UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI, PROTECȚIEI SOCIALE ȘI PERSOANELOR VÂRSTNICE
AMPOSDRU

Fondul Social European
POS DRU 2007-2013

Instrumente Structurale
2007-2013

MINISTERUL EDUCAȚIEI NAȚIONALE
OIPOSDRU

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Investeşte în oameni !

FONDUL SOCIAL EUROPEAN

Proiect cofinanţat din Fondul Social European prin Programul Operaţional Sectorial pentru Dezvoltarea Resurselor Umane 2007 – 2013

Axa prioritară 1: „Educaţia şi formarea profesională în sprijinul creşterii economice şi dezvoltării societăţii bazate pe cunoaştere”

Domeniul major de intervenţie 1.5 "Programe doctorale şi post-doctorale în sprijinul cercetării"

Titlul proiectului: „Q-DOC- Creșterea calității studiilor doctorale în științe inginerești pentru sprijinirea dezvoltării societății bazate pe cunoaștere”

Contract : POSDRU/107/1.5/S/78534
Beneficiar: Universitatea Tehnică din Cluj-Napoca

FACULTATEA DE CONSTRUCȚII

Ing. Ioana D. Balea (Krucke)

TEZĂ DE DOCTORAT

STRATEGII DE OPTIMIZARE A STRUCTURILOR METALICE BAZATE PE ALGORITMI GENETICI

Conducător ştiinţific,

Prof.Univ.Dr.Ing. George M. Bârsan

2015

CUPRINS

1 INTRODUCERE

1.1 MOTIVAȚIA ALEGERII TEMEI

Motivația acestei lucrări a fost dată de observația prevalenței studiilor de analiză și unei insuficiente atenții acordate dezvoltării proiectării capabile să folosească eficient noile metode dezvoltate de designul conceptual și optimizare. În domeniul mecanicii structurilor este posibilă analiza unei structuri aflate sub acțiunea unei solicitări date, pentru a se obține valorile exacte ale tensiunilor, deformațiilor și frecvențelor naturale. Nu este însă evident cum structura poate fi configurată geometric și proporționată pentru a ne asigura că ea este cea mai eficientă în satisfacerea cerințelor de rezistență, serviceabilitate, estetică, etc, impuse.

Economia de cost, de forță de muncă, aplicarea de metode noi, simplificate, utilizarea de materiale inovative și tehnologii ecologice, forme și design remarcabile; toate acestea sunt trăsături fundamentale ale optimizării structurale. Noile tendințe și cercetarea în acest domeniu au fost conduse, în ultimele decenii, de aplicarea cunoștințelor și observațiilor obținute din studiul proceselor naturale, a organismelor, a structurilor și materialelor, de la nivelul particulelor subatomice la comportamentul insectelor și animalelor, a anatomiei, a relațiilor ecologice din habitate naturale, și apoi aplicarea acestor cunoștințe la designul structurilor și mediului construit. Rezultatele sunt extrase din analiza atentă și sistematică a modurilor în care natura a proiectat structuri. Pe această bază putem dezvolta criterii și strategii pentru a evolua construcțiile într-o manieră asemănătoare, în mod eficient și sustenabil, găsind resurse noi, și răspunzând la mediul dinamic în care structurile sunt plasate.

O structură ușoară necesită mai puțin material pentru construcție și astfel reușește să asigure utilizarea maximă, rațională a resurselor. Utilizând geometrii optimale pentru structuri este asigurată rezistența superioară a acestora și în acelați timp sunt reduse consumurile și pierderile. În natură întâlnim nenumărate exemple de optimizare. Structura fagurelui este un exemplu de aranjare compactă eficientă. În cazul metalelor și aliajelor, atomii iau pozițiile care necesită consumul minim de energie posibil prin formarea de celule unitare care definesc structura cristalină a materialului. Pentru stâlpi, optimizarea nu este o tendință nouă. Tehnicile de optimizare sunt în prezent utilizate în majoritatea domeniilor industriale, cum sunt: industria aeronautică, industria constructoare de automobile, industria electrică, industria chimică, etc.

Câteva exemple de aplicații industriale diversificate ale tehnicii de optimizare sunt enumerate mai jos:

1. greutate, vibrații, zgomot și optimizarea consumului de combustibil la automobile, reducerea costurilor de fabricație, precum și îmbunătățirea calității;

2. proiectarea aeronavelor și a structurilor aerospațiale de greutatea minimă;

3. proiectarea de structuri, cum ar fi poduri, turnuri, baraje pentru un cost minim;

4. proiectarea optimă a diferitelor componente mecanice cum ar fi legături, came, mașini-unelte, etc;

5. proiectarea optimă a rețelelor electrice;

6. optimizarea producției, a planificării, și a controlului, etc.

Figura 1.1 *Aceasta este o fotografie aeriană a unui râu care curge printr-un deșert, în Baja California, Mexic. Se poate observa similaritatea dintre acesta și ramurile copacilor. Acest tip de ramificare se regăsește în numeroase locuri diferite din natură, vasele de sânge fiind un alt exemplu unde se observă modelul de ramificare. Tiparele în natură sunt relații care apar atât pentru forma structurii cât și a proceselor. Atunci când procesele naturale întâlnesc forme care funcționează bine, tiparele par să se imite reciproc.*

Când optimizarea topologică este realizată fără a ține cont de restricții de fabricație, structuri foarte atractive sunt adesea produse, însă acestea nu pot fi realizate prea ușor. Figura 1.2 este un astfel de exemplu, unde peste un milion de variabile de proiectare au fost utilizate. Această structură a fost optimizată pentru a minimiza energia de deformare sub încărcări. Este de remarcat faptul că optimizarea topologică produce rareori designul final, chiar dacă sunt utilizate restricții de fabricație. Acest lucru se datorează faptului că optimizarea topologică, în mod normal, nu include restricții de tensiuni. Cu toate acestea, ajută la identificarea căilor de descărcare a forțelor aplicate și oferă un punct foarte bun de plecare pentru optimizarea formei și optimizarea dimensională.

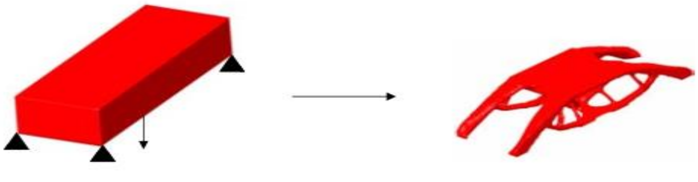


Figura 1.2 *Exemplu de optimizare topologică a unei grinzi.*

Pornind de la această premiză, în lucrarea de față, sunt realizate studii numerice care utilizează forme obținute prin optimizare topologică care folosesc mai departe la cautarea unui optim structural prim optimizare dimensională cu variabile discrete.

În general în cazul în care se pretinde ca s-a optimizat o structură, în fapt, doar câteva părți structurale alese sunt optimizate. Proiectanții caută, pentru dimensiunea minimă a secțiunii transversale, satisfacerea codului de proiectare, încearcă să găsească numărul minim de șuruburi necesare într-o anumită conexiune de oțel, caută aria minimă necesară de oțel pentru armarea unei grinzi de beton, etc. Toate părțile structurale sunt proiectate optim, dar aceasta nu înseamnă că întreaga structură este optimizată pentru, spre exemplu, costurile materialelor, timpul de construcție, prețul forței de muncă, etc. Am încercat în această lucrare să testez cele mai des utilizate tipuri de algoritmi evoluționiști și să realizez o analiză critică a modului în care aceștia pot fi aplicați cât mai eficient în domeniul optimizării structurilor metalice.

1.2 OBIECTIVE

Obiectivul optimizării structurale este maximizarea performanței unei structuri sau a unei componente structurale. Aceasta este motivată de resursele limitate, impactul asupra mediului și de concurența tehnologică, care cere structuri ușoare, ieftine și de înaltă performanță. Designul optim reprezintă cel mai bun proiect fezabil care satisface criteriile de performanță prescrise prin norme și tipologia particulară proiectului (Muller, 2002). Proiectarea optimală a structurilor are ca scop realizarea unor construcții la prețuri mici și consum mic de materiale, respectând cerințele de siguranță, funcționalitate și exploatare (Petrina, 1982) (Poterasu, și alții, 1984) (Hristache, și alții, 1981).

În cadrul lucrării am vizat următoarele obiective:

Elaborarea unui studiu privind stadiul actual al cunoașterii în domeniul optimizării structurale evolutive.

Prezentarea analizei cercetărilor efectuate și sistematizarea cunoștințelor într-o formă coerentă ușor aplicabilă în rezolvarea problemelor de optimizare cu algoritmi genetici.

Prezentarea sistematică a procesului de calcul iterativ de optimizare și analiza sub formă de funcții multi-nivel ("nested"), alături de folosirea celor mai utili algoritmi evoluționiști în domeniul structurilor.

Descrierea unei proceduri de realizare a analizei structurale în MATLAB, a posibilităților oferite de Global Optimization Toolbox și a metodelor de utilizare a algoritmilor de optimizare în mediul de programare vizuală parametrică Grasshopper (McNeel Rhinoceros).

Realizarea unui program de optimizare care implementează un algoritm genetic simplu în MATLAB.

Realizarea unui program de optimizare care implementează o strategie multi-nivel prin utilizarea a trei tipuri de algoritmi într-un program capabil de parametrizare a structurilor (Rhino Grasshopper). Etapa de evaluare a indivizilor s-a realizat printr-o modelare numerică, folosind funcții fitness dar am prezentat și posibilitatea alegerii unor soluții bune, chiar dacă nu optime global, pe criterii estetice, utilizatorul având astfel ultimul cuvânt de spus în alegerea soluției finale.

1.3 STRUCTURA LUCRĂRII

Capitolul 1 Introducere Primul capitol constă în prezentarea motivației alegerii temei, a obiectivelor și a structurii lucrării de față.

Capitolul 2 Optimizare globală vs. optimizare structurală Am introdus acest capitol pentru a sublinia, de la început, legătura dintre optimizarea globală și optimizarea structurală, pentru a putea pregăti terenul descrierii diferitelor metode de optimizare globală ce pot fi aplicate, în diferite faze ale proiectării, în domeniul ingineriei structurale.

Capitolul 3 Formularea matematică a problemelor de optim În acest capitol am realizat o introducere în optimizarea structurală. Se arată că, prin descompunerea problemelor în trei componente, și anume, model structural, model de optimizare și algoritm de optimizare, dificultatea rezolvării acestora poate fi redusă considerabil. Am discutat modul în care sunt generate coordonatele punctelor ce reprezintă soluțiile candidat în spațiul de proiectare și cum se poate determina apartenența acestora la subspațiul admisibil prin formularea de restricții. Am prezentat formularea matematică directă și variațională a problemelor de optim și am trecut în revistă metodele tradiționale de optimizare după cum urmează: metode unidirecționale, metode bazate pe gradientul funcției, metode de programare liniară, metoda de penalizare, metode de liniarizare și metode de programare geometrică și stocastică.

Capitolul 4 Forme de optimizare structurală Acest capitol prezintă formele optimizării structurale cel mai des întâlnite în literatura de specialitate, caracteristicile fiecărei formulări și tipul de structuri la care se pretează fiecare. Pentru fiecare formă de optimizare (topologică, a formei, dimensională, topografică și diferite combinații între acestea) sunt menționați și cei mai des utilizați algoritmi de căutare a soluțiilor optime.

Capitolul 5 Algoritmii stocastici Capitolul cuprinde documentația sintetizată, și o analiză critică a celor mai frecvent utilizate metode evolutive de optimizare. Sunt descriși algoritmii genetici (GA), optimizarea cu colonie de furnici (ant colony optimization), călirea simulată (simulated annealing SA), căutarea tabu (tabu search), optimizare cu roiuri de particule (particle swarm optimization PSO), harmony search și metode hibride. Accentul este pus pe GA, ESO și SA, aceștia fiind algoritmii utilizați în lucrarea de față pentru propunerea hibridizării între o metodă de optimizare bazată pe tehnici de descompunere a obiectivelor și un algoritm de adaptare a parametrilor.

Capitolul 6 Strategii de optimizare Este dată o definiție a strategiilor de optimizare și sunt descrise direcțiile identificate de autoare în care domeniul optimizării structurale evoluționiste poate fi extins și dezvoltat - design structural specializat, îmbunătățiri aduse algoritmilor și formularea obiectivelor optimizării. Sunt prezentate dificultățile întâlnite în cazul aplicării algoritmilor evoluționiști la rezolvarea problemelor de optimizare multi-modală, este discutată eficiența și necesitatea unei precizii sporite a analizei structurale în condițiile în care, cel puțin în faza conceptuală a designului, există multiple incertitudini legate de variabilele structurii și mai apoi sunt prezentate exemple de utilizare a inspirației formelor naturale pentru designul unor construcții structural superioare și eficiente (argumentul pro bio-mimetism)..

