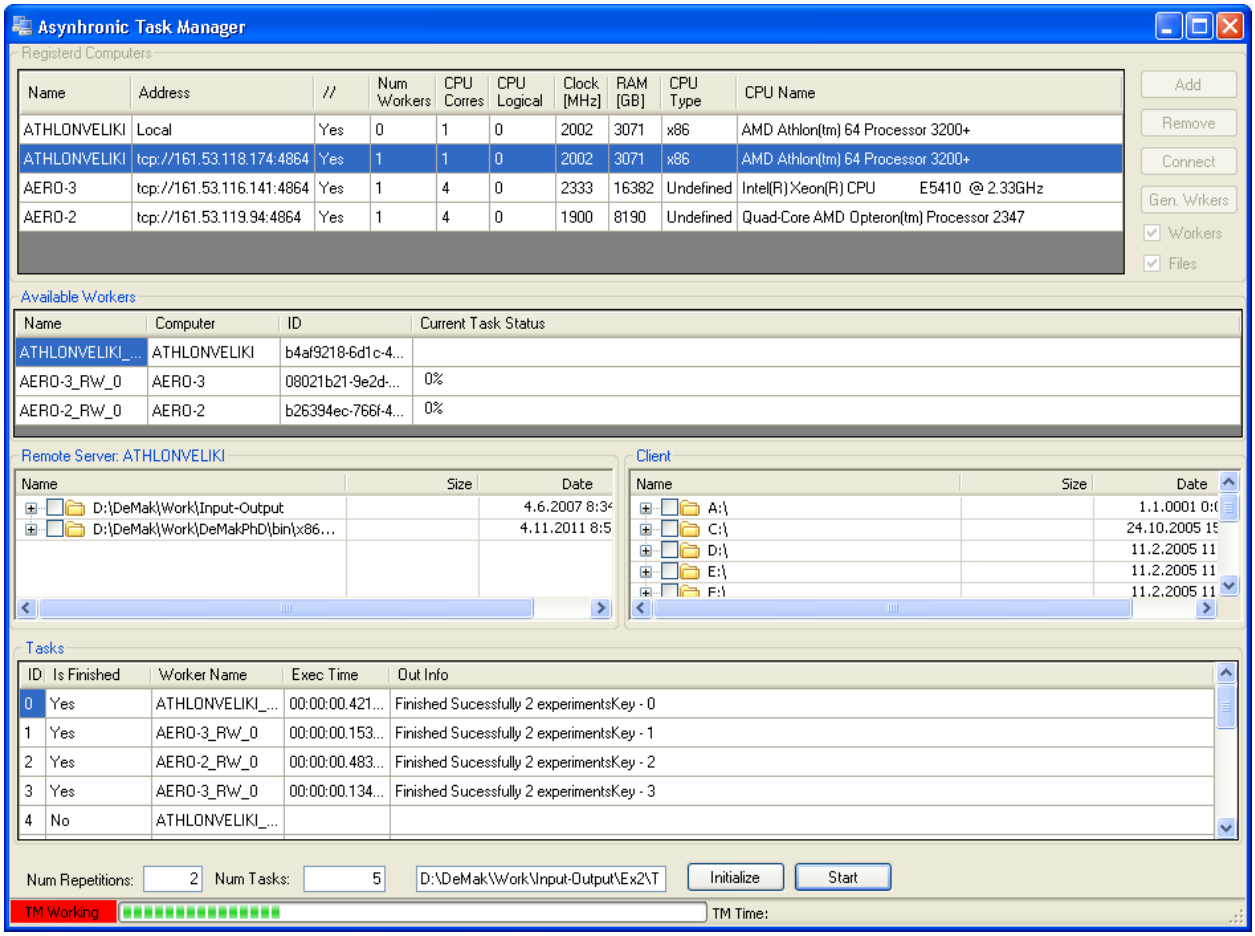
Kao što je prethodno navedeno paralelno izvršavanje sekvenci u DeClusteru je ostvareno na trima razinama: 1) generiranjem nove dretve, generiranjem nove aplikacijske domene i 3) generiranjem .NET Remoting servera i klijenta. Navedeni načini rada su izvedeni adekvatnim implementacijama koncepta sučelja `ITaskWorker` prikazanog na Sl. 7.19.

Pri tome se 1 i 2 ostvaruju kreiranjem instance `LocalTaskWorker`-a (vidi Sl. 7.21), dok kreiranjem instance `RemoteTaskWorker` - a kreira rješavač u potpuno odvojenom procesu koji može biti i na drugom računalu. Pri tome je za ostvarivanje veze preko granica procesa zadužena odgovarajuća implementacija `IDeCluRemote` koncepta sučelja.



Sl. 7.21. Dijagram klasa rješavača pojedinog zadataka (TaskWorker)

Na Sl. 7.22 nalazi se prikaz okruženje *DeCluster* sustava pri izvođenju numeričkih simulacija potrebnih za generiranje surogat modela odziva



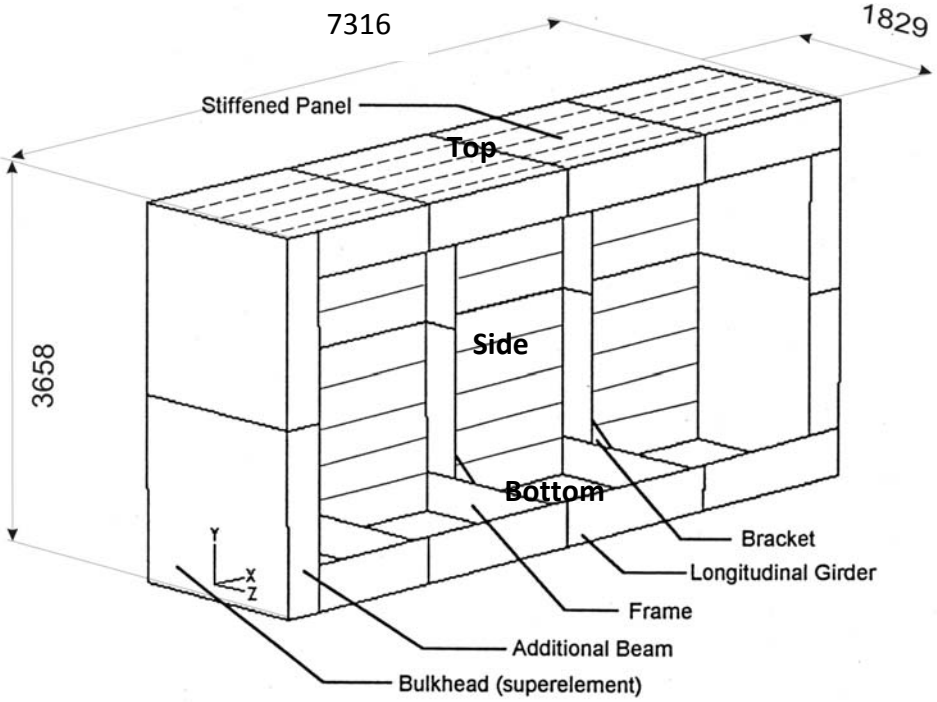
Sl. 7.22. Prikaz sučelja DeCluster sustava pri rješavanju pokusa potrebnih za generiranje surogat modela odziva

8. PRIMJENA RAZVIJENOG OKRUŽENJA NA TESTNIM PRIMJERIMA

U ovom poglavlju prikazana je primjena razvijenog programskog okruženja za projektiranje složenih sustava na primjerima s područja projektiranja složenih tankostjenih konstrukcija. Prvo će na jednostavnom primjeru prizmatičnog pontona biti prikazani razni testovi kojima je zadatak bio istražiti osnovne ideje prilagodbe surogat modeliranja i dekompozicije na tankostjenoj konstrukciji. Nakon toga slijede primjer projektiranja strukture broda za prijevoz putnika i vozila (Ropax) te primjer konceptualnog projektiranja nadzvučnog zrakoplova (SBJ – Supersonic business jet).

8.1. Projektiranje tankostjene konstrukcije jednostavnog pontona

Ovaj primjer demonstrira projektiranje jednostavne ali realne vertikalno i poprečno simetrične konstrukcije pontona kvadratičnog poprečnog presjeka. Model jednostavnog pontona izrađen je u programskom sustavu MAESTRO (MAESTRO, 2011) i zapravo je jedan od primjera koji se isporučuje zajedno s instalacijom sustava. Na Sl. 8.1 mogu se vidjeti osnovne dimenzije i građevni elementi konstrukcije. Na ovom primjeru su testirane i istraživani razni aspekti unaprjeđenja projektiranja prikazanih u poglavlju 6.



Sl. 8.1. Jednostavni ponton

8.1.1. Model za analizu tankostjene konstrukcije pontona

MAESTRO model konstrukcije pontona sastoji se od tri voja, te nekoliko dodatnih ljuskastih i grednih elemenata, a modeliran je kao polovinski simetrični model oko vertikalne osi simetrije. Maksimalni iznos zadanih tlakova je isti za sve vojeve.

Zadana vanjska opterećenja iznose:

1) tlak na svaki voj u iznosu 0.0827 MPa;
2) alternirajući (pregib, progib) vertikalni moment savijanja -2.2032e+010 Nmm

U modelu su definirane tri podstrukture koje će se optimizirati: gornji pokrov (*top*), bok (*side*), dno (*bottom*). U nastavku teksta podstrukture će biti referencirane prema njihovim nazivima na engleskom jeziku radi lakšeg praćenja dobivenih podataka na MAESTRO modelu, jer su na njemu korišteni engleski nazivi podstruktura kao i izmjera strukturnih elemenata.

8.1.2. Formulacija osnovnog optimizacijskog problema

U nastavku je dana formulacija osnovnog, jednorazinskog, optimizacijskog problema koji uključuje sve varijable i ograničenja.

Projektne varijable

Projektne varijable su izmjere osnovnih nosivih elemenata konstrukcije: oplate (*plating*), ukrepa oplate (*stiffeners*), jakih uzdužnih nosača (*girders*) te jakih poprečnih nosača (*frames*). U Tbl. 8.1 i Tbl. 8.2 dane su početne vrijednosti varijabli s propisanim gornjim i donjim granica.

Tbl. 8.1 Propisane gornje i donje granice oplate pontona

	TPL [mm]		
	P0	min	max
Top	11.0	4.0	25
Side	9.5	4.0	25
Bottom	11.0	4.0	25

Tbl. 8.2 Propisane gornje i donje granice grednih dijelova konstrukcije

		HW [mm]			TW [mm]			BF [mm]			TF [mm]		
		P0	min	max	P0	min	max	P0	min	max	P0	min	max
Top	Stiffener	152	50	250	6.5	5.0	15.0	101	50	250	6.5	6	26
	Girder	457	152	760	9.5	5.0	25.0	228	228	406	19.0	6	26
	Frame	381	152	760	9.5	5.0	25.0	203	203	304	19.0	6	26
Side	Stiffener	152	50	250	6.5	5.0	15.0	101	50	250	6.5	6	26
	Frame	457	152	760	9.5	5.0	25.0	228	228	406	19.0	6	26
Bottom	Stiffener	152	50	250	6.5	5.0	15.0	101	50	250	6.5	6	26
	Girder	457	152	760	9.5	5.0	25.0	228	228	406	19.0	6	26
	Frame	381	152	760	9.5	5.0	25.0	203	203	304	19.0	6	26

Ograničenja

Zadana ograničenja su lokalni faktori podobnosti konstrukcije definirani u Tbl. 8.3 koji se provjeravaju programskim sustavom MAESTRO, a njihov opis moguće je pronaći u [95]

Tbl. 8.3 Ograničenja – kriteriji podobnosti konstrukcije pontona

No.	Description	γ
1	Panel Collapse - Stiffener Flexure	1.04
2	Panel Collapse - Combined Buckling	1.04
3	Panel Collapse - Membrane Yield	1.224
4	Panel Collapse - Stiffener Buckling	1.04
5	Panel Yield - Tension Flange	1.04
6	Panel Yield - Tension Plate	1.224
7	Panel Yield - Compression Flange	1.04
8	Panel Yield - Compression Plate	1.224
9,10	Panel Serviceability - Plate Bending	1.0
11	Panel Failure - Local Buckling	0.9
12	Girder Collapse Tripping	1.04
13	Girder Collapse Compression in Flange	1.04
14	Girder Collapse Compression in Plate	1.04
15	Girder Yield Compression in Flange	1.224
16	Girder Yield Compression in Plate	1.224
17	Girder Yield Tension in Flange	1.224
18	Girder Yield in Tension in Plate	1.224
19-21	Frame Collapse, Plastic Hinge	1.6
22-24	Frame Yield, Compression in Flange	1.224
25-27	Frame Yield, Tension in Flange	1.224
28-30	Frame Yield, Compression in Plate	1.224
31-33	Frame Yield, Tension in Plate	1.224

Projektni atributi

Atributi korišteni u ovom primjeru su masa i dvije lokalne mjere sigurnosti (Tbl. 8.4), s tim da je kao cilj u optimizacijskim problemima korištena samo masa.

Tbl. 8.4 Atributi – mjere kvalitete konstrukcije

Oznaka	**Tip**	**Komentar**
WGT	Minimizirati	Masa konstrukcije
GMean	-	$g_{<0.05}$
GMin	-	$g_{\min 5\%}$

Definicija korištenih mjera sigurnosti dana je u nastavku. Prva od mjera je $g_{<0.05}$ a računa se prema izrazu (8.1) kao suma svih faktora podobnosti, na svim evaluacijskim elementima (*patch*) podstrukture, koji imaju vrijednost manju od 0.05. Mjera daje indikaciju količine narušenosti podobnosti.

$$g_{<0.05} = \sum_{i=1}^{n_p} \sum_{j=1}^{n_c} g_{ij}\Big|_{<0.05}; \tag{8.1}$$

Druga mjera sigurnosti, $g_{\min 5\%}$ računa srednju vrijednost najgorih 5% svih faktora podobnost i na svim *patchevima* podstrukture s tim da se ne uzimaju u obzir faktori podobnosti koji imaju vrijednost > 0.15. Mjera daje indikaciju sigurnosti podstrukture, s tim da ta mjera ne prelazi vrijednost 0.15.

$$g_{\min 5\%} = \frac{\sum_{i=1}^{n_{5\%}} g_{ij}\Big|_{5\%}}{n_{5\%}}; \text{uz } g_{ij}\Big|_{5\%} \leq 0.15 \tag{8.2}$$

8.1.3. Primjena surogat modela za aproksimaciju odziva

S obzirom da je osnovna ideja unaprjeđenja procedure projektiranja tankostjenih konstrukcija bazirana na korištenju surogata odziva tankostjene konstrukcije, prvo su izrađeni testovi točnosti više različitih tipova surogat modela s ciljem utvrđivanja primjerenosti pojedinog tipa surogat modela za primjenu pri optimizaciji. U Tbl. 8.5 definirane su oznake testiranih surogata.

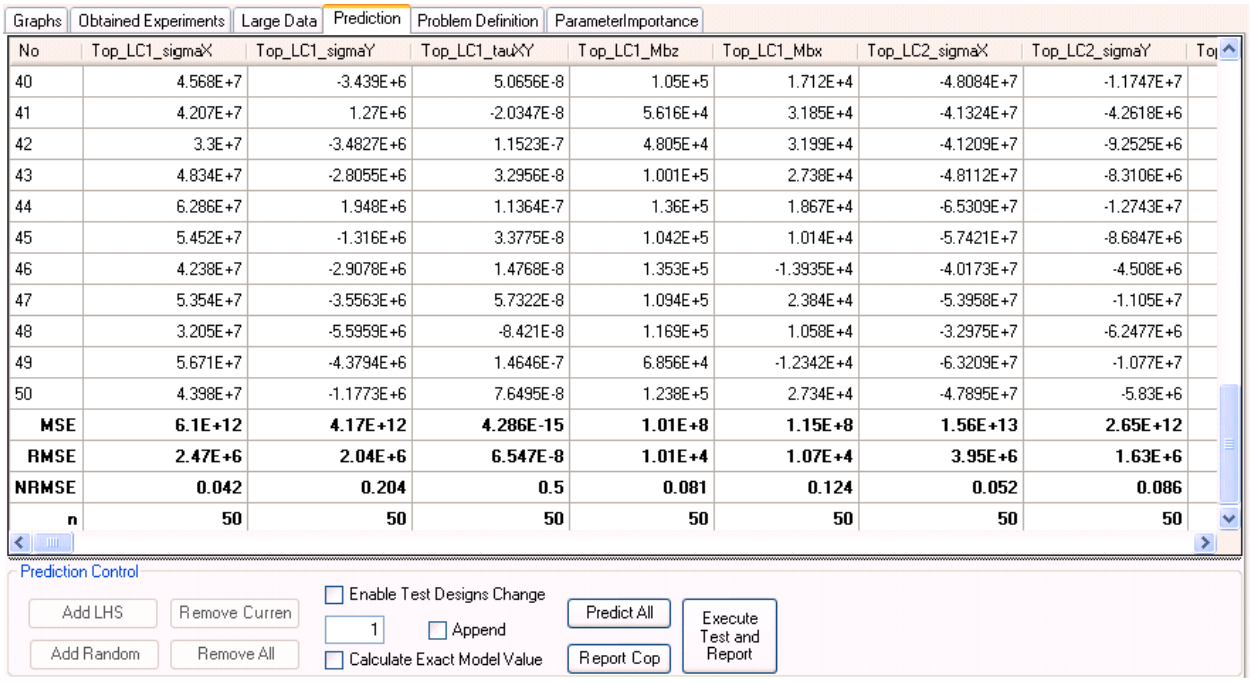
Kao što je i opisano u potpoglavlju 6.3.2 broj parametara korištenih za generiranje surogat modela reduciran je korištenjem samo jednog kontrolnog parametra za svaki gredni deskriptor konstrukcije, što znači da je ukupno definirano 11 parametara, od čega je 8 tzv. kompozitnih parametara koji definiraju više deskriptora poprečnog presjeka grednih elemenata.

Tbl. 8.5 Legenda oznaka surogat modela

Oznaka	**Opis**
P2	Puni polinomski surogat drugog reda
PIn	Polinomski surogat s interakcijskim članvima
P(2)	Polinomski surogat drugog reda bez interakcijskih članova
PLn	Linearni polinomski surogat
RSM	RBF, jednostavni (*Simple)*, multikvadratična bazna funkcija
RSI	RBF, jednostavni (*Simple*), inverzna multikvadratična bazna funkcija
RSP	RBF, jednostavni (*Simple*), *thin plate spline* bazna funkcija
RTM	RBF s trendom, multikvadratična bazna funkcija
RTI	RBF s trendom, inverzna multikvadratična bazna funkcija
RTP	RBF s trendom*, thin plate spline* bazna funkcija
KSE	Kriging, jednostavni (*Simple*), eksponencijalni model variograma
KSG	Kriging, jednostavni (*Simple)*, Gausov model variograma
KSC	Kriging, jednostavni (*Simple*), Cauchy model variograma
KOE	Kriging, obični (*Ordinary*), eksponencijalni model variograma
KOG	Kriging, obični (*Ordinary*), Gausov model variograma
KOC	Kriging, obični (*Ordinary*), Cauchy model variograma
KTE	Kriging, s trendom, eksponencijalni model variograma
KTG	Kriging, s trendom, Gausov model variograma
KTC	Kriging, s trendom, Cauchy model variograma

8.1.3.1. Usporedba surogat modela generiranih pomoću 100 računalnih simulacija (LHS – corr plan pokusa)

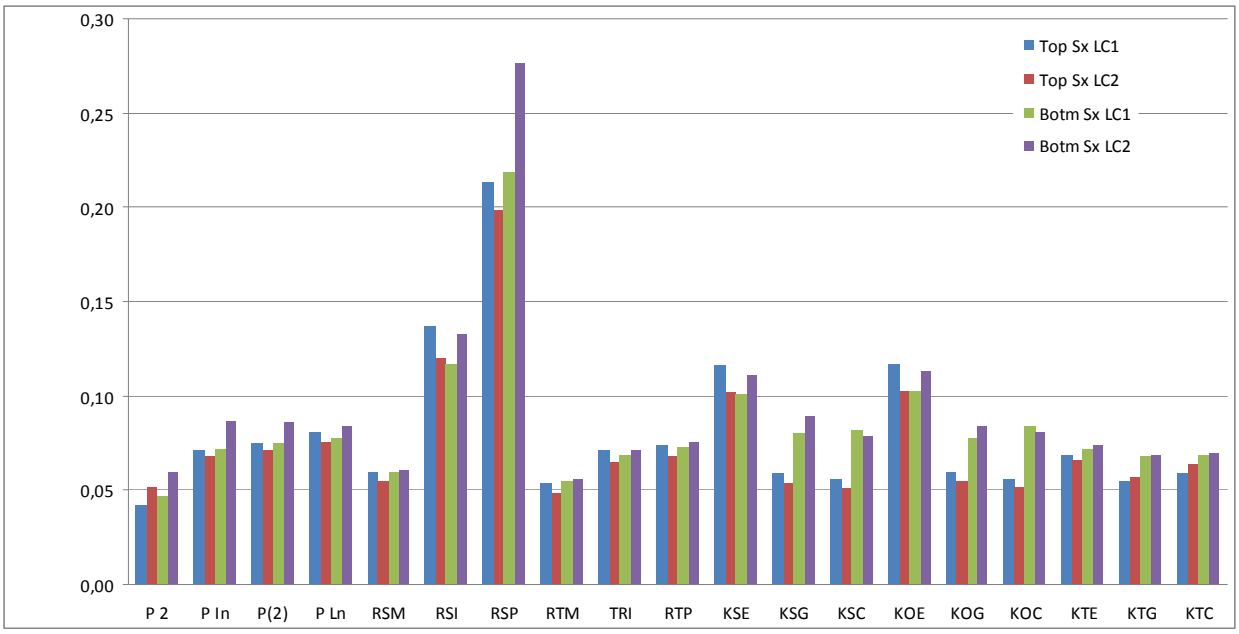
Za generiranje surogat modela korišteno je 100 pokusa definiranih LHS metodom korištenjem MATLAB-ove funkcije `lhsdesign` s kriterijem minimalne korelacije parametara. Za validaciju modela korišteno je 50 novih pokusa definiranih LHS metodom korištenjem MATLAB-ove funkcije `lhsdesign` s `minmax` kriterijem, kako bi se dobio pouzdani model točnosti predviđanja surogata. Kao mjere točnosti korišteni su korijen sume kvadrata odstupanja - RMSE i normalizirani korijen sume kvadrata odstupanja – NRMSE. Prvi je dan kao mjera apsolutnog odstupanja surogat modela u domeni ispitivanja, dok je NRMSE korišten zbog svog jasnog prezentiranja relativne greške odstupanja. Na Sl. 8.2 može se vidjeti prikaz sučelja OCTOPUS Designera za validaciju surogat modela u kojem su izvršeni svi testovi koji se nalaze u nastavku.



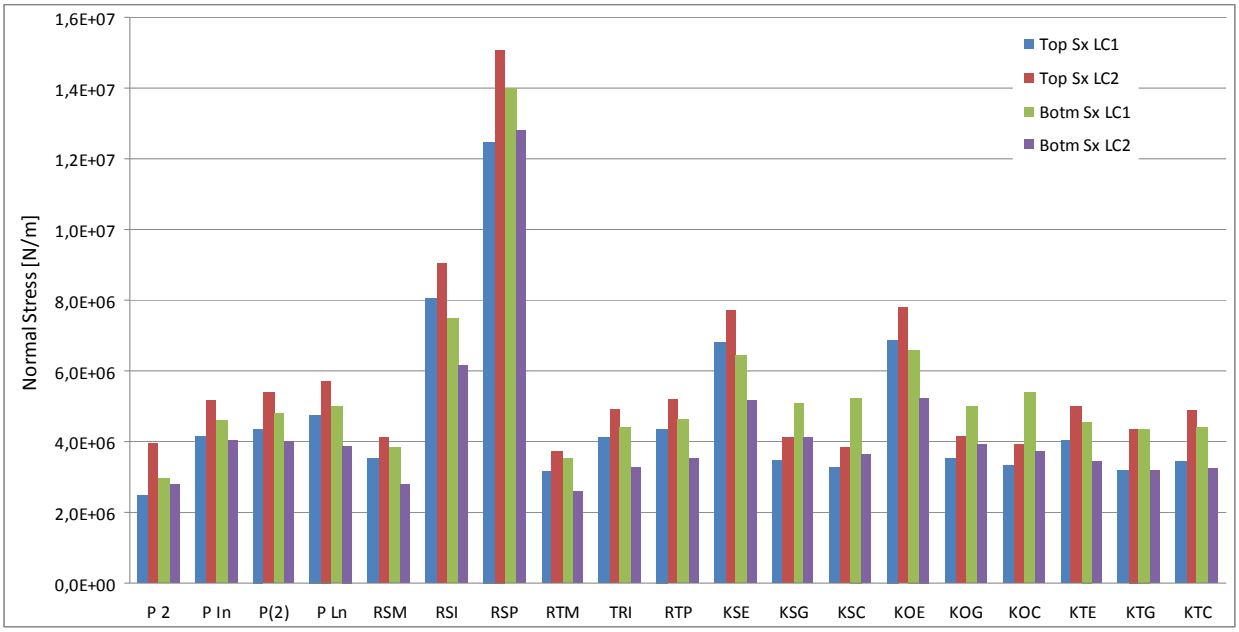
No	Top_LC1_sigmaX	Top_LC1_sigmaY	Top_LC1_tauXY	Top_LC1_Mbz	Top_LC1_Mbx	Top_LC2_sigmaX	Top_LC2_sigmaY	To
40	4.568E+7	-3.439E+6	5.0656E-8	1.05E+5	1.712E+4	-4.8084E+7	-1.1747E+7	
41	4.207E+7	1.27E+6	-2.0347E-8	5.616E+4	3.185E+4	-4.1324E+7	-4.2618E+6	
42	3.3E+7	-3.4827E+6	1.1523E-7	4.805E+4	3.199E+4	-4.1209E+7	-9.2525E+6	
43	4.834E+7	-2.8055E+6	3.2956E-8	1.001E+5	2.738E+4	-4.8112E+7	-8.3106E+6	
44	6.286E+7	1.948E+6	1.1364E-7	1.36E+5	1.867E+4	-6.5309E+7	-1.2743E+7	
45	5.452E+7	-1.316E+6	3.3775E-8	1.042E+5	1.014E+4	-5.7421E+7	-8.6847E+6	
46	4.238E+7	-2.9078E+6	1.4768E-8	1.353E+5	-1.3935E+4	-4.0173E+7	-4.508E+6	
47	5.354E+7	-3.5563E+6	5.7322E-8	1.094E+5	2.384E+4	-5.3958E+7	-1.105E+7	
48	3.205E+7	-5.5959E+6	-8.421E-8	1.169E+5	1.058E+4	-3.2975E+7	-6.2477E+6	
49	5.671E+7	-4.3794E+6	1.4646E-7	6.856E+4	-1.2342E+4	-6.3209E+7	-1.077E+7	
50	4.398E+7	-1.1773E+6	7.6495E-8	1.238E+5	2.734E+4	-4.7895E+7	-5.83E+6	
MSE	**6.1E+12**	**4.17E+12**	**4.286E-15**	**1.01E+8**	**1.15E+8**	**1.56E+13**	**2.65E+12**	
RMSE	**2.47E+6**	**2.04E+6**	**6.547E-8**	**1.01E+4**	**1.07E+4**	**3.95E+6**	**1.63E+6**	
NRMSE	**0.042**	**0.204**	**0.5**	**0.081**	**0.124**	**0.052**	**0.086**	
n	**50**	**50**	**50**	**50**	**50**	**50**	**50**	

Sl. 8.2. Prikaz sučelja OCTOPUS Designera za validaciju surogat modela

Na Slikama Sl. 8.3 do Sl. 8.10 dan je grafički prikaz dobivenih rezultata na gornjoj (*top*) i donjoj (*bottom*) podstrukturi, dok je ispod samih slika dan komentar analize.

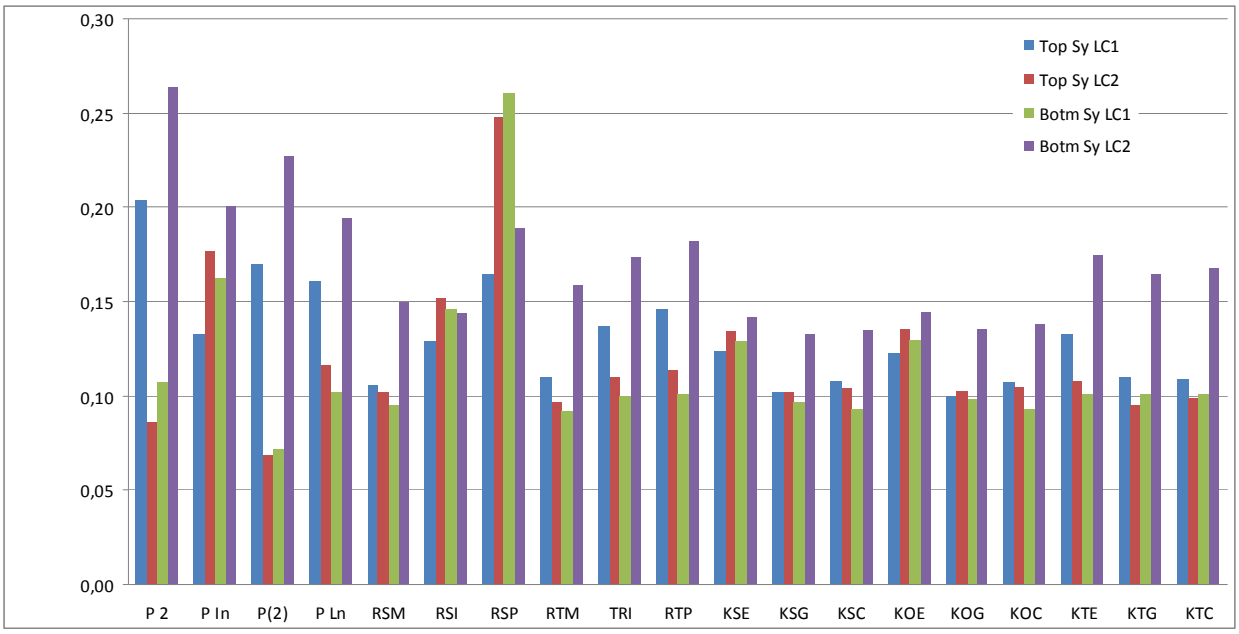


Sl. 8.3. NRMSE greška testiranih surogata za normalno naprezanje σ_x na palubi i dnu

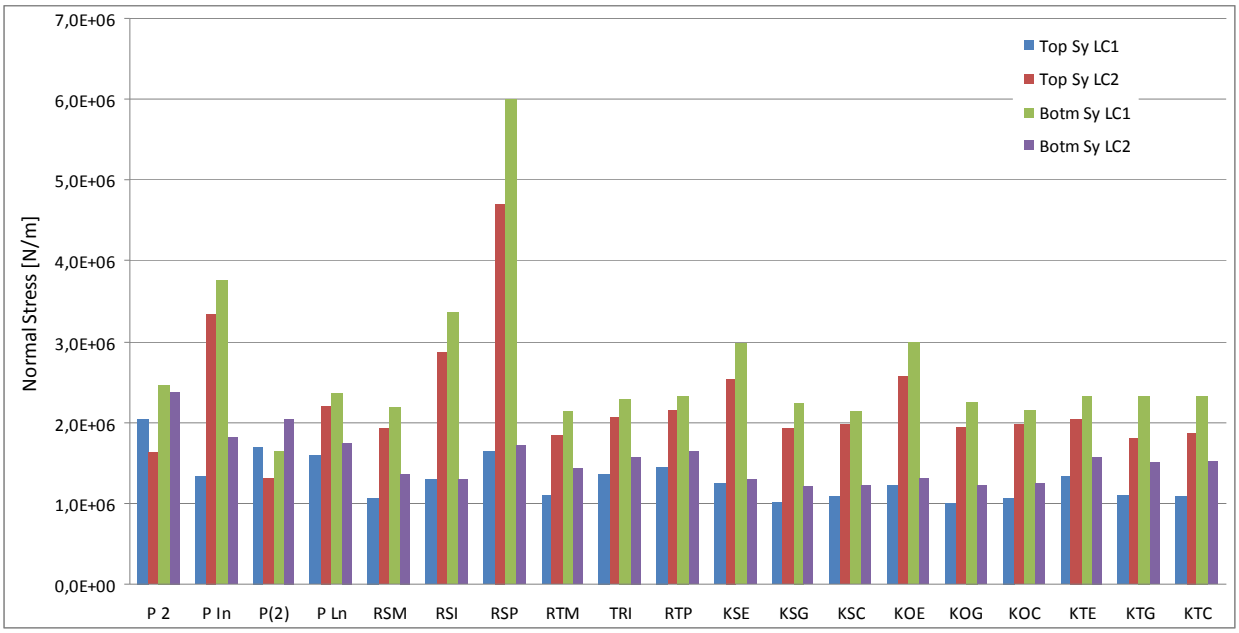


Sl. 8.4. RMSE greška testiranih surogata za normalno naprezanje σ_x na palubi i dnu

Prvo je dan prikaz točnosti normalnog naprezanja σ_x zbog toga što ovo naprezanje ima najveći utjecaj na podobnost konstrukcije. Iz dijagrama je vidljivo da polinom drugog stupnja najbolje opisuje ovaj odziv. Prema NRMSE mjeri, vidljivo je da je greška za ovaj model za oba slučaja opterećenja, na obje podstrukture oko 5% . Istovremeno RMSE greška daje da je apsolutna srednja mjera greške oko 3 MPa . Vidljivo je da ni ostali surogat modeli ne odstupaju značajno te da je relativna mjera greške oko 7% dok je apsolutna srednja mjera greške oko 6 MPa što je potpuno prihvatljivo za predloženu namjenu. Jedino značajnije odstupanje daju RBF jednostavni modeli s inverznom multikvadratičnom funkcijom i splajnom.