Capitolul 7 Variabile de proiectare și variabile de optimizare În acest capitol am dorit să subliniez importanța deosebită pe care o au cele două categorii de variabile în cadrul procesului de optimizare structurală. Într-o etapă timpurie a procesului de proiectare (conceptuală și faza de definire a proiectului), este de o mare importanță găsirea celei mai bune topologii structurale posibile, în contextul obiectivelor de proiectare și a restricțiilor. Identificarea parametrilor structurali esențiali și apoi a criteriilor de alegere a formei structurii depind de condițiile care trebuie satisfăcute de structură și au o importanță decisivă asupra rezultatului optimizării. Merită menționat aici faptul că criterii “absolute” de optimizare nu există și nici nu par a fi de dorit. După alegerea acestor variabile de proiectare si a unui algoritm de optimizare, deoarece majoritatea algoritmilor au un set de parametri care le controlează comportarea, această alegere devine și mai dificilă. Alegerile referitoare la valorile acestor parametrii pot avea un impact major asupra performanței algoritmului.

Capitolul 8 Restricțiile de proiectare și funcția obiectiv Capitolul continuă descrierea unor alte componente esențiale ale unei strategii de optimizare structurale eficiente – traducerea restricțiilor de proiectare și a formulării funcției obiectiv într-un hiperspațiu de căutare pe care algoritmul de optimizare să îl poate explora. Aici am discutat posibilitatea transformării formulării problemei de optimizare cu restricții într-o problemă fără restricții prin încorporarea acestora în funcția obiectiv, prin diverse metode.

Capitolul 9 Modelare Parametrică Capitolul cuprinde descrierea modului în care modelarea parametrica poate oferi o soluție, în contextul descris, la problema numărului mare de variabile necesare descrierii unei structuri. Modelele parametrice sunt capabile sa descrie geometrii complexe utilizând un număr relativ redus de variabile, lăsând totodată loc pentru o marjă mare de variație. Aceste soluții care pot fi explorate cu ajutorul modelării parametrice pot fi însă în număr foarte mare, iar problema devine găsirea în rândul acestora a celor mai bune din punct de vedere al performanțelor dorite. Pentru acest tip de căutare, algoritmii genetici sunt foarte potriviți, datorită capacității modelului parametric de a utiliza un număr relativ mic de variabile. Pornind de la această premiză, este descris modul în care autoarea a realizat formularea parametrică a problemelor de optimizare și diferențele cele mai fundamentale rezultate din această formulare a problemei în Grasshopper față de un mediu de programare tradițional cum este Matlab. Sunt discutate toate cele trei componente ale strategiei de optimizare – modelarea variabilelor (parametrică), definirea problemei de optimizare prin crearea unei funcții fitness specifice problemei și alegerea și implementarea algoritmului de optimizare.

Capitolul 10 Implementarea strategiilor de optimizare structurală cu algoritmi genetici

Este descris programul elaborat în platforma Matlab, modul în care au fost alese variabilele de optimizare și restricțiile incluse în funcția fitness prin metoda penalizărilor. Este ales un algoritm genetic simplu pentru căutarea soluțiilor iar parametrii algoritmului sunt ficși, stabiliți înainte de începerea optimizării. Se observă convergența algoritmului și sunt obținute rezultate comparabile cu cele din literatură pentru problemele cu 8, 9 și respectiv 120 de variabile, constând în dimensiunile secționale ale elementelor.

Am trecut apoi la prezentarea formulării parametrice în Grasshopper. Structura a fost analizată cu ajutorul unor componente din Karamba, o bibliotecă FEM încorporată în mediul parametric al Grasshopper. Acest fapt face ușoară combinarea modelelor parametrice, calculului cu element finit și a algoritmilor de optimizare, comunicarea între aceste componente și biblioteci fiind făcută prin programare vizuală (descrisă în capitolul 9). Se propune hibridizarea între o metodă de optimizare bazată pe tehnici de descompunere a obiectivelor și un algoritm de adaptare a parametrilor. Includerea în funcția fitness a unor valori care lucrează una împotriva celeilalte nu trebuie neapărat să constituie o problemă insurmontabilă. Cheia este atribuirea unor greutăți în funcție de importanță, care să poată ghida algoritmul spre soluțiile preferate de proiectant. Noua abordare este validată pe probleme de test și apoi aplicată la structuri mai complexe. Rezultatele sunt comparate cu cele obținute de algoritmi din literatură. Rezultatele au fost obținute prin utilizarea SA (călirii simulate) pentru găsirea unei soluții suficient de bune, care mai apoi este preluată ca punct de plecare pentru GA.

Capitolul 11 prezintă concluziile care încheie această teză și oferă câteva direcții de cercetare viitoare.

2 OPTIMIZARE GLOBALĂ VS. OPTIMIZARE STRUCTURALĂ

Optimizarea structurală implică determinarea variabilelor de proiectare, care controlează forma, proprietățile materialului sau dimensiunile unei structuri, astfel încât să respecte anumite restricții și să îmbunătățească anumite proprietăți pentru a obține structuri optime (Valery, 1999).

Atunci când ne ocupăm de probleme de inginerie, se pot discuta două domenii diferite de optimizare:

- Primul este numit *optimizare globală* (Global optimization). Prin acest termen se va înțelege optimizarea uneia sau mai multor funcții, fără o cunoaștere a-priori a problemei exprimate prin aceste funcții (numite uneori funcții "black-box").

- Al doilea domeniu, numit *optimizare structurală*, poate fi descris ca o știință aplicată, unde metodele din domeniul optimizării globale sunt aplicate la un model al unei structuri sau al unui material.

În procesul proiectării structurilor, în diverse domenii inginerești, proiectanții aleg cea mai bună variantă decizională, la fiecare pas, legată de aspecte structurale și non-structurale, cum ar fi rigiditatea, rezistența, serviceabilitatea, proprietățile estetice. Cu alte cuvinte, aceștia iau decizii pentru a realiza cel mai bun design, astfel încât procesul proiectării structurale poate fi privit ca design optimal chiar dacă nu urmărește expres găsirea unui optim. Optimizarea structurală este privită ca aplicarea metodelor de optimizare în proiectarea structurală.

Problema tipică de optimizare structurală este formulată ca minimizarea unei funcții obiectiv (funcții de cost), de obicei reprezentând greutatea structurii sau volumul acesteia. Luând în considerare modul în care se poate rezolva această problemă de optimizare generală, o abordare ar fi alegerea unor multiple combinații de variabile de proiectare și apelarea la un program de analiză pentru a evalua fiecare dintre acestea, spre a alege una cu cele mai bune valori ale funcției obiectiv și care, de asemenea, îndeplineşte toate restricțiile. Aceasta ar fi o abordare clasică de căutare aleatorie sau versiunea modernă cunoscută sub numele de căutare genetică (Hajela, 1990).

O altă abordare ar fi perturbarea fiecărei variabile de proiectare și evaluarea funcției obiectiv şi a restricțiilor. Astfel se poate determina sensibilitatea (gradientul) designului în raport cu variabilele. Cu ajutorul acestor informaţii, putem matematic (numeric) determina modul în care se pot modifica variabilele de proiectare spre a îmbunătăţi modelul în timp ce obiectivul satisface restricțiile. Există o multitudine de astfel de metode "pe bază de gradient" și un număr considerabil de software-uri disponibile în prezent (Vanderplaats, 2004).

Proprietățile mecanice, ce includ deplasările de noduri, tensiunile în elemente, frecvențe de vibrație, încărcări de flambaj sunt luate drept variabile de proiectare. Problema de optimizare structurală poate fi formulată, ca alternativă, pentru a urmări maximizarea unei proprietăți mecanice, supusă unor restricții de cost. Deși există multiple formulări ale problemei de optimizare, ex. design pentru greutate minimă, design pentru rigiditate maximă, termenul de optimizare structurală sau design optimal se referă la toate tipurile de probleme de optimizare asociate designului structural.

Designul[1] optim se realizează în mai multe faze consecutive:

Proiectarea conceptuală este faza în care are loc identificarea configurației de bază a sistemului structural împreună cu ansamblul obiectivelor. Este important, de asemenea, să se identifice domeniile de variație ale valorilor parametrilor ce descriu sistemul, astfel încât pentru orice parametru cu valori din domeniul corespunzător, sistemul să satisfacă funcțiile identificate în pasul precedent. Prin urmare, se identifică mulțimea parametrilor ce descriu diverse sisteme admisibile.

Proiectarea optimă are ca obiectiv alegerea parametrilor rămași nedeterminați în pasul precedent. Acești parametri trebuie să aibă valori în domeniile definite de restricțiile tehnologice și de funcțiile sistemului. Criteriul pentru alegerea parametrilor sistemului este, de cele mai multe ori, minimizarea costului, a greutății, a consumului anumitor materiale, maximizarea eficienței etc.

Mai trebuie specificat faptul că proiectarea unui sistem structural este un proces caracterizat de proprietatea că parcurgerea etapelor sale poate declanșa contrareacții (feedback). Asta înseamnă că după parcurgerea unei etape este posibil să nu se treacă la etapa următoare, ci să se reia procesul de proiectare de la o anumită etapă anterioară, sau chiar de la început, de atâtea ori până când sunt îndeplinite anumite restricții, impuse în etapa curentă. Acest proces iterativ de proiectare se oprește doar atunci când se consideră că structura simulată poate fi aplicată în

[1] Design este folosit in sensul complet din limba engleză (plan, proiect, design (și industrial), desen, schiță, schemă, proiectare, construcție, sinteză, concepție, tip, model, calcul; (TH) a proiecta, a executa un proiect / un plan, a desena, a calcula.

realitate. Se subliniază că această ultimă decizie este mai mult de natură umană decât de programare matematică.

Condiții necesare pentru implementarea designului optimal al structurilor:

1. Existența unei funcții pentru proiectarea optimă a elementelor structurale specifice, cum ar fi grinzi de oțel, grinzi de beton, prinderile din oțel, blocuri de fundaţie, etc. De obicei, dimensiunile minime, mărimea sau numărul elementelor sunt valorile căutate. Elementul trebuie să îndeplinească criteriile corespunzătoare codurilor de proiectare.
2. Trebuie să existe posibilitatea parametrizării structurii. Proiectantul trebuie să decidă, ceea ce este fixat ca dimensiune în structură și ceea ce poate fi schimbat - deschideri, adâncimi, dimensiuni ale secțiunilor transversale, grosimi de plăci și pereţi, sarcini, etc. Fiecare trăsătură care poate varia trebuie să poată fi descrisă de un parametru independent. Alte dimensiuni pot fi dependente de parametri, creând un model structural inteligent parametrizat.
3. Trebuie să fie posibilă definirea funcției obiectiv. Ea poate fi: greutatea oţelului structural necesar, volumul de beton utilizat, greutatea armăturii, dar poate fi, de asemenea, deplasarea maximă sau oricare altă caracteristică structurală sau estetică. Situaţia ideală este dacă sistemul este capabil să calculeze o valoare globală precum costul total al construcției.
4. Trebuie să existe capacitatea evaluării funcției obiectiv pentru setul specific de parametri. Aceasta înseamnă că o funcție capabilă să interpreteze setul de parametri și să returneze o valoare obiectiv trebuie să fie disponibilă.
5. Rezolvatorul optimizării - un instrument care generează diferite seturi de parametri, calculează funcția obiectiv și propune în cele din urmă setul optim de parametri trebuie să fie creat.

Metoda elementului finit poate fi folosită ca nucleu numeric pentru rezolvarea generală a problemelor de calcul pentru cele mai diverse tipuri de structuri și solicitări. Există o multitudine de programe de calcul care folosesc MEF, acestea furnizând toate datele necesare pentru a fi procesate în algoritmul de optimizare. Pentru a lucra în regim integrat este necesară folosirea unor automatisme software pentru definirea problemei și rezolvarea ei automată folosind MEF. De asemenea, este necesar ca programul să poată prelua automat datele din programul de element finit şi să le folosească mai departe. Formularea problemei de optimizare ar trebui făcută automat. Întrucât rezolvarea problemelor folosind MEF este un proces costisitor din punct de vedere al calculului, este necesar să se minimizeze numărul de rulări ale modelului folosind MEF. Pentru a evita deteriorarea modelului structural modelat în urma ajustării geometriei sau topologiei în procesul de optimizare, este necesară definirea unui set de restricții suplimentare față de restricțiile ce țin de configurația structurii (tensiuni, eforturi, deplasări). Pentru o bună poziţionare a procesului de căutare în spaţiul soluțiilor este foarte utilă analiza senzitivităţii. Folosind analiza senzitivităţii, spaţiul de căutare este redus la şablonul sugerat de coeficienţii de senzitivitate.

3 FORMULAREA MATEMATICĂ A PROBLEMELOR DE OPTIM

Problemele de optimizare pot fi rezolvate prin aplicarea "conceptului de trei coloane" (Three-Columns Concept). Cele trei coloane sunt **modelul structural, modelul de optimizare și algoritmul de optimizare**. Optimizarea automată a structurilor este o sarcină complexă și dificil de organizat. Conceptul prezentat a fost dezvoltat de Eschenauer [Eschenauer,2007]. Acesta pornește de la ideea descompunerii problemei în subprobleme care pot fi rezolvate direct, iar conceptul a fost dezvoltat pentru a lucra cu algoritmi de programare matematică (dar este valid și în cazul tehnicilor de soluționare de tipul algoritmilor genetici).

Modelul structural, necesar pentru traducerea structurii reale în vederea realizării procesului de optimizare computerizată, descrie matematic sau numeric comportamentul fizic al structurii, adică răspunsul la încărcări, sau proprietăți structurale cum sunt frecvențe proprii de vibrație sau greutate. În cazul în care structura este modelată prin FEM variabilele de stare ale problemei sunt deplasările nodale (u). Alte valori care ne pot interesa, cum ar fi tensiunile, sunt calculate din valorile deplasărilor în etapa de postprocesare.

Problemele de optimizare reale sunt în general neliniare și cu restricții, iar algoritmii care le pot rezolva se bazează pe proceduri iterative care pornesc de la un design inițial (x_0) și produc vectori de variabile de design îmbunătățiți (x_k). Procedura este oprită când un anumit criteriu de convergență predefinit este satisfăcut. Numeroase studii au arătat că alegerea celui mai bun **algoritm de optimizare** se face în strânsă legătură cu problema tratată.

Modelul de optimizare face legătura intre modelul structural și algoritmul de optimizare. Modelul de evaluare are rolul de a evalua designul în funcție de obiectivul de optimizare și de starea restricțiilor (active sau nu) din valorile variabilelor de stare și alte informații obținute din modelul structural.

Obiectivul optimizării este adesea formulat ca o funcție obiectiv scalară f, sau, în cazul optimizării multicriteriale, ca un vector f.

Restricțiile designului sunt formulate sub formă de funcții de restricție incluse în vectorii g (inegalități) și h (restricții de tip egalitate). Modelul de evaluare se poate baza pe variabilele de stare u (când sunt luate în considerare tensiunile) sau alte variabile care influențează designul (necesare calculării greutății totale, de exemplu). Modelul de optimizare mai conține definițiile variabilelor și transformările acestora sub denumirea de *parametrizare*. Pozițiile nodurilor aflate la granița domeniului modelului structural definesc forma acestuia și se modifică în timpul unui proces de optimizare a formei. Forma unei structuri sau a unui design este definită explicit în termeni de variabilele de design x. Modelul de design descrie relația matematică dintre variabilele de analiză y și variabilele de design x.

Adițional, acestea din urmă pot fi transpuse în variabile de transformare z în scopul adaptării problemei de optimizare la unele cerințe ale algoritmului de optimizare. Analiza senzitivitații demonstrează susceptibilitatea obiectivului și restricțiilor față de mici schimbări ale variabilelor de design. Această informație e folosită la controlul algoritmului de optimizare și la alegerea unui design.

Evaluarea designului. În optimizarea structurilor este folosită, în general, MEF pentru obținerea răspunsului structural la încărcări sub anumite condiții limită. Soluția sistemului de ecuații oferă soluția primară în termeni de grade de libertate nodale (în cazul structurilor acestea se traduc prin deplasări), iar din acestea, alte valori pot fi obținute (tensiuni). Valorile tensiunilor pot fi utilizate pentru a formula un obiectiv când se dorește maximizarea rezistenței unui element, sau la formularea restricțiilor când dorim minimizarea greutății unei structuri și asigurarea rezistenței necesare acesteia. În cazul general, răspunsul structural e necesar atât pentru evaluarea obiectivului cât și a restricțiilor.

3.1 EVALUAREA MODELULUI

3.1.1 Spaţiul și subspaţiul de proiectare

Prin convenţie, se consideră structura ca fiind un punct într-un spaţiu de proiectare abstract. În acest spaţiu, coordonatele punctului corespunzător structurii sunt dimensiunile geometrice ale acesteia şi constantele de material. Aceste coordonate care vor fi denumite *parametrii structurii*, pot fi numere reale, funcţii sau vectori (mulţimi total ordonate de numere reale). Pentru o în-ţelegere mai profundă a spaţiului figurativ de proiectare, se prezintă, mai jos, parametrii ce sunt utilizaţi de proiectant pentru a specifica o structură.

Parametri geometrici:

- geometria secțiunii transversale a elementelor structurale unidimensionale;
- forma axei longitudinale a elementelor structurale unidimensionale;
- forma geometrică a suprafeţei mediane a plăcii sau membranei;

- legea de variație a grosimii plăcii;
- forma conturului plăcii sau membranei;
- poziţia spaţială a nodurilor unei grinzi cu zăbrele sau ale unui cadru;
- localizarea spaţială a elementelor componente ale structurii.

Constante de material:

- modulul de elasticitate;
- densitatea;
- coeficientul de conductivitate și de dilatare termică;
- coeficienţii legilor constitutive care stabilesc legătura dintre tensiuni și deformaţiile elastice, elasto-plastice, vîsco-elastice, etc.;
- tensiunile de cedare ale materialului la diverse solicitări;
- constantele de oboseală ale materialului;
- constantele de anizotropie ale materialului.

Starea de pretensionare a unei structuri poate, de asemenea, fi considerată ca un parametru de calcul. Evident, această trecere în revistă a parametrilor structurii nu este completă, însă include parametrii cei mai frecvent utilizaţi în proiectare.

O problemă particulară generată de optimizarea structurii este aşa-numitul *subspaţiu de proiectare*.

În situaţia concretă în care se dorește proiectarea unei grinzi, pentru început, proiectantul decide dinainte dacă grinda va fi o grindă cu secţiune I, sau grindă cu zăbrele. Această alegere implică restricții asupra parametrilor de proiectare ca, de exemplu, înălţimea maximă a grinzii I. De asemenea, deşi nu este strict necesar, proiectantul îşi alege dinainte materialul folosit, adăugându-se astfel noi restricții.

Constantele materialului pot fi introduse printre variabilele de proiectare ce vor fi determinate în procesul de optimizare, însă trebuie subliniat că puţini autori au abordat acest aspect, existând puţine lucrări care abordează aceste probleme.

3.1.2 Subspaţiul admisibil

Subspaţiul de proiectare ce conţine structura satisface un număr de cerinţe, necesar pentru acceptabilitatea funcţională a structurii, aflată sub acţiunea solicitărilor ce decurg din îndeplinirea rolului funcţional. În general, condițiile impuse asupra rezistenței, rigidităţii, duratei de viaţă, ş.a. limitează răspunsul structurii la solicitarea dată.

Aceste condiții pot fi, însă, concepute ca restricții ce împart subspaţiul de proiectare într-un *subspaţiu admisibil* şi un *subspaţiu neadmisibil*.

Printre restricțiile cele mai frecvent întâlnite în literatură, menționez:

- tensiuni maxime;
- deformaţie maximă;
- coeficient de siguranţă maxim la pierderea stabilităţii, sau la rupere;
- minimum de senzitivitate la imperfecţiuni de execuţie, de montaj, etc.;
- minimumul frecvenţei fundamentale de oscilaţie proprie;
- maximul vitezei de deformare în curgerea plastică staţionară;
- maximul duratei de viaţă sub solicitări ciclice;
- greutate sau volum minim;
- rigiditate maximă la diverse solicitări (încovoiere, torsiune etc.);
- moment de inerție maxim;
- solicitări de stabilitate maximă;
- ductilitate maximă la solicitări dinamice.

Am constatat că diferite teorii de rupere sunt luate în consideraţie, în concordanţă şi pe baza unor indicatori de material, solicitare, etc., prin restricții adecvate din subspaţiul de proiectare.

Restricțiile sunt exprimate ca limite de funcții definite pe subspaţiul de proiectare, acest subspaţiu fiind delimitat numai implicit. Am observat că dificultăţi de calcul apar atunci când solicitările sunt aleatoare sau dinamice, în cazul unor tipuri de solicitări diferite, restricțiile fiind diferite pentru fiecare dintre acestea. Acesta este, în mod obişnuit, cazul când se consideră diferite suprasarcini, în condiţii de exploatare.

Cel mai adesea, restricțiile asupra limitelor răspunsului nu sunt de natură fizică ci rezultă din reglementări sau standarde. Când este cazul, o problemă de proiectare optimă este cea a senzitivității soluției optime la mici modificări în aceste standarde. Privind lucrurile și prin prisma acestui ultim aspect menţionat, se pune și problema optimizării standardelor sau a reglementărilor.

În formularea matematică a problemei, restricțiile apar în mod obişnuit sub formă de inegalităţi.

Drept restricții se pot considera: ecuațiile de echilibru şi de compatibilitate, (ecuații diferenţiale cu derivate parţiale sau ecuații diferenţiale ordinare), inegalităţi algebrice de tip unilateral sau bilateral (suprafeţe, dimensiuni, momente de inerţie etc), tensiuni normale, tangenţiale, principale, echivalente, critice la stabilitate elastică, în regim static sau dinamic, deformaţii locale sau generale, viteza critică de deformare plastică etc., sau de tip izoparametric cum ar fi: volum constant, deformare constantă, potenţial elastic constant etc.

După ce variabilele de proiectare au fost alese, problema de proiectare optimă poate fi formulată astfel:

Să se găsească S *astfel* încât:

$$\begin{cases} f_k(S) = 0, & k = 1,2,...,m \\ h_j(S) \leq 0, & j = 1,2,...,n \\ \min F(S) \end{cases} \qquad (\ 3\text{-}1)$$

unde S este un punct în spaţiul de proiectare, caracterizat de variabilele alese. În multe probleme există condițiile impuse funcţionalelor f_k şi h_j, datorită restricțiilor impuse răspunsului structurii la solicitări, însă unele dintre acestea pot să fie exprimate prin delimitări ale subspaţiului de proiectare. Funcția obiectiv este notată cu F.

Existenţa soluției și a unicităţii acesteia, când există, pentru problema definită, la modul general, prin (3-1), este o chestiune deschisă la care numai în rare cazuri se poate răspunde pe baza intuiţiei. Din (3-1) rezultă că, dacă S este optim, pentru mici variaţii δS în domeniul subspaţiului de proiectare, există relațiile:

$$\begin{cases} \delta f_k(S) = 0, & k = 1,2,...,m \\ \delta h_j(S) \leq 0, & pt. indici\ j\ la\ care\ h_j(S) = 0 \\ \delta F(S) \geq 0 \end{cases} \qquad (\ 3\text{-}2)$$

Această formulare variaţională dă condiţia necesară pentru existenţa unei soluții optime.

Condițiile din formularea variaţională (3-2) pot fi exprimate printr-o altă formă mult mai folosită: Se presupune că variabilele de proiectare sunt p numere reale, astfel încât spaţiul de proiectare poate fi interpretat ca un spaţiu euclidian p-dimensional.

Fie S o soluție admisibilă și δS o variație arbitrară în domeniul subspaţiului de proiectare. Cum $f_k(S) \leq 0$, variaţia δS este normală la toţi vectorii $\nabla f_k(S)$ ($k = 1,2,...,m$). În mod similar, restricțiile descrise prin inegalităţile $h_j(S) \leq 0$ sugerează că δS nu are componentă în direcția pozitivă a lui $\nabla h_j(S)$.

Prin urmare, se poate deduce că pentru orice numere reale λ_k şi orice $\gamma_j \geq 0$, proiecţia lui δS pe vectorul

$$\sum_{k=1}^{m} \lambda_k \nabla f_k(S) + \sum_{j=1}^{n}{}^{*} \gamma_j \nabla h_j(S) \qquad (3\text{-}3)$$

este negativă. În relația (3-3), simbolul $\sum_{j=1}^{n}{}^{*}$ indică faptul că însumarea este restricţionată la acele valori ale indicelui j valorile lui j pentru care $h_j(S) = 0$. Cu alte cuvinte, orice vector care are o componentă pozitivă pe direcția vectorului dat de relația (3-3) se găseşte în subspaţiul neadmisibil.