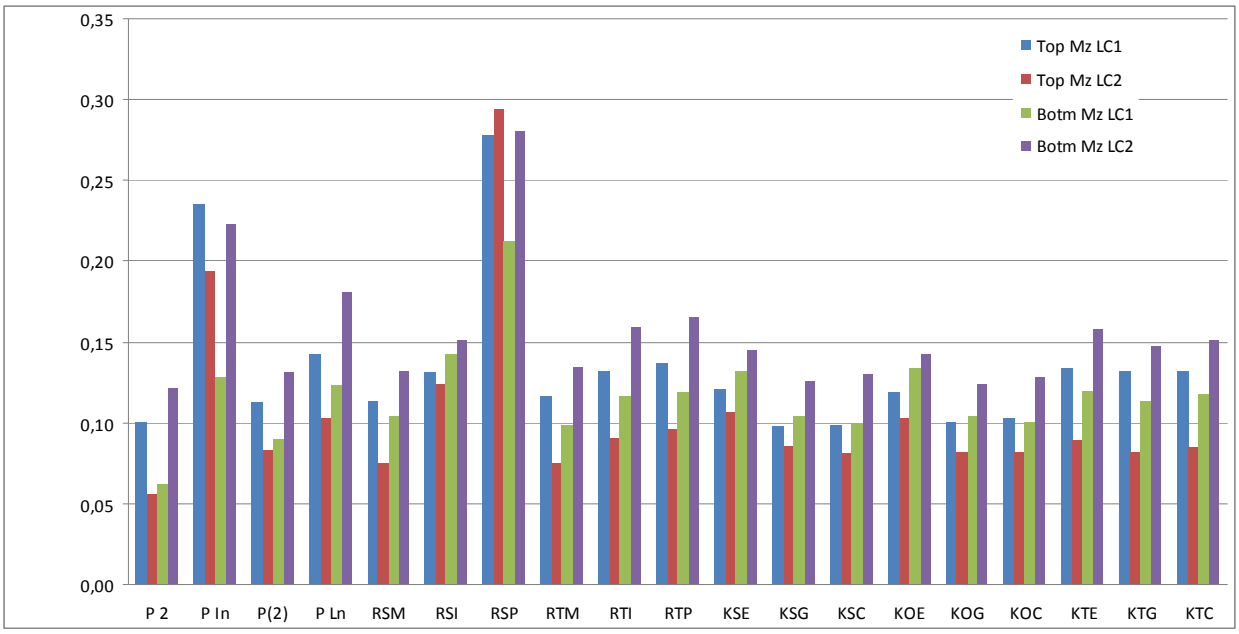


Sl. 8.5. NRMSE greška testiranih surogata za normalno naprezanje σ_y na palubi i dnu

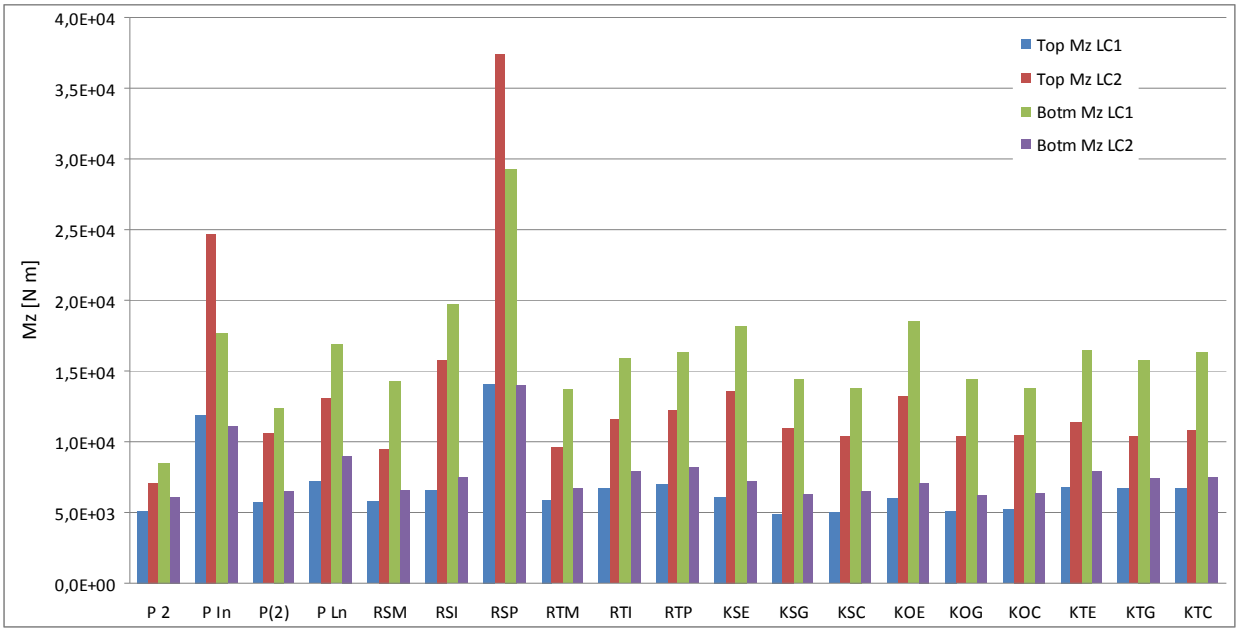


Sl. 8.6. RMSE greška testiranih surogata za normalno naprezanje σ_y na palubi i dnu

Na gornjim slikama dan je prikaz analize točnosti normalnog naprezanja σ_y. Iz dijagrama je vidljivo da je normalizirana greška značajno veća no u slučaju naprezanja σ_x.no da je RMSE mjera greške na nižoj razini. Ovakav rezultat može se objasniti s tim što je razina ovog naprezanja niža, ali i promjenjivija po podstrukturi.

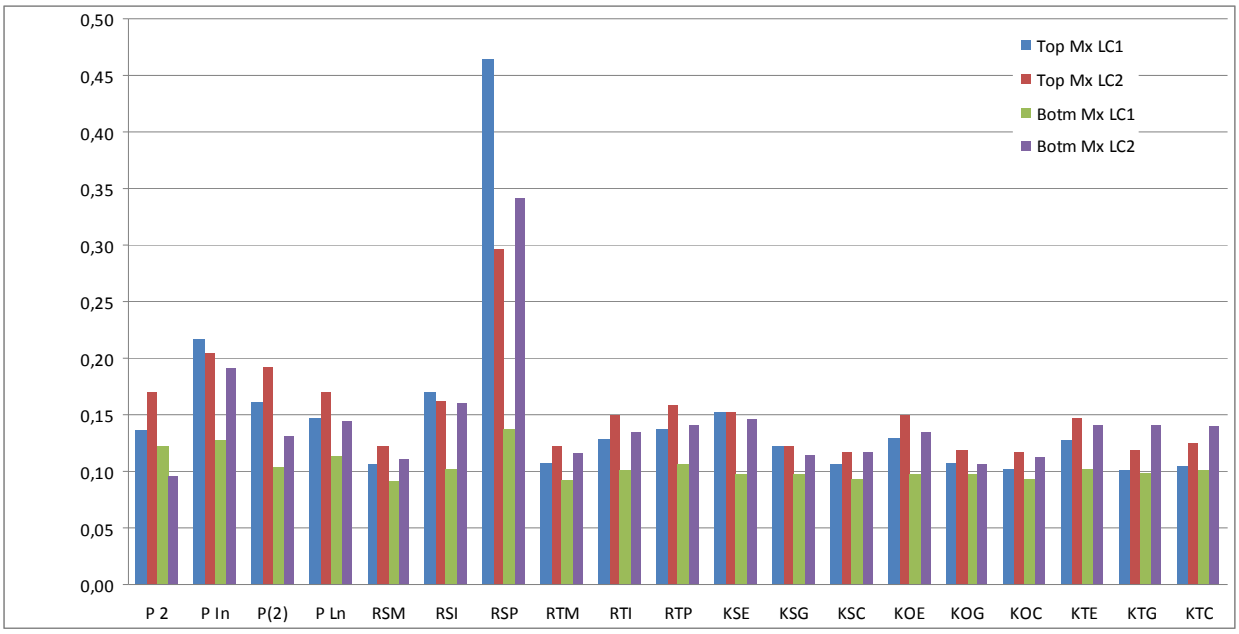


Sl. 8.7. NRMSE greška testiranih surogata za moment savijanja uzdužnih jakih nosača Mz na palubi i dnu

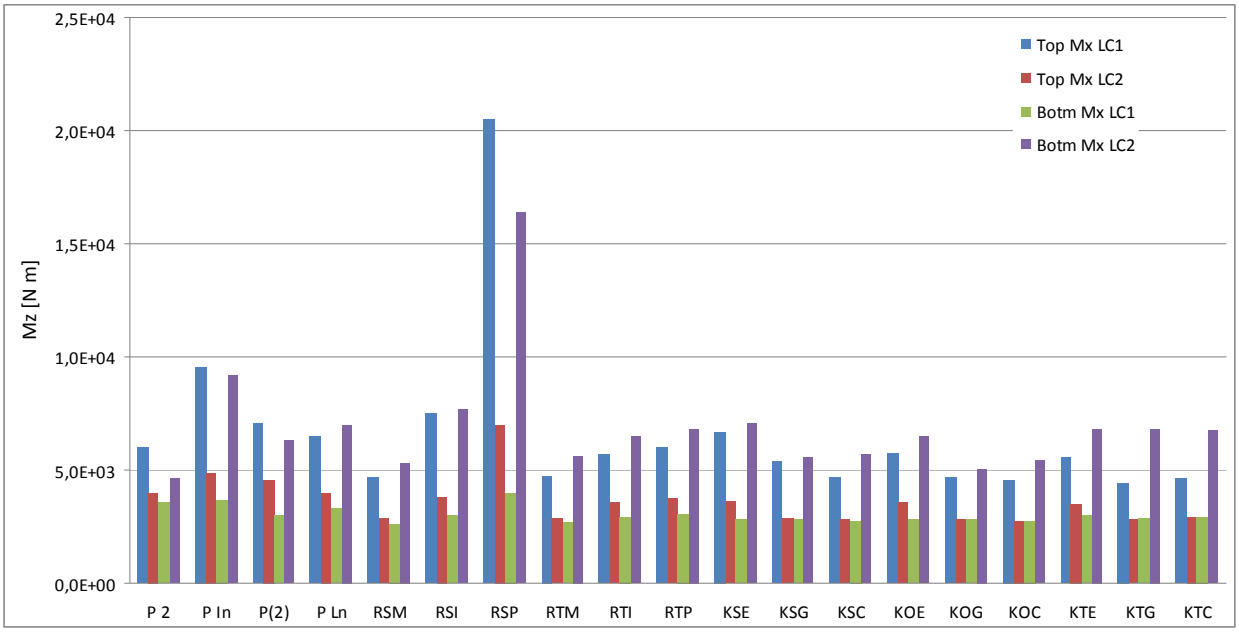


Sl. 8.8. RMSE greška testiranih surogata za moment savijanja uzdužnih jakih nosača Mz na palubi i dnu

Na gornjim slikama dan je prikaz analize točnosti momenta savijanja jakih nosača *Mz*. Slično kao i kod surogata naprezanja σ_y primjećuje se da je NRMSE relativno loš. Najvjerojatniji uzrok ovakvog rezultata je promjenjivost distribucija momenta po podstrukturi. Kao što je napomenuto u potpoglavlju 6.3.1, ovu grešku je moguće reducirati frekventnijim kalibriranjem lokalnih koeficijenata pojedinog surogata.



Sl. 8.9. NRMSE greška testiranih surogata za moment savijanja poprečnih jakih nosača Mx na palubi i dnu



Sl. 8.10. RMSE greška testiranih surogata za moment savijanja poprečnih jakih nosača Mx na palubi i dnu

Na gornjim slikama dan je prikaz analize točnosti momenta savijanja jakih nosača *My*,s tim da su komentari isti kao i za prethodno opisano savijanje jakih nosača *Mz.*

Zaključak analize točnosti surogat modela

Iz prikazanih rezultata može se zaključiti da su primarna naprezanja konstrukcije vrlo dobro aproksimirana s gotovo svim surogat modelima, dok su sekundarna normalna naprezanja te momenti savijanja nešto niže razine točnosti, najvjerojatnije zbog promjenjivije distribucije naprezanja po podstrukturi što je moguće kompenzirati tijekom optimizacije kalibriranjem lokalnih koeficijenata pojedine evaluacijske jedinice – *patcha*.

Također se može primijetiti da s izuzetkom jednostavnog RBF-a s *thin plate spline* baznom funkcijom sve metode surogat modeliranja podjednako uspješno aproksimiraju odziv. Pri tome je bitno napomenuti da to znači da kompleksnije surogat metode, poput Kriginga i RBF-a, nemaju značajno bolju točnost od linearnog i čisto kvadratičnog modela, a da zahtijevaju značajno više vremena za generiranje modela. To je naročito uočljivo kod implementiranih verzija Kriging-a koje zahtijevaju za red veličine više vremena od jednostavnih polinomskih surogat modela.

8.1.4. Usporedba rješavanja optimizacijskih problema sa i bez surogata strukturnih odziva

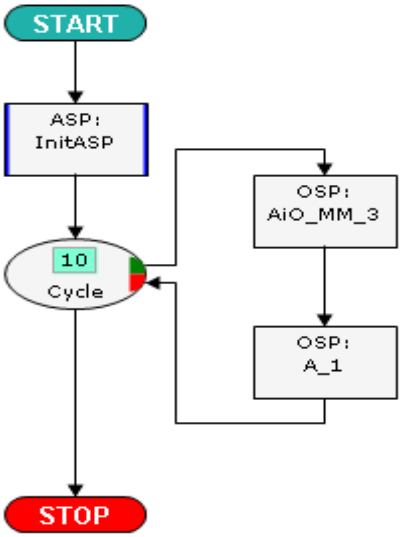
Kako bi se utvrdila korisnost predloženih surogat modela odziva izvršena je usporedba rješavanja nekoliko različito formuliranih optimizacijskih problema. Sve formulacije uključuju inicijalnu MKE analizu nakon koje slijedi 10 ciklusa optimizacije. U okviru svakog ciklusa rješavat će se 1 (za AIO formulacije) ili 3 (za dekomponirane formulacije) optimizacijska potproblema nakon kojih slijedi MKE analiza s evaluacijom podobnosti u fiktivnom AIO problemu koji ne provodi optimizaciju već samo poziva spomenutu analizu i . Problem je nazvan fiktivnim zato što samo bilježi stvarno stanje svih komponenti optimizacijskog problema, a ne provodi nikakvu dodatnu optimizaciju.

Simulacijska sekvenca svakog potproblema koji ne koristi surogat modele odziva uključuje evaluaciju svih evaluacijskih elemenata uz pretpostavku nepromijenjenih pomaka MKE modela. Time se pokušala ostvariti optimizacija poput one u tzv. *Legacy* verziji programa MAESTRO (vidi potpoglavlje 6.2.2.3). Da bi se spriječilo „zabijanje“ u nepodobno područje u prvim ciklusima optimizacije , potrebno je optimizacijskom problemu zadati još i ograničenje na maksimalnu promjenu varijabli unutar jednog ciklusa, jer se u suprotnom može dogoditi da optimizacijski algoritam ne može zadovoljiti ograničenja zbog nerealnog stanja pomaka iz prethodnog ciklusa. Simulacijska sekvenca svakog potproblema koji koristi surogat modele odziva uključuje predviđanje svih strukturnih odziva koji su potrebni za izračun evaluacijskih elemenata te evaluaciju podobnosti.

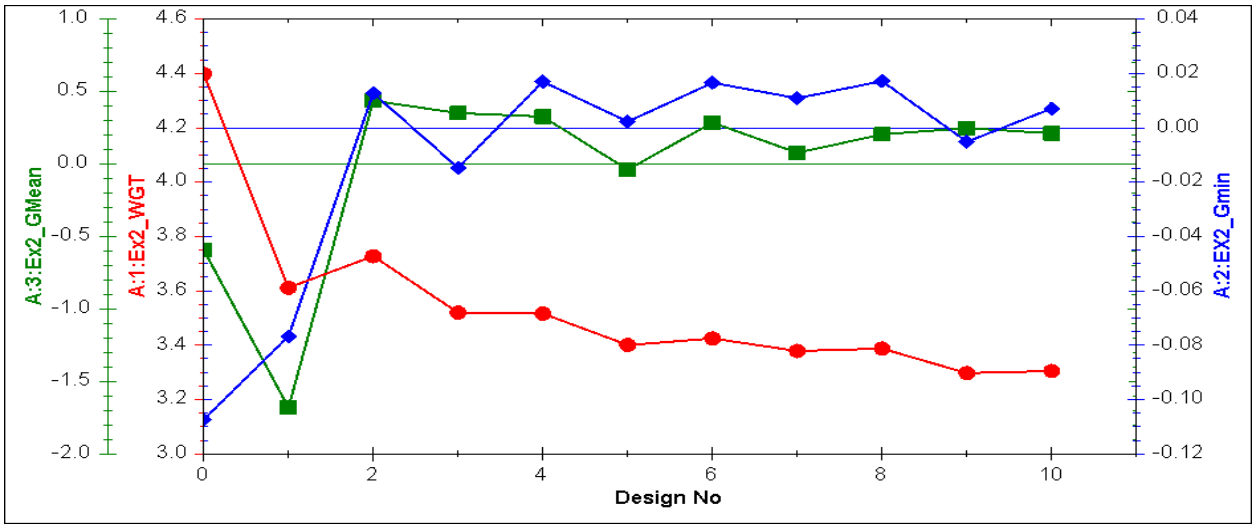
Svim optimizacijskim potproblemima je dodijeljena ista optimizacijska metoda – SQP tj. NLPQLP implementacija s maksimalno 20 iteracija.

8.1.4.1. Optimizacija AiO bez surogata

U ovoj formulaciji, sve komponente optimizacijskog problema sadržane su u jednom optimizacijskom potproblemu AiO_MM_3 (vidi Sl. 8.11), koji pri svakoj iteraciji poziva evaluaciju svih evaluacijskih elemenata uz pretpostavku nepromijenjenih pomaka MKE modela. Nakon optimizacije AiO_MM_3 problema slijedi MKE (A_1) analiza čime se dobiva novo stanje pomaka. Ostvareni rezultati kroz iteracije vidljivi su na Sl. 8.12.



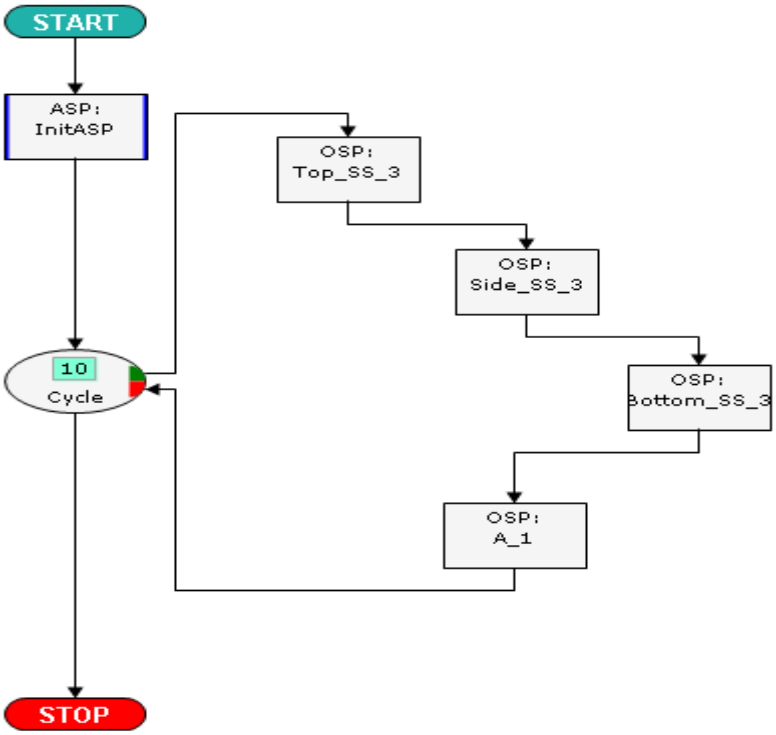
Sl. 8.11. Sekvenca rješavanja AiO formulacije bez surogata



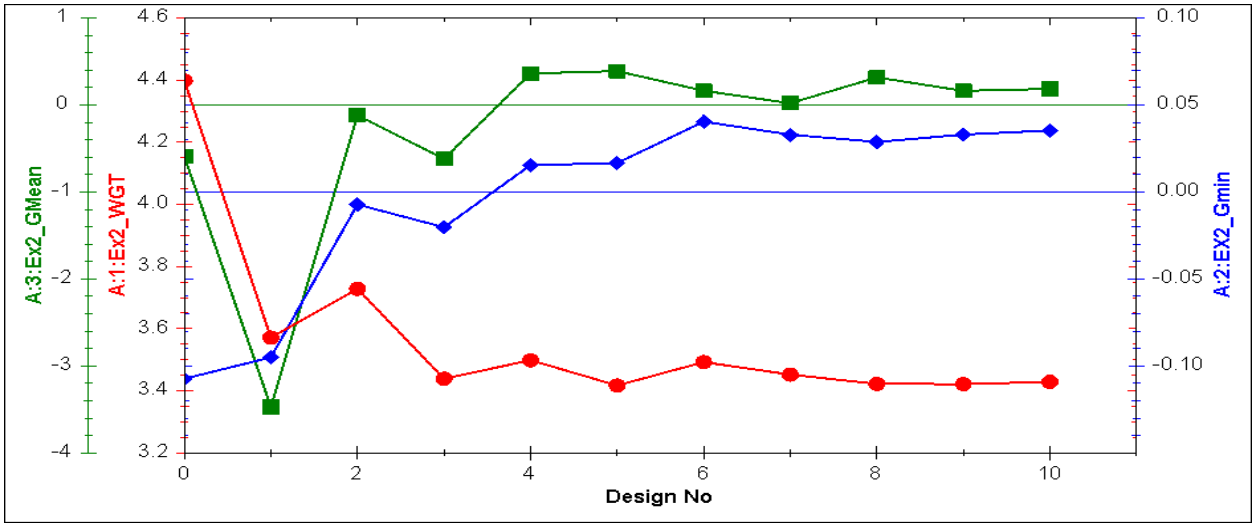
Sl. 8.12. Ostvarene vrijednosti atributa kroz iteracije za AiO bez surogata (bilježeno u A_1)

8.1.4.2. Optimizacija dekomponiranog problema bez surogata

U ovoj formulaciji komponente optimizacijskog problema podijeljene su po podstrukturama, čime je postavljeno tri optimizacijska potproblema: Top_SS_3, Side_SS_3,Bottom_SS_3 (vidi Sl. 8.13), koji se optimiziraju nezavisno uz pretpostavku nepromijenjenih pomaka MKE modela. Ovakav pristup u približno simulira rješavanje optimizacijski problem poput onog u tzv. *Legacy* verziji MAESTRA- kad se koristi samo lokalna optimizacija vojeva (vidi potpoglavlje 6.2.2.3). Nakon optimizacije svih potproblema slijedi MKE (A_1) analiza čime se dobiva novo stanje pomaka. Ostvareni rezultati kroz iteracije vidljivi su na Sl. 8.14.



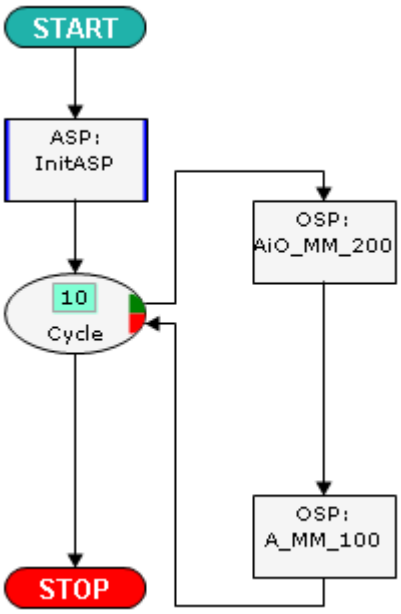
Sl. 8.13. Sekvenca rješavanja dekomponiranog problema bez surogata



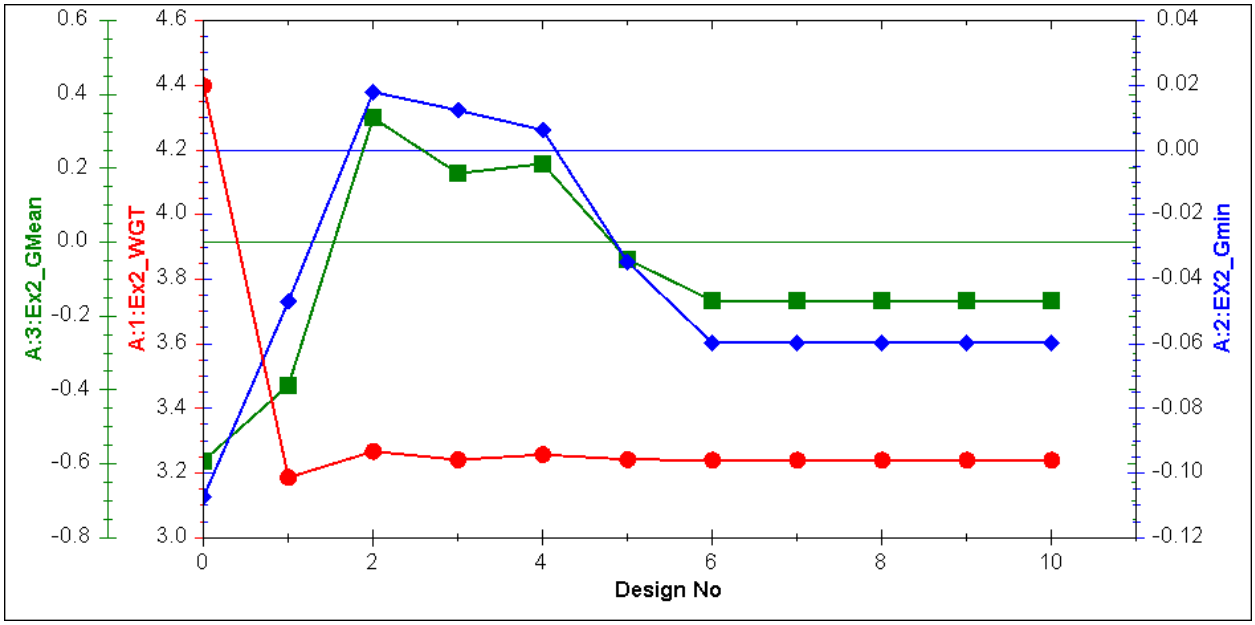
Sl. 8.14. Ostvarena rješenja dekomponiranog problema bez surogata (Bilježeno u A_1)

8.1.4.3. Optimizacija AiO sa globalnim surogatom odziva

U ovoj formulaciji u odnosu na 8.1.4.1, svaka računalna simulacija u AiO problemu uključuje predviđanje odziva na evaluacijskim jedinicama globalnim surogat modelom. Svi korišteni surogat modeli odziva su polinomski surogati drugog rede dobiveni pokusima opisanim u potpoglavlju 0. Sekvenca rješavanja ovog problema prikazana je na Sl. 8.15, dok su ostvarena rješenja prikazana Sl. 8.16.