În scopul descreșterii funcției obiectiv F, este necesar să se producă o mişcare din sens pozitiv în sensul negativ al direcţiei ∇F.

Dacă această direcţie (- ∇F) este direcția vectorului dat de relația (3-3), o deplasare în subspaţiul admisibil va descrește funcția obiectiv. Prin urmare, la punctul de optim, -∇F are direcția identică cu direcția vectorului (3-3). Utilizând acest fapt, se deduce că dacă S este soluție optimală, atunci există o mulţime de numere reale λ_k şi de numere pozitive $\gamma_j \geq 0$, astfel, încât are loc ecuaţia:

$$-\nabla \Phi(S) = \sum_{k=1}^{m} \lambda_k \nabla f_k(S) + \sum_{j=1}^{n}{}^{*} \gamma_j \nabla h_j(S) \qquad (3\text{-}4)$$

Relaţia (3-4) este cunoscută sub numele de *condiţia Kuhn-Tucker*. Se observă că, dacă nu există restricțiile inegalități, λ_k poate fi interpretat ca multiplicator Lagrange.

Pentru o problemă fără restricții, condiţia (3-4) se reduce la $\nabla \Phi = 0$. Ca toate soluțiile staţionare, însă, condițiile (3-2) și (3-4) *nu pot asigura optimul global*.

Utilizarea unor teste adiţionale asigură însă optimul global. În particular, dacă spaţiul de proiectare admisibil este convex şi dacă funcția obiectiv este fie convexă, fie concavă, atunci unele teoreme ale programării neliniare pot da informaţii importante despre optimul global și/sau poziţiile soluțiilor posibile.

Pot exista probleme care din punct de vedere matematic sunt total diferite de cea formulată prin relațiile (3-1), dar care exprimă acelaşi model fizic. În acest context, de exemplu, problema determinării celui mai înalt stâlp posibil, de material și volum dat (considerând și flambajul sub greutatea proprie), este, din punct de vedere principial, identică cu problema determinării volumului minim al stâlpului, de material și înălţime date. Deşi problemele sunt, în fond, identice, formularea lor cu ajutorul relaţiilor (3-1) este diferită. Acest lucru prezintă un interes deosebit, deoarece, inevitabil, una din formulări conduce la o soluție mai uşor de obţinut.

Pentru unele modele speciale de structuri (ca de exemplu o grindă elastică pentru care rigiditatea la încovoiere este proporţională cu masa), cu una sau mai multe restricții ce sunt caracterizate prin principiile de extrem ale teoriei structurilor (principiul lui Rayleigh și principiul minimului energiei potenţiale), condițiile necesare obținute prin metode variaţionale pot fi suplimentate cu condiții suficiente. Depinzând de principiul de minim al structurii cu caracter global sau local, condiţia rezultată este, de asemenea, suficientă pentru un optim global sau local.

Faptul că formularea directă, dată de relația (3-1) şi formularea variaţională, dată de relația (3-2) a problemei de proiectare optimă sunt esenţial diferite, afectează alegerea metodelor folosite la rezolvarea problemei.

Din punct de vedere principal, problemele formulate prin relația (3-1) sunt rezolvate, prin utilizarea unor procedee iterative în care, la fiecare iteraţie se obţine o soluție „mai bună” decât cea obţinută la iteraţia anterioară.

Problemele formulate variaţional conduc, pe de altă parte, la sisteme de ecuaţii diferenţiale cu condiții pe contur. Numai în cazuri cu totul excepţionale (de regulă, când problema prezintă suficiente simetrii), o soluție este dată sub formă analitică cunoscută. De regulă, se aplică algoritmi pentru obținerea de soluții numerice.

Datorită faptului că ecuațiile diferenţiale sunt adesea neliniare şi nu au soluții regulate (adesea apar singularităţi pe contur), aceste probleme prezintă un grad sporit de dificultate.

În consecinţă, există o diferenţă importantă între formularea variaţională (3-2) şi formularea mai generală (3-1). Formularea variaţională poate da o soluție (presupunând că ea există) *optimală* (sau mai curând, staţionară), funcţiile (3-1) fiind aplicate la *orice* proiectare admisibilă. Această diferenţă devine pregnantă atunci când soluția este singulară sau nu există.

În sens larg, aceasta înseamnă că o soluție bazată pe formularea (3-1) poate conduce la o proiectare „mai bună”, chiar dacă nu la „cea mai bună”.

Procedeele iterative menționate mai sus sunt metodele programării matematice și cele formulate de R.L. Fox. Trăsătura lor comună este generarea unui şir de puncte în subspaţiul de proiectare

$$S_1, S_2, ..., S_q, ... \quad (3\text{-}5)$$

începând cu un punct arbitrar S_1.

Pasul ΔS_q din $S_{q+1} = S_q + \Delta S_q$ este determinat folosind gradienţii funcţiilor restricții ∇f_k, ∇h_j şi funcţia obiectiv $\nabla \Phi$ în punctul S_q. Diferenţa între diferitele metode constă în relația dintre gradienţi și pasul ΔS_q.

În cel mai simplu caz, problemele cu restricții liniare şi funcție obiectiv liniară pot fi rezolvate prin metode ale programării liniare. Pentru restricții liniare şi tipuri particulare de funcții obiectiv neliniare există metodele programării pătratice.

Cazurile mult mai generale de probleme neliniare pot fi rezolvate fie cu *metode directe,* fie cu *metode indirecte.*

3.2 METODE DE OPTIMIZARE

În procesul tipic de optimizare a structurilor finit dimensionale, proprietățile secționale, localizarea nodurilor și poziționarea elementelor structurale sunt alese ca variabile ale problemei. Există numeroase metode de optimizare, care pot fi clasificate în:

- **Metodele analitice** de optimizare utilizează teorii matematice de calcul și metode variaționale în studiul optimului pentru formele geometrice simple ale elementelor structurale, cum ar fi grinzi, bare, plăci. Aceste metode pot fi folosite cu succes pentru componente structurale singulare, dar nu sunt posibil de utilizat la structuri complexe. Cu acest tip de metode optimul este calculat foarte exact prin soluționarea unui sistem de ecuații și inecuații ce exprimă condițiile de optim.

- **Metodele numerice** sunt reprezentate de metode de programare în cadrul aplicațiilor matematice. Cele mai noi cercetări în domeniu [Murren,2011] sunt legate direct de creșterea aproape exponențială a capacității de calcul a computerelor și au ca direcții de dezvoltare:

- programarea liniară;
- programarea neliniară;
- programarea dinamică;
- proceduri neconvenționale.

Optimizarea cu această clasă de metode se face printr-un proces iterativ, definindu-se o stare inițială folosită ca punct de start pentru o căutare sistematică în scopul îmbunătățirii structurii. Procesul iterativ este stopat când toate criteriile sunt satisfăcute, astfel încât configurația curentă obținută să fie cât mai aproape de optimul căutat.

Clase de metode de optimizare:

- Metode directe;
- Metode bazate pe optimalitatea Kuhn-Tucker;
- Metode de penalizare;
- Metode de punct interior de urmărire a traiectoriei centrale.

Principalele mărimi ce trebuie evaluate în cadrul metodelor de programare matematică ce folosesc tehnici derivative sunt: *Gradientul* şi *Hessianul* funcției obiectiv, coeficienţii Lagrange, *Jacobianul* restricțiilor. Toate aceste mărimi joacă un rol important în determinarea admisibilităţii soluțiilor şi a existenţei acesteia. Punctul candidat la optim trebuie să se afle în domeniul fezabil (gradientul restricțiilor trebuie să fie liniar independente). Din studii ale metodelor de mai sus se poate trage concluzia că optimul poate fi găsit prin rezolvarea unor ecuații diferenţiale cu o formă clar precizată. Această observaţie ne poate conduce la concluzia parţial adevărată că optimul ar putea fi găsit întotdeauna cu ajutorul unui algoritm clar formulat. Totuși, trebuie precizat că ecuațiile ce definesc condițiile de optimalitate sunt supuse unor condiții extrem de restrictive. Din acest motiv, o alternativă la aceste metode sunt cele de **căutare directă**. Spre deosebire de metodele derivative ce presupun calculul unor mărimi complexe, metodele de căutare directe folosesc cicluri de calcul cu un cost computațional mic.

Metodele directe de căutare permit optimizarea funcțiilor pentru care nu putem aplica metodele derivative de optimizare. Metodele de căutare evaluează funcția f în k puncte {x} urmărind evoluţia funcției în scopul găsirii punctului de optim.

Situaţiile în care se recomandă folosirea uneia dintre metodele directe de căutare sunt următoarele:

- funcția f nu este derivabilă,
- derivatele sunt foarte greu de evaluat sau sunt discontinue,
- nu este necesară o soluție foarte precisă a problemei.

Alegerea uneia dintre metodele prezentate se face în funcție de tipul de problemă practică ce trebuie rezolvată. O tehnică destul de des utilizată în rezolvarea problemelor simple este cea care presupune că setul de restricții este activ în punctul de optim, caz în care acestea sunt considerate ca egalități, fiind utilizate pentru a elimina variabilele libere. În concluzie, numărul de restricții active poate, în general, să fie cel mult egal cu numărul de variabile libere. În general, în problemele mai complexe, numărul total de restricții este mai mare decât numărul de variabile, fiind dificil de a cunoaște care constante sunt active în punctul de optim.

O soluție optimă “x” a problemei de optimizare structurală este caracterizată prin proprietatea că nu există alte soluții fezabile într-o vecinătate apropiată lui “x”, ce corespunde unei valori minime a funcției obiectiv. Din punct de vedere matematic, acest concept se exprimă prin condițiile Kuhn-Tucker (Kuhn, Tucker, 1951):

$$\frac{\partial f(x)}{\partial x_i} - \sum_{j=1}^{m} u_j \frac{\partial g_j}{\partial x_i} = 0 \quad i = 1...m \qquad (3\text{-}6)$$

$u_j g_j(x) = 0 \;\; j = 1...m$

$u_j \geq 0 \;\; j = 1...m$

Este important de remarcat faptul că aceste condiții sunt valabile doar pentru problemele de programare neliniare convexe.

Pentru a rezolva problema de optimizare structurală au fost dezvoltate diferite tehnici, putând fi amintite **trei abordări** în modul de rezolvare.

Astfel, [Moses (1968)], și [Romstad și Wang (1978)] au construit aplicații bazate pe metoda Simplex de programare liniară. În lucrările lor, acești autori aproximează o problemă de programare neliniară cu o secvență de probleme de programare liniară. [Gellatly și Gallagher (1976)], și [Moses și Onoda (1979)] au utilizat metode de tip “direcții posibile” sau “direcții fezabile” pentru a rezolva problema de optimizare structurală.

O a treia categorie de metode de programare neliniară este bazată pe așa-numitele **funcții de penalizare**. Acest tip de metodă este utilizată de [Schmidt și Fox (1975)], folosind tehnici de penalizare exterioare, în timp ce [Kavlie, Moe și Kowalik (1979)] aplică tehnici de penalizare interioară. *Ideea "funcțiilor de penalizare" constă în transformarea problemei de optimizare cu restricții, într-o problemă fără restricții, prin adăugarea la funcția obiectiv a unor termeni suplimentari, care să înlocuiască efectul restricțiilor.* Astfel o problemă de minimizare fără restricții poate fi rezolvată cu o funcție transformată, care are forma generală:

$$P(x, f_k) = f(x) = f_k \sum_{i=1}^{m} G(g_i(x)) \qquad (3\text{-}7)$$

unde al doilea termen al ecuației este denumit *termen de penalizare.*

[Fiacco și McCormick (1969)] au adus o contribuție importantă la dezvoltarea acestei abordări a problemei de optimizare, numind-o tehnică de minimizare secvențială fără restricții. Unul din elementele comune pentru clasele de metode de programare neliniară amintite este faptul că folosesc *variabile continue*.

În activitatea curentă de proiectare, însă, multe variabile sunt limitate de *valori discrete*, cum ar fi grosimile tablelor sau plăcilor, diverși parametri geometrici (lungimi, diametre), ș.a., și în lipsa unor metode eficiente de cuantificare a acestor mărimi discrete, este totuși acceptată formularea continuă a problemelor de optimizare, a căror soluție este în final rotunjită. Acest mod de lucru furnizează rezultate satisfăcătoare pentru problemele de optimizare de mici dimensiuni, dar poate da soluții relativ depărtate de optim dacă numărul de variabile crește foarte mult.

O problemă de *optimizare cu variabile discrete* este formulată asemănător cu problema generală, și anume:

Să se determine minimul funcției: f (x)

cu restricțiile: $g_j(x_i) = 0 \quad j=1,2,\dots m$ (3-8)

unde: $x_i^{\inf} \le x_i \le x_i^{\sup} \quad i=1,2,\dots n$ (3-9)

$x_i \in D_i$ (3-10)

în care: $f(x)$ este funcția obiectiv; $g_j(x)$ sunt restricțiile; x_i este vectorul variabilelor de proiectare; $x_i^{\inf}$ și $x_i^{\sup}$ reprezintă limita superioară și inferioară a variabilelor; *m* este numărul de restricții; D_i este mulțimea finită de variabile discrete.

Problema de optimizare formulată matematic prin relațiile (3-8)-(3-10) este, în general o problemă de programare neliniară, fiind studiate și folosite diverse tehnici de rezolvare.

Este important de subliniat că majoritatea algoritmilor au ca cerință o valoare inițială pentru variabile, fiecare evaluare a funcției necesitând de fapt o nouă analiză a structurii. Din acest motiv, dacă se lucrează cu structuri complexe, este necesar un număr mare de analize cu elemente finite, deci un consum mare de timp și resurse, eficient numai în cazul folosirii unui computer și a unui program rapid.

Deci pentru rezolvarea eficientă a problemei ar fi necesare o aproximare de calitate a problemei, precum și rezolvarea într-un număr redus de pași, existând soluții, cum ar fi :

- Reducerea numărului de variabile prin realizarea de legături între acestea, abordare rezonabilă, deoarece în practică o serie de variabile au aceeași valoare (table și plăci de aceeași grosime, din motive constructive și tehnico-economice cum ar fi ușurința în aprovizionare,etc.), și reducerea restricțiilor prin luarea în considerare doar a celor critice la fiecare iterație.
- Utilizarea de funcții de aproximare pentru reprezentarea restricțiilor, din punct de vedere matematic fiind folosite serii Taylor. Această tehnică de rezolvare generează o formă de aproximare a restricțiilor în funcție de variabile, bazându-se pe constatarea că vectorii de răspuns structural, cum ar fi tensiuni sau deplasări sunt cvasiliniari în raport cu variabilele, deși în practică restricțiile de proiectare sunt în general neliniare în raport cu variabilele.
- Utilizarea unei tehnici de generare aproximativă [Arora, 1997], la care răspunsul structurii la încărcarea exterioară, definit prin deplasări, frecvențe, etc., devine în problema de optimizare ca primă aproximare. În acest mod este creată o problemă explicită neliniară, a cărei soluționare necesită mai puțin de 10 pași.

În vederea creșterii eficienței metodei este folosită o strategie duală, în care optimizarea cu variabile discrete este realizată după optimizarea cu variabile continue. Statistic a fost stabilit că acestă metodă duală este cu cel puțin un ordin de mărime mai eficientă decât alte metode în cazul problemelor de optimizare cu mai mult de 20 de variabile. *Algoritmii clasici de optimizare* oferă posibilitatea optimizării unei structuri prin următoarele clase de metode: *metoda Simplex*, *metoda direcțiilor fezabile* sau *metoda funcțiilor de penalizare*.

Ca element de noutate în toate aceste metode este posibilitatea folosirii unei mulțimi discrete pentru variabilele de proiectare, lucru care reprezintă o abordare pragmatică a procesului de optimizare structurală, prin posibilitatea obținerii de valori care au aplicabilitate in practică. De asemenea, toți acești algoritmi clasici au ca element de legătură utilizarea metodei elementului finit ca procedură de calcul a tensiunilor și deformațiilor structurii analizate. Analiza acestor metode, atât prin prisma faptului că se folosește MEF pentru calculul deplasărilor și tensiunilor, cât și în ceea ce privește ușurința în aplicare, precum și necesitatea unei anumite accesibilități hardware și software, duce la concluzia că pot fi considerate două variante de lucru în vederea optimizării structurale: fie utilizarea unui modul de optimizare cuplat cu un modul de analiză structurală cu MEF, fie crearea unui modul de optimizare propriu, cuplat cu un modul de analiză structurală cu MEF, care să răspundă unor anumite cerințe specifice.

4 FORME DE OPTIMIZARE STRUCTURALĂ

În conformitate cu Steven Grant [Steven, 2003], patru forme diferite de **optimizare structurală** pot fi distinse. Fiecare poate fi rezolvată cu o strategie de optimizare distinctă, dar rezolvarea problemelor reale, de obicei, solicită o combinaţie a acestor forme.

4.1.1 Optimizarea topologică

Prin optimizarea topologiei înţelegem găsirea unei structuri fără a cunoaşte forma sa finală în prealabil [Bendsøe și Sigmund, 2003].

Figura 4.1: Exemplu de design optimal pentru stâlp

Doar condițiile externe, criteriile de optimalitate și restricțiile sunt cunoscute. Acest tip de probleme vin de obicei din domeniul ingineriei mecanice, unde conceperea unor piese pentru mașini sau avioane sunt temele de proiectare cele mai frecvente. Structurile reprezentative din ingineria civilă servesc drept instrument de decizie în alegerea unui sistem static adecvat al unei structuri noi. Ele sunt aplicate mai ales la structurile articulate, în cazul în care coordonatele nodale ale îmbinărilor sunt variabilele de optimizare. Luând în considerare poziţia suporturilor şi a funcțiilor obiectiv, sisteme istorice bine-cunoscute pot fi descoperite prin optimizare topologică.

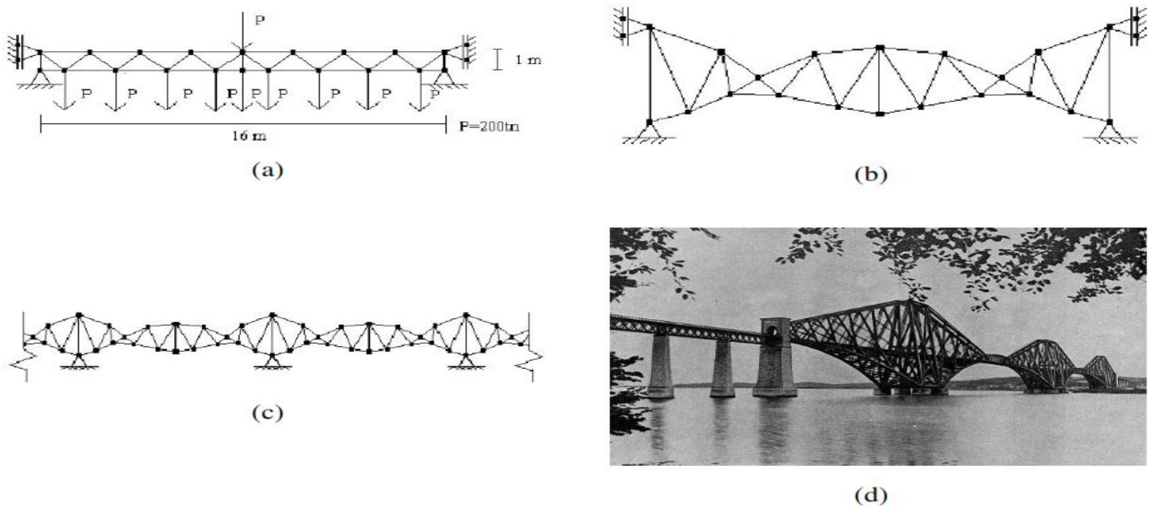


Figura 4.2: (a) Diagrama de calcul a problemei, (b) soluția optimală a problemei, (c) configurația optimizată formată prin concatenarea modulelor de bază și (d) First of Forth Bridge, construit 1883–1890 ca un exemplu de optimizare topologică prezentat în [Gil și Andreu, 2001].

Exemplul tipic pentru această formă de optimizare în domeniul construcțiilor metalice este plasarea elementelor din oțel. Cu alte cuvinte, căutăm cel mai potrivit model pentru o structură în care poziţia elementelor de oţel nu este cunoscută în avans. În acest caz, obiectivul este, de obicei, reducerea la minimum a cantității de oţel, supus cerinţelor structurale. În primii ani de optimizare numerică procedura tradiţională pentru rezolvarea acestor probleme a fost proiectarea pentru tensiune limita (fully stressed design), însemnând ca toate elementele structurii sub încărcări să fie cât mai aproape posibil de limitele materialului.

În primii ani de optimizare numerică procedura tradiţională pentru rezolvarea acestor probleme a fost design la tensiune limită (fully stressed design), astfel încât tensiunile în toate elementele sunt menite să fie cât mai aproape posibil de limitele materialului. Dezavantajul este vizibil pentru cazurile de încărcare multiple sau mai multe cazuri de sprijinire. În prezent, metodele cele mai frecvent utilizate pentru rezolvarea acestei categorii de probleme sunt criteriile de optimalitate, abordare bazată pe teoria dualității sau programare convexă [Olhoff, 1996], omogenizarea combinată cu metodele de programare matematică [Allaire, 2002] sau [Cherkaev, 2000], optimizare structurală evolutivă (ESO) [Xie şi Steven, 1997] - o altă metodă bazată pe eliminarea elementelor ineficiente din mesh-ul de elemente finite, automate celulare - o veche metoda de simulare dinamică studiată încă din anii 1960 [von Neumann, 1966] bazată pe construirea de sisteme bloc cu comportament predefinit [Wolfram, 2002] şi, în cele din urmă, Algoritmii Evolutivi (EAs) bazați pe principiile selecției naturale.

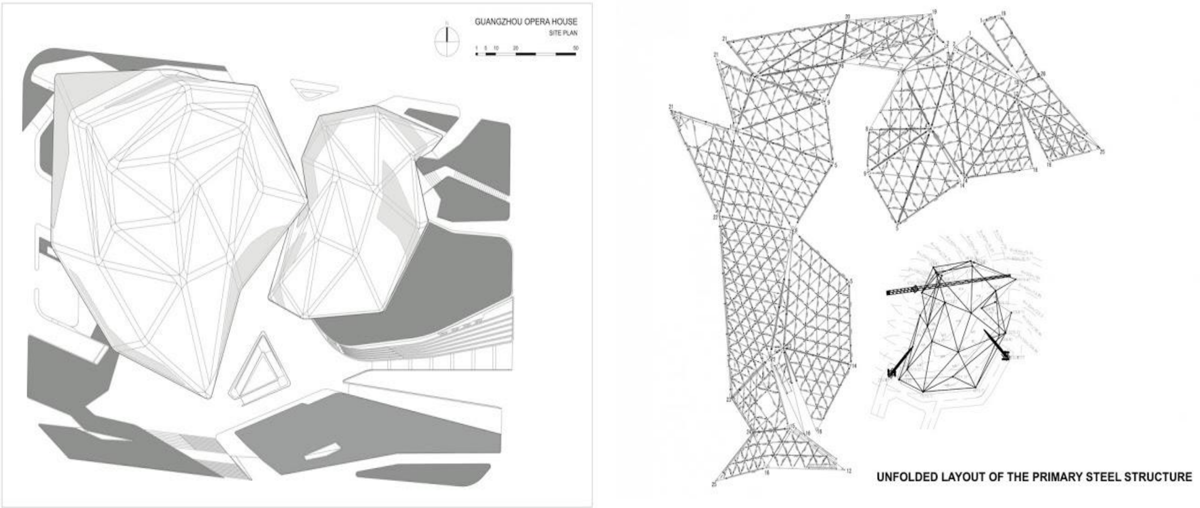


Figura 4.3: Guangzhou Opera House, Arhitect: Zaha Hadid, Structura: Shanghai Tongking (SHTK), China „Siguranţa fiind numărul unu între obiectivele noastre, dorim să reducem greutatea oţelului pentru a încerca să facem costul structurilor metalice apropiat de cel al clădirilor din beton, în condiții similare.”

4.1.2 Optimizarea formei

În această formă de optimizare, topologia structurii este cunoscută a-priori, dar poate exista o parte și/sau un detaliu al structurii, în care, de exemplu, tensiuni mari pot produce probleme. Prin urmare, obiectivul este, de obicei, găsirea celei mai bune forme, care va duce la distribuirea tensiunilor cât mai eficient. Parametrii de formă sunt dimensiuni ale pieselor optimizate sau un set de variabile care descriu forma (de exemplu coeficienţii de funcții spline). Din punct de vedere matematic, două reprezentări de variabile - continue și discrete - pot fi găsite în domeniul optimizării formei. Prezentarea generală a primului caz poate fi găsită în [Sokolowski și Zolesio, 1992], şi al doilea caz rezumat în [Bauer şi Gutkowski, 1995]. Algoritmii disponibili pentru rezolvarea acestor probleme sunt programarea matematică [Haslinger și Neittaanmaki, 1996], din nou ESO; o nouă metodă în acest context, este creșterea biologică simulată bazată pe definiţia de temperatură "falsă", sau artificială [Mattheck şi Burkhardt , 1990], şi din nou Algoritmii Evolutivi (EAs).