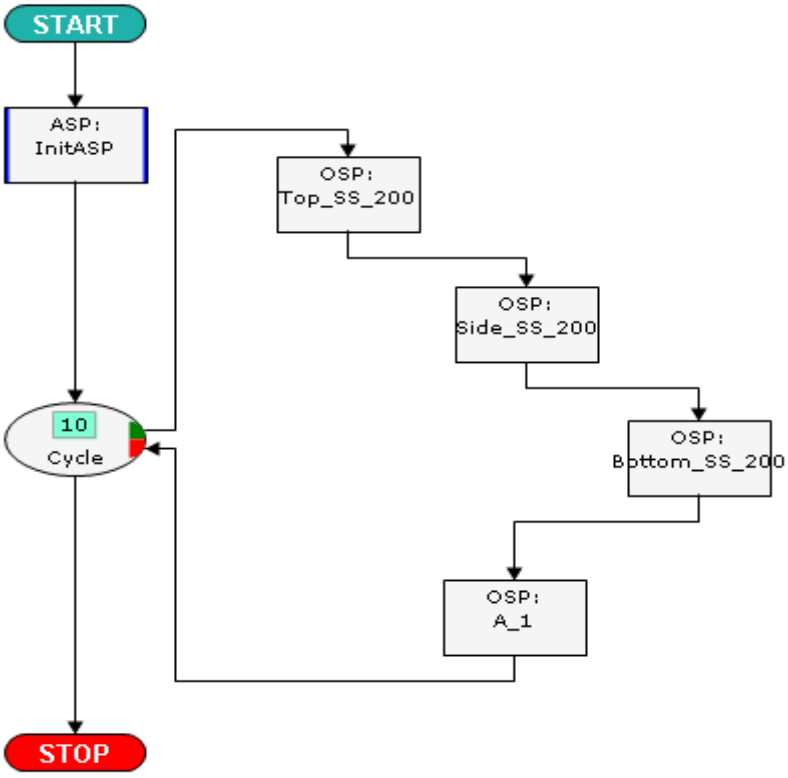
Sl. 8.15. Sekvenca rješavanja AiO problema sa surogat modelom strukturnog odziva



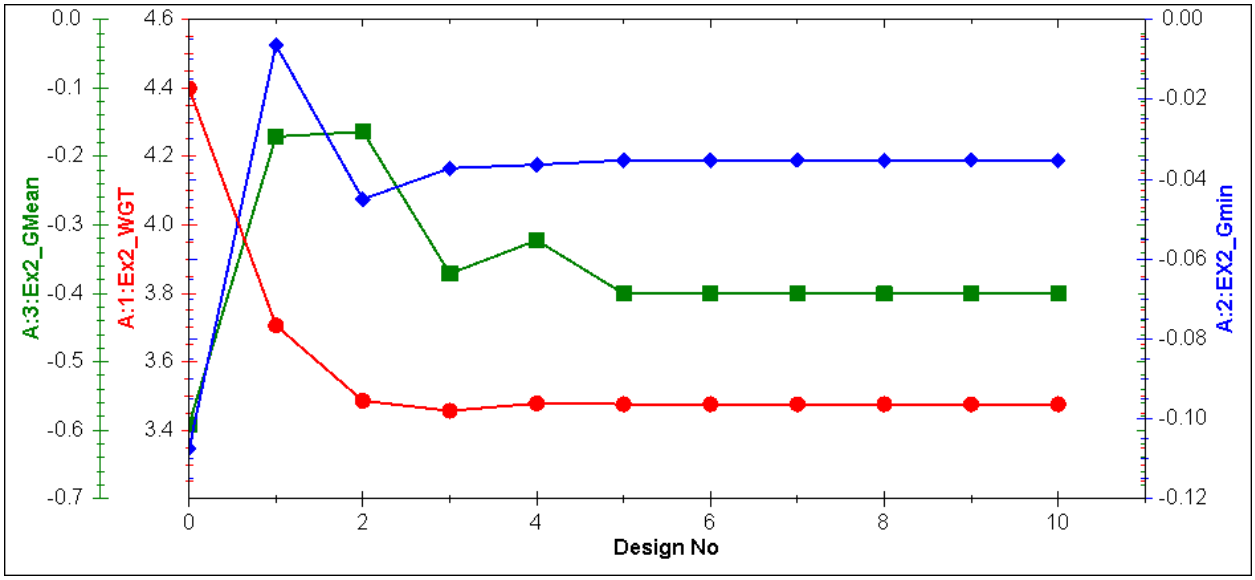
Sl. 8.16.Ostvarena rješenja AiO problema sa surogat modelima odziva (Bilježeno u A_MM_100)

8.1.4.4. Optimizacija dekomponiranog problema sa globalnim surogatima odziva

U ovoj formulaciji u odnosu na , svaka računalna simulacija u AiO problemu uključuje predviđanje odziva na evaluacijskim jedinicama globalnim surogat modelom. Kao i u prethodnoj formulaciji, svi korišteni surogat modeli odziva su polinomski surogati drugog rede dobiveni pokusima opisanim u potpoglavlju 0. Sekvenca rješavanja ovog problema prikazana je na Sl. 8.17, dok su ostvarena rješenja prikazana Sl. 8.18.



Sl. 8.17. Sekvenca rješavanja dekomponiranog problema sa surogat modelima odziva



Sl. 8.18. Ostvarena rješenja dekomponiranog problema sa surogatima odziva (Bilježeno u A_1)

8.1.4.5. Analiza i usporedba ostvarenih rezultata formulacija u potpoglavljima 8.1.4.1-8.1.4.4

Iz rezultata prikazanim u potpoglavljima 8.1.4.1-8.1.4.4, može se vidjeti da optimizacija potpomognuta surogatima stabilno konvergira kroz samo nekoliko iteracija, dok se kod optimizacije bez surogata primjećuje stalni uzorak prebačaj – podbačaj.

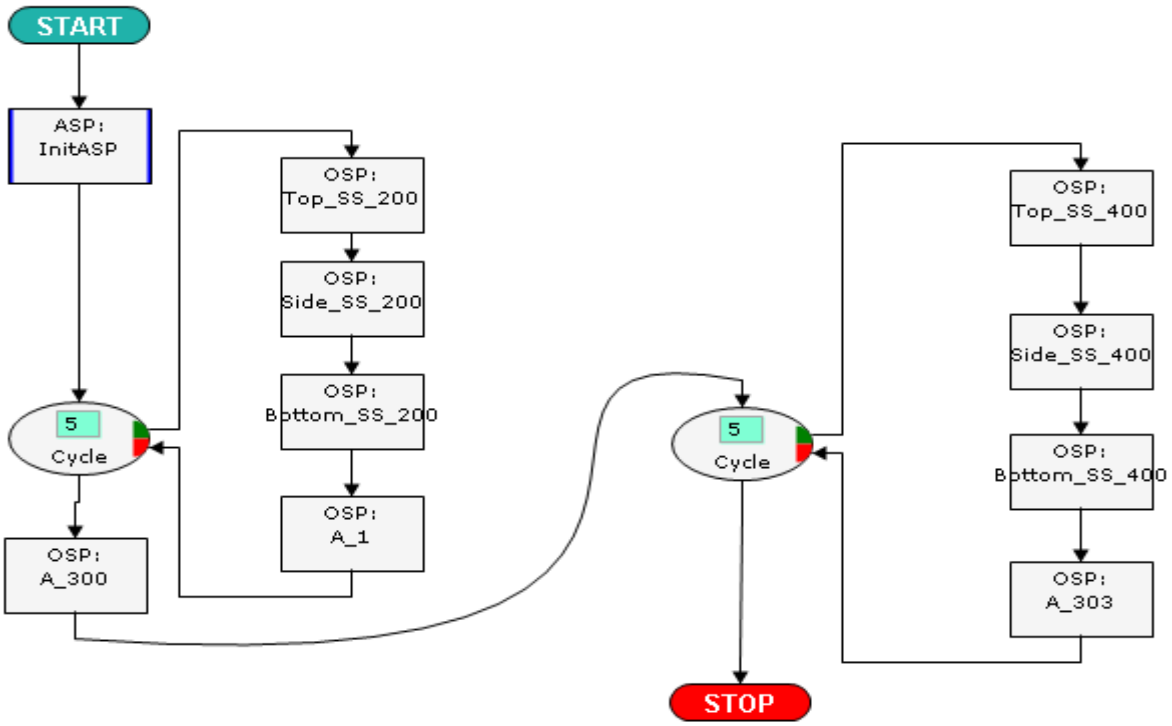
Takovo ponašanje je i očekivano s obzirom da tijekom optimizacije potproblema bez surogat modela strukturnih odziva nema informacije o promjeni stanja naprezanja odnosno promjeni stanja pomaka. Zbog toga optimizacijski problem, u kojem se promjenom stanja varijabli mijenja samo izdržljivost tankostjene konstrukcije, svojom konvergencijom do optimuma dovodi sustav do preopterećenog ili podopterećenog stanja. Ipak, mora se primijetiti da unatoč tome sustav polako konvergira i da u okviru promatranog broja iteracija konvergira na praktično primjenjive rezultate.

S druge strane optimizacijski problemi sa surogat modelima iznimno brzo konvergiraju čak i kod dekomponiranog modela. Isto tako vidljivo je i da su optimizacijski problemi sa surogat modelima strukturnih odziva zapravo konvergirali u donekle nezadovoljena rješenja zbog netočnosti surogat modela. Kako bi se ta netočnost kompenzirala, potrebno je nakon nekoliko globalnih ciklusa kalibrirati lokalne koeficijente pojedine evaluacijske jedinice da kompenziraju razliku između stvarnog odziva i aproksimiranog odziva, kao što je i opisano u potpoglavlju 6.3.1. U idućem potpoglavlju predstavljena je upravo takova sekvenca rješavanja projektnog problema.

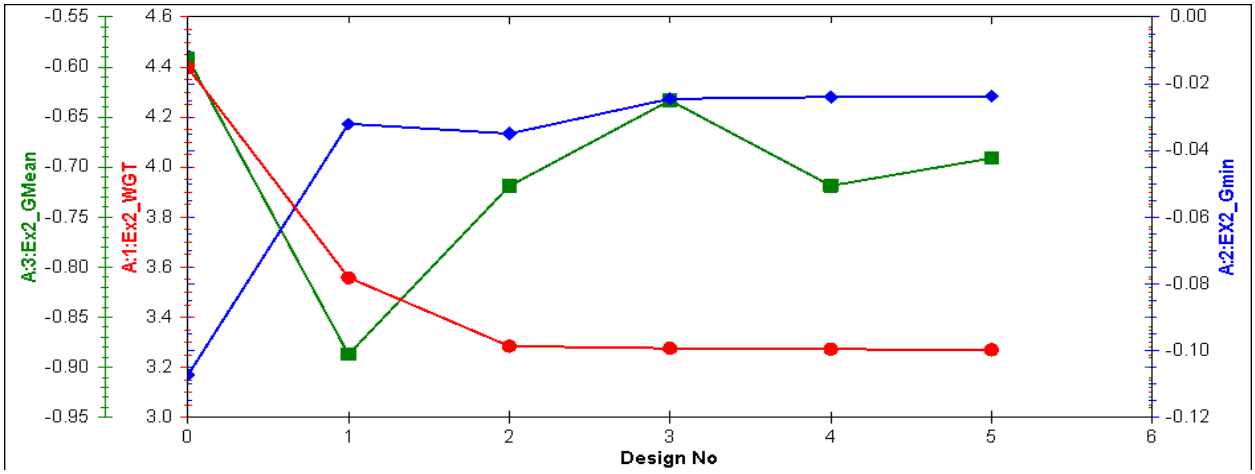
8.1.4.6. Optimizacija dekomponiranog problema sa globalnim i lokalnim surogatom odziva

U sekvenci rješavanja prikazanoj na Sl. 8.19 može se vidjeti da se sastoji od lijevog dijela koji zapravo odgovara sekvenci prikazanoj u potpoglavlju 8.1.4.4 , s tim da je zadano 5 ciklusa tog dijela sekvence, dok je preostalih 5 iskorišteno za rad s lokalno kalibriranim surogat modelima odziva. U desnom dijelu sekvence u A_303 vrši se kalibracija nakon MKE proračuna i evaluacije sa realnim stanjem naprezanja.

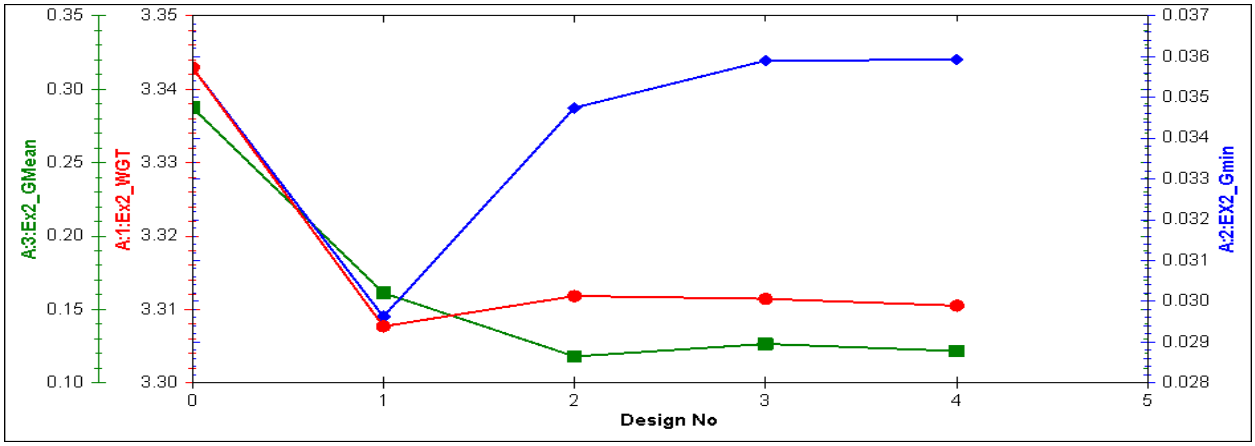
Ostvareni rezultati u prvih 5 ciklusa koji rade s globalnim surogatima strukturnih odziva mogu se vidjeti na Sl. 8.20 dok su na Sl. 8.21 prikazane ostvarene vrijednosti atributa u zadnjih 5 iteracija s lokalno kalibriranim surogat modelima odziva. Iz prikazanih dijagrama se vidi da je predložena sekvenca stabilno konvergirala na granicu podobnog područja.



Sl. 8.19. Sekvenca rješavanja problema koji uključuje lokalno kalibriranje surogat modela odziva



Sl. 8.20. Ostvarena rješenja dijela sekvence koji radi s globalnim surogatom (bilježeno u A_1)



Sl. 8.21. Ostvarene vrijednosti atributa u dijelu sekvence koji radi s lokalno kalibriranim surogatima odziva (bilježeno u A_303)

Iz provedenih testova može se zaključiti da korištenje surogat modela odziva doprinosi brzini konvergencije, no i da je istovremeno moguće ostvariti bolje vrijednosti projektnih ciljeva. Naravno preduvjet za to je mogućnost izvođenja dovoljnog broja računalnih simulacija potrebnih za generiranje surogat modela strukturnih odziva.

8.2. Projektiranje tankostjene konstrukcije broda za prijevoz putnika i vozila – ROPAX

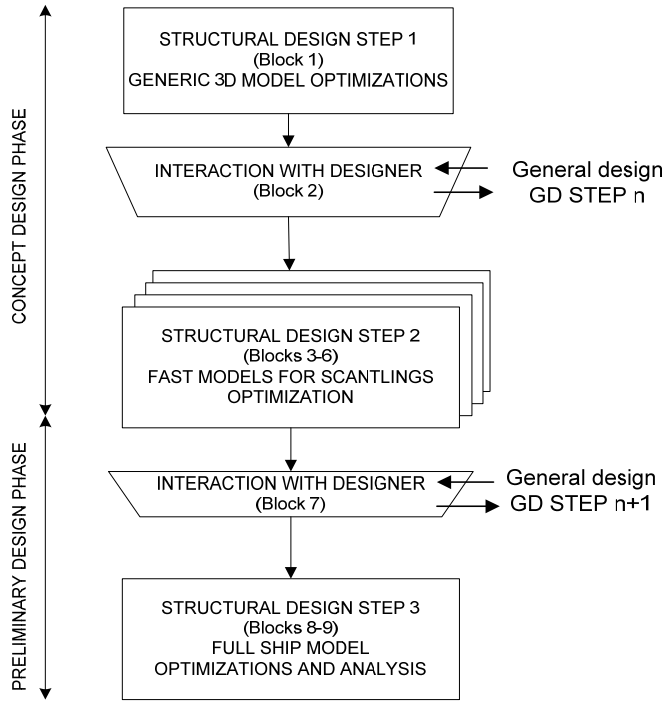
Projekt RoPax broda prikazanog na Sl. 8.22 dio je EU FP6 projekta IMPROVE [5].

Sl. 8.22. EU projekt IMPROVE RoPax

Glavne karakteristike projektiranog broda:

- Ukupna duljina 193 m
- Duljina između okomica 180 m
- Širina 29.8 m
- Projektirani gaz 7.5 m
- Blok koeficijent. 0.53
- Pokusna brzina 24.5 kn
- Snaga glavnog motora *(MCR)* 14940 kW
- Izlazna snaga aktivnog kormila 5000 kW

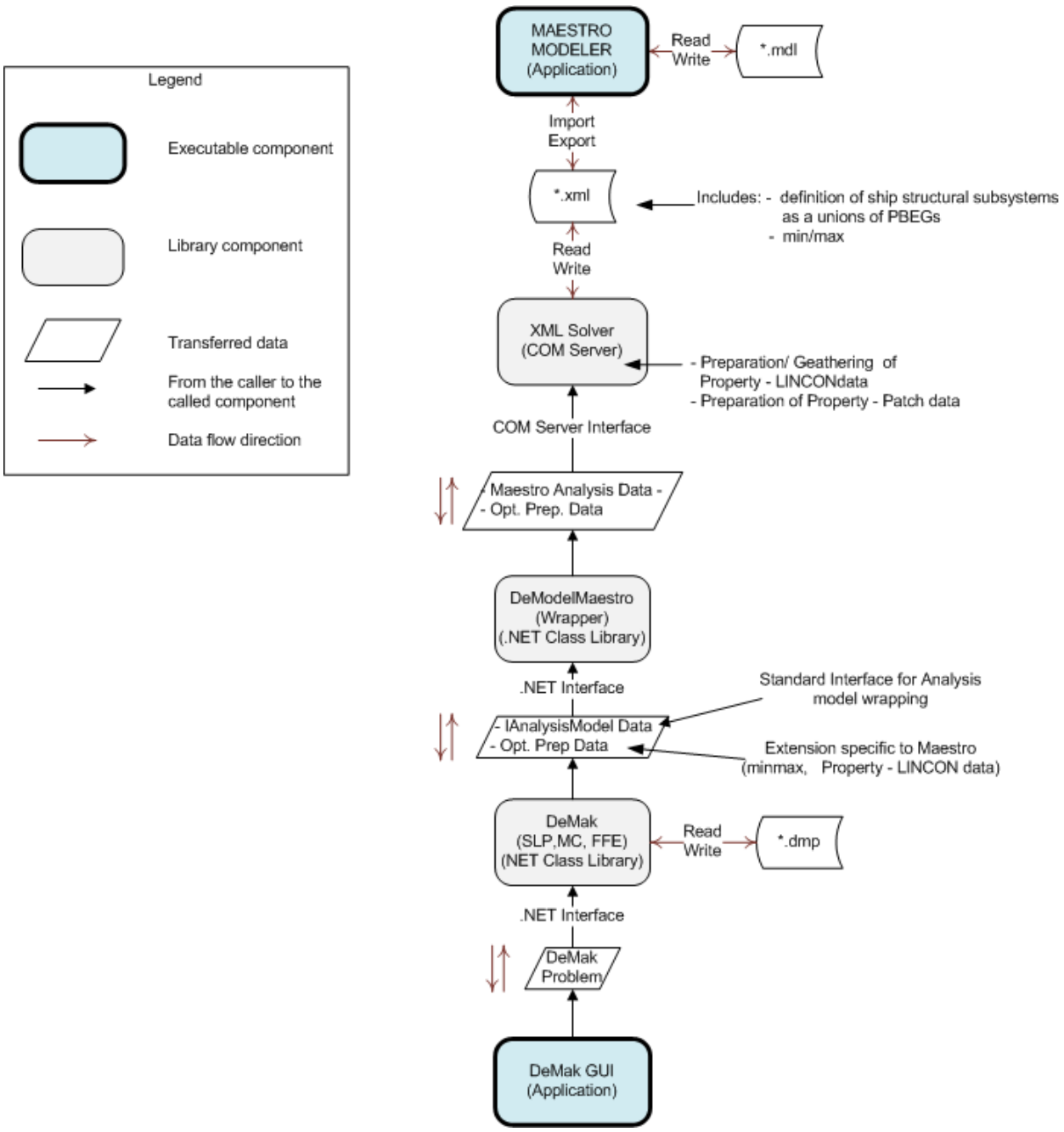
U radu [76] na strukturnom modelu ovog broda prikazana je metodologija projektiranja tankostjene konstrukcije koja se sastoji od tri koraka: topološke optimizacije generičkih 3D modela [78], optimizacije dimenzija strukturnih elemenata korištenjem brzih 2D modela analize i finalne optimizacije na potpunom (full ship) 3D MKE modelu broda (Sl. 8.23). U ovom radu će se razmatrati mogućnost primjene 3D MKE modela u drugom koraku predložene metodologije korištenjem metoda projektiranja složenih sustava u okviru razvijenog programskog okruženja OCTOPUS Designer.



Sl. 8.23. Metodologija projektiranja strukture višepalubnih brodova [76]

8.2.1. Model za analizu tankostjene konstrukcije RoPax-a

Parcijalni model za analizu tankostjene konstrukcije broda izrađen je u programu MAESTRO [75]. Osnovne karakteristike programa MAESTRO mogu se pronaći u Tbl. 8.6. Za razliku od optimizacija provedenih u okviru koraka 1 i 3 projekta IMPROVE koje su provedene sa starom (legacy) verzijom programa s integriranom optimizacijom [74] , u ovom radu jest korišten novi, komponentizirani rješavač čija integracija unutar okruženja za projektiranje DeMak jest projekt koji se trenutno odvija u suradnji *s DRS Defense Solutions, LLC, Advanced Marine Technology Center (AMTC)*. Opis integracije sustava MAESTRO s DeMak (OCTOPUS Designer) okruženjem nalazi se na Sl. 8.24.

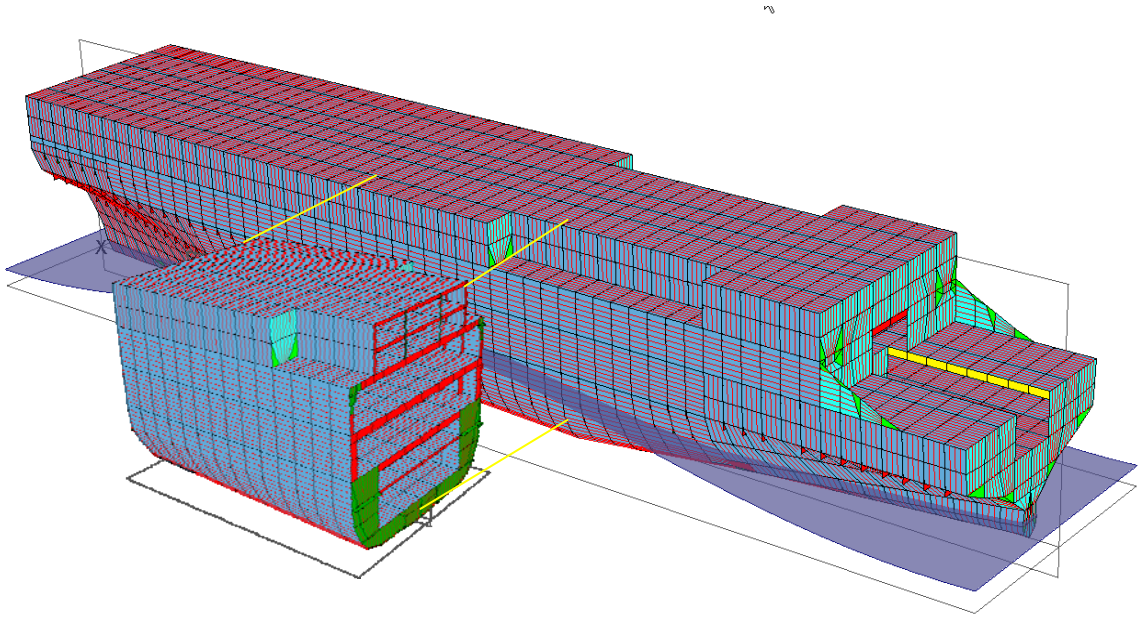


Sl. 8.24. Opis integracije sustava MAESTRO s DeMak (OCTOPUS Designer) okruženjem

Strukturni MKE model RoPax broda korišten u ovom radu, donekle je modificirani model iz koraka 1 projekta IMPROVE i može se vidjeti na Sl. 8.25.

Tbl. 8.6 Osnovne karakteristike MAESTRO programa [5]

META SYSTEM	CHARACTERISTICS OF SUBSYSTEMS (FUNCTIONS)						
	#.	SHORT DESCRIPTION	APPLIED TO	INPUT DATA	OUTPUT VALUES	INTERNAL NAME	TIMING (APPROX.)
Physical (structure) (**Φ**)	1	Structural geometry definition	Maestro Module*	GA, ship lines, decks position	Node coordinates member incidences	MAESTRO Part - 1	interactive
	2	Structural scantlings definition	GP**	scantlings	Strake and C.S. data	MAESTRO Part - 1	interactive
	3	Material properties definition	GP	Material zones, and qualities		MAESTRO Part - 1	interactive
Environment (Loads) (**ε**)	1,NLC	MAESTRO loading module	GP, M. Module	Mission profile, wave data,….	Load vectors	MAESTRO Part - 3	10 ms
Response (Displacements, stresses) (**ρ**)	1	3D FEM calculation of thin-walled structure using macroelements	Ship Structure (partial or full-ship)	Φ-1,2,3, ε-1	Stresses & displacements	MAESTRO Part –4-5	~1-10 min/mod
Adequacy (Safety criteria) (**α**)	1	Stiffened Panel Criteria Library (Hughes)	GP	Φ, ε, ρ	Panel Adequacy Criteria Parameters (g)	MAESTRO Part - 6 S/R EVAL, ()	10 ms
	2	'Stiffened Panel Criteria Library (Paik)	GP	Φ, ε, ρ	Panel Adequacy Criteria Parameters (g)	ULSAP***	10 ms
	3	Ultimate long. strength of hull girder (Hughes).	M. Module	Φ, ε	Hull girder ultimate vertical bending moment	MAESTRO Part -8	5 min
	4	Ultimate long. strength of hull girder (Paik)	M. Module	Φ, ε	Hull girder ultimate vertical bending moment	ALPS-HULL***	5-10 min
Quality (KPI) (**Ω**)	1	Structural weight calculation	M. Module	Φ-1,2,3	Ship module weight	WEIGHT	1ms
	2	Structural cost calculation	M. Module	Φ-1,2,3	Ship module cost	COST	1ms
	3	Position of VCG	M. Module	Φ-1,2,3	Ship VCG	VCG	1ms



Sl. 8.25. Model strukture RoPax broda s označenom kontrolnom konstrukcijom za optimizaciju

8.2.1.1. Model opterećenja

Karakteristika razmatranog broda je relativno fina forma i kontinuiran raspored vlastite težine i tereta što rezultira situacijom da se uvijek na mirnoj vodi nalazi u pregibnom stanju tj. ima višak uzgona na sredini, a težine na krajevima. Uslijed takve distribucije statičkog opterećenja, konstrukcija je opterećena vrlo visokim pregibnim momentom savijanja na mirnoj vodi. Kombinacija maksimalnog pregibnog momenta savijanja na mirnoj vodi i maksimalnog valnog pregibnog momenta daje maksimalna uzdužna naprezanja. Kombinacija minimalnog pregibnog momenta savijanja na mirnoj vodi i maksimalnog valnog progibnog momenta otvara mogućnost pojave tlačnih naprezanja u gornjim palubama. To se po svaku cijenu nastoji izbjeći, jer tlačna naprezanja u gornjim palubama nadgrađa koje su većinom od vrlo tankih limova mogu uzrokovati probleme s izvijanjem panela palube. Za razmatrani brod konstrukcija i pri stanju krcanja s minimalnim pregibnim momentom i maksimalnim valnim progibnim momentom u blago progibnom stanju. Struktura paluba nadgrađa obično se projektira na takav način da je izdržljivost ukrepljenog panela uslijed tlačnih naprezanja minimalno 30 N/mm^2.

Distribucija poprečne sile na mirnoj vodi obično slijedi uobičajenu distribuciju s maksimalnim vrijednostima u području oko 0.25 L i 0.75 L od krmene okomice. Projektne veličine poprečne sile dobiju se zbrajajući maksimalnu vrijednost poprečne sile uslijed vala s maksimalnom vrijednosti poprečne sile na mirnoj vodi.

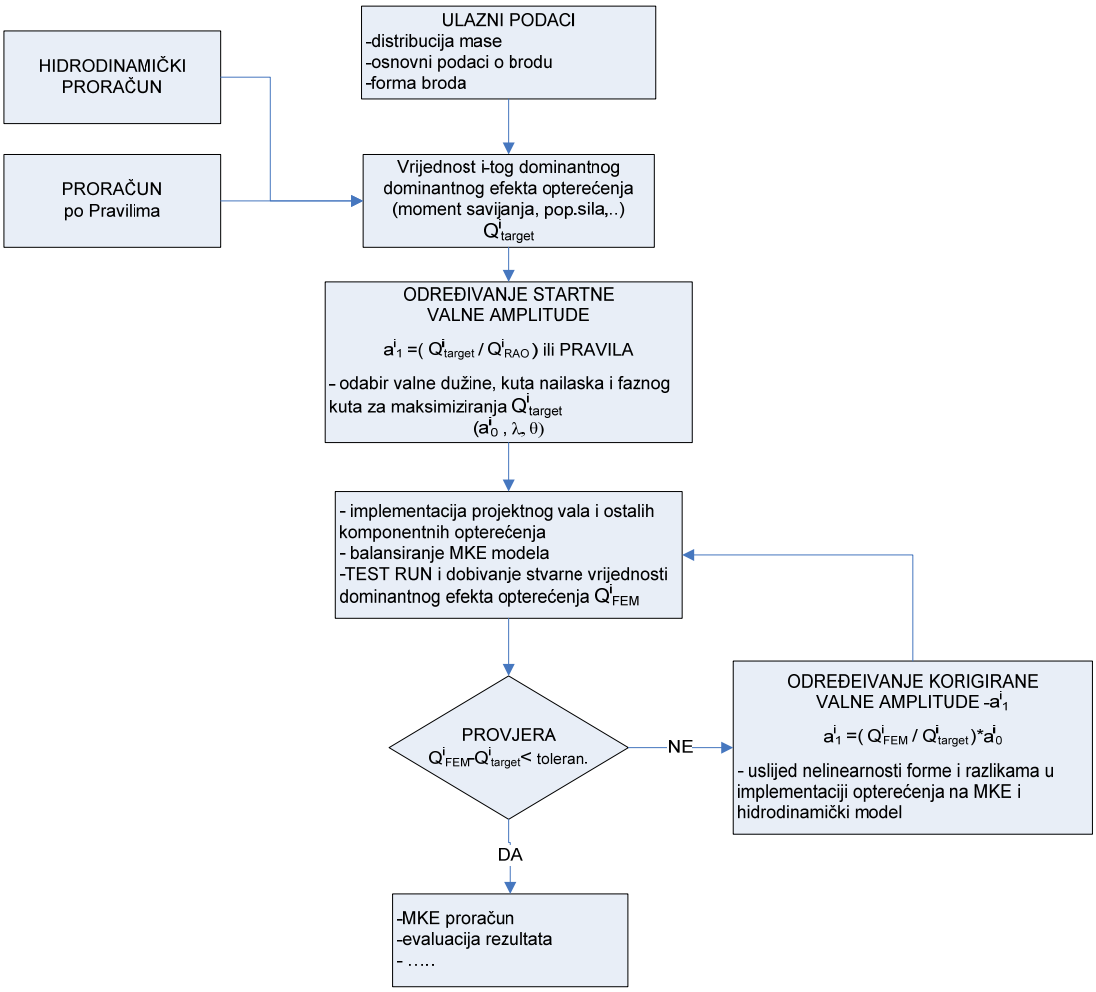
Jedna od prednosti 3-D *MKE* modela broda je što se izbjegava definiranje rubnih uvjeta sila/pomaka što je prisutno kod parcijalnih modela na njihovim krajevima. Model mora biti u ravnoteži tako da su reakcije u točkama u kojima je fiktivno oslonjen zanemarivo male. Ukupni moment savijanja (statički i valni), te poprečne sile direktno proizlaze iz distribucije statičkih i dinamičkih komponenata opterećenja.

Komponente opterećenja i način na koji su implementirani kod opterećivanja 3-D *MKE* modela broda dani su u nastavku, Te se sam postupak temelji na preporukama ta direktnu kalkulaciju klasifikacijskog društva Bureau Veritas, [96].

Statičko opterećenje idealizirane konstrukcije obično je potrebno uvećati i prilagoditi težini lakog broda prema knjizi Trima i stabiliteta [97], za razmatrani slučaj krcanja. Forma *MKE* modela prilično vjerno slijedi stvarnu formu broda, a razlike u istisnini do 2% smatraju se prihvatljivima. Distribucija hidrostatičkog tlaka direktno je definirana gazom broda i program je automatski implementira. Opterećenje po palubama obično se eksplicitno zadaje u formi tlaka ili masenog opterećenja. U slučaju zadanog vektora ubrzanja mnogo pogodnije je zadati opterećenje u obliku produkta distribucije masa i odgovarajuće akceleracije. Za konkretni brod to je opterećenje je definirano prema stanju krcanja iz knjige Trima i Stabiliteta. Težina tereta u tankovima proizlazi iz volumena tanka i gustoće tekućine. Mase glavnog stroja i krupnije opreme definiraju se na svojim točnim pozicijama kao koncentrirane mase.

Valno inducirana opterećenja i implementacija na 3-D *MKE* model cijeloga broda

Prilikom modeliranja valnog opterećenja treba imati na umu da se ono generira s mnogo više nesigurnosti nego strukturni model. Direktne metode proračuna raznih valnih opterećenja zbog nesigurnosti modeliranja još uvijek daju velika rasipanja u rezultatima čak i vertikalnog valnog momenta savijanja . Za praktičnu implementaciju valnih opterećenja na 3-D *MKE* model moguće je koristi razne metodologije koje koriste direktni proračun valnih opterećenja, te njegovu implementaciju na *MKE* model cijelog broda, npr. Metoda projektnih valova [96] koristi elemente deterministčkog pristupa u određivanju ekvivalentnog projektnog vala kojim će se opteretiti *MKE* model, te je korištena u ovoj studiji, vidi Sl. 8.26. Dominantni efekt opterećenja tj. ciljana vrijednosti kojim je opterećen *MKE* model je ukupni (valni + statički) vertikalni moment savijanja za pregibno stanje, gdje je valni moment savijanja izračunat prema Pravilima BV – *Zahtjev S11* [98].



Sl. 8.26. Shematski prikaz procedure određivanja amplitude projektnog vala

Ekvivalentni projektni val je pravilni sinusoidalni val (s uključenim dodatkom za *Smithov* efekt) sljedećih karakteristika: valna duljina - λ, visina vala - h_w (dvostruka vrijednost amplitude vala), faza vala - ξ_C, definira poziciju brijega od ishodišta. Potrebno je istaknuti da, zbog nelinearnosti forme (odstupanje stjenki boka iznad vodne linije od okomitosti) posebno na krajevima, stvarni valni moment savijanja se u pregibnom stanju smanjuje, a u progibnom povećava. Vrijednost vertikalnog valnog momenta dobivenog prema izrazu iz *Zahtjeva S11* ipak nije skalirana za izračunati faktor već je ukupna vrijednost vertikalnog momenta savijanja reducirana prema Pravilima [96].

Dinamička komponente ubrzanja također su izračunate prema Pravilima (Bureau Veritas, 2007b) te su kao akceleracijski vektor implementirani u model.

Od dva stanja krcanja (definiranih u Knjizi trima i stabiliteta) formirana su tri kritična slučaja opterećenja prema zahtjevima Pravilima [98].

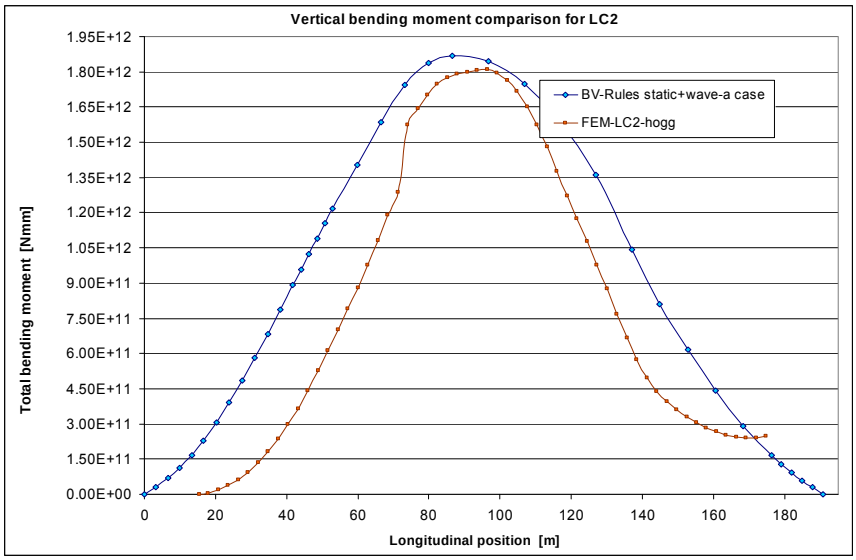
- LC1- Stanje krcanja_4 (Puno opterećenje +100% zaliha)-bez dinamičkog faktora i s maksimalnim pregibnim momentom
- LC2 – Stanje krcanja_4 (Puno opterećenje +100% zaliha)-s dinamičkim faktorom i s maksimalnim pregibnim momentom
- LC3 - Stanje krcanja_5- artificijelno stanje krcanja gdje su svi kamioni koncentrirani oko glavnog rebra i izazivaju minimalni pregibni statički moment savijanja, te valni progibni moment

Sažeta tablica za kritične slučajeve opterećenja prikazana je u Tbl. 8.7

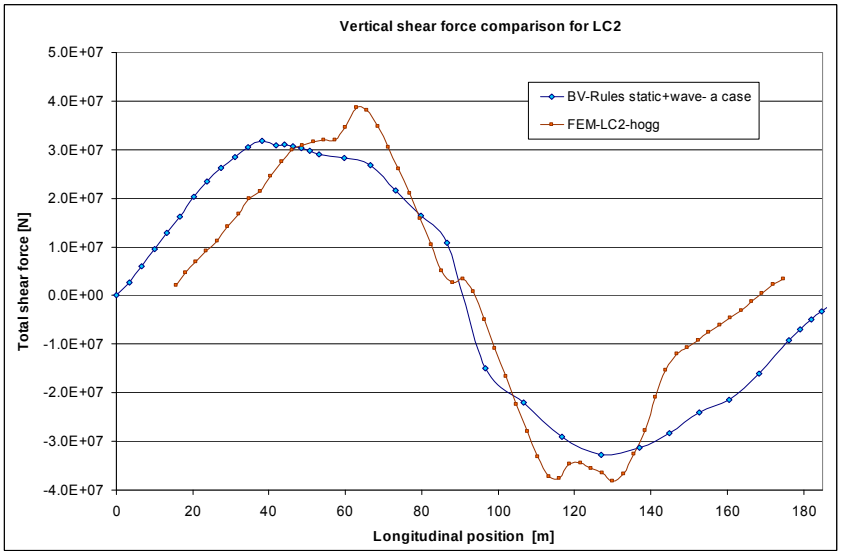
Tbl. 8.7 Slučajevi opterećenja i glavne komponentne

	LC1	LC2	LC3
	"a" $_{HOGG}$	"b" $_{HOGG}$	"a" $_{SAGG}^{*}$
M_{STATIC} [Nmm]	1.220E+12	1.220E+12	5.000E+11
$M_{W,V\text{-}RULE}$ [Nmm] (*0.675)	6.158E+11	6.158E+11	-8.421E+11
$M_{TOTAL,V\text{-}RULE}$ [Nmm]	1.836E+12	1.836E+12	-3.421E+11
$M_{TOTAL,V\text{-}MAESTRO}$ [Nmm]	1.856E+12	1.843E+12	-3.523E+11
$M_{H\text{-}RULE}$ [Nmm]	N/A	N/A	N/A
$M_{H\text{-}MAESTRO}$ [Nmm]	N/A	N/A	N/A
$p_{CL\ BOTT}$ [N/mm^2]	0.1014	N/A	0.0463
$p_{CL\ BOTT\text{-}MAESTRO}$ [N/mm^2]	0.1124	0.1310	0.0462
a_{HEAVE} [m/s^2]	N/A	1.690	N/A
* - artificial loading condition – minimum hogging			

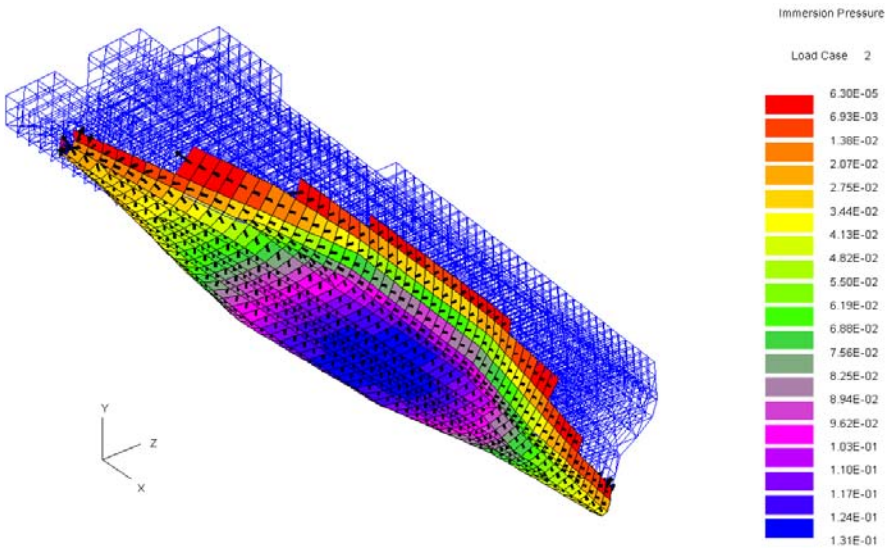
Na slikama i prikazane su usporedbe ostvarenog i prema Pravilima zahtijevanog momenta savijanja i poprečne sile, dok je na slici prikazana distribucija tlaka (statički+dinamički) na vanjsku oplatu za LC2.



Sl. 8.27. Postignuti (statički+valni) vertikalni moment savijanja i usporedba sa Pravilima za LC2



Sl. 8.28. Postignuta (statička+valna) vertikalna poprečna sila i usporedba sa Pravilima za LC2



Sl. 8.29. Distribucija vanjskog tlaka (statički +dinamički) duž modela za LC2

8.2.2. Projektni model RoPax-a

U nastavku će biti definirani optimizacijski modeli s tim da će se na početku formulirati osnovni optimizacijski model, nakon čega će slijediti opisi testiranih optimizacijskih modela i njihovih sekvenci rješavanja.

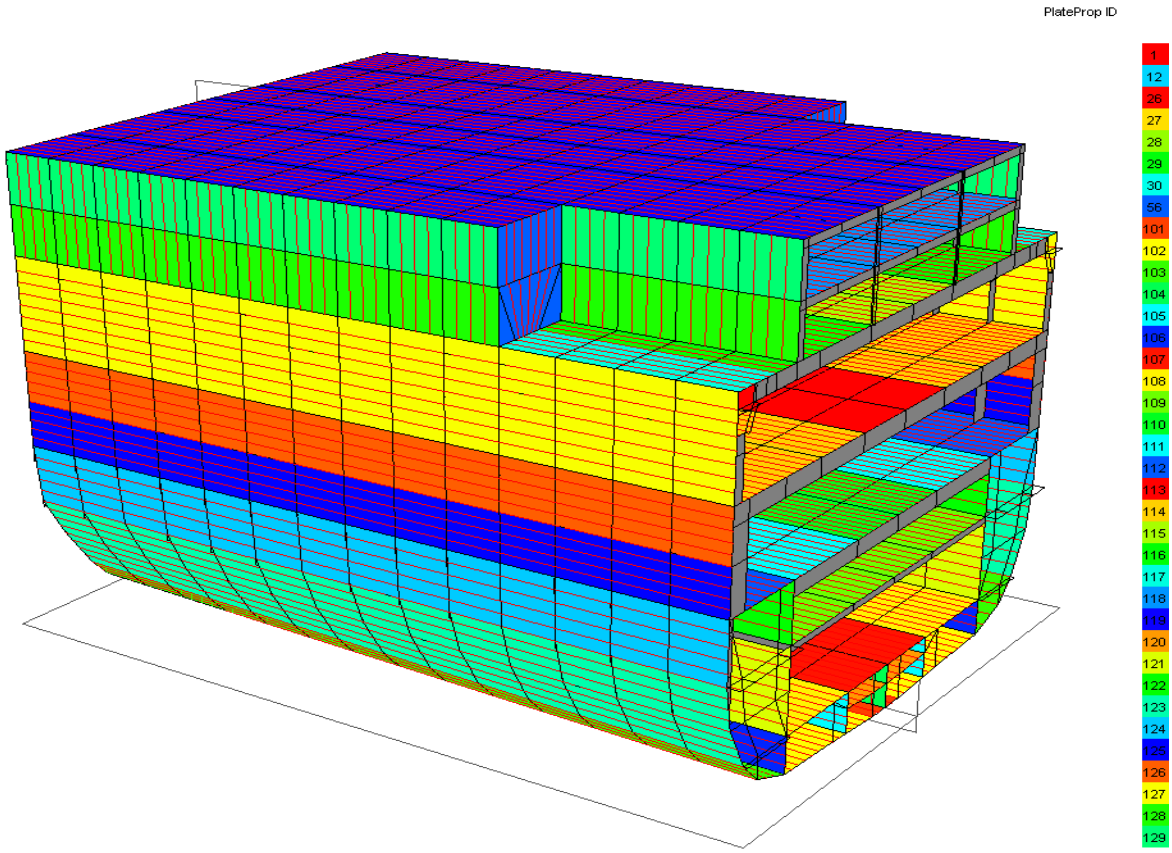
8.2.2.1. Formulacija osnovnog optimizacijskog modela

U nastavku je dana definicija osnovnih komponenti optimizacijskog problema bez dodatnih komponenti koje zahtjeva pojedina metoda rješavanja.

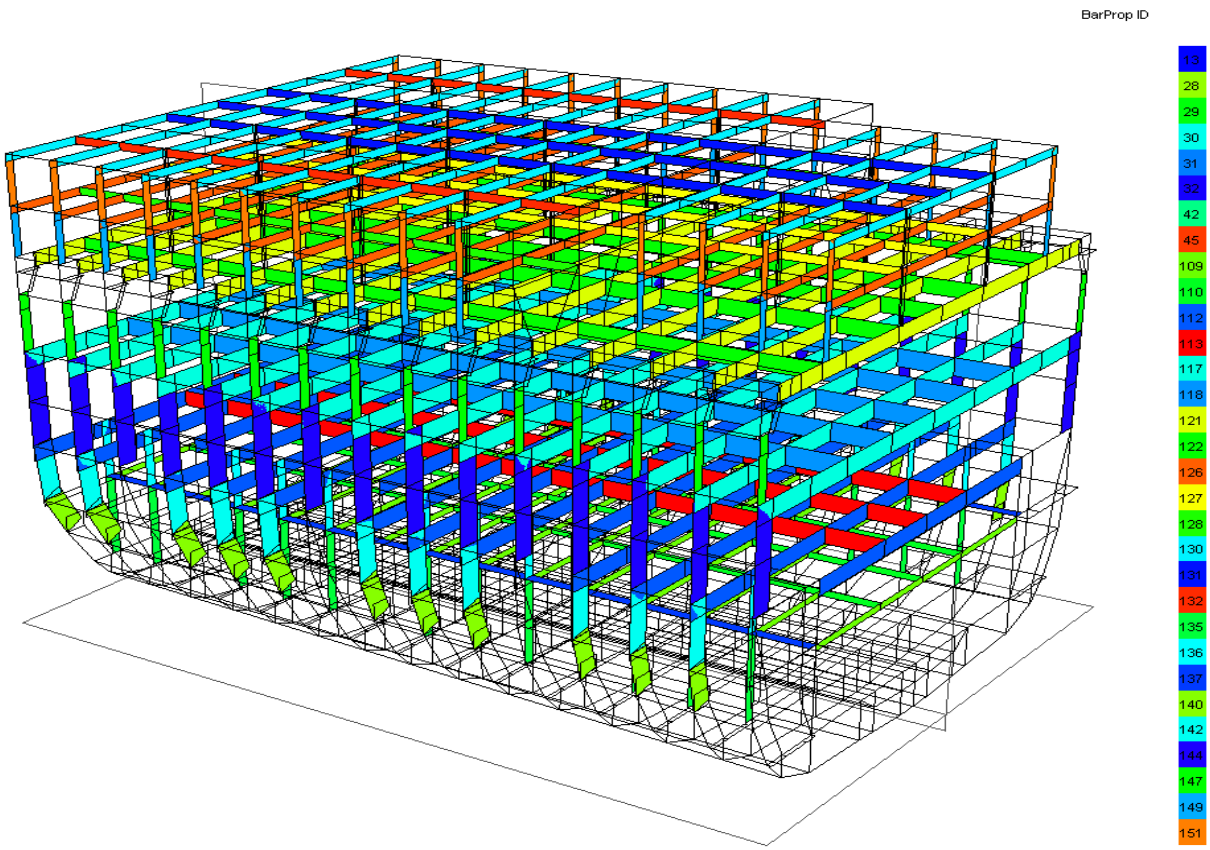
Projektne varijable

Projektne varijable su dimenzije elemenata strukture modula S1M1. Na Sl. 8.30 može se vidjeti kako su definirana svojstva (properties) konačnih elemenata koji modeliraju konstrukciju broda a koja su kroz generički model spojena s projektnim varijablama. Na Sl. 8.32 mogu se vidjeti osnovne podstrukture (s imenima na engleskom jeziku kako bi se mogli pratiti na modelu analize) na poziciji glavnog rebra RoPax koje su definirane direktno koristeći informacije o svojstvima konačnih elemenata.

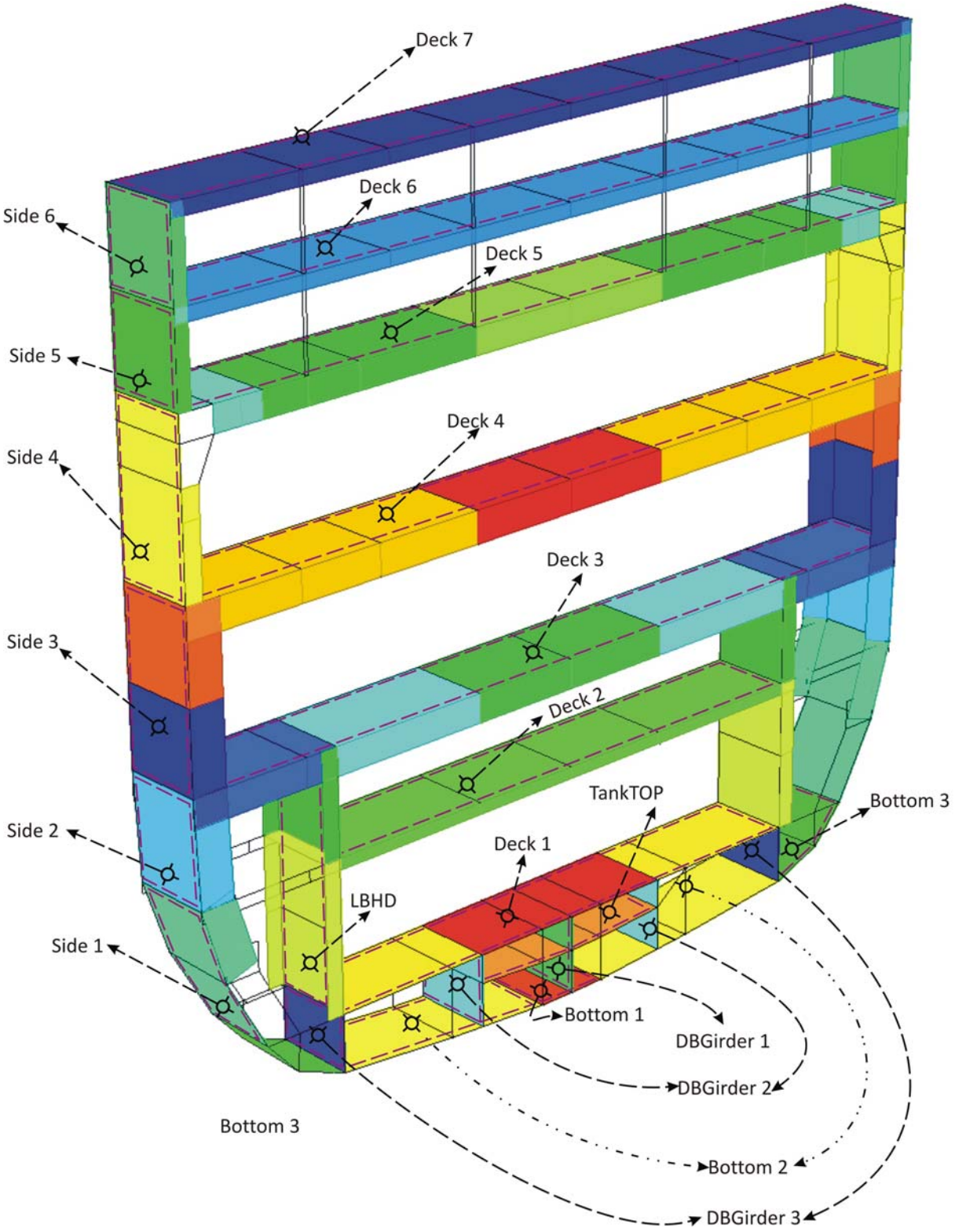
Definicija gornjih i donjih granica varijabli na modulu S1M1 može se vidjeti na Tbl. 8.8 do Tbl. 8.10.



Sl. 8.30. Definirana svojstva ljuskastih konačnih elemenata na kontrolnoj optimizacijskoj konstrukciji



Sl. 8.31. Definirana svojstva grednih konačnih elemenata na kontrolnoj optimizacijskoj konstrukciji



Sl. 8.32. Osnovne podstrukture glavnog na poziciji glavnog rebra RoPax-

Tbl. 8.8 Definicija donjih i gornjih granica varijabli oplate (plate) i ukrepa (*stiffeners*)

S1M1 Substructure Names	Plate thickness				BBS (NS)				Stiffeners HP			
	Item	P 0	Min	Max	Item	P 0	Min	Max	Item	P 0	Min	Max
Keel	1	15,5	14,0	19,0	21	640	600	900	41	HP 300x11	240x10	300x14
Bottom Inner	2	13,0	10,5	15,0	22		600	900	42	HP 300x11	240x10	300x14
Bottom Outer + Bilge	3	13,0	11,0	15,0	23	650	600	900	43	HP 280x11	220x10	280x13
Inner Bott Gird. Dwn	4	14,0	10,0	18,0	24	500	600	900	44	FB 150 X 10	100 X 6	200 X 15
Inner Bott Gird. Upp	5	14,0	8,5	18,0	25	493	400	700	45	HP 280x11	220x10	280x13
Inner Bott LBHD	6	11,0	10,0	15,0	26	670	600	900	46	HP 140x7	220x10	300x14
HFO tank top	7	13,0	11,0	18,0	27	640	600	900	47	HP 320x12	240x10	320x14
Deck 1 Inner	8	11,0	9,5	18,0	28	640	600	900	48	HP 260x10	220x10	260x13
Deck 1 Outer	9	13,0	9,5	18,0	29		600	900	49	HP 280x11	260x12	280x13
Side 1	10	12,0	10,0	16,0	30	Multi spac.	600	900	50	HP 260x10	200x9	260x13
LBHD D1-D2	11	13,0	11,5	17,0	31	Multi spac.	600	900	51	HP 280x11	200x9	280x13
LBHD D2-D3	12	11,0	9,0	15,0	32	650	600	900	52	HP 260x10	160x7	260x13
Deck 2	13	6,0	4,0	10,0	33	640	600	900	53	HP 100x7	140x7	100x8
Deck 3 Inner + Outer	14	13 (11.0)	10,0	15,0	34	640	600	900	54	HP 240x10	200x12	240x12
Side 2 + Side 3	15	11,0	7,5	17,0	35	Multi spac.	400	700	55	HP 140x8	120x6	160x8
Side 4	16	9 AH	7,0	12,0	36	700	400	700	56	HP 140x7 AH	80x5	140x9
Shear Strake	17	9 AH	FIX		37	Multi spac.	400	700	57	HP 140x7 AH	FIX	
Deck 4 Inner + Outer	18	9.5 AH	10,0	12,0	38	Multi spac.	400	700	58	HP 260x10	180x8	260x13
Deck 5 Inner + Outer	19	7,0	5,0	12,0	39	Multi spac.	400	700	59	HP 140x8	120X7	140x9
Superstructure	20	6,0	5,0	12,0	40	Multi spac.	600	900	60	HP 120x6	60x5	140x9

Tbl. 8.9 Definicija donjih i gornjih granica varijabli jakih uzdužnih nosača (*longitudinal girders*)

S1M1 Substructure Names	Longitudinal girder												
	HGW				TGW			BGF			TGF		
	Item	P 0	Min	Max	P 0	Min	Max	P 0	Min	Max	P 0	Min	Max
Deck 2	1	380	100	400	6	5	10	150	50	250	10	5	20
Deck 3 Inner + Outer	2	970	600	1200	10	5	15	150	50	600	30	5	30
Deck 4 Inner + Outer	3	1070	300	1100	10	5	12	150	25	200	30	5	30
Deck 5 Inner + Outer	4	880	500	1200	10	5	12	180	25	250	12	5	40
Superstructure	5	460	200	400	7	5	10	100	25	150	10	5	12

Tbl. 8.10 Definicija donjih i gornjih granica varijabli jakih poprečnih nosača (*transverse frames*)

S1M1 Substructure Names	Transverse Frame												
		HFW			TFW			BFF			TFF		
	Item	P 0	Min	Max	P 0	Min	Max	P 0	Min	Max	P 0	Min	Max
Side 1	1	1220	700	1300	10	5	12	200	50	500	20	5	25
LBHD D1-D2	2	600	400	650	12	5	12	200	50	250	20	5	25
LBHD D2-D3	3	600	400	650	12	5	12	200	50	250	20	5	25
Deck 2	4	380	100	500	6	5	10	200	50	250	15	5	20
Deck 3 Inner + Outer	5	970	600	1200	12	5	15	400	50	600	30	5	30
Side 2 + Side 3	6	1220	800	1300	10	5	12	600	200	600	30	5	30
Side 4	7	620	400	650	10	5	12	200	50	250	20	5	26
Shear Strake													
Deck 4 Inner + Outer	8	1070	400	1200	12	5	12	600	50	600	30	5	30
Deck 5 Inner + Outer	9	880	600	1200	7	5	14	350	150	450	20	5	40
Superstructure	10	460	200	400	7	5	10	100	25	150	10	5	12

Ograničenja

Zadana ograničenja su lokalni faktori podobnosti konstrukcije definirani u Tbl. 8.3 koji se provjeravaju programskim sustavom MAESTRO, a njihov opis moguće je pronaći u [95]

Tbl. 8.11 Ograničenja – kriteriji podobnosti konstrukcije RoPax-a

No.	Description	γ
1	Panel Collapse - Stiffener Flexure	1.04
2	Panel Collapse - Combined Buckling	1.04
3	Panel Collapse - Membrane Yield	1.224
4	Panel Collapse - Stiffener Buckling	1.04
5	Panel Yield - Tension Flange	1.04
6	Panel Yield - Tension Plate	1.224
7	Panel Yield - Compression Flange	1.04
8	Panel Yield - Compression Plate	1.224
9,10	Panel Serviceability - Plate Bending	1.0
11	Panel Failure - Local Buckling	0.9
12	Girder Collapse Tripping	1.04
13	Girder Collapse Compression in Flange	1.04
14	Girder Collapse Compression in Plate	1.04
15	Girder Yield Compression in Flange	1.224
16	Girder Yield Compression in Plate	1.224
17	Girder Yield Tension in Flange	1.224
18	Girder Yield in Tension in Plate	1.224
19-21	Frame Collapse, Plastic Hinge	1.6
22-24	Frame Yield, Compression in Flange	1.224
25-27	Frame Yield, Tension in Flange	1.224
28-30	Frame Yield, Compression in Plate	1.224
31-33	Frame Yield, Tension in Plate	1.224

Projektni atributi

Atributi korišteni u ovom primjeru su masa i dvije lokalne mjere sigurnosti (Tbl. 8.4).