Denis Weare și Robert Phelan (RIBA Publishing, 2008) au calculat că modul cel mai eficient de a împărţi un spaţiu în celule de volum egal minimizând în acelaşi timp suprafaţa specifică între ele a fost prin utilizarea unui aranjament suprapus compus din 75% forme cu 14 feţe și 25% forme cu 12 feţe. Dar din moment ce structura rezultată va avea 22.000 de elemente din oțel conectate la 12.000 de noduri, generarea unui model real bazat pe această idee depăşeste posibilitățile proiectării convenţionale.

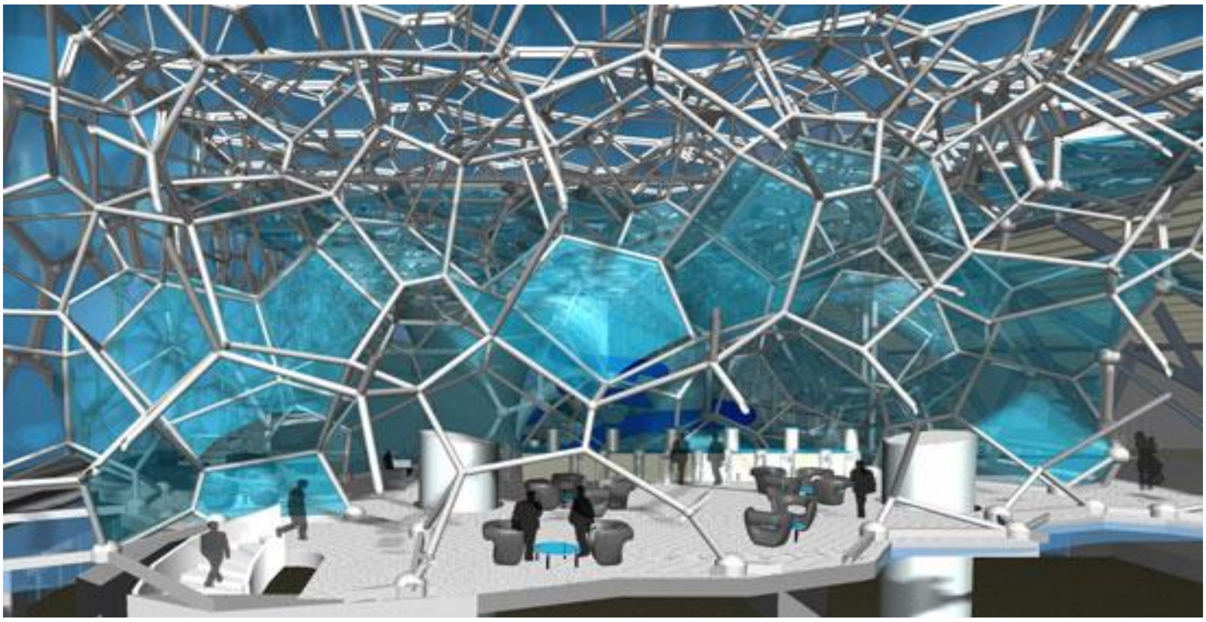


Figura 4.4: Designul optimal al Water Cube (China) a fost determinat prin analizarea a multiple configurații ale miilor de elemente structurale din oțel și a conexiunilor (nodurilor).

În schimb, în conformitate cu acești autori, pentru a manipula dinamic acest sistem geometric complex, în cadrul biroului de design al firmei Arup a fost conceput și formulat parametric un software care a automatizat procesul de desen și de analiză. Bazat pe restricţii specifice de proiectare și aproximativ 190 de scenarii de încărcare, acest algoritm verifică iterativ distribuţia forţelor prin întreaga structură pe baza dimensiunilor specifice ale elementelor, permiţând echipei să testeze diferite configuraţii de proiectare și să primească feedback-ul în 25 de minute. Rezultatul a fost o cladire spectaculoasă, cu o structură sofisticată, care este optimizată din punct de vedere al raportului greutate de material - rezistenţă. În plus faţă de avantajele structurale, Arup a estimat că a economisit 10 milioane de dolari la costurile de proiectare, comparativ cu metodele tradiţionale de proiectare.

4.1.3 Optimizarea dimensională

Acestea sunt combinate pentru a atinge criteriile de optimalitate dorite. În cadrul acestui domeniu două grupe principale de structuri pot fi diferențiate:

- **Structuri discrete**. Aici pot apărea structuri articulate și structuri cu legături rigide. În cazul structurilor din oțel, aproape toate problemele posibile de optimizare au fost supuse unei anumite forme de investigaţie. Pentru a enumera o serie de probleme rezolvate cu succes, optimizarea structurilor cu legături semi-rigide [Kameshki și Saka, 2001], optimizarea împotriva flambajului [Rong et al, 2001.], sau găsirea unei greutăți minime în cazul folosirii unui număr minim de profile de oţel într-un design [Greiner et al, 2001.] şi [Greiner et al, 2003.]. Multe exemple de probleme de dimensiuni mici din acest domeniu servesc drept benchmarks pentru diferite tipuri de algoritmi de optimizare, cum sunt grinda cu zăbrele din 10 bare [Belegundu, 1982] şi grinda cu zăbrele spaţială din 25 de bare [Adeli și Kamal, 1986], acestea fiind cele mai des citate. Aici, toate variabilele sunt selectate din setul discret de valori admisibile predefinite.

- **Structuri continue**. Acest grup cuprinde structuri asemănătoare grinzilor - definite de variabile continue, care nu sunt cunoscute în avans, în contrast cu cazul anterior exemplul de bază este o grinda cu momente de inerţie definite ca variabile continue [Lagaros et al, 2002.]. Încă o dată, metodele disponibile de optimizare sunt programarea matematică pe bază de gradient, criteriile de optimalitate, metode hard-kill, cum sunt cele menționate anterior: ESO şi din nou EA. Ca o consecinţă a definiţiilor introduse de mai sus, putem distinge o formă suplimentară de optimizare structurală. În cazul în care o variabilă de design - dimensiunea unui element sau valoarea unei proprietăţi materiale - poate ajunge la valoarea zero, adică nu este necesară în structură și poate fi eliminată, atunci acest tip de optimizare este adesea numit Optimizare de Configuraţie, de exemplu [Kirsch, 1995]. Piatra de temelie a acestei abordări este aşa-numita structură de baza (ground structure), care defineşte toate poziţiile posibile ale nodurilor şi setul tuturor elementelor / conexiunilor posibile între aceste noduri [M. E. Stavroulaki, 1997]. Apoi, scopul este eliminarea de elemente ineficiente pentru a obţine o structură optimă. În cazul în care coordonatele de noduri sunt de asemenea necunoscute, atunci aceasta formulare devine parte din optimizarea topologiei, a se vedea secţiunea 1.2.1. Prin urmare, optimizarea configurației poate fi văzută ca punct de legătură între cele două tipuri menționate anterior, de optimizare.

4.1.4 Optimizarea topografică

Această formă este cel mai puțin investigată parte a optimizarii structurale. Aici se pot întâlni căutarea de forme eficiente pentru structuri de tip shell, membrană sau cort. Doar câteva lucrări pe această temă pot fi găsite în literatura de specialitate, de exemplu [Goslingt şi Lewist, 1996] sau [Schwarz et al, 2001]. Metodele programării matematice sunt cunoscute ca singurele soluții eficiente pentru acest tip de probleme de optimizare.În calculele de optimizare a structurilor se operează cu o serie de noțiuni și concepte ale teoriei matematice a optimizării, care capătă semnificații specifice corespunzătoare scopului urmărit și restricțiilor impuse.

5 ALGORITMII STOCASTICI

Designul optim generat de aceste metode este dependent de mai mulți factori: designul de la care se pornește, numărul iterațiilor de optimizare și gradul de aleatoriu al metodei. În cazul general, nu este cunoscută configurația optimă globală, iar rezultatele obținute în cursul a două rulări al aceluiași algoritm stocastic pot fi diferite. De aceea este necesară executarea multiplă a optimizării pentru a putea fi evaluată performanța designului. Metodele stocastice pot fi evaluate referitor la performanță și eficacitate luând în considerare acuratețea rezultatelor, cât sunt acestea de robuste și care este costul computațional.

După mai multe rulări ale algoritmului poate fi măsurată acuratețea în funcție de greutatea medie a cadrului optim obținut. Robustețea se va măsura în funcție de deviația standard a greutăților structurilor, iar costul computațional se măsoară în numărul de analize structurale necesare obținerii rezultatelor. Un algoritm bun nu se remarcă doar prin generarea de modele structurale mai ușoare, ci și prin generarea acestora în mod consecvent cu un cost computațional rezonabil. Metodele stocastice utilizate în literatura de specialitate includ algoritmii genetici (GA), optimizare cu colonie de furnici (ant colony optimization), călire simulată (simulated annealing SA), căutarea tabu (tabu search), optimizare cu roiuri de particule (particle swarm optimization PSO), harmony search și metode hibride. Deși acestea au dovedit că au rezultate bune în obținerea de configurații pentru cadre optime, niciuna dintre aceste metode nu s-a dovedit a fi superioară celorlalte în termenii celor trei caracteristici metrice: acuratețe, robustețe și eficiență computațională.

Optimizarea structurală poate fi formulată în mai multe moduri – într-un spațiu de variabile discrete sau continue, cu funcții obiectiv care calculează costurile în diferite moduri, și o multitudine de posibile restricții. Pentru a genera un context pentru formularea problemei și a algoritmilor utilizați în studiile de caz, am prezentat în continuare un studiu al literaturii și avantajele și dezavantajele celor mai utilizate metode de formulare. Problemele de optimizare alese pentru acest studiu au un număr semnificativ de variabile discrete alese din domenii care variază intre zeci și sute de posibilități. Funcția obiectiv aleasă este discontinuă, ceea ce face nepractică utilizarea metodelor de gradient. Algoritmii stocastici sunt recunoscuți pentru performanța lor în optimizarea structurală în spații de căutare ample, cu variabile discrete.

Multiple metode stocastice au fost aplicate în optimizarea structurilor, însă fiecare algoritm are aceeași formă de bază în care soluția cea mai eficientă este îmbunătățită gradual cu fiecare generație. La fiecare iterație, un număr stabilit de vectori de design este generat și valoarea fitness a fiecăruia este evaluată cu ajutorul funcției obiectiv și a restricțiilor.

Cele mai bune soluții sunt selectate și utilizate la crearea generației următoare de soluții candidat. Procesul este repetat până când un criteriu stabilit de convergență este îndeplinit. Datorită acestor asemănări cu teoria darwinistă - supraviețuirea celor mai potrivite soluții (numite „fittest”), reținerea celor dorite și înmulțirea soluțiilor optime din mulțimea acestora - a algoritmilor utilizați la optimizarea cu variabile discrete, metodele se numesc algoritmi evoluționiști.

Aceștia oferă numeroase avantaje: nu necesită calculul gradienților și matricelor hessiene - făcându-i eficienți în identificarea soluțiilor pentru funcții neliniare și cu "vârfuri" ascuțite. Generarea aleatorie, prin încercări consecutive, a soluțiilor candidat, permite algoritmilor să realizeze o căutare eficientă a spațiilor unor probleme cu multe posibile variabile de design.

Deoarece fiecare generație de soluții iterative este stocastic derivată din precedenta, acești algoritmi sunt buni candidați pentru calculul paralel, unde multiple lanțuri de soluții pot fi calculate pe procesoare paralele.

Cel mai însemnat punct slab al acestor algoritmi este costul computațional mare, dependența lor de parametri specifici care controlează nivelul variabilității de la o generație la alta, și - deoarece optimalitatea și convexitatea funcției obiectiv nu pot fi verificate cu gradienți și matrice hessiene - incapacitatea de a determina cu siguranță dacă soluția obținută este optimul global [Murren,2011].