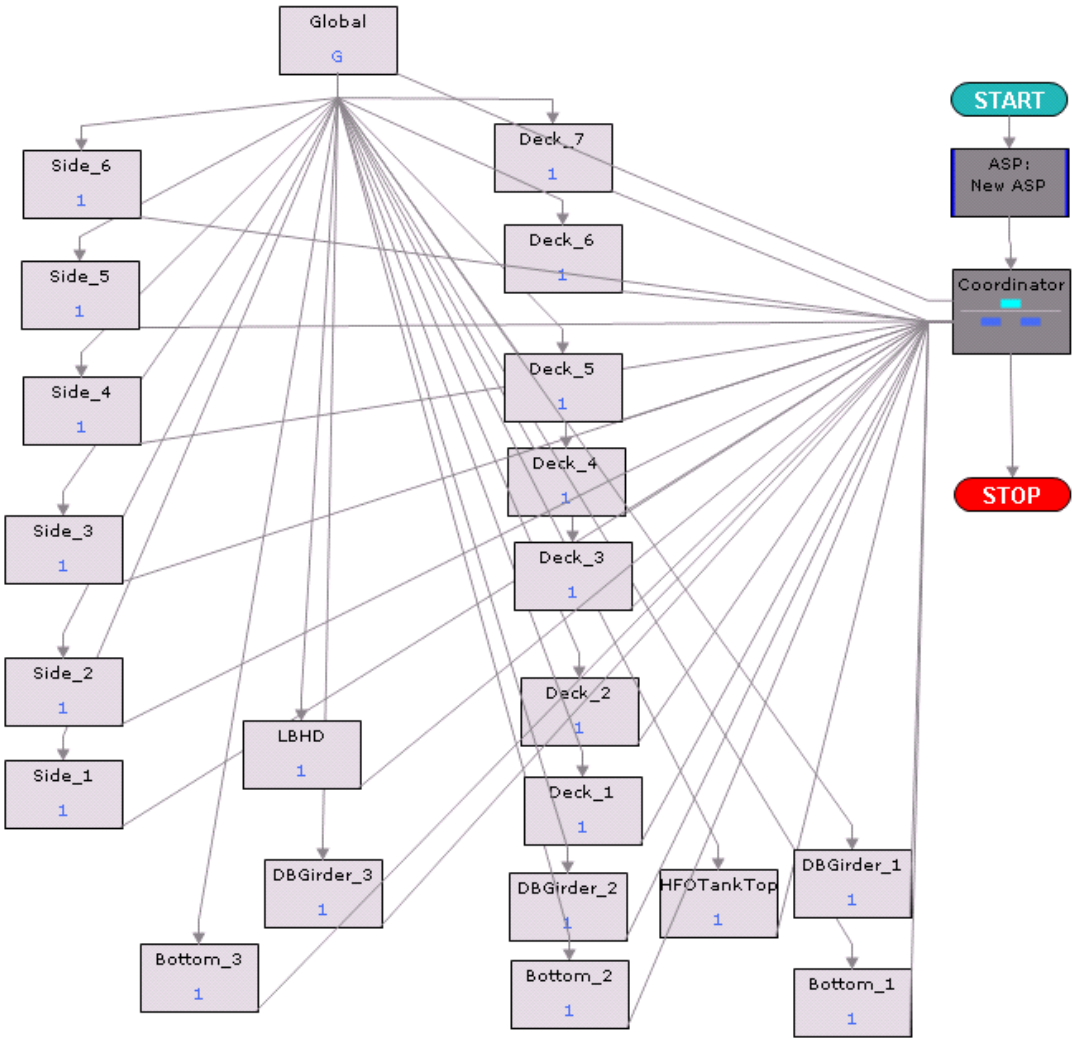
Tbl. 8.12 Atributi – mjere kvalitete konstrukcije RoPax-a

Oznaka	Tip	Komentar	Mjerna jedinica
WGT	Minimizirati	Masa konstrukcije	kg
GMean	Maksimizirati	$g_{<0.05}$	
GMin	Maksimizirati	$g_{\min 5\%}$	

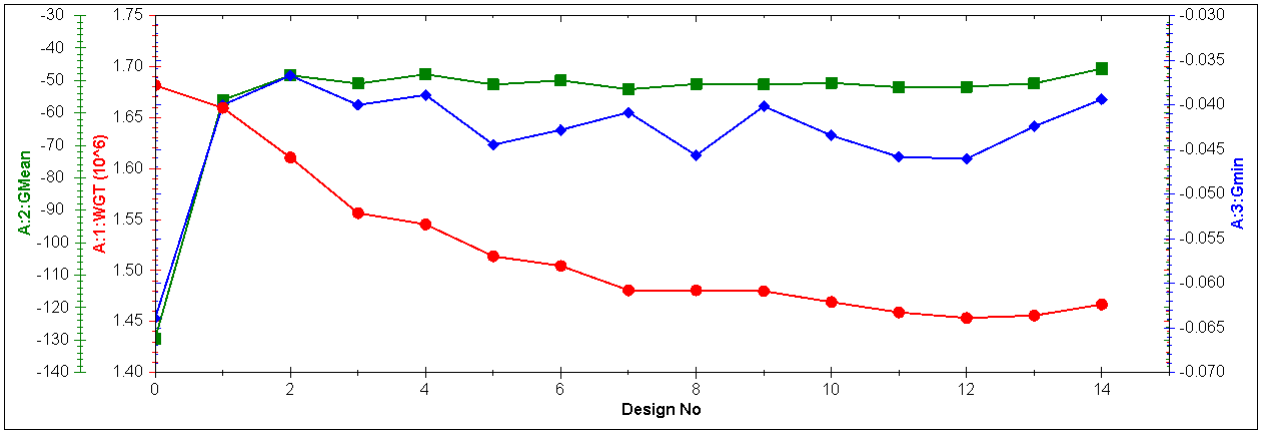
8.2.2.2. Optimizacija RoPax-korištenjem kažnjavanja udaljavanja deskriptora potproblema od stanja za koje je izračunato polje pomaka

Kako bi se testirala ova ideja korišten je novorazvijeni dio OCTOPUS Designera za dekompoziciju i koordinaciju dekomponiranog problema. Korištena je ATC metoda s tim da je globalni problem zapravo nije pravi optimizacijski problem već samo problem analize definiranih komponenti projektnog problema. Služi isključivo za pokretanje MKE analize te kao drugi podsustav pri definiranju ograničenja konzistentnosti na deskriptore podstruktura.

Optimizacijski potproblemi na podstrukturama imaju nametnuto kažnjavanje udaljenosti od stanja za koje je izračunato polje pomaka čime se sprečava osciliranje parnih i neparnih ciklusa usklađivanja potproblema s novim poljem pomaka.



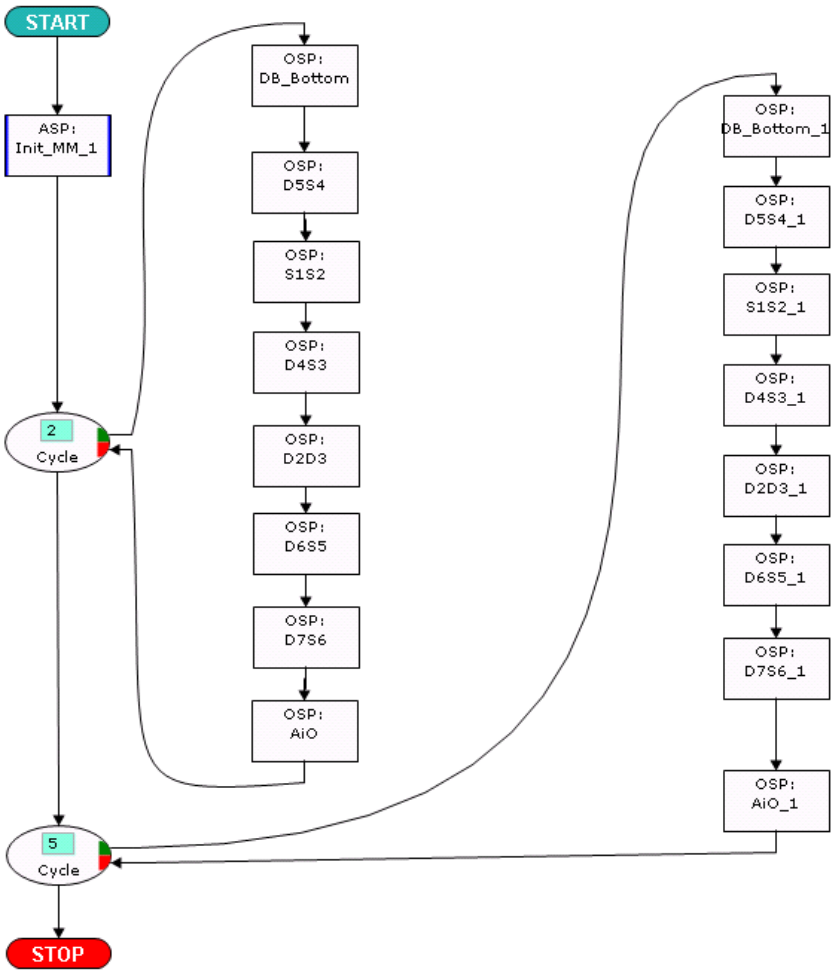
Sl. 8.33. Definicija dekompozicijskog problema RoPax-a s virtualnim globalnim sustavom



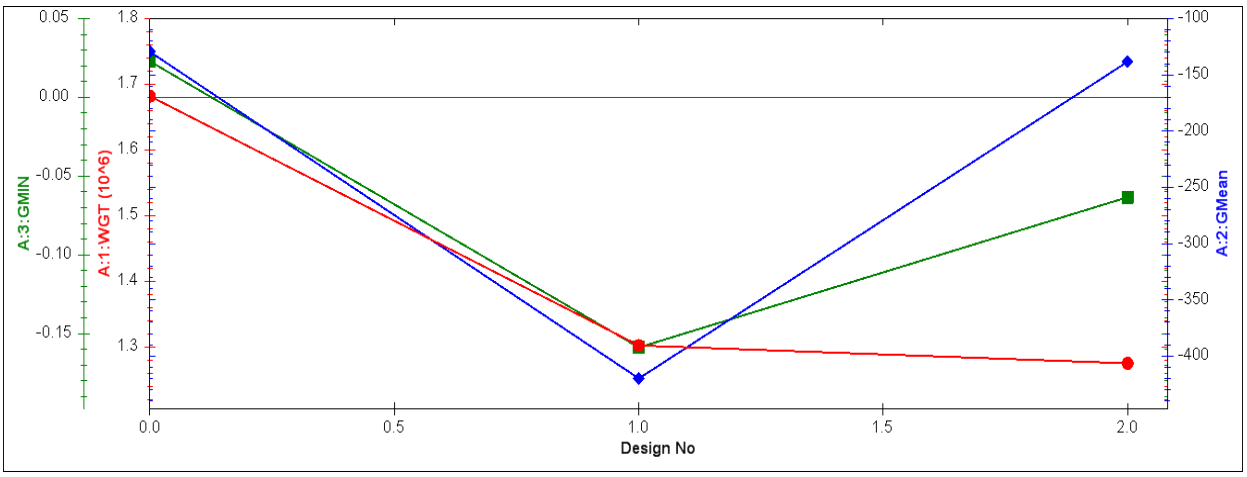
Sl. 8.34. Ostvarene vrijednosti atributa RoPax-a kroz iteracije (bilježeno u A_1)

8.2.2.3. Optimizacija RoPax-a-korištenjem surogat modela odziva

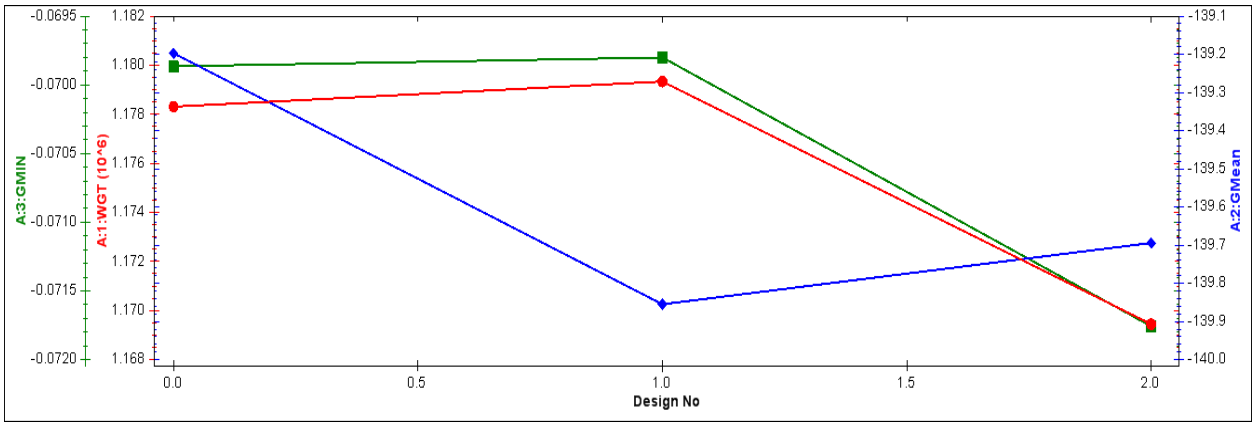
Na Sl. 8.35 prikazana je sekvenca rješavanja problema korištenjem surogat modela odziva. Korišteni surogat model generiran je pomoću 500 pokusa određenih LHS metodom (MATLAB funkcija `lhsdesign` s opcijom `corr`). Primjenom kompozitnih grednih parametara, broj parametara je smanjen na 81 (sa 237). Za sve odzive je korišten čisti kvadratni polinomski surogat model. Ostvareni rezultati kroz iteracije nalaze se na Sl. 8.36



Sl. 8.35. Sekvenca rješavanja problema RoPax-a sa surogat modelima odziva



a) Ostvarene vrijednosti atributa nakon dva ciklusa optimizacije s globalnim surogat modelom

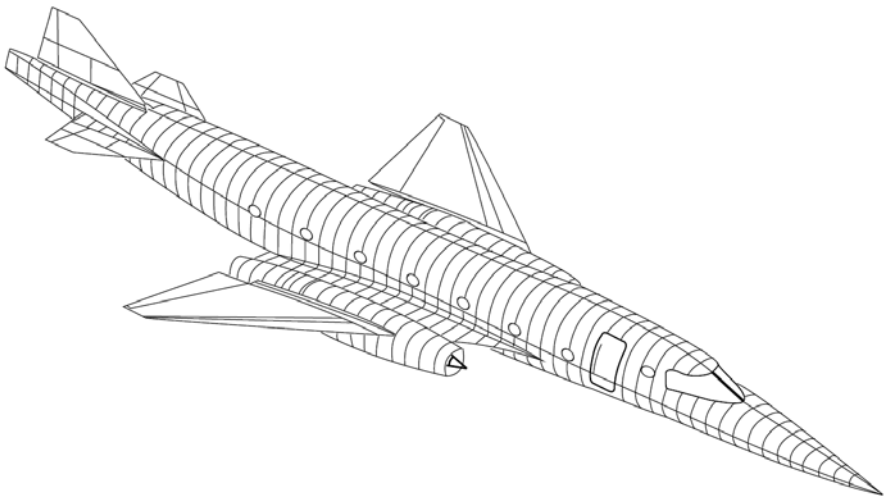


b) Ostvarene vrijednosti atributa nakon tri ciklusa optimizacije s lokalno umjerenim surogatom

Sl. 8.36. Ostvarene vrijednosti atributa RoPax-a kroz iteracije

8.3. Konceptualno projektiranje nadzvučnog poslovnog zrakoplova – SBJ

Primjer konceptualnog projektiranje nadzvučnog zrakoplova preuzet je iz [99]. Originalna verzija primjera s nešto detaljnijim opisom modula analize može se pronaći u [100]. Primjer se neko vrijeme nalazio na službenoj listi testnih primjera (verzija 1996-2007) za multidisciplinarne probleme laboratorija MODEL (Multidisciplinary Optimization and Design Engineering Laboratory) sveučilišta Buffalo, NY http://www.eng.buffalo.edu/Research/MODEL/mdotestsuite.html, gdje je klasificiran kao problem klase 3/3 tj. problem s velikim brojem varijabli i ograničenja te računalno zahtjevan za izvršavanje. Osim navedenih radova ovaj problem, odnosno njegove modificirane verzije korištene su u brojnim radovima npr. , no međusobne usporedbe rezultata nisu moguće jer optimizacijski modeli nisu identični, najčešće zbog različito definiranih gornjih i donjih granica projektnih varijabli. Zbog toga će se u ovom radu rezultati primjene dekompozicije i koordinacije ATC-om uspoređivati s rezultatima dobivenim jednorazinskom IDF metodom .



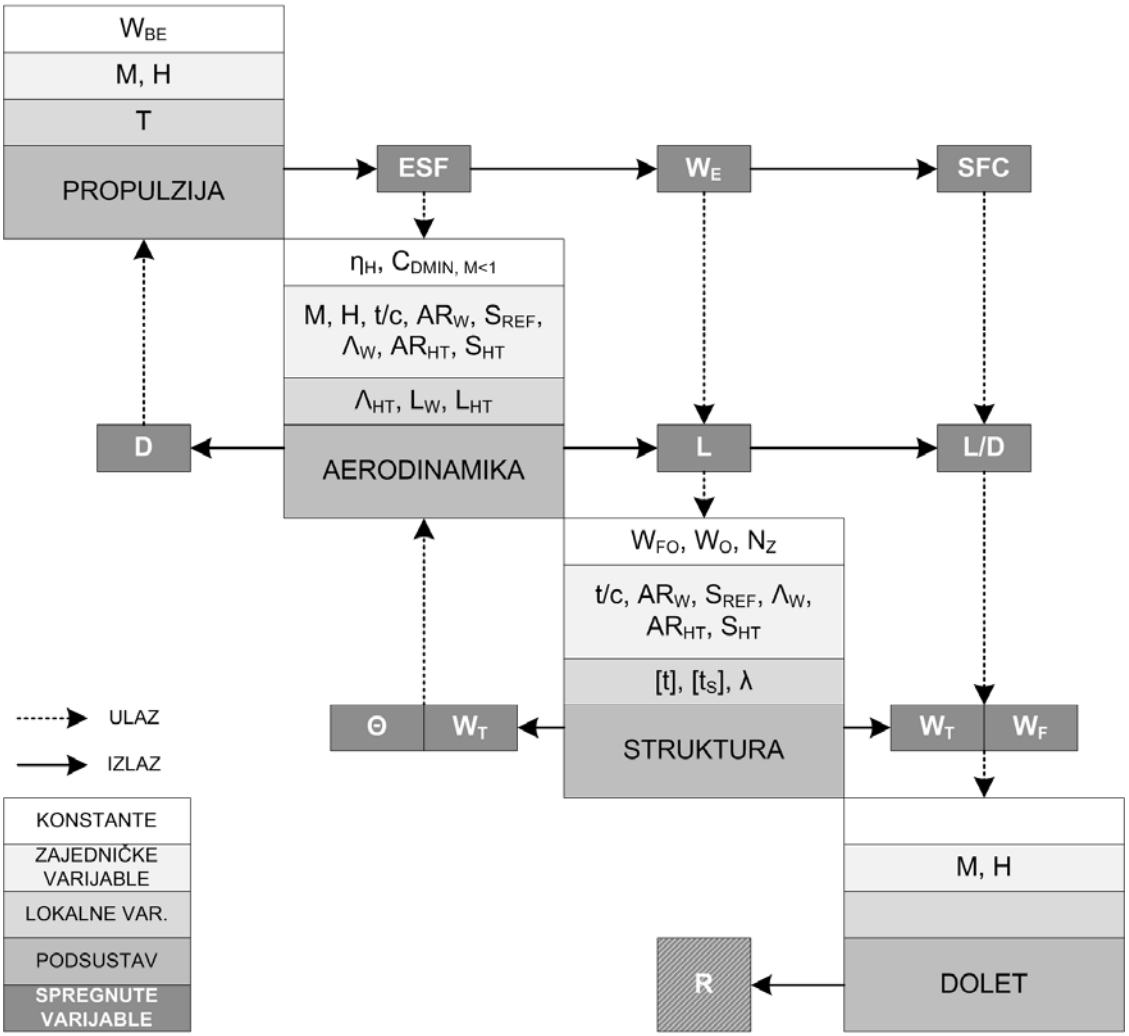
Sl. 8.37. Ilustrativni prikaz nadzvučnog zrakoplova [99]

8.3.1. Model analize

Kao što je već navedeno, model za analizu preuzet je u najvećoj mjeri iz [99] i [100], dok je prijevod na hrvatski jezik moguće pronaći u seminarskom radovima [101] i [102] Za proračun odziva modela analize nadzvučnog zrakoplova zadužena su 4 modula:

1. modul propulzije (PROP),
2. modul aerodinamike (AERO),
3. modul strukture (STR),
4. modul performansi (PERF).

Svaki modul (podsustav-PS) računa na temelju ulaznih deskriptora odgovarajuće izlazne veličine kao što je moguće vidjeti na dijagramu, s tim da se jasno vidi da su izlazne veličine jednog modula ujedno izlazne veličine drugog modula i obrnuto. Stoga je jasno da je za dobivanje rješenja jedne analize neophodno koristi sistemsku analizu tj. neku od iterativnih metoda.



Sl. 8.38. Tok podataka modela nadzvučnog zrakoplova

Radi lakšeg snalaženja u Tbl. 8.13 izdvojeno su prikazane samo oznake korištene u ovom primjeru. Važno je napomenuti da kako bi se osigurala usporedivost rezultata ostvarenih kroz optimizacije u ovom radu s radovima iz literature, nije korišten SI sustav oznaka, već originalno korišteni anglosaksonski sustav iz rada [99], kao što su to učinjeno i u drugim radovima koji su koristili ovaj primjer.

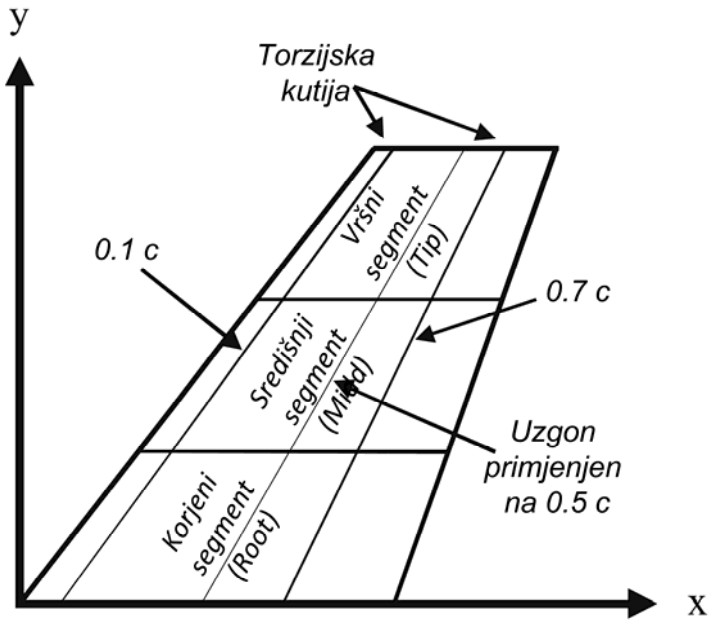
Tbl. 8.13 Popis korištenih oznaka u na primjeru projektiranja nadzvučnog poslovnog zrakoplova

Oznaka	Značenje oznake
AR_{HT}	vitkost horizontalnog repa
AR_W	vitkost krila
C_{Lw}	koeficijent uzgona krila
D	otpor zrakoplova
d_{HT}	udaljenost a/d centra repnih površina do težišta zrakoplova
d_W	udaljenost od a/d centra krila do težišta zrakoplova
ESF	faktor skaliranja motora
h	visina torzijske kutije krila
H	visina leta
l	dužina torzijske kutije krila
L	ukupni uzgon zrakoplova
L_{HT}	uzgon horizontalnih repnih površina
L_W	uzgon krila
M	Machov broj
Nz	faktor maksimalnog opterećenja
R	dolet (NM)
SBJ	nadzvučni poslovni zrakoplov (supersonic business jet)
SFC	specifična potrošnja goriva
S_{HT}	površina horizontalnog repa
S_{REF}	referentna površina krila
t	debljina kore sendviča panela torzijske kutije krila
T	potisak
t/c	relativna debljina aeroprofila
t_s	ukupna debljina sendviča panela torzijske kutije krila
V	brzina zrakoplova
W_E	masa motora
W_F	masa goriva
W_T	ukupna masa zrakoplova
λ	suženje krila
Λ_{HT}	kut strijele repnih površina
Λ_W	kut strijele krila

8.3.1.1. Modul za analizu strukture

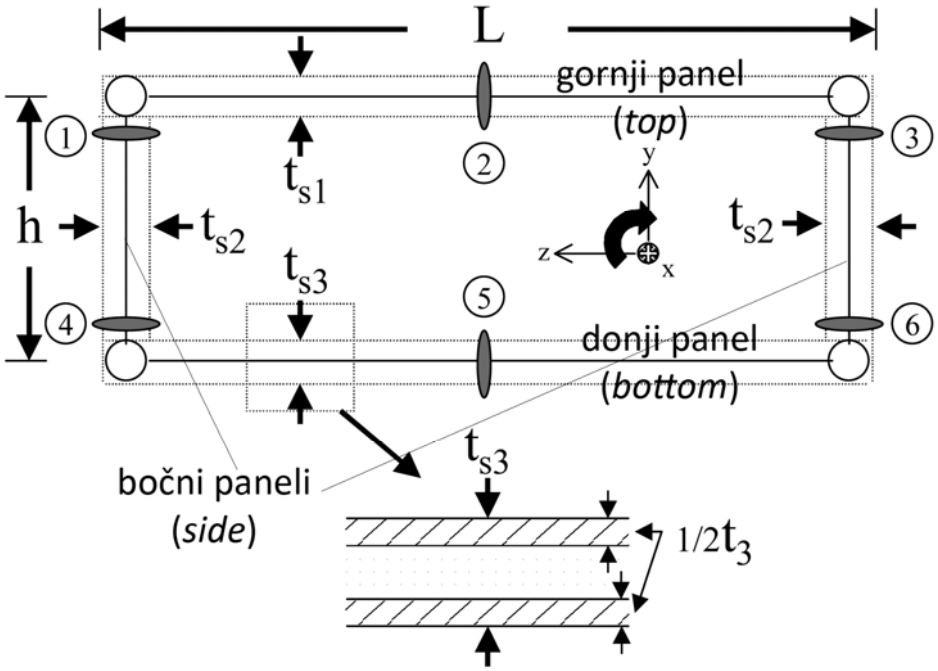
Modul za analizu strukture namijenjen za analizu strukturnog odziva i podobnosti te izračun mase strukture krila. U okviru ovog modula računa se i ostala masa zrakoplova pomoću integriranih regresijskih modela koji su generirani na osnovu podataka o sličnim zrakoplovima.

Strukturni model krila sastoji se od višedijelne torzijske kutije (TK) krila čiji odziv se računa korištenjem ELAPS modela (*Equivalent Laminated Plate Solution*), [103]. Rebra koja razdvajaju sekcije s nezavisnim dimenzijama torzione kutije okomita su na napadni rub krila u korijenu krila, na 1/3 polu-raspona i na 2/3 polu-raspona krila (Sl. 8.39). Proračun uvijanja krila uzima u obzir uvijanje uzrokovano torzijom i uvijanje uzrokovano savijanjem strelastog krila.



Sl. 8.39. Tri segmenta modela krila

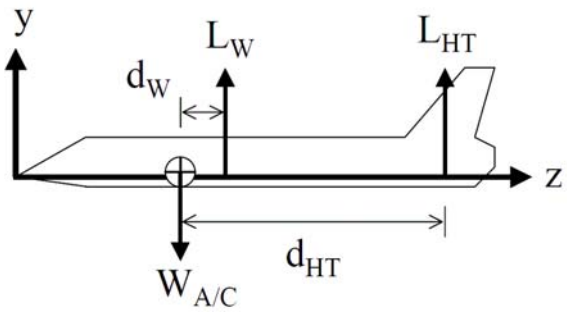
Torziona kutija modelirana je kao metalna sendvič konstrukcija kao što je prikazano na Sl. 8.40. Deskriptori torzione kutije su debljina kore sendviča t i ukupna debljina sendviča t_s za gornji i donji panel te za bočne panele. Analiza naprezanja i deformacije kao i ocjena lokalnih ograničenja izvijanja je održana na elementarnoj razini u skladu sa uobičajenim primjenama tankostjenih konstrukcija [104]. Kriteriji podobnosti strukture krila uključuju i dopuštene odnose dimenzija torzijske kutije koji osiguravaju tehnološku izvodivost konstrukcije krila. U radu [100] nalaze se detaljniji opis korištenih kriterija podobnosti, kao i još neki detalji vezani za proračun odziva.



Sl. 8.40. Osnovne dimenzije torzione kutije krila

8.3.1.2. Modul za analizu aerodinamike

Aerodinamički modul je zamišljen tako da uzima u obzir promjene otpora uzrokovane uvijanjem krila kao i promjenom mase zrakoplova. Kako bi se ostvarila fleksibilnost modula za primjenu na nekoj od metoda optimizacije analiza aerodinamike je svedena na najosnovnije jednadžbe.



Sl. 8.41. Prikaz sila na zrakoplov u ravnotežnom horizontalnom letu

Postavljanjem jednadžbi za ravnotežu sila i momenata uz poznatu ukupnu masu i kut uvijanja krila, te odabrane deskriptore geometrije zrakoplova i hvatišta sila računaju se ukupni uzgon i otpor zrakoplova te njihov odnos. Detaljni prikaz jednadžbi ovog modula nalazi se u[100].

8.3.1.3. Modul za analizu propulzije

Propulzijski modul je zasnovan na polinomskoj aproksimaciji podataka dvoosovinskog turboventilatorskog motora s miješanjem struja koji je prilagođen radu pri visokim Mach-ovim brojevima. Potisak *T* je neovisna ulazna varijabla koja može varirati po cijelom rasponu vrijednosti za svaku kombinaciju visine i Mach-ovog broja leta. Mach-ov broj i visina leta su druge ulazne varijable koje variraju između 1,4 i 1,8 (Mach-ov broj), te 30000 i 60000 ft (visina leta). Numeričkim eksperimentima je dokazano da unutar ovih granica aproksimacija daje rezultate visoke točnosti uz specifičnu potrošnju goriva (SFC) kao zavisnu varijablu [100].

Uvedena je postavka gasa do koje se dolazi tako da se neto potisak motora pri nekoj visini leta i Mach-ovom broju svede na bezdimenzijski oblik u odnosu na maksimalni potiska na danim uvjetima. Primjerice, postavka gasa od 0.01 označava rad u praznom hodu dok vrijednost 1 označava maksimalni potisak za odabrane uvjete leta. Kako maksimalni potisak varira sa uvjetima leta , uvedena je i druga polinomska aproksimacija kojom se i postavka gasa svodi na bezdimenzijski oblik.

Osim specifične potrošnje goriva, propulzijski modul kao izlaz ima i masu motora. Za određivanje mase motora koristi se faktora skaliranja motora (ESF):

$$ESF = \left(\frac{T}{T_{ref}} \right)_{h=0} \tag{8.3}$$

gdje je:

T- ukupni potisak

T_{ref} – potisak referentnog stanja oko kojeg su generirane polinomske aproksimacije.

Ukupna masa motora W_E računa se korištenjem izraza:

$$W_E = W_{E_{ref}} \cdot ESF^{1.05} \tag{8.4}$$

gdje je:

W_{Eref} – referentna masa motora.

Kako bi se što vjernije prikazao utjecaj motora na otpor zrakoplova, ESF također ulazi u modul aerodinamike gdje se računa otpor kućišta motora.

Kako bi se osiguralo da modul daje pouzdane izlazne vrijednosti, neophodno je ograničiti njegovu primjenu na područje oko referentnog stanja za koje su generirani aproksimacijski modeli. Stoga se interno u modulu provjerava zadovoljenje ovog uvjeta, a rezultat provjere se u formi kriterija podobnosti nalazi u listi izlaza ovog modula.

8.3.1.4. Modul za proračun performansi

Modul za proračun performansi jest vrlo jednostavan i zapravo se sastoji samo od proračuna doleta korištenjem Breguet-ove jednadžbe:

$$R = \frac{V\left(\frac{L}{D}\right)}{SFC}\ln\left(\frac{W_T}{W_T - W_F}\right) \tag{8.5}$$

8.3.2. Projektni model nadzvučnog zrakoplova

U nastavku će biti definirani optimizacijski modeli s tim da će se na početku formulirati osnovni optimizacijski model, nakon toga referentni IDF model koji će služiti za usporedbu, nakon čega će slijediti opisi optimizacijskih modela kojima će se testirati odrađeni aspekti primijenjenih i modificiranih metoda za projektiranje složenih sustava.

8.3.2.1. Formulacija osnovnog optimizacijskog modela

Ovaj optimizacijski model uključuje definiciju osnovnih komponenti optimizacijskog problema bez dodatnih komponenti koje zahtjeva pojedina metoda rješavanja. Drugim riječima, ovaj model obuhvaća komponente za MDF formulacije problema (vidi potpoglavlje 5.4.1.1).

Projektne varijable

Kao što su na Sl. 8.38 posebno prikazani zajednički i lokalni deskriptori, tako će i ovdje biti posebno prikazana definicija lokalnih i zajedničkih varijabli, iako se u MDF formulaciji sve varijable tretiraju na isti način. Prednost odvojenog prikazivanja različitih tipova varijabli bit će vidljiva na formulacijama višerazinskih modela. U

Tbl. 8.14 definirano je 8 zajedničkih varijabli, dok je u Tbl. 8.15 definirano 23 lokalne varijable čime su za ovaj problem definirana 31 varijabla.

Tbl. 8.14 Zajedničke varijable

Oznaka	Min	Max	Značenje	Mjerna jedinica
M	1,4	1,8	Mach-ov broj	-
H	45000	60000	visina leta	ft
t/c	0,05	0,12	relativna debljina aeroprofila	-
AR_W	2,5	5	vitkost krila	-
S_{REF}	200	800	referentna površina krila	ft^2
ΛW	40	70	strijela krila	deg
AR_{HT}	2,5	8,5	vitkost repa	-
S_{HT}	50	150	površina repa	ft^2

Tbl. 8.15 Lokalne varijable podsustava

Podsustav		Oznaka	Min	Max	Značenje	Mjerna jedinica
PROP		T	0,1	1,0	potisak	lb
AERO		Λ_{HT}	40	70	strijela repa	deg
		L_W	0,01	0,2	položaj krila	%MAC
		L_{HT}	1	3,5	položaj repnih površina	%MAC
STR		λ	0,1	0,4	suženje krila	-
STR	**Root/Midd/Tip**	$(t_s)_{TOP}$	0,1	9	ukupna debljina sendviča gornjeg panela TK	in
		$(t)_{TOP}$	0,1	4	debljina kore sendviča panela gornjeg panela TK	in
		$(t_s)_{Bottom}$	0,1	9	ukupna debljina sendviča donjeg panela TK	in
		$(t)_{Bottom}$	0,1	4	debljina kore sendviča panela donjeg panela TK	in
		$(t_s)_{Side}$	0,1	9	ukupna debljina sendviča bočnih panela TK	in
		$(t)_{Side}$	0,1	4	debljina kore sendviča panela bočnih panela TK	in

Projektni ciljevi

Kao i u originalnom primjeru kao cilj jest definiran samo dolet zrakoplova R (vidi Tbl. 8.17). Iz izraza (8.5) vidljivo je da odabir doleta za cilj zapravo izravno povezuje najvažnije izlaznih veličine svih disciplina (aerodinamike, strukture i propulzije) te se na taj način osigurava postizanje optimuma cjelokupnog sustava. No naravno odabir funkcije cilja mora biti prvenstveno vezan za specifikaciju misije naručitelja projekta osnivanja zrakoplova. U ovom primjeru to je isključivo maksimizacija doleta, no ukoliko to misija zahtjeva potrebno je kroz određeni način višekriterijskog projektiranja omogućiti generiranje višekriterijske funkcije cilja ili što je još poželjnije generiranje Pareto fronte s koje bi se tehnikama višekriterijskog projektiranja odabralo željeno rješenje.

Tbl. 8.16 Korišteni projektni cilj

Opt. Model	Podsustav	Oznaka	Tip	Težinski koeficijent	Značenje	Mjerna jedinica
SBJ-R	**PERF**	R	Maksimizirati	1	dolet	lb

Projektna ograničenja

Projektna ograničenja definirana na ovom primjeru uz osiguravanje funkcionalnosti zrakoplova, osiguravaju i da se matematički model koristi u okviru područja točnosti koje odgovara konceptualnom projektiranju. Sva ograničenja su direktno u modelu analize svedena na bezdimenzijski oblik te se moguće vrijednosti nalaze i intervalu [-1,1] s tim da su ograničenja zadovoljena ukoliko je vrijednost >= 0.

Tbl. 8.17 Projektna ograničenja

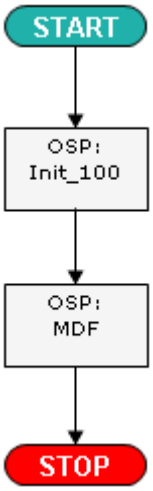
Podsustav		Ograničenje	Značenje	Komentar
PERF		g_R	minimalni dolet	R>= 2500 nm (samo u SBJ_MO)
AERO		g_Pg	gradijent tlaka krila	Pg<=1.1
AERO		g_Cl	omjer koef. uzgona	2Clh<Clw
PROP		g_Throt	postavka gasa	Throt< Throt_max
PROP		g_Temp	temperatura motora (TIT)	Temp< Temp_max
PROP		g_ESF	faktor skaliranja motora	0,5≤g_ESF≤1,5
PROP		g_We	masa motora	100≤g_We≤30000
STR		g_tw	uvijanje krila	0,2≤g_tw≤50
STR	**Root/Midd /Tip**	g_Y_1:6	dopuštena naprezanja na kontrolnim točkama TK	$\sigma \leq \sigma_{dop}$
		g_B_1:3,5	dopušteno izvijanje na kontrolnim točkama TK	$\sigma \leq \sigma_{krit}$
		g_T,S,B_sr	dimenzija sendviča	t<(ts+0.1t)
		g_Box	dimenzija kutije krila	0.5(ts+t)<h

8.3.2.2. MDF optimizacijski model

Kao što je gore navedeno, MDF formulacija problema odgovara osnovnom optimizacijskom problemu. Konzistencija spregnutih odziva izvedena je unutar samog modela analize iterativnim postupkom.

Sekvenca rješavanja optimizacijskog problema

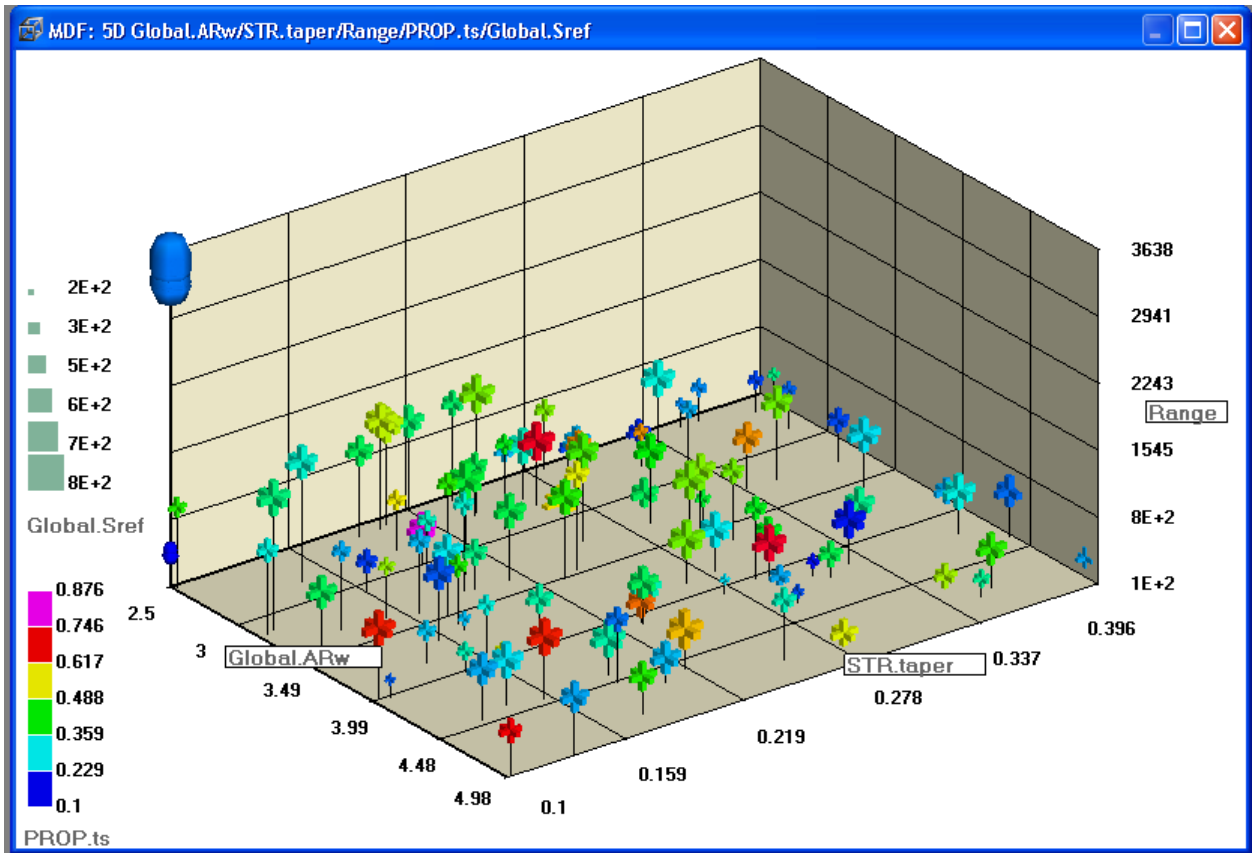
Za rješavanje zadanog optimizacijskog odabran je optimizacijski algoritam NLPQLP. Kao što je navedeno u potpoglavlju 3.2.2.1 ovaj algoritam spada u skupinu SQP algoritama. S obzirom da algoritam nema garantirano pronalaženje globalnog optimuma, već lokalnog, sam algoritam je startan s više slučajnih, približno podobnih startnih projekata.



Sl. 8.42. Definicija sekvence rješavanja za SBJ MDF problem u OCTOPUS Designeru

Rezultati SBJ-MDF-R modela

Rezultati ostvareni MDF formulacijom prikazani su na Sl. 8.43 zajedno s inicijalnih 100 rješenja. Od početnih 100, 99 je uspješno konvergiralo i zadovoljilo sva projektna ograničenja. Od tih 99 rješenja dva rješenja su konvergirala na izrazito inferiorni lokalni optimum ~600 nm .

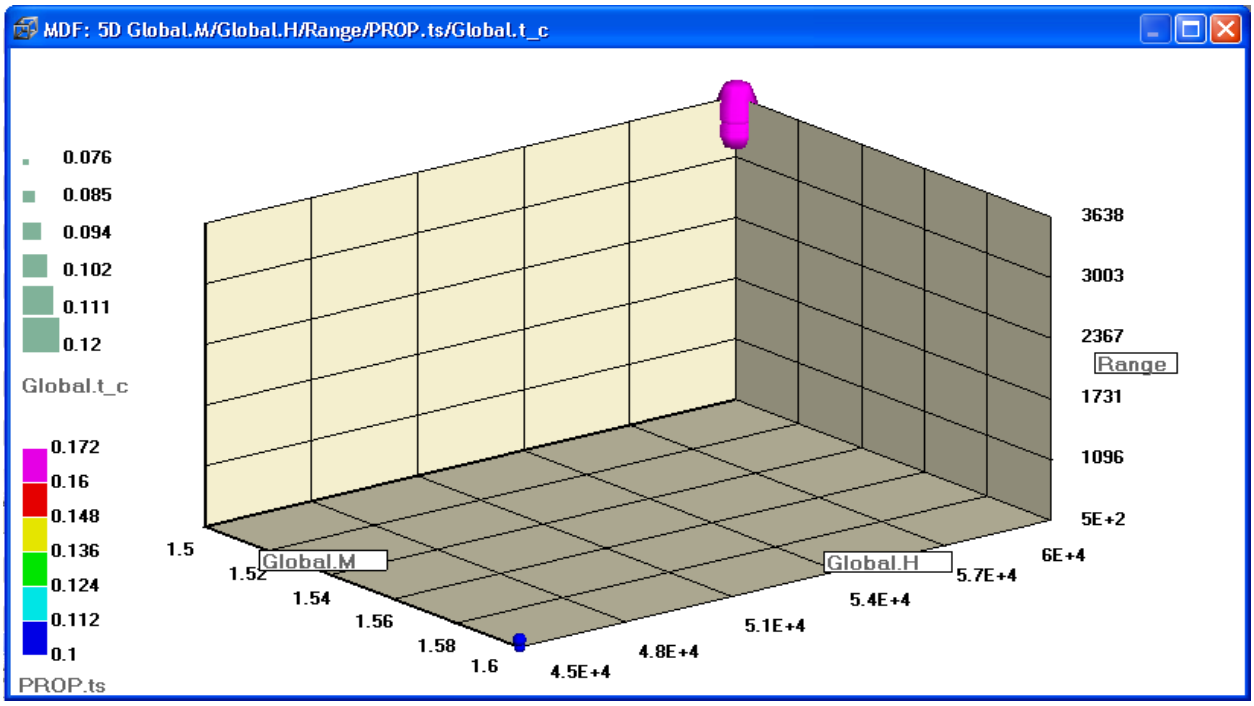


Sl. 8.43. Prikaz početnih 100 rješenja zajedno sa završnih 99 MDF rješenja

U Tbl. 8.18 nalaze se postignuti doleti grupirani po iznosu doleta za preostalih 97 projekata. Na Sl. 8.44 nalaze se prikazano 99 konvergiranih rješenja.

Tbl. 8.18 Postignuti doleti konvergiranih rješenja MDF formulacije grupirani po iznosu doleta

Broj projekata	Donja granica područja doleta	Gornja granica područja doleta
5	3279	3400
16	3400	3500
66	3500	3600
10	3600	3638
97	3279	3600



Sl. 8.44. Prikaz 99 konvergiranih rješenja MDF formulacije

8.3.2.3. IDF optimizacijski model

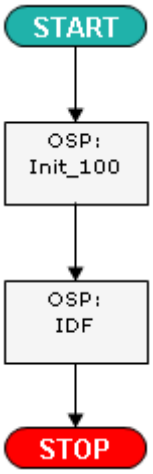
U IDF optimizacijskom modelu osim komponenti osnovnog optimizacijskog modela, optimizacijski model proširuje se spregnutim varijablama te pripadajućim ograničenjima jednakosti.

Tbl. 8.19 Spregnute varijable (IDF formulacija)

Deskriptor (Varijabla)	Izlazna vrijednost	Oznaka	Min	Max	Značenje	Mjerna jedinica
PROP	**AERO**	*D*	100	70000	otpor	lb
AERO	**PROP**	*ESF*	0,5	1,5	koeficijent dimenzioniranja motora	-
AERO	**STR**	*Θ*	0,2	50	kut uvijanja krila	deg
AERO, PERF	**STR**	W_T	5000	100000	ukupna masa	lb
STR	**PROP**	W_E	100	30000	masa motora	lb
PERF	**STR**	W_F	0,1	0,9	masa goriva	lb
STR	**AERO**	*L*	5000	100000	uzgon	lb
PERF	**AERO**	*L/D*	0,1	10	uzgon/otpor	-
PERF *	**PROP**	*SFC*	1	4	specifična potrošnja goriva	-

Sekvenca rješavanja optimizacijskog problema

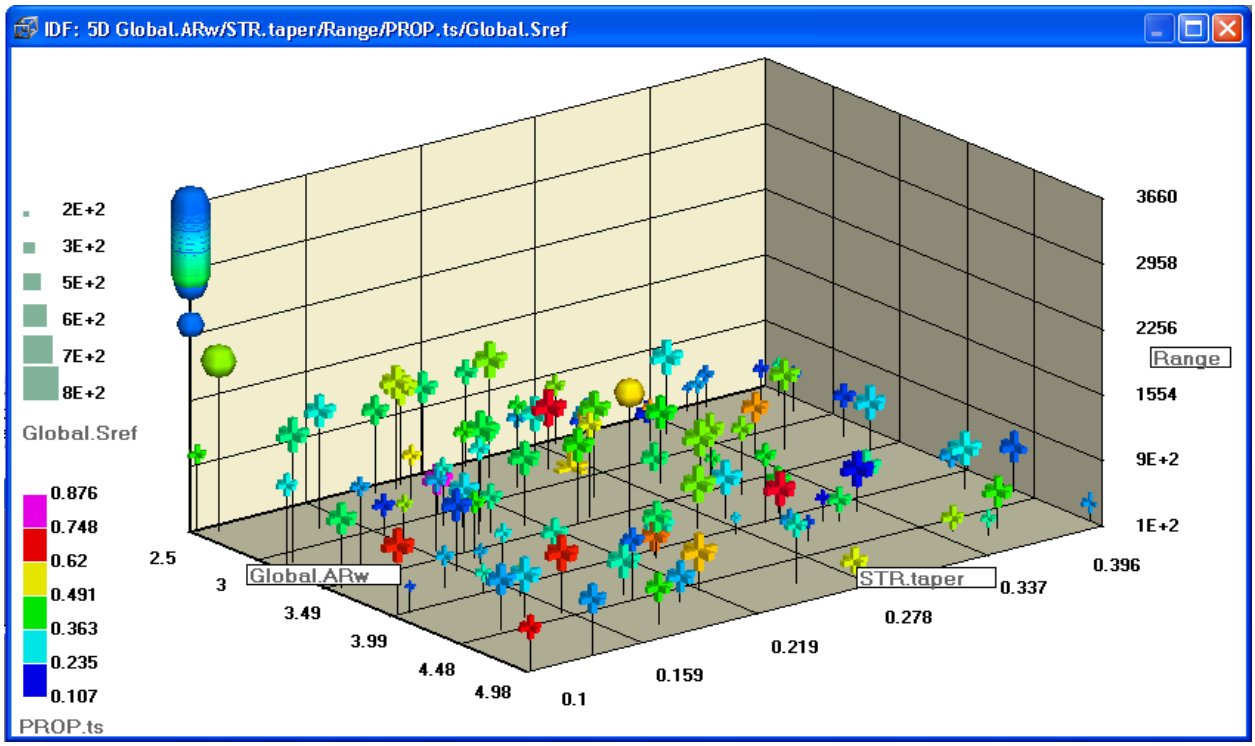
Za rješavanje zadanog optimizacijskog odabran je optimizacijski algoritam NLPQLP, pokrenut s istih početnih rješenja kao i kod MDF formulacije.



Sl. 8.45. Definicija sekvence rješavanja za SBJ IDF problem u OCTOPUS Designeru

Rezultati SBJ-IDF-R modela

Rezultati ostvareni IDF formulacijom prikazani su na Sl. 8.46 zajedno s inicijalnih 100 rješenja. Od početnih 100, 98 je uspješno konvergiralo i zadovoljilo sva projektna ograničenja i ograničenja konzistentnosti.



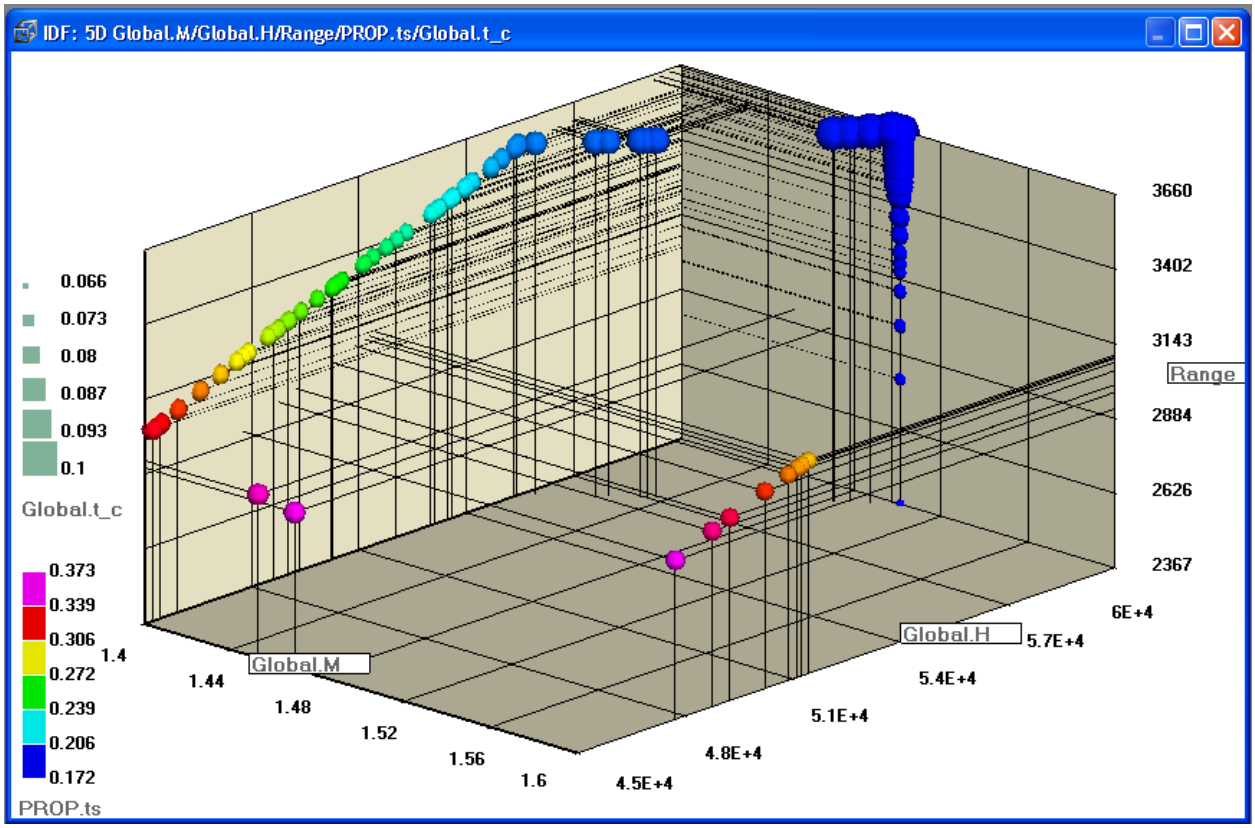
Sl. 8.46. Prikaz početnih 100 rješenja zajedno sa završnih 100 IDF rješenja

U Tbl. 8.20 nalaze se postignuti doleti grupirani po iznosu doleta. Na Sl. 8.47 nalaze se prikazana konvergirana rješenja. Može se zaključiti da je veliki broj rješenja konvergirao prema lokalnim optimumima, dok se oko 10 posto nalazi u području oko 3650 nm tj. jako blizu globalnog optimuma od 3660 nm.

Tbl. 8.20 Postignuti doleti konvergiranih rješenja IDF formulacije grupirani po iznosu doleta

Broj projekata	Donja granica područja doleta	Gornja granica područja doleta
8	2367	3000
58	3000	3500
17	3500	3600
15	3600	3660
98	2367	3660
9	3638	3660

Može se primijetiti da je u odnosu na MDF formulaciju mnogo više rješenja konvergiralo prema globalno suboptimalnim lokalnim optimumima, no da je istovremeno oko 10% rješenja IDF formulacije nešto bolje od najboljeg rješenja dobivenog MDF formulacijom. Vrijeme trajanja optimizacije obje formulacije je podjednako s tim da treba imati na umu da se sa stajališta sistemske analize radi o relativno jednostavnom problemu te je iterativna petlja unutar MDF formulacije konvergirala relativno brzo i robusno.



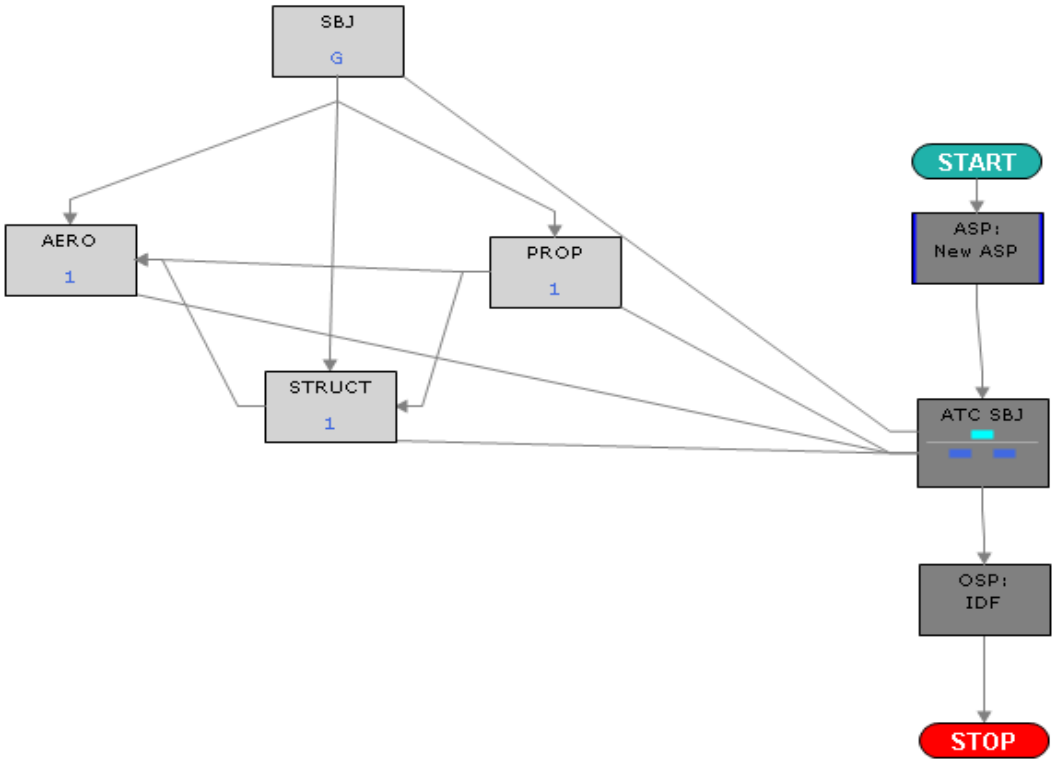
Sl. 8.47. Prikaz 98 konvergiranih rješenja IDF formulacije

8.3.2.4. Jednociljni projektni model dekomponiran na 2 nivoa s ATC koordinacijskom metodom – SBJ-R-2L-ATC

Iz prije danih rješenja dobivenih MDF i IDF formulacijom ustanovljeno je da globalni maksimum doleta ukupnog problema oko 3660 nm, s tim da postoji značajan broj lokalnih maksimuma u području 3300 do 3600 nm. U ovom potpoglavlju je ukupni problem definiran koristeći dekompoziciju prema proračunskim disciplinama s ciljem da se provjeri mogućnost korištenja dekompozicije i koordinacije u OCTOPUS Designeru. S obzirom na vremensku zahtjevnost rješavanja ovako definiranog problema u nastavku će biti dano rješenje startano sa samo jednog početnog projekta (prvo iz skupine početnih rješenja).

Na lijevoj strani Sl. 8.48 vidljiv je dekomponirani model nadzvučnog poslovnog zrakoplova, dok je na desnoj strani definirana sekvenca rješavanja s ATC koordinacijskom metodom. IDF formulacija problema se nalazi u sekvenci samo radi usporedbe ostvarenog doleta tom formulacijom za isti početni projekt. Varijable svakog podsustava dane su u Tbl. 8.14 i Tbl. 8.15, dok su ograničenja dana u Tbl. 8.17. Konzistentnost spregnutih podataka osigurava ATC metoda formulacijom danom izrazima (5.30) do (5.34) odnosno modifikacijom danom u (5.35).

Spregnuti podaci po podsustavima mogu se vidjeti na Sl. 8.49.



Sl. 8.48. Definicija sekvence rješavanja za – SBJ-R-2L-ATC problem u OCTOPUS Designeru

Decomposiotion systems:SBJ and AERO consystency

Consistency Pairs Short Info	
Num Pairs	4
First	SBJ
Second	AERO
Edit Consistency	Click to Edit!
Pair 0	DV: PERF.M; -> Par: AERO.M;
Pair 1	DV: PERF.H; -> Par: AERO.H;
Pair 2	Par: PERF.L_D; <- Rp: AEROOut.L_D;
Pair 3	DV: PERF.Wt; -> Par: AERO.Wt;

Decomposiotion systems:SBJ and PROP consystency

Consistency Pairs Short Info	
Num Pairs	3
First	SBJ
Second	PROP
Edit Consistency	Click to Edit!
Pair 0	DV: PERF.M; -> Par: PROP.M;
Pair 1	DV: PERF.H; -> Par: PROP.H;
Pair 2	Par: PERF.SFC; <- Rp: PROPOut.SFC;

Decomposiotion systems:SBJ and STRUCT consystency

Consistency Pairs Short Info	
Num Pairs	2
First	SBJ
Second	STRUCT
Edit Consistency	Click to Edit!
Pair 0	Par: PERF.Wf_Wt; <- Rp: STROut.Wf_Wt;
Pair 1	DV: PERF.Wt; <- Rp: STROut.Wt;

Decomposiotion systems:STRUCT and AERO consystency

Consistency Pairs Short Info	
Num Pairs	8
First	STRUCT
Second	AERO
Edit Consistency	Click to Edit!
Pair 0	Rp: STROut.twst; -> Par: AERO.twst;
Pair 1	Par: STR.L; <- Rp: AEROOut.L;
Pair 2	DV: STR.ARht; <-> DV: AERO.ARht;
Pair 3	DV: STR.ARw; <-> DV: AERO.ARw;
Pair 4	DV: STR.Sht; <-> DV: AERO.Sht;
Pair 5	DV: STR.Sref; <-> DV: AERO.Sref;
Pair 6	DV: STR.Sw_w; <-> DV: AERO.Sw_w;
Pair 7	DV: STR.t_c; <-> DV: AERO.t_c;

Decomposiotion systems:PROP and STRUCT consystency

Consistency Pairs Short Info	
Num Pairs	1
First	PROP
Second	STRUCT
Edit Consistency	Click to Edit!
Pair 0	Rp: PROPOut.We; -> Par: STR.We;

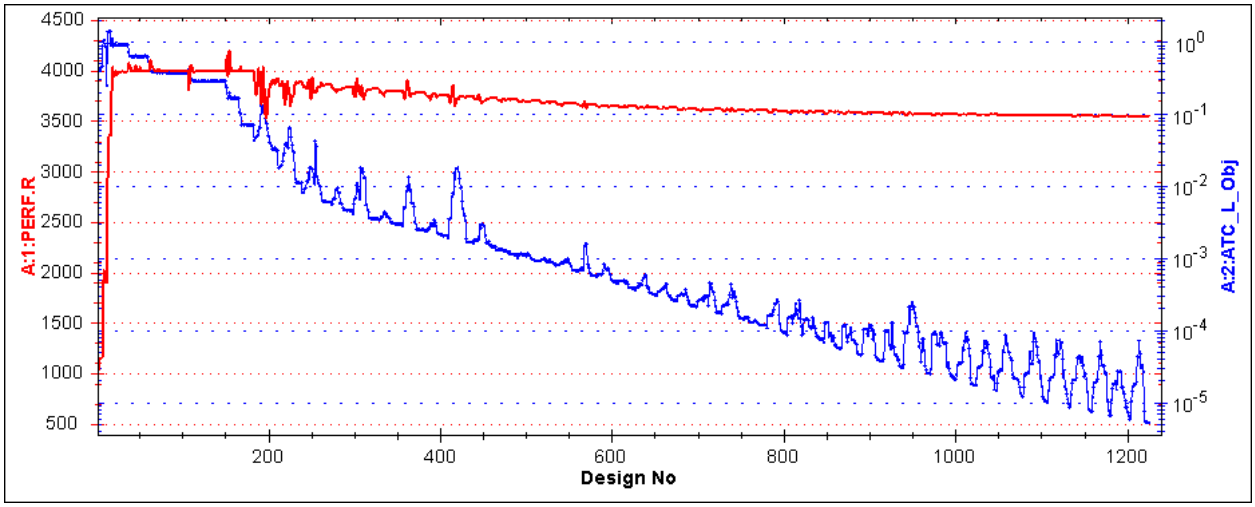
Decomposiotion systems:PROP and AERO consystency

Consistency Pairs Short Info	
Num Pairs	2
First	PROP
Second	AERO
Edit Consistency	Click to Edit!
Pair 0	Par: PROP.D; <- Rp: AEROOut.D;
Pair 1	Rp: PROPOut.ESF; -> Par: AERO.ESF;

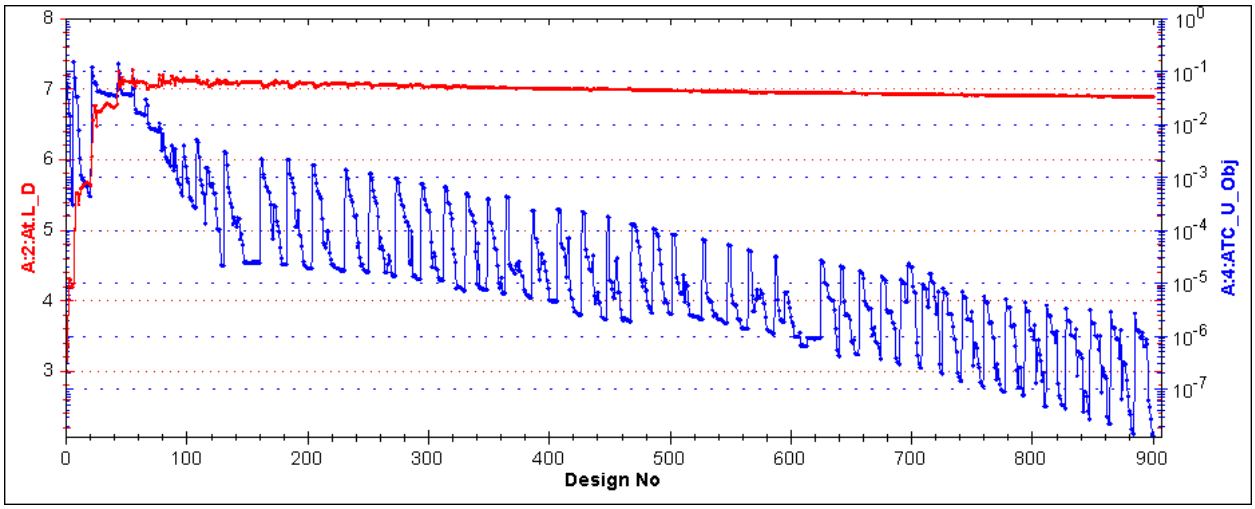
Sl. 8.49. Definicija spregnutih podataka između podsustava u OCTOPUS Designeru

Rezultati SBJ-R-2L-ATC modela

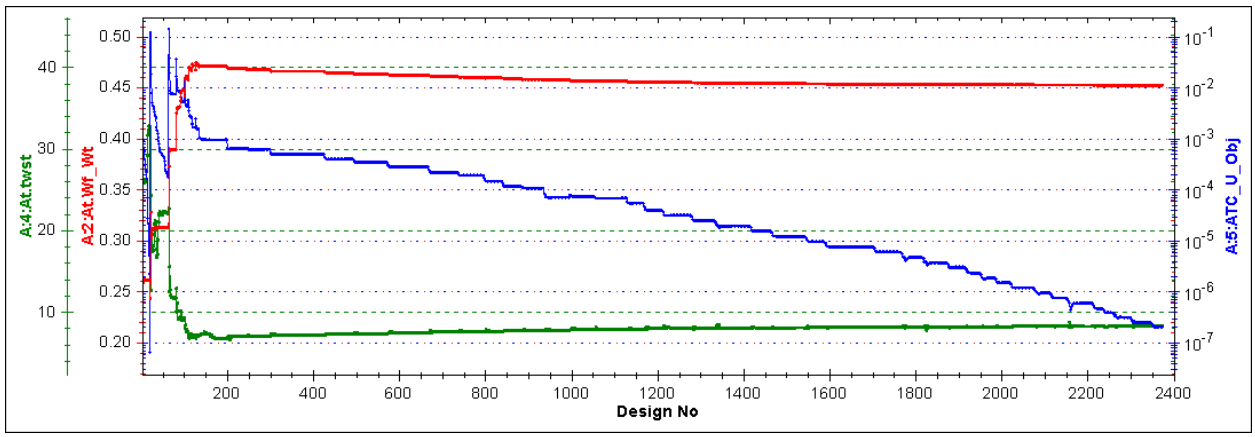
Ostvarena rješenja kroz 50 iteracija mogu se vidjeti na slijedećim slikama. Na slikama se može vidjeti kako je mjera konzistentnosti sustava, koja je u ATC metodi korištena kao cilj na svim podrazinama, konvergirala tijekom 50 iteracija. Napomena: svaka od ATC iteracija uključuje određeni broj iteracija optimizacijskog algoritma na tom podsustavu te je stoga broj rješenja na X osi aditivan kroz iteracije. S obzirom da svaki podsustav samostalno konvergira prema optimumu pojedine iteracije ATC algoritma, ukupni broj izračunatih rješenja jest različit za svaki sustav.