Algoritmii stocastici încorporează restricțiile, în mod tipic, sub forma funcțiilor de penalizare:

$$W(x) = f(x)\left(1 + \sum_{i=1}^{n_c} p_i \beta_i\right); \quad x \in R^{n_{var}} \qquad (\ 5\text{-}1)$$

$$\beta_i(x) = \begin{cases} 0 & g_i(x) \leq 0 \\ > 0 & g_i(x) > 0 \end{cases} \qquad (\ 5\text{-}2)$$

Unde x este un vector de n_{var} variabile de design reprezentând locația secțiunilor disponibile în lista de n_s posibilități. Astfel, fiecare variabilă poate fi aleasă din aceste n_s posibilități. Se asigură satisfacerea restricțiilor prin aplicarea factorului de penalizare $(1 + \sum_{i=1}^{n_c} p_i \beta_i)$ la funcția de cost $f(x)$. Restricțiile $g_i(x)$ sunt exprimate în termeni de funcția auxiliara $\beta_i(x)$ într-o manieră în care $\beta_i(x) = 0$ când restricția este satisfăcută și >0 în caz contrar.

5.1 ALGORITMII EVOLUȚIONIȘTI

Sunt o categorie de metode numerice stocastice bazate pe analogii cu genetica. Paradigmele AE au fost dezvoltate de cercetători începând cu 1960. Principiile evoluționiste au fost implementate în algoritmi cu ajutorul cărora se pot soluționa probleme de optimizare. Diferența între algoritmii evoluționiști și algoritmii tradiționali constă în crearea unei populații de puncte. Prin adaptarea de generații succesive și a unui număr mare de indivizi, algoritmii evoluționiști efectuează o căutare directă și eficientă (Sivanandam, și alții, 2008).

Metodele care sunt cel mai frecvent aplicate în arhitectură și inginerie sunt:

- Algoritmi genetici (GAs)
- Strategii evoluționiste (ESs)
- Calcul evoluționist interactiv (IEC)

Aceste metode au punct comun în utilizarea unor generații de populații de soluții pentru căutarea în spațiul de soluții a celor care corespund cel mai bine criteriilor funcțiilor obiectiv. Performanța unei soluții este măsurată prin fitness. Populația evoluează gradual prin încrucișare, mutație și selecție spre soluții mai bune.

În lucrarea de față, soluțiile individuale descriu forme structurale, geometrice, și sunt reprezentate prin variabile. Variabilele sunt reprezentate de un “cromozom” asupra căruia sunt aplicați operatorii genetici pentru a crea indivizi noi mai performanți.

Cu cat e nevoie de mai multe variabile pentru a descrie geometria structurii, cu atât lungimea șirului cromozomului va fi mai mare. Mărimea populației este proporțională cu lungimea cromozomului iar numărul de generații necesare pentru convergență depinde de ambii factori. Așadar, cu cât e nevoie de mai multe variabile de proiectare cu atât problema devine mai intensivă computațional (Goldberg, Deb & Clark 1991).

5.1.1 Algoritmii genetici

Algoritmilor genetici li s-a acordat o atenție deosebită datorită potențialului de a reprezenta o modalitate nouă de optimizare (Mitsuo, și alții, 2008).

Primul cercetător al teoriei algoritmilor genetici a fost John Holland, care i-a descris în cartea „Adaptation în Natural and Artificial Systems” în 1975. Domeniul algoritmilor evoluționiști include strategii evoluționiste (ES), programare evoluționistă (EP), viață artificială (AL), programare genetică (GP). În 1960 Ingo Rechenberg și Hans-Paul Schwefel au dezvoltat ideea strategiilor evoluționiste. În același timp Lawrence Fogel și alții au pus bazele programării evoluționiste. Aceste teorii aveau în comun ideea procesului de mutație și selecție. Inspirați din teoria evoluționistă a lui Darwin, Bremermann și Fraser au folosit teoria recombinării (crossover) [Hayalioglu, 2001].

Algoritmii genetici sunt construiți pentru a putea efectua căutarea structurilor din ce în ce mai bune, iar această procedură necesită o funcție obiectiv – funcția „fitness” a cărei valoare este asociată unui șir numit „individ”.

GA utilizează trei operații de bază pentru crearea unei noi generații: *selecție, încrucișare și mutație* grupate sub denumirea de reproducere. Operația de reproducere cuprinde copierea sau modificarea unor indivizi dintr-o generație în alta, în funcție de valoarea funcției „fitness”. Funcția de selecție poate fi implementată într-un algoritm în diferite forme. Cea mai simplă formă a funcției se bazează pe teoria „roulette wheel” [Sivanandam, 2008].

GA sunt avantajoși și eficienți când:

- spațiul de căutare este mare, complex sau dificil de definit,
- domeniu de răspuns este redus sau condițiile sunt dificil de codat pentru a obține un spațiu de răspuns concentrat,
- metodele tradiționale de optimizare nu generează soluții satisfăcătoare.

Printre avantajele folosirii GA merită menționate ușurința cu care se pot aplica tipuri de restricții arbitrare și varietatea mare a posibilelor funcții obiectiv. Toate aceste lucruri pot fi manipulate ca și componente de penalizare a funcției de fitness, făcând ușoară adaptarea algoritmului la cerințele specifice ale unei varietăți de obiective generale [Sivanandam,2008].

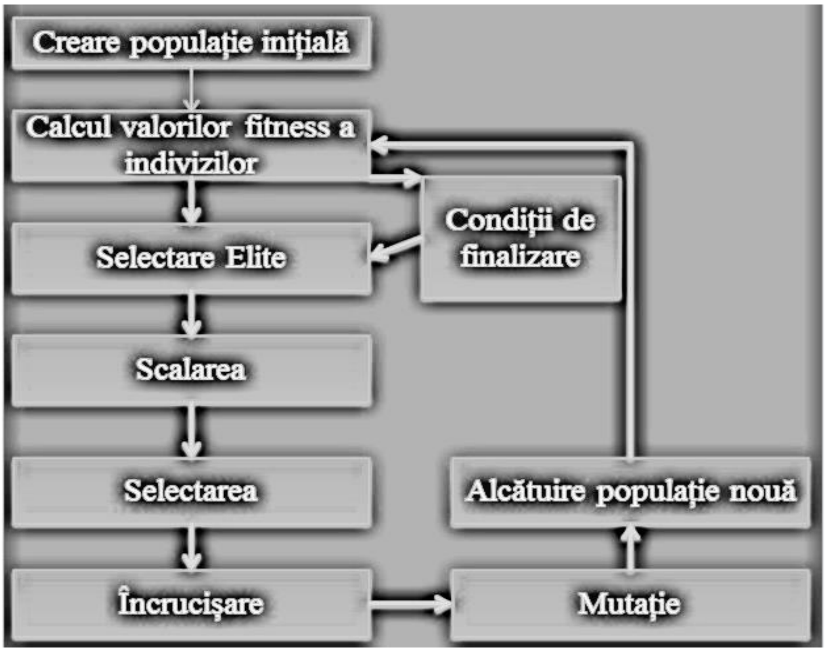


Figura 5.1 Schema unui algoritm genetic simplu

Mecanismele fundamentale care realizează legătura dintre algoritmul genetic şi problema care trebuie rezolvată sunt următoarele:

- codificarea problemei în termeni de cromozomi,
- funcția de evaluare, care furnizează o măsură a calității fiecărui cromozom în contextul problemei respective.

Codificarea se realizează de obicei prin șiruri de cifre binare. S-a demonstrat că acest mod de codificare este robust, în sensul adaptării lui la o mare varietate de probleme practice. Ceea ce i se reproșează uneori este precizia soluției, limitată la numărul de biţi, pe care se face reprezentarea. Alegerea unui număr suficient de mare de biţi pentru reprezentarea valorilor reale din problemă poate înlătura însă acest dezavantaj.

Mai jos sunt prezentate programe scrise în MATLAB, care fac conversia zecimal-binar și invers.

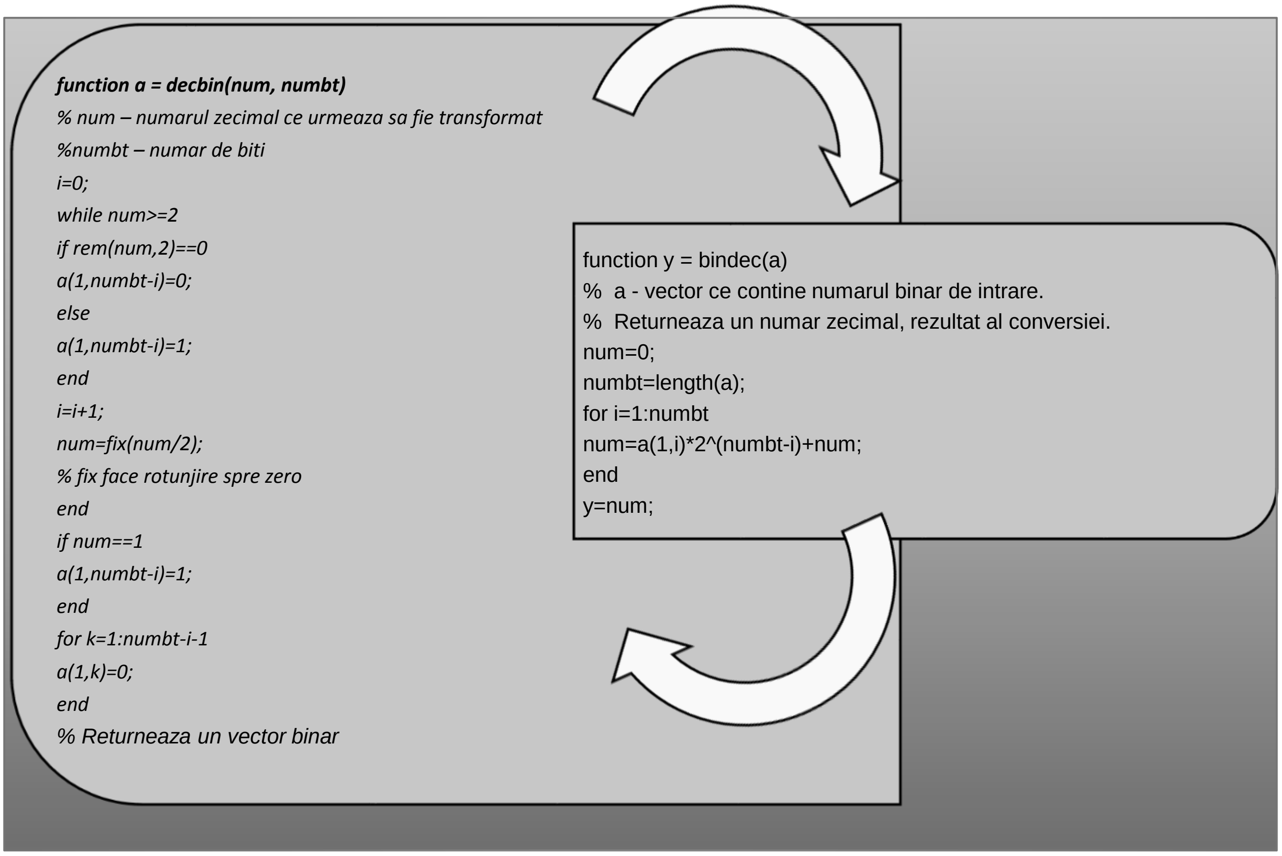


Figura 5.2 Exemplu de funcție de transcriere a soluțiilor candidat din zecimal în binar si viceversa.

Funcția de evaluare primeşte la intrare şirul de cromozomi și returnează numere sau liste de numere ce reprezintă performanţa cromozomilor. Ea are rolul mediului înconjurător pentru evoluţia naturală.

Structura unui algoritm genetic fundamental este dată mai jos:

1. Se iniţializează populaţia de cromozomi.

2. Se evaluează fiecare cromozom din populație. Se selectează părinții noii populații.

3. Se creează o nouă generație de cromozomi prin împerecherea cromozomilor selectați, folosind operatori genetici.

4. Se șterg membrii populației inițiale, pentru a fi înlocuiţi cu noua generaţie.

5. Se evaluează noii cromozomi și se inserează în noua populaţie.

6. Dacă timpul de căutare nu s-a terminat, se merge la pasul 3. În caz contrar, se opreşte execuţia algoritmului.

Modul de reprezentare a populaţiei de cromozomi, modul de evaluare a cromozomilor şi modul de reproducere sunt componente esenţiale ale algoritmului genetic şi sunt prezentate în cele ce urmează.

Valoarea fitness a unui individ, într-un algoritm genetic este dată de funcția obiectiv (funcția fitness). În cazul optimizării multicriteriale funcția obiectiv se determină destul de dificil. Pentru problemele de optimizare multicriteriale există o problemă legată de evaluarea soluției optime. Când procesul GA de căutare începe, populaţia este supusă unei evaluări cu ajutorul funcţiei fitness. În funcție de această evaluare se va alcătui noua populație. La acel moment, în fiecare generaţie, soluţiile relativ bune sunt reproduse şi soluțiile cu valoare fitness mică sunt abandonate. Pentru a distinge soluţiile avem nevoie de o funcție de evaluare (funcția fitness), aceasta având un rol important în procesul evoluționist alături de mecanismele de scalare. Atunci când se evaluează funcția fitness a unor indivizi, avem nevoie de o procedură de decodare [Bendsœ, 1988].