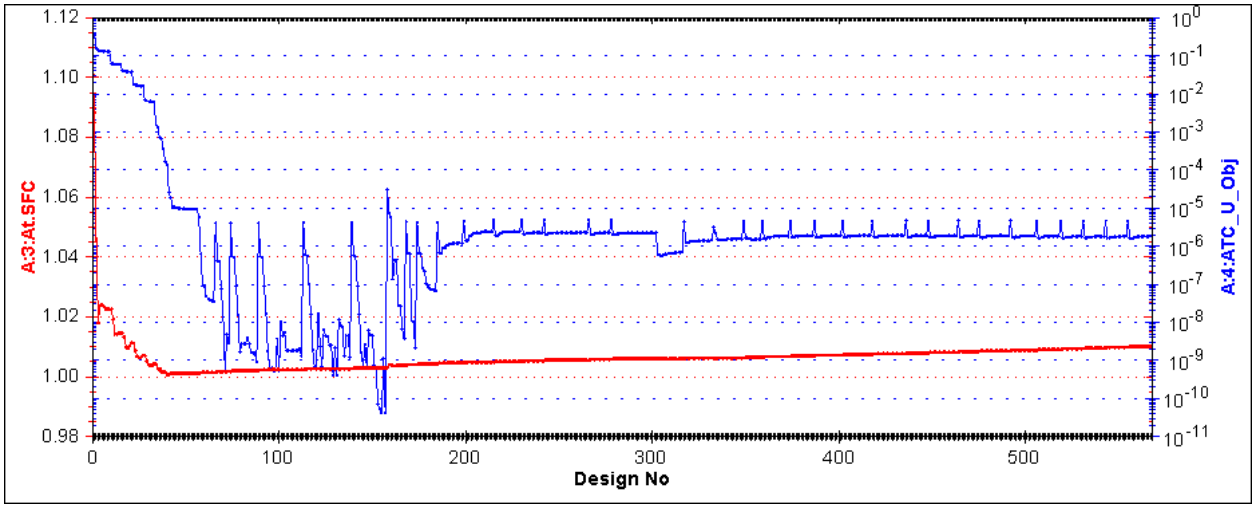
Sl. 8.50. Konvergencija globalnog sustava (SBJ-PERF)



Sl. 8.51. Konvergencija AERO podsustava



Sl. 8.52. Konvergencija STRUCT podsustava

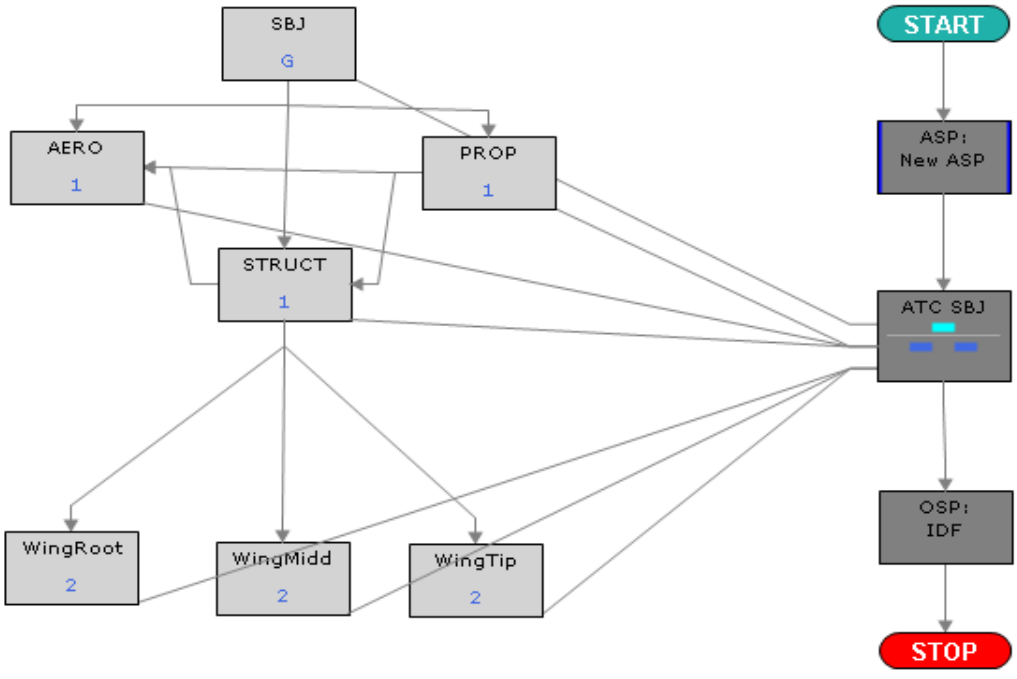


Sl. 8.53. Konvergencija PROP sustava

Ostvareno rješenje, koje je nakon 50 ATC iteracija konvergiralo na prihvatljivu razinu nekonzistentnosti globalnog sustava ($5E^{-5}$), ima dolet 3550 nm što znači da se radi o lokalnom optimumu u području u kojem je i većina MDF i IDF rješenja. Ukupno trajanje optimizacije je nekoliko puta duže (~ 15-30 minuta) nego kod MDF i IDF formulacija (~3-5 minuta) na istom računalu (AMD Athlon 3200+, 2GB RAM). Treba imati na umu da se pravi benefiti dekompozicije vide tek u paralelnom radu jer se sustavi kroz iteracije mogu istovremeno optimizirati, dok je globalni sustav zadužen za distribuiranje informacija podsustavima o ostalim podsustavima nakon svake iteracije. Isto tako, ovdje se radi o primjeru koji ima gotovo sve sprege koje postoje i u problemima iz inženjerske prakse, no sami moduli analize su relativno trivijalni. Izuzetak jest strukturni modul koji ima relativno složenu metodu proračuna globalnih pomaka krila ELAPS modelom [103].

8.3.2.5. Jednociljni projektni model dekomponiran na 3 nivoa s ATC koordinacijskom metodom – SBJ-R-3L-ATC

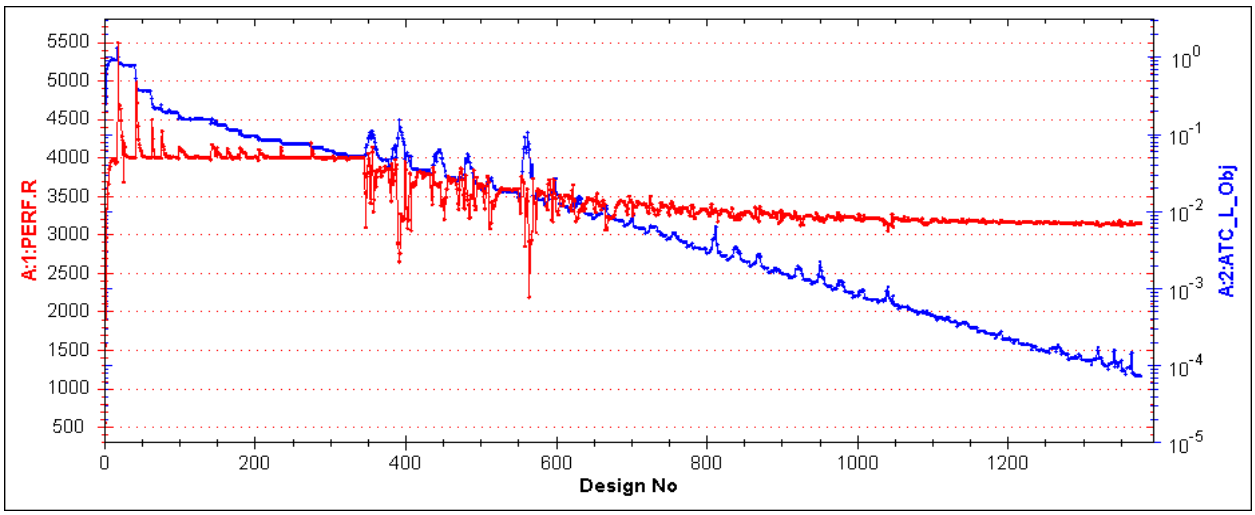
Ovaj projektni model je što se tiče prve dvije razine dekomponiran na isti način kao i model iz prethodnog potpoglavlja s tim da je strukturni podsustav (STRUCT) dodatno dekomponiran na globalni dio, koji je ostao u podsustavu STRUCT, i lokalne podsustave koji predstavljaju segmente strukture krila tj. podstrukture (vidi Sl. 8.39). Na Sl. 8.54 s lijeve strane se vidi dekomponirani model dok je s desne strane definirana sekvenca rješavanja s ATC koordinacijskom metodom. Vidljivo je da u ovom slučaju ATC metoda mora upravljati s 3 razine sustava, što ne predstavlja problem jer formulacija ove metode dopušta neograničen broj razina. Kao što je to simbolično prikazano na Sl. 8.54 SBJ sustav je na G (globalnoj) razini tj. razini 0 i spojen je s najgornjom linijom pridruživanja s ATC metodom, dok su podsustavi AERO, STRUCT i PROP na razini 1 te su spojeni sa srednjom linijom pridruživanja s ATC blokom. Podsustavi segmenata krila nalaze se na razini 2 te su spojeni najdonjom linijom pridruživanja s ATC blokom.



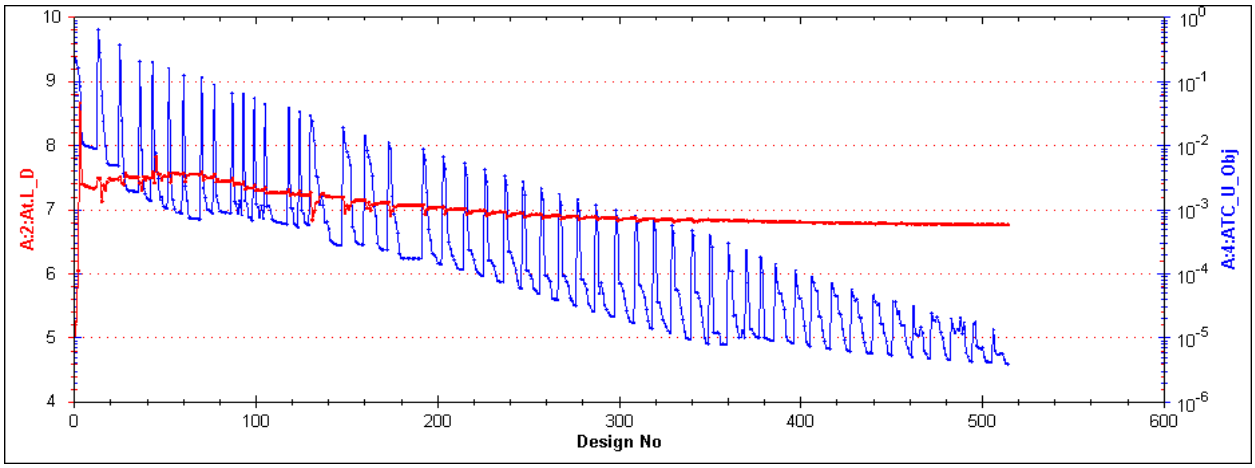
Sl. 8.54. Definicija sekvence rješavanja za SBJ-R-3L-ATC problem u OCTOPUS Designeru

Rezultati SBJ-R-3L-ATC modela

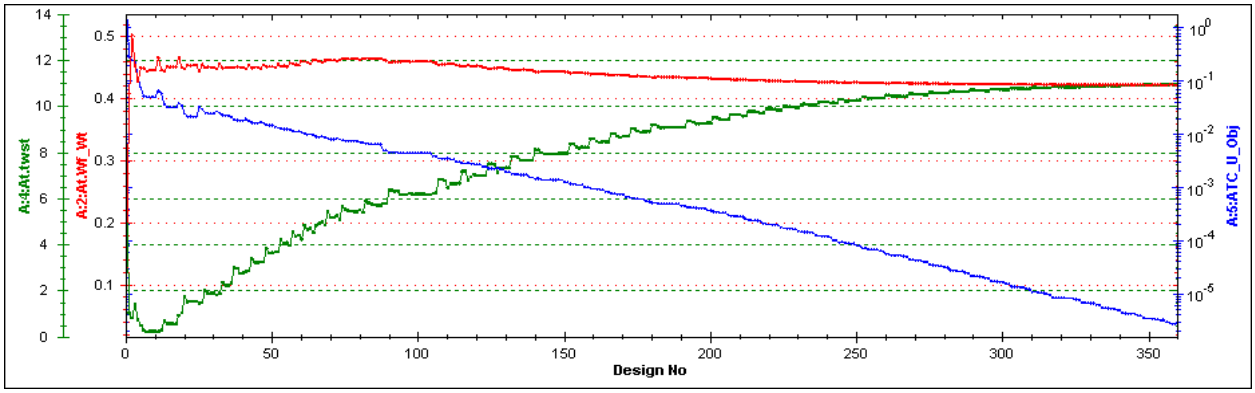
Ostvarena rješenja kroz 50 iteracija mogu se vidjeti na slijedećim slikama. Na dijagramima se može vidjeti kako je mjera konzistentnosti sustava, koja je u ATC metodi korištena kao cilj na svim podrazinama, osim na globalnoj gdje je funkcija cilja zbroj normaliziranog doleta i nekonzistentnosti sustava razine 1, konvergirala tijekom 50 iteracija te kako su se mijenjali osnovni atributi pojedinog podsustava kroz iteracije.



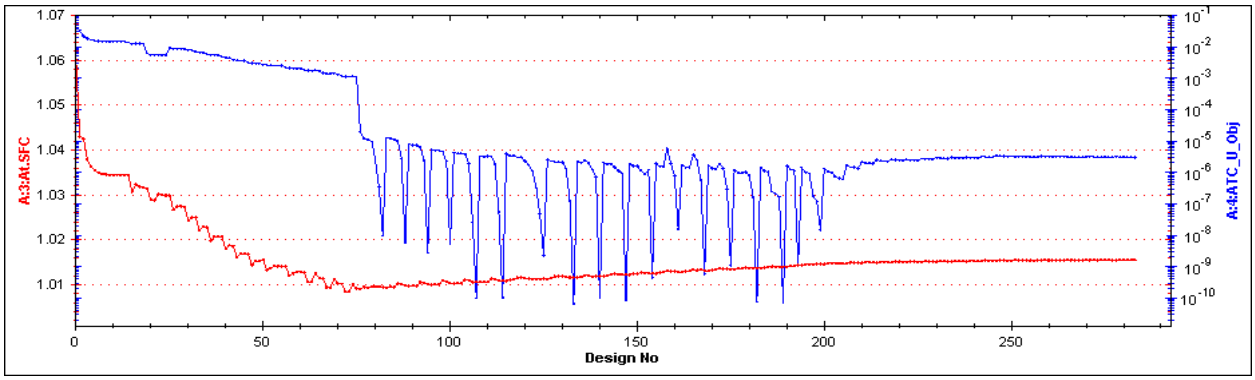
Sl. 8.55. Konvergencija globalnog sustava (SBJ-PERF)



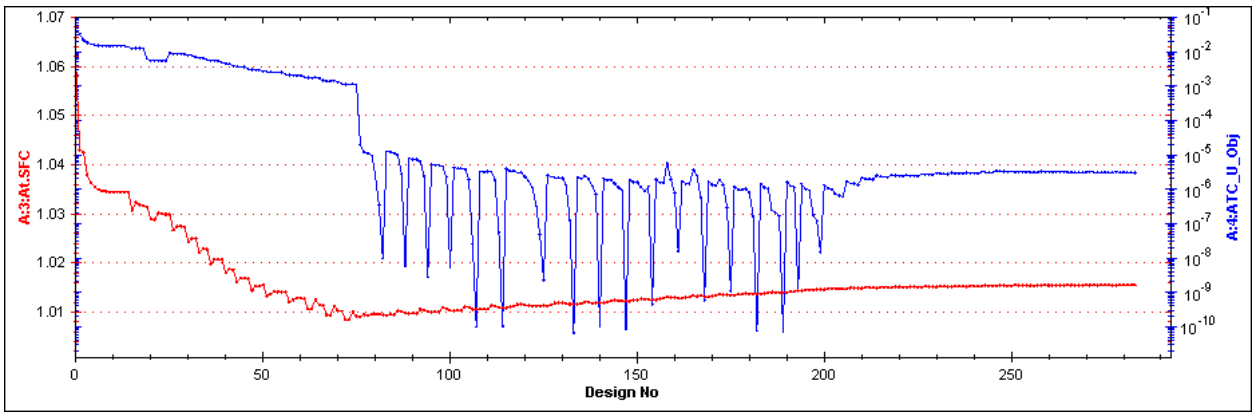
Sl. 8.56. Konvergencija AERO podsustava



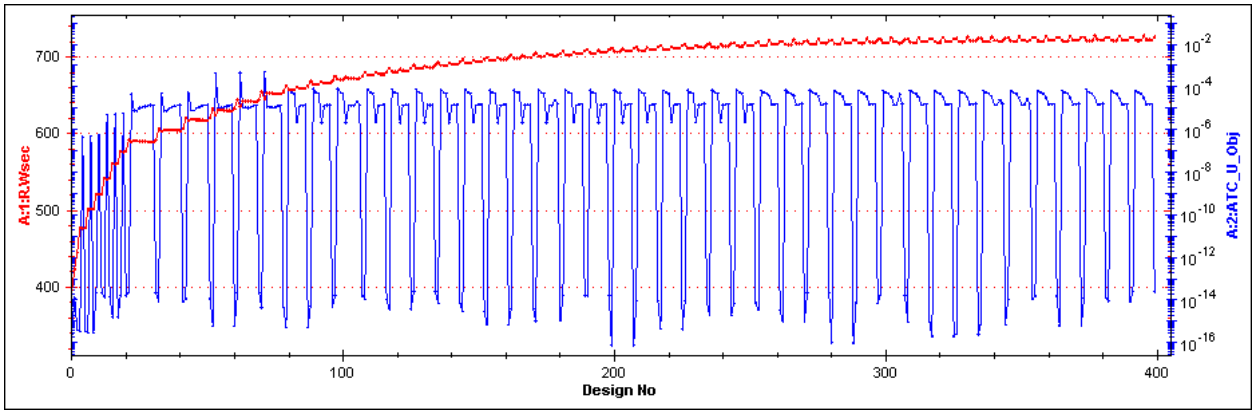
Sl. 8.57. Konvergencija STRUCT podsustava



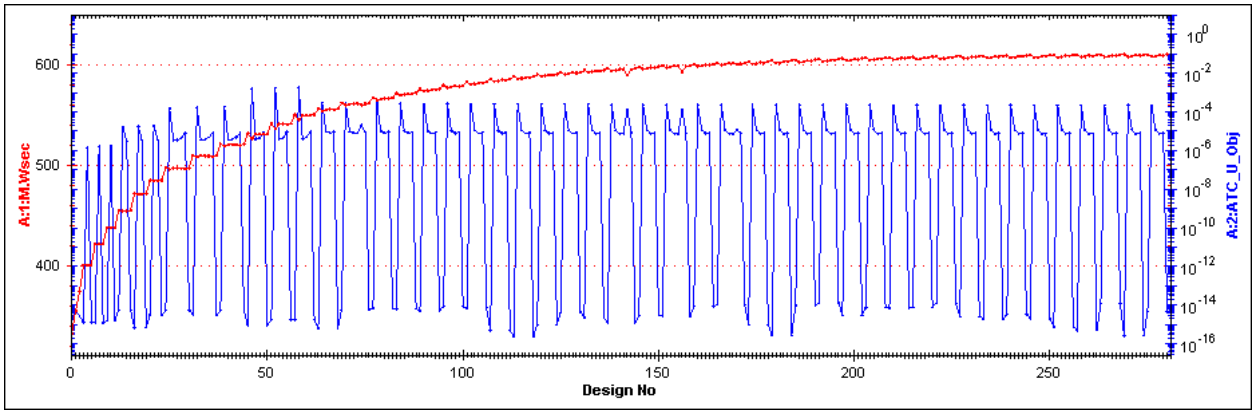
Sl. 8.58. Konvergencija PROP sustava



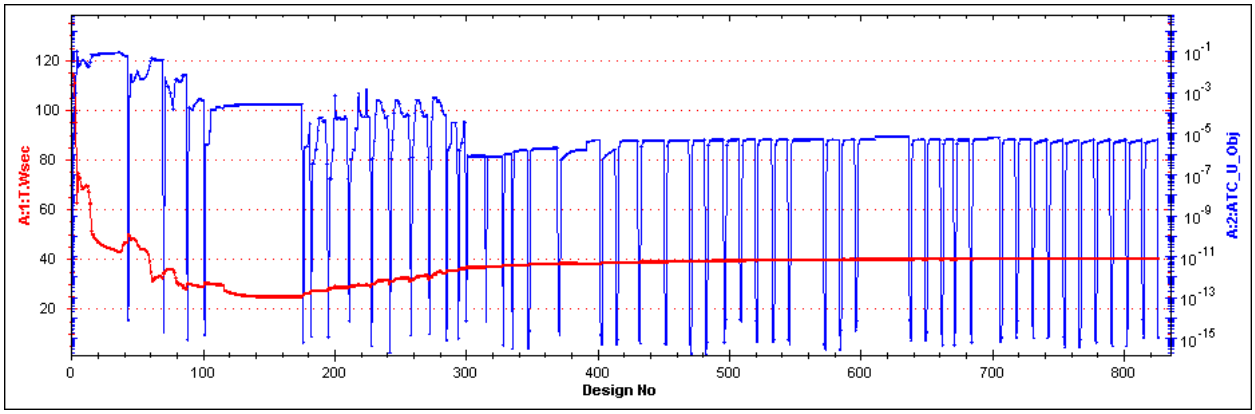
Sl. 8.59. Konvergencija PROP sustava



Sl. 8.60. Konvergencija WingRoot sustava



Sl. 8.61. Konvergencija WingMidd sustava



Sl. 8.62. Konvergencija WingTip sustava

Ostvareno rješenje, koje je nakon 50 ATC iteracija konvergiralo na prihvatljivu razinu nekonzistentnosti globalnog sustava ima dolet 3150 nm što znači da se radi o lokalnom optimumu nešto lošijeg doleta nego kod problema dekomponiranog na dvije razine. Ukupno trajanje optimizacije je značajno kraće (~ 6-9 minuta) nego kod dekompozicije na dva nivoa iz prošlog potpoglavlja. To je u najvećoj mjeri uzrokovano podjelom ukupnog broja varijabli krila na tri potproblema s jednakim brojem varijabli što je smanjilo ukupni broj poziva simulacijske sekvence prilikom numeričkog računanja derivativna faktora podobnosti neophodnih za rad korištene SQP optimizacijske metode (NLPQLP). Pri tome nije došlo do gubitka informacija o utjecaju jednog segmenata krila na ostale segmente jer su svi povezani preko njima globalnog sustava (STRUCT) te su se ti utjecaji kroz ATC iteracije prenijeli u susjedne podstrukture.

8.3.2.6. Zaključci

Jedan od osnovnih uočenih nedostataka same ATC metode jest kontrola konstanti penalizacije o kojima umnogome ovisi brzina konvergencije. Stoga je, kako bi se optimizacija uspješno odvijala, bilo neophodno testirati na koji način različite vrijednosti konstanti penalizacije utječu na konvergenciju. Pri tome nije samo važan odnos penalizirane nekonzistentnosti u odnosu na globalnu funkciju cilja nego i međusobni odnos pojedinih članova koji ulaze u ukupnu mjeru nekonzistentnosti podsustava ili nadsustava. U ovom primjeru bilo je neophodno proučiti same proračunske module i njihove interakcije kako bi se uspješno procijenile konstante penalizacije. Najveći problem je to što je za osiguranje konzistentnosti, nekonzistentnost pojedinih spregnutih podataka potrebno penalizirati s visokim iznosima konstanti penalizacije čime problem postaje „nezdravo“ kondicioniran (*ill - conditioned*) [105].

U zadnjih nekoliko godina izdano je nekoliko radova kroz koje je ovaj problem adresiran uz prijedloge unaprjeđenja, a najčešće je spominjan pristup s augmetiranim Lagrangeovim multiplikatorima [72], koji iako zahtjeva dodatne resurse za izračun Lagrangeovih multiplikatora, omogućuje bržu i sigurniju konvergenciju.

9. ZAKLJUČAK RADA

Složene tankostjene konstrukcije, poput zrakoplova i broda, traže primjenu novih tehnologija kako u gradnji i opremanju, tako i u procesu projektiranja. Da bi bili konkurentni na svjetskom tržištu, proizvođači/projektni uredi moraju moći brzo generirati različite projektne varijante i proizvesti kvalitetne projekte u ranoj projektnoj fazi. Pri tome potrebno je imati na umu da projektanti u ranim fazama projektiranja raspolažu vrlo ograničenom količinom deskriptora (parametara) projekta koja se tijekom procesa projektiranja povećava. Istovremeno, upravo u ranim fazama projektiranja donosi se najveći broj odluka povezanih s cjeloživotnim troškovima proizvoda.

Na početku rada dan je prikaz osnovnih karakteristika projektiranja složenih tankostjenih konstrukcija, poput zrakoplova i broda, u kontekstu zahtijevane sigurnosti koja je regulirana mjerodavnim tijelima. Naglašena je važnost utjecaja odluka u ranim fazama projektiranja na cjeloživotne troškove. Opisane su glavne karakteristike alata za analizu tankostjene konstrukcije u konceptualnoj i preliminarnoj fazi projektiranja prema provedenom istraživanju postojećih radova te dugoročne koncepcije programskog sustava OCTOPUS.

U nastavku dan je opis metoda sinteze koje je moguće primijeniti pri projektiranju složenih sustava. Prvo je dana generalna formulacija optimizacijskog problema te optimizacijske metode koje su sposobne rješavati probleme iz inženjerske prakse.

Izložene su osnove aproksimacijskog, tj. surogat modeliranja odziva modula za analizu, čijim je korištenjem, umjesto zahtjevnih modula analize, moguće reducirati ukupno vrijeme projektiranja složenog problema. Razrađen su glavna obilježja osnovnih koraka surogat modeliranja: planiranja računalnih pokusa, izvođenja računalnih simulacija, generiranja/treniranja surogat modela te validacije surogat modela. Prikazane su najčešće korištene formulacije surogat modela korištenih u radu: odzivnih površina (polinomski surogat), Kriginga te radijalnih baznih funkcija (RBF-a). Prema obrađenoj literaturi dane su osnovne smjernice korištenja spomenutih surogat modela.

Detaljno je prikazana primjena dekompozicije složenog problema kojom se originalni problem dijeli na više povezanih, jednostavnijih potproblema koji se mogu optimizirati paralelno na višeprocesorskim računalima ili klasterima računala. Prema relevantnim radovima tog područja opisane su osnovne karakteristike dekompozicije prema: načinu održavanja konzistentnosti spregnutih sustava (jaka i slaba formulacija) te prema načinu rasprezanja (hijerarhijska, nehijerarhijska). Isto tako navedeni su glavni načini izvođenja koordinacije te

njihova obilježja koja su usko povezana s načinima održavanja konzistentnosti spregnutih sustava. Nakon toga su prema istraženoj literaturi navedene dvorazinske i višerazinske koordinacijske metode koje se najčešće koriste pri projektiranju složenih tehničkih sustava. Prema nedavno objavljenim zaključcima istraživača toga područja, dani su perspektivni smjerovi razvoja tog područja. U njima je naglašena potreba za formalnom klasifikacijom spregnutosti problema prema jačini i količini spregnutosti te prilagođavanjem metoda trendovima masovnog paralelnog računarstva.

U poglavlju 6 na osnovi analize karakteristika pojedinih metoda sinteze te karakteristika samog projektiranja složenih tankostjenih konstrukcija predloženo je unapređenje procesa projektiranja tankostjenih konstrukcija dekompozicijom na funkcionalne jedinice-podstrukture (unutar kojih su sadržani elementi čija svojstva su bitnije spregnuta) te povezivanje odnosno koordinacija podstruktura surogat modelom strukturnih odziva. Također je istaknuta važnost sagledavanja problema projektiranja tankostjene konstrukcije sa stajališta količine i jačine spregnutosti pojedinih dijelova strukture. Predložena je mogućnost mjerenja jačine spregnutosti pojedinih podsustava korištenjem analize utjecajnosti parametara polinomskog surogat modela kao alternativa najčešće korištenom pristupu pomoću parcijalnih derivacija.

U poglavlju 7 korištenjem tehnika objektno-orijentirane paradigme, svojstva i ponašanje optimizacijskih metoda, dekompozicije i koordinacije potproblema/podsustava te surogat modeliranja prenesena su u objektni model čime je omogućeno intuitivno korištenje tih tehnika u okruženju za potporu pri projektiranju OCTOPUS Designer. Omogućeno je da se korištenjem interaktivnog zadavanja i oblikovanja implementiranih metoda sinteze u određenoj mjeri istražuju i zadaju i nove metode/tehnike koordinacije, odnosno sekvence rješavanja problema/potproblema čak i bez implementiranja novog programskog koda. Nasljeđivanjem osnovnih klasa: Optimizer (za optimizacijske metode), CoordinationSystem i CoordinationMethodBase (za koordinacijske metode) te SurogatBase (za surogat modele), moguće je na jednostavan način uvesti i potpuno nove metode sinteze. S obzirom na karakteristike arhitekture današnjih računala kod kojih se ukupna proračunska snaga više bitno ne povećava povišenjem radnog takta procesora, već povećanjem broja radnih jezgri, jedna od bitnih karakteristika jest mogućnost paralelnog izvršavanja određenih sekvenci naredbi. U okviru okruženja za potporu pri projektiranju OCTOPUS Designer, to je omogućeno na trima razinama: 1) generiranjem nove dretve, 2) generiranjem nove aplikacijske domene i 3) generiranjem .NET Remoting servera i klijenta. Time je omogućen fleksibilan odabir pojedine tehnike prema zadatku koji je potrebno obaviti.

Na kraju rada dani su primjeri složenih tankostjenih konstrukcija na kojima je prikazano sljedeće: mogućnosti interakcije projektanta s projektnim problemom, korištenje razmatranih metoda sinteze te uspješnost predloženih unapređenja projektiranja tankostjenih konstrukcija.

Na primjeru jednostavnog pontona testirane su osnovne postavke predloženih unaprjeđenja. Validacijom točnosti predviđanja dvadesetak podvrsta polinomske regresije, Kriginga i RBF-a utvrđeno je da uz nekoliko izuzetaka većina surogat modela daje podjednaku razinu točnosti predviđanja odziva. Utvrđeno je da i najjednostavniji polinomski surogat modeli (linearni i čisti kvadratični) nemaju bitno manju točnost od polinomskih surogata višeg reda, odnosno od boljih podvrsta Kriginga i RBF-a. Također je utvrđeno da je točnost predviđanja primarnih (uzdužnih) naprezanja izuzetno dobra, no da je predviđanje sekundarnih naprezanja te momenata savijanja na evaluacijskim elementima nešto niže razine točnosti. Najvjerojatniji uzrok tome jest promjenjivost distribucije tog naprezanja i momenta po podstrukturama. Predloženo je, a kasnije na primjeru i dokazano, da je tu pogrešku prilikom primjene u optimizaciji moguće efikasno neutralizirati kalibriranjem lokalnih koeficijenata pojedinog surogata u završnim ciklusima optimizacije. Optimizacijama te usporedbom ostvarenih rezultata nekoliko predloženih sekvenci rješavanja utvrđeno je da korištenje surogat modela poboljšava kako kvalitetu rješenja, tako i stabilnost konvergencije .

Na primjeru projektiranja realnog broda iz inženjerske prakse (RoPax) potvrđeno je poboljšanje kvalitete i stabilnosti optimizacijskog procesa upotrebom surogat modela odziva čak i kod visokodimenzijskih problema. Također je potvrđeno da je postupak realno primjenjiv s obzirom da se korištenjem redukcije broja parametara uz pomoć kompozitnih grednih parametara ukupni broj kontrolnih parametara surogat modela bitno smanjuje. Time je omogućeno da se za konceptualne, pa čak i preliminarne modele, potreban broj pokusa za generiranje jednostavnijih polinomskih modela generira na samo jednom računalu u „prekonoćnom radu“. Programski sustav razvijen u ovom radu, DeCluster (asinkroni rješavač višestruko paralelnih zadataka), ovisno o broju spojenih računala, omogućuje proračun dovoljnog broja pokusa/simulacija za puni kvadratni polinomski surogat, tijekom samo nekoliko sati za visokodimenzijski primjer projektiranja broda (RoPax).

Na primjeru konceptualnog projektiranja nadzvučnog poslovnog zrakoplova prikazana je mogućnost korištenja unaprijeđenog sustava za potporu pri projektiranju OCTOPUS Designer na višedisciplinarnim optimizacijskim (MDO) problemima, tj. problemima kod kojih je dekompozicija izvedena prema disciplinama. Prikazana je usporedba dobivenih rješenja jednorazinskih MDF i IDF metoda, s dvorazinskom i trorazinskom ATC metodom.

Tijekom istraživanja i primjene odabranih metoda sinteze, na primjerima koji imaju karakteristike realnih problema iz inženjerske prakse, utvrđeni su i perspektivni smjerovi istraživanja koji bi mogli rezultirati dodatnim unaprjeđenjem procesa projektiranja tankostjenih konstrukcija te je predloženo da se:

- istraži korištenje nekih drugih veličina, poput momenta inercije podstruktura, kao veličina kojima se mjeri i kažnjava udaljenost dekomponiranog strukturnog optimizacijskog potproblema od zadnjeg izračunatog stanja strukturnog odziva, umjesto u ovom radu korištenog kažnjavanja proporcionalnog s udaljenošću samih deskriptora
- istraži mogućnosti poboljšanja kvalitete surogat modela strukturnog odziva odabirom karakterističnih pozicija pojedinog surogata odziva na osnovi analize distribucije strukturnog odziva prototipa
- istražiti mogućnosti korištenja strukturnog odziva niže razine točnosti (npr. OCTOPUS Analizator) kao zamjena za vremenski zahtjevne 3D MKE modele (npr. MAESTRO) u pojedinim fazama sekvence rješavanja problema
- istraži mogućnost unošenja informacija o utjecaju pojedinih parametara pri mjerenju udaljenosti pozicije, za koju se želi izvršiti predviđanje, od postojećih projekata prema kojima je surogat model kreiran, pri radu sa surogat modelima temeljenih na mjerenju ukupne normalizirane udaljenosti od poznatih projekata (Kriging, RBF)
- istražiti mogućnost automatskog prilagođavanja težinskih faktora za penalizaciju nekonzistentnosti nadsustava i podsustava na osnovi podataka iz prethodnih koraka ATC metode, s obzirom da kvaliteta penalizacije izrazito utječe na brzinu konvergencije osnovne ATC metode

ZAKLJUČAK DOKTORSKOG RADA

Provedena istraživanja dala su sljedeće znanstvene doprinose i spoznaje na području sinteze složenih tankostjenih konstrukcija:

- Provedena je klasifikacija koncepata iz domene optimizacije složenih sustava, što obuhvaća formulacije optimizacijskih problema za takve sustave te postupke primjenjive za njihovo rješavanje.
- Na osnovi analize karakteristika pojedinih metoda sinteze te karakteristika samog projektiranja složenih tankostjenih konstrukcija izvedeno je unapređenje procesa projektiranja tankostjenih konstrukcija dekompozicijom na funkcionalne jedinice - podstrukture (unutar kojih su sadržani elementi čija su svojstva bitnije spregnuta) te povezivanje, odnosno koordinacija podstruktura surogat modelom strukturnih odziva.
- Korištenjem tehnika objektno-orijentirane paradigme, svojstva i ponašanje optimizacijskih metoda, dekompozicije problema te surogat modeliranja, a u određenoj mjeri i integriranih modula analize, preneseni su u okruženje za potporu pri projektiranju. Time je omogućeno intuitivno korištenje tih metoda pri rješavaju problema projektiranja složenih sustava općenito, pa tako i pri projektiranju tankostjenih konstrukcija.
- Unapređenjem vizualnog, prezentacijskog sloja računalnog okruženja za potporu pri višekriterijskom projektiranju dan je projektantu cjelovitiji uvid u odvijanje projektnog procesa na osnovi fleksibilno, interaktivno definiranih sekvenci rješavanja projektnih potproblema te mogućnost efikasnijeg odabira preferiranih projektnih rješenja.

LITERATURA

[1] Sandberg, A., Strömberg, U., (1999) "Gripen: with focus on availability performance and life support cost over the product life cycle", Journal of Quality in Maintenance Engineering, Vol. 5 Iss: 4, pp.325 – 334

[2] Kitarović, S., (2012) Analiza uzdužne granične nosivosti u konceptualnoj sintezi tankostjenih konstrukcija, Doktorski rad (u pripremi).

[3] Keane, A. and Nair, P., (2005) Computational approaches for aerospace design: the pursuit of excellence, John Wiley & Sons

[4] Žanić, V., Das, P.K., Pu, Y. and Faulkner, D., (1993): Multiple criteria synthesis techniques applied to reliability based design of SWATH ship structure, (Chapter 18), Integrity of Offshore Structures 5, Faulkner, Das, Incecik, Cowling, editors, EMAS Scientific Publications, Glasgow, pp. 387-415.

[5] IMPROVE (2009), Design of Improved and Competitive Ships using an Integrated Decision Support System for Ship Production and Operation – Proceedings of Final Workshop, Žanić, V. i Andrić, J. (ur.), Dubrovnik.

[6] Pelletier, J. L. and Vel, S. S., 2006, "Multi-Objective Optimization of Fiber Reinforced Composite Laminates for Strength, Stiffness and Minimal Mass," Computers and Structures, 84, pp. 2065-2080.

[7] Sinha and Kaushik (2007). Reliability-based multiobjective optimization for automotive crashworthiness and occupant safety. Structural and Multidisciplinary Optimization, 33(3):255-268.

[8] Hansen, L.U., Horst, P., (2008): Multilevel optimization in aircraft structural design evaluation, Computers & Structures, Vol 86 p. 104–118, 2008

[9] Romanoff, J., Klanac, A. (2007), Design Optimization of Steel Sandwich Hoistable Car-Decks Applying Homogenized Plate Theory, 10th Symposium on Practical Design of Ships and Other Floating Structures, PRADS 2007, Huston, USA, pp.839-846.

[10] Rao, S. S., (1996), „Engineering Optimization - Theory and Practice“ (3rd Edition).. John Wiley & Sons.

[11] Žanić, V., Prebeg, P. and Kitarović, S., (2007), Decision Support Problem Formulation for Structural Concept Design of Ship Structures, MARSTRUCT Conference, Glasgow, pp. 499-512.

[12] ISSC, (2006), Technical Committee IV.2, "Design Methods", Proc. of the 16th International Ship and Offshore Structures Congress, Southampton, UK

[13] ISSC, (2009), Technical Committee IV.1, " Design Principles and Criteria", Proc. of the 17th International Ship and Offshore Struc-tures Congress, Seoul, Korea.

[14] Dantzig, G. (1974.), Linear Programming and Extensions, Princeton University Press,

[15] Žanić,V. (1980): Projektiranje brodske konstrukcije generiranjem optimalnih strukturnih varijanti na elektroničkom računalu, Doktorska disertacija, FSB Zagreb.

[16] Schittkowski, K. (2010). NLPQLP: A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search - User's Guide, Version 3.1.

[17] Powell, M. J. (1978). The convergence of variable metric methods for nonlinearly constrained optimization calculations. U O. Mangasarian, R. Meyer, & S. R. eds., Nonlinear Programming 3. Academic Press.

[18] Kennedy, J., & Eberhard, R.C., (1995), Particle swarm optimization. In Proceedings of IEEE international conference on neural networks, Piscataway, NJ, USA (pp. 1942-1948).

[19] Venter, G. and Sobieski, J.(2002): Particle Swarm Optimization, AIAA 2002-1235, 43rd AIAA/ASME/ASCE/ AHS/ASC, Structures, Structural Dynamics, and Materials Conference, Denver, CO.

[20] Coello C.A.C. and Lechunga M.S. (2002). MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In Proceedings of the IEEE World Congress on Computational Intelligence, Hawaii, May 12-17, 2002. IEEE Press.

[21] Važić, B., Posavec, D., Bingula, S., Pijaca, N. i Popović, S., (2008): Implementacija Multiobjective Particle Swarm Optimization optimizacijskog algoritma u OCTOPUS DESIGNER, Seminarski rad iz kolegija Osnivanje konstrukcije zrakoplova, FSB Zagreb, Hrvatska.

[22] Knowles J.D. and Corne D. , (2000), Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary Computation, 8(2) :149-172.

[23] Mostaghim S. and Teich J., (2003), Strategies for finding good local guides in multi-objective particle swarm optimization. In IEEE Swarm Intelligence Symposium, pages 26–33.

[24] Huang, V.L., Suganthan, P.N. and Liang, J.J.,(2006): Comprehensive Learning Particle Swarm Optimizer for Solving Multiobjective Optimization Problems, School of Electrical

and Electronic Engineering, Nanyang Technological University, 639798 Singapore, INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS,VOL. 21, 209–226, 2006.

[25] C. A. Coello Coello, G. B. Lamont and D. A. Van Veldhuizen, (2007). Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York, second edition.

[26] Važić, B. (2009), Quality Measures of Pareto Fronts Obtained By Multiobjective Optimization Algorithms, Projekt, Sveučilište u Zagrebu, Hrvatska,.

[27] Fang, K. Li R., Sudjianto A., (2006), Design and Modeling for Computer Experiments, Chapman & Hall - Taylor & Francis Group.

[28] Sacks, J., Schiller, S. B. and Welch, J. (1989a). Designs for computer experiments, Technometrics 31(1): 41-47.

[29] Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989b). Design and analysis of computer exerimenis, Statistical Science 4 (4): 409-435.

[30] Simpson, T. W., Maurey, T. M., Korte, J. J., and Mistree, F., "Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," AIAA Journal, Vol. 39, No. 12, 2001,pp. 2233–2241.

[31] Wang, G. G., Shan, S. (2007), "Review of Metamodeling Techniques in Support of Engineering Design Optimization," ASME Transactions, Journal of Mechanical Design, Vol. 129, No. 4, April 2007, pp. 370-380.

[32] Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization. Prog Aerosp Sci 41:1–28

[33] Montgomery , D. C.,(1991) "Design and Analysis of Experiments", John Wiley&Sons, Singapore,.

[34] Cajner, H., (2011), Višekriterijsko adaptivno. Oblikovanje planova pokusa, Doktorski rad, Fakultet strojarstva i brodogradnje, Zagreb.

[35] MATLAB (2011), Program documentation, www.mathworks.com/help/index.html

[36] Žanić, V., Kitarović, S.; Prebeg, P. (2010): Safety as Objective in Multicriterial Structural Optimization // Proceedings of the ASME 2010. Shanghai, 2010.

[37] Viana, F. A. C., Venter, G., & Balabanov, V. (2009). An algorithm for fast optimal latin hypercube design of experiments. International Journal for Numerical Methods in Engineering, 82(October 2009), 135-156.

[38] *Hengl, T. (2009). A Practical Guide to Geostatistical Mapping. Second edition (p. 293). University of Amsterdam.* http://spatial-analyst.net/book/

[39] Krige, D.G. (1951): A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. Journal of the Chemical, Metallurgical and Mining Society of South Africa, 52, 119-139.

[40] Kleijnen, J. (2009). Kriging metamodeling in simulation: A review. European Journal Of Operational Research, 192(3), 707-716

[41] Neuman, S. P. and Jacobson, E. A. (1984). Analysis of nonintrinsic spatial variation by residual kriging with application to regional groundwater levels. Journal of the International Association for Mathematical Geology, 16, 499–521.

[42] Powell, M. J. D. (1992). The theory of radial basis functions approximation in 1990. In W. A. Light (Ed.), Advances in Numerical Analysis Volume II Wavelets Subdivision algorithms and radial basis functions (pp. 105-210). Oxford University Press.

[43] Carlin, M. (1992). Radial Basis Function Networks and Nonlinear Data Modelling. Proc Neural Networks and their Applications NeuroNimes92, 62363.

[44] Buhmann, M. D. (2003). Radial basis functions: theory and implementations. System. Cambridge University Press. doi:10.1017/CBO9780511543241

[45] Jin, R., Chen, W., Simpson, T. W., (2001) Comparative studies of metamodelling techniques under multiple modelling criteria, Structural and Multidisciplinary Optimization, 23, 1-13.

[46] Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of surrogates. Struct Multidisc Optim 33(3):199–216

[47] Žanić, V., Andrić, J., Frank, D. (2003.) Structural Optimization Method for the Concept Design of Ship Structures, International Marine Design Conference, IMDC 2003, Athens, Greece, Vol 2, pp.205-218.

[48] Chen J., Goldberg D. E. , Sastry K., Ho S., (2002): Fitness Inheritance in Multi-Objective Optimization, Proceedings of the Genetic and Evolutionary Computation Conference, 319–326.

[49] www.flightglobal.com, Airbus 380, http://www.flightglobal.com/imagearchive/Image.aspx?GalleryName=Cutaways/Civil%20Aviation/Civil%20Aviation%201949-2006&Image=Airbus-A380-800 Datum pristupa 1. ožujka 2011.

[50] www.anchoryachts.com/preview.php?ID=241, Putnički brod, Datum pristupa 1. ožujka 2011.

[51] Minnett, S i Taylor, C (2008) http://www.cadinfo.net/aerospace/building-the-worlds-largest-passenger-aircraft-wings. Datum pristupa 1. ožujka 2011.

[52] Schoeffler, J. D. (1970), Multilevel Systems and Decomposition for the Solution of Static Optimization Problems; Decomposition and Multilevel Methods for On-Line Computer Control of Industrial Processes, in D. A. WISMER, Ed., Optimization Methods for Large-Scale Systems, McGraw-Hill, New York, 1970.

[53] de Wit, A.J. and van Keulen F.(2010) Overview of Methods for Multi-Level and/or Multi-Disciplinary Optimization, Proceedings of the 51th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Orlando, USA.

[54] de Wit, A.J., (2009): A unified approach towards decomposition and coordination for multi-level optimization. PhD thesis, Delft University of Technology, 2009.

[55] Agte, J., Weck, O., Sobieszczanski-Sobieski, J., Arendsen, P., Morris, A., & Spieck, M. (2010). MDO: assessment and direction for advancement—an opinion of one international group. Structural and Multidisciplinary Optimization, 40(1-6), 17-33

[56] Allison J., Kokkolaras M., and Papalambros P.,(2005), On The Impact of Coupling Strength on Complex System Optimization for Single-Level Formulations, Proceedings of DETC'05 2005 ASME Design Engineering Technical Conferences Long Beach, California, USA, September 24-28, 2005

[57] Kirsch, U. (1997). Two-level optimization of prestressed structures. Engineering Structures, 19(4):309–317.

[58] Vanderplaats, G., Yang, Y., and Kim, D. (1990). Sequential linearization method for multilevel optimization, American Institute of Aeronautics and Astronautics Journal, 28:290–295.

[59] Tosserams, S., Etman, L. F. P., & Rooda, J. E. (2008). A classification of methods for distributed system optimization based on formulation structure. Structural and Multidisciplinary Optimization, 39(5), 503-517.

[60] Allison James T., Kokkolaras Michael, and Papalambros Panos Y.. (2009) 'Optimal Partitioning and Coordination Decisions in Decomposition-based Design Optimization.' ASME Journal of Mechanical Design, Vol. 131, No. 8.

[61] *Sobieszczanski-Sobieski J, James BB, Dovi AR (1985) Structural and multidisciplinary optimization by multilevel decomposition. AIAA J 23(11):1775–1782*

[62] Sobieszczanski-Sobieski, J.,(1991), "Optimization by Decomposition in Structural and Multidisciplinay Applications," presented at the NATO-Advanced Study Institute on Optimization of Large Structural Systems, Sept. 23-Oct. 4 1991, Berchtesgaden, Germany.

[63] Liu, B., Haftka, R.T., and Watson L.T., “Global-Local Structural Optimization Using Response Surfaces of Local Optimization Margins,” Structural and Multidisciplinary Optimization, 27(5), 352-359, 2004.

[64] Schutte, J.F. and Haftka, R.T., (2010) “Global structural optimization of a stepped cantilever beam using quasiseparable decomposition”, Engineering Optimization, Vol. 42(4), 347-367.

[65] Braun R. D, Moore, A.A., Kroo, I. M (1996), Use Of The Collaborative Optimization Architecture For Launch Vehicle Design , Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization

[66] Kroo, Ilan, (2004) “Distributed Multidisciplinary Design and Collaborative Optimization,” VKI Lecture Series: Optimization Methods and Tools, von Karman Institute.

[67] Alexandrov, N.M.; Lewis, R.M. (2002), Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design, AIAA Journal , vol.40 no.2 (301-309).

[68] Kim, H.M., Michelena, N.F., Papalambros, P.Y., and Jiang, T., "Target Cascading in Optimal System Design," Transaction of ASME: Journal of Mechanical Design, Vol. 125, pp. 481-489, 2003

[69] Michelena, N., Park, H., and Papalambros, P. (2003). Convergence properties of analytical target cascading. American Institute of Aeronautics and Astronautics Journal, 41:897–905.

[70] de Wit, A. and van Keulen, F. (2007). Numerical Comparison of Multi-Level Optimization Techniques. In 3rd AIAA Multidisciplinary Design Optimization Specialist Conference, Waikiki in Honolulu, Hawaii, 23-26 April.

[71] Kim, H. M., Chen, W., and Wiecek, M., (2006) “Lagrangian Coordination for Enhancing the Convergence of Analytical Target Cascading,” AIAA Journal, Vol. 44, No. 10, pp.2197 - 2207.

[72] Tosserams, S., Etman, L. F. P., Papalambros P.Y., Rooda, J. E. (2006), An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. Structural and Multidisciplinary Optimization, 31: 176–189.

[73] OCTOPUS manuals (2009), Software documentation, University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, Croatia.

[74] Hughes, O.F., Mistree,F., Žanić,V., (1980), A Practical Method for the Rational Design of Ship Structures, Journal of Ship Research, Vo. 24, No. 2, pp.101-113.

[75] MAESTRO Version 9.2. (2011). Program documentation, DRS Technologies, http://www.maestromarine.com/

[76] Žanić, V., Andrić, J., Prebeg, P., Stipčević M., Pirić K., (2010) RoPax Structural Design-Multi-level Decision Support Methodology, 11th International Symposium on Practical Design of Ships and Other Floating Structures, PRADS 2010, Rio de Janeiro, Brazil.

[77] Andrić, J. (2007), Metodologija konceptualnog projektiranja brodskih konstrukcija s interakcijom trup-nadgrađe, Doktorski rad, Fakultet strojarstva i brodogradnje, Zagreb.

[78] Žanić, V., Andrić, J., Prebeg, P., Stipčević, M., Kitarović, S., Pirić, K., Grgić, M., Hadžić, N., et al. (2009), D7.2 Report presenting the identified parameters having larger positive or negative impact on the optimum solution/RDMM procedure-RoPax ship // EU FP6 Project IMPROVE Devliverable D7.2 / Žanić, Vedran i Andrić, Jerolim (ur.). Zagreb : FSB, Zagreb, 2009. 1-188

[79] Žanic, V., Grubisić, I. and Trincas, G., (1992). "Multiattribute Decision Making System based on random Generation of Nondominated Solutions, Proceedings of PRADS 92, Elsevier Applied Science, Vol. 2, pp. 1443-1460.

[80] Das, P.K., V.Žanić, V., Faulkner, D.(1992), "Reliability- Based Design Procedure of Stiffened Cylinder Using Multiple Criteria Optimisation Techniques", Offshore Technology Conference, Houston, Paper 7326, 1992.; pp.297-311.

[81] Grubišić, I., Trincas G., Žanić V., (1993)"Efficient Solution of the Multiattribute Design Problem Applied to Fast Passenger Vessel ", Second Symposium on 'High Speed Marine Vehicles' HSMV ' 93, Naples: pp.323-336.

[82] Trincas G, Žanić V, Grubišić I., (1994), Comprehensive Concept of Fast Ro-Ro Ships by Multiattribute Decision-Making, Proceedings of 5th International Marine Design Conference, IMDC'94,.

[83] Grubišić, I. Žanić, V. Trincas, G.: "Sensitivity of Multiattribute Design to Economy Environment: Shortsea Ro-Ro Vessels", IMDC'97; The Sixth International Marine Design Conference; 23-25 June 1997., Univ. of Newcastle; pp. 201-216.

[84] Žanić V. (2005) Decision Support Techniques in Structural and General Ship Design, Proceedings of the International Congress of Marine Research and Transportation -

ICMRT 2005 / Cassella, Pasquale; Cioffi Pasquale (ur.), Napulj, Italija, 1-14 (invited lecture).

[85] Hozmec, J. (2003), Proračun pouzdanosti tankostjenih konstrukcija, Magistarski rad, Fakultet strojarstva i brodogradnje, Zagreb.

[86] http://msdn.microsoft.com/en-us/library/52f3sw5c%28v=vs.90%29.aspx

[87] Žanić, V. et al.,(2006), "Integrirani programski sustav za brodske konstrukcije-TP-03/0120-23- Završno izvješće o rezultatima projekta", Fakultet strojarstva i brodogradnje, Zagreb.

[88] Herceg, N. (2009), Modeliranje tehničkih problema korištenjem surogatnih modela, Diplomski rad, Fakultet strojarstva i brodogradnje, Zagreb.

[89] Frank, D.: Programska ljuska za višekriterijalno projektiranje, Diplomski rad, Sveučilište u Zagreb, Hrvatska, 2000

[90] Prebeg, P., Kitarović, S. and Žanić, V., (2009) The Design Methodology with the Sequencer for Efficient Design Synthesis of Complex Engineering Systems // Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, USA.

[91] http://en.wikipedia.org/wiki/Unified_Modeling_Language

[92] Čudina, P.: (2010),"Nova metodologija projektiranja trgovačkih brodova", Doktorski rad; Fakultet strojarstva i brodogradnje, Zagreb.

[93] Diraneh (2010), http://dranieh.com/NET/Assemblies.htm http://www.skilldrive.com/book/DOTNETinSamples.htm#_Toc112335378

[94] Rammer, I. i Szpuszta, M., (2005) Advanced .NET Remoting 2nd Edition, Apress.

[95] Hughes, O. F. (1983) Ship Structural Design: A Rationally-Based, Computer-Aided, Optimization Approach, John Wiley&Sons, NY, 1983.

[96] Bureau Veritas (2007), Analysis Based on 3-D Models, Pt B, Ch 7, App. 1-3, June 2000.

[97] Trim & Stability Book (2009), IMPROVE RoPax, ULJANIK Shipyard, Pula, Croatia.

[98] Bureau Veritas (2007). Rules for the Classification of Steel Ships - June 2007 Edition, Bureau Veritas, France.

[99] Sobieszczanski-Sobieski J, Altus DT, Phillips M, Sandusky R (2003) Bi-level System Synthesis (BLISS) for concurrent and distributed processing. AIAA Paper AIAA-2002-540.

[100] Agte, J., Sobieszczanski-Sobieski, J., and Sandusky, R. (1999); "Supersonic Business Jet Design through Bi-Level Integrated System Synthesis," SAE 1999-01-5622, World Aviation Congress.

[101] Semenov, O., (2011a), Izdvajanje programskih modula za projektiranje nadzvučnog poslovnog zrakoplova, Seminarski rad.

[102] Semenov, O., (2011b), Više disciplinarna, jednorazinska, optimizacija nadzvučnog poslovnog zrakoplova, Seminarski rad.

[103] Giles, G.L.: "Equivalent Plate Modeling for Conceptual Design of Aircraft Wing Structures", Proc. 1st AIAA Aircraft Engineering Technology and Operations Congress, Los Angeles, CA, AIAA Paper No. 95-3945, September 1995.

[104] Bruhn, E.F., "Analysis and Design of Flight Vehicle Structures," S.R. Jacobs & Associates, Inc., 1973.

[105] Tosserams, S. (2008). Distributed optimization for systems design. PhD thesis, Eindhoven University of Technology.

KRATKI ŽIVOTOPIS

Pero Prebeg rođen je u Kutini 28. srpnja 1976. gdje je završio osnovnu i srednju tehničku školu. Godine 1995/96. upisao je Studij zrakoplovstva na Fakultetu strojarstva i brodogradnje u Zagrebu. Tijekom studija, 2001. godine , proveo je tri mjeseca na studentskoj praksi na Zavodu za primijenjenu termodinamiku, Universidad Politécnica de Valencia. Dobitnik je nagrade Davorin Bazjanec za najboljeg studenta Studija zrakoplovstva 1997/98., Rektorove nagrade za rad Computer Aided Design of a Two Seat Small Aerobatic Airplane, 2000/01. te Medalje Fakulteta strojarstva i brodogradnje Sveučilišta u Zagrebu za uspjeh tijekom studija 2002. Diplomirao je u svibnju 2002. godine iz područja analize tankostjenih konstrukcija.

Od ožujka 2003. godine zaposlen je kao znanstveni novak na Zavodu za brodogradnju i pomorsku tehniku Fakulteta strojarstva i brodogradnje u Zagrebu. Školske godine 2002/03. upisao je poslijediplomski studij na istom fakultetu, smjer Teorija konstrukcija.

U okviru znanstvenoistraživačkog rada bavi se metodama projektiranja tankostjenih konstrukcija te je sudjelovao u nekoliko zahtjevnih strukturnih analiza i optimizacija složenih brodskih konstrukcija metodom konačnih elemenata. Uz istraživački rad na projektu Višekriterijski projektni modeli u osnivanju i konstrukciji broda i zrakoplova, aktivno je sudjelovao na europskim FP6 projektima IMPROVE i DeLight, STIRP projektu Integrirani programski sustav za brodske konstrukcije (modul OCTOPUS-DESIGNER) te na projektu razvoja računalnog projektnog sustava CREST - CSR(BC) u suradnji s Hrvatskim registrom brodova. Koautor je desetak znanstvenih i stručnih radova iz područja projektiranja i strukturne analize tankostjenih konstrukcija na međunarodnim konferencijama i časopisima.

Sudjeluje u održavanju vježbi i pismenih ispita na matičnom fakultetu iz redovitih kolegija: Konstrukcija broda I, Konstrukcija zrakoplova II, Čvrstoća broda, Čvrstoća i pouzdanost nosivih konstrukcija, Osnivanje zrakoplova I i II.

Služi se engleskim jezikom. Oženjen je.

CURRICULUM VITAE

Pero Prebeg was born in Kutina July 28 1976. He graduated Aerospace Engineering in 2002 at Faculty of Mechanical Engineering and Naval Architecture in Zagreb. He received the Medal of the Faculty of Mechanical Engineering and Naval Architecture for success during the study. His research involves complex system design with focus on a thin-walled structures design, where he is a co-author of fifteen scientific papers in international conferences and journals. In addition to a work on research project on Multi-criteria Design Models for the Ship and Aircraft Design, he actively participated in the European FP6 project IMPROVE and Delight, STIRP The Integrated Software System for Ship Design, and development of a software system for ship structural analysis CREST in collaboration with the Croatian Register of Shipping.

Fluent in English. Married.