O componentă necesară în aplicarea algoritmilor genetici este modul de manipulare a restricțiilor, deoarece operațiile algoritmilor genetici asupra indivizilor creează indivizi nefezabili [Bendsœ, 1988]. Algoritmii genetici generează indivizi care urmează să fie testați cu ajutorul funcției fitness și a restricțiilor.

Procesul de **scalare** are rolul de a evita o convergență prematură a algoritmului sau o terminare înceată. De obicei la începutul algoritmului variația între indivizi este mare și doar o mică parte dintre ei sunt mai buni decât restul indivizilor. Cu o selecție conform valorii fitness nescalate aceștia se vor multiplica repede și vor împiedica algoritmul să exploreze spațiul soluțiilor, acest fenomen este cunoscut drept o convergență prematură [Melanie, 1996].

Scalarea liniara a valorii fitness se face cu relația:

$$f' = af + b$$

Cu această metodă valoarea fitness medie a indivizilor trebuie să se păstreze și după scalare. Pentru a înlătura posibilitatea ca indivizii superiori să domine procesul de scalare trebuie respectată egalitatea:

$$f'max = C * fav^{`}$$

Unde: C – reprezintă numărul indivizilor cu valoarea fitness optimă.

Metoda de scalare sigma are rolul de a exercita o presiune constantă asupra procesului. Valoarea fitness a individului se recalculează în funcție de valoarea fitness medie a populației și de deviația standard [Arora, 1997].

$$ExpVal(i,t) = \begin{cases} 1 + \frac{f(i) - \bar{f}(t)}{2\sigma(t)} & if\ \sigma(t) \neq 0 \\ 1 & if\ \sigma(t) = 0 \end{cases}$$

$ExpVal(i,t)$ reprezintă valoarea scalată a individului i, $f(i)$ reprezintă valoarea fitness a individului, iar $\bar{f}(t)$ este valoarea medie a populației și $\sigma(t)$ este deviația standard a fitness-ului populației [Arora, 1997].

Scalarea prin metoda puterii se aplică cu ajutorul relației:

$$f' = f^k$$

Unde: k – constantă (1.005)

Această metodă se folosește împreună cu metoda de selectare „roulette wheel” .

Selecţia este un operator genetic care stabileşte şirurile populaţiei curente care vor fi alese pentru a transmite materialul lor genetic generaţiei următoare.

Există trei tehnici de selecţie:

• cea mai utilizată este selecţia proporţională, care modelează mecanismul selecţiei naturale, în care cromozomii cu o evaluare mai mare au o şansă mai mare de a fi aleşi. Cunoscută și sub numele de principiul ruletei, această tehnică presupune parcurgerea următoarelor etape:

1. Se stabilește funcția de evaluare pentru fiecare cromozom din populație $feval(x_i)$

2. Se însumează toate funcțiile de evaluare $feval = \sum_i feval(x_i)$

3. Cromozomilor li se atribuie aleator numerele naturale i.

Apoi, următorii pași se repetă până la crearea unui număr suficient de perechi de cromozomi:

4. Se generează numerele aleatoare n și m, astfel ca $1 \leq n, m \leq feval$,

5. Se alege cromozomul x_i, unde i este cel mai mic număr care satisface relația: $\sum_{j \leq i} feval(x_j) \geq n$

6. Se alege cromozomul x_j, ca la pasul 5, cu m în loc de n

7. Se stabilește perechea de cromozomi x_i și x_j.

Această modalitate de selecţie însă poate genera serioase probleme dacă un cromozom din populaţie are o funcţie de evaluare de valoare mult mai mare decât a celorlalţi cromozomi, aceasta fiind departe de optim, iar atunci selecţia proporţională va raspândi foarte repede caracteristicile acestui cromozom în populaţie. În câteva generaţii populaţia ar putea fi alcătuită numai din astfel de cromozomi și algoritmul genetic nu ar mai putea evolua, deci optimul nu mai poate fi găsit. Acest fenomen este cunoscut sub numele de *convergenţă prematură*.

O altă problemă o constituie gradientul scăzut al funcției de evaluare spre sfârşitul căutării. Treptat soluția optimă este preluată de întreaga populaţie. Efectul este cunoscut sub numele de *terminare lentă* (slow finishing).

O altă tehnică de selecţie este selecţia pe baza rangului, în care probabilitatea de a fi ales este o funcție liniară de locul ocupat de individ (cromozom) în cadrul populaţiei. Avantajul constă în faptul că nu mai este necesară interpolarea permanentă a evaluării ca în cazul precedent. Un caz special de selecţie de acest tip este selecţia prin trunchiere, prin care se elimină o parte din cromozomii cu cea mai slabă evaluare, iar în locul lor se generează alţii, după diferite scheme posibile. Un exemplu de selecție prin trunchiere este prezentat în continuare:

1. Din populaţia actuală se elimină n cromozomi care au evaluarea cea mai slabă.

2. Se generează un nou cromozom x, folosind principiul ruletei

3. Dacă x diferă de toţi ceilalţi cromozomi ai populaţiei actuale, atunci el este inclus în populaţie; în caz contrar, este supus operatorului de “mutație” (explicat mai jos) până ce devine diferit de ceilalţi cromozomi și este inclus în populaţie.

O altă metodă euristică de selecţie este selecţia elitistă, care reţine întotdeauna cei mai buni cromozomi ai populaţiei (de regulă unul singur). Ea garantează convergenţa asimptotică spre un minim global, dar rata de convergenţă este variabilă în funcție de problemă. Elita poate introduce un efect de dominanţă asupra populaţiei care să ducă la o stagnare timpurie a procesului de evoluţie. Soluţia constă în utilizarea operatorului de mutaţie pentru reducerea acestui efect.

Încrucişarea (crossover) este operatorul necesar pentru construcţia noilor indivizi ai populaţiei.

Populaţia intermediară, formată din n cromozomi, este împărţită în n/2 perechi şi operatorul de încrucişare este aplicat fiecărei perechi cu o anumită probabilitate χ. Valoarea lui χ este de obicei mai mare de 0,6 și de cele mai multe ori se alege $\chi= 1$.

Noii indivizi ai populaţiei sunt generaţi prin combinarea unor părţi alternative de material genetic provenind din două şiruri părinte a_1 și a_2. Cea mai simplă schemă este încrucişarea cu un singur punct. Dacă numărul de biţi din cromozomul-şir este L=numbt, atunci punctul de încrucişare este ales aleator între 1 și L. Această schemă de încrucişare este prezentată mai jos.

1. Se generează aleator un număr natural p în intervalul [1 , L (a_i)] ;

2. Se generează noua pereche de cromozomi a_{1new} şi a_{2new} , după cum urmează:

Pentru i≤p se execută următoarea secvenţă:

$a_{1new}(i) = a_2(i)$ și $a_{2new}(i) = a_1(i)$

Pentru i>p se execută următoarea secvență:

$a_{1new}(i) = a_1(i)$ și $a_{2new}(i) = a_2(i)$

Mutaţia permite algoritmului genetic să găsească noi soluții în cadrul populaţiei și îl protejează împotriva pierderii de informaţie în cazul unor încrucişări nepotrivite. Rata mutaţiei este foarte redusă, probabilitatea mutaţiei având valori cuprinse între 0,001 şi 0,01. Dacă operatorul de selecţie reduce diversitatea în populaţie, cel de mutație determină o nouă creştere a diversităţii. Cu cât probabilitatea mutaţiei este mai mare, cu atât mai redus este riscul convergenţei premature, dar apare un nou risc datorită faptului că o rată mare de mutație va transforma algoritmul genetic într-un algoritm de căutare aleatoare.

O schemă tipică de mutație pentru un cromozom $x = \langle x_1, \dots, x_n \rangle$ este:

1. Se generează aleator un număr z astfel încât 1≤z≤n;

2. Se selectează gena x_z ;

3. $x_z = 1 - x_z$.

```
function y = mutate(a)
% Modifica aleator un bit al vectorului a.
numbt = length(a);
z=fix(rand*(numbt-1)+1);
a(z)=1-a(z);
y = a;
```

Criterii de convergență:

Numărul maxim de generații – algoritmul genetic se oprește la îndeplinirea unui număr maxim de generații, stabilit de utilizator.

Timpul de executare – algoritmul genetic se poate opri după o anumită perioadă setată de utilizator.

Nicio schimbare a valorii fitness – valoarea minimă sau maximă a funcției fitness rămâne constantă pentru un număr de generații stabilit.

Stall generations- algoritmul se oprește dacă nu apare nicio îmbunătățire în funcția obiectiv pentru un număr de generații consecutive.

Stall time limit- algoritmul se opreşte dacă nu apare nici o îmbunătățire în funcția fitness pe parcursul unui interval de timp echivalent în secunde cu stall time limit.

Best individual - criteriu de convergență pentru determinarea celui mai bun individ oprește căutarea dacă valoarea minimă fitness dintr-o populație scade sub valoarea de convergență. Acest lucru aduce procesului de căutare rapiditate, garantând cel puțin o soluție fezabilă.

Tehnici de manipulare a restricțiilor:

Strategiile de respingere elimină indivizii nefezabili creați printr-un proces evoluționist. O astfel de abordare este limitată în cazul în care populația inițială conține mulți indivizi nefezabili care trebuie îmbunătățiți pentru a trece în domeniul fezabil. Deseori pentru ca algoritmul să ajungă la un optim, individul trebuie să traverseze un domeniu nefezabil [Bendsœ, 1988].

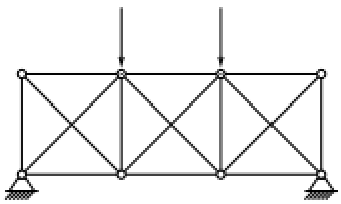
Strategiile de reparare implică selectarea unui individ nefezabil și prin anumite procese de reparare se transformă într-un individ fezabil. Aceste strategii depind de existența unor proceduri deterministice de reparare a soluțiilor nefezabile înlocuindu-i cu indivizi fezabili. Dificultatea acestor metode constă în definirea unui algoritm de reparare pentru fiecare problemă de optimizare în parte. Orvosh şi Davis a definit regula de 5%, această regulă euristică prevede ca GA cu o procedură de reparare oferă cele mai bune rezultate atunci când 5% din cromozomi sunt reparați înlocuind indivizii nefezabili. Michalewicz a raportat că regula de înlocuire de 15% indicată pentru probleme de optimizare numerică cu restricții neliniare.

Aplicând *strategiile de modificare a operatorilor genetici* vom genera indivizi fezabili, astfel încât algoritmul genetic lucrează în domeniul fezabil. Michalewicz a subliniat faptul că de multe ori astfel de sisteme sunt mult mai fiabile decât oricare alți algoritmi genetici bazați pe metodele de penalizare. Avantajul strategiile descrise anterior constă în eliminarea generării de soluții nefezabile, dar au dezavantajul că nu consideră puncte din afara regiunilor fezabile. Pentru problemele cu restricții majore soluțiile nefezabile pot ocupa un procent mare din totalitatea soluțiilor. Într-un astfel de caz, soluțiile fezabile pot fi greu de găsit dacă vom limita căutarea în zona regiunilor fezabile [Bendsœ, 1988].

Pentru ca GA să opereze cu soluțiile nefezabile se pot aplica strategii de penalizare a funcției obiectiv. Aceste strategii transformă problema de optimizare cu restricții într-o problemă de optimizare fără restricții cu ajutorul unei funcții de penalizare .

5.1.1.1 Problemele tipice la care sunt aplicați algoritmii genetici

Includ designul grinzilor cu zăbrele de greutate minimă, optimizare topologică, analiza limitelor, design cu număr minim de bare.



m bare de lungime l și arii secționale x_i

N noduri; nodurile 1,...,n sunt libere, nodurile n+1,...,N sunt fixate

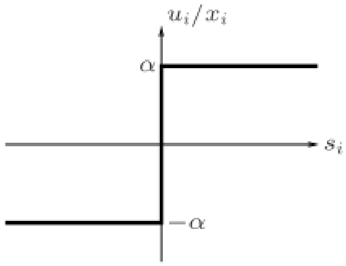
încărcări externe: forțe $f_i \in R^2$ la nodurile i=1,...,n.

Probleme de design:

- dată fiind topologia (pozițiile barelor și a nodurilor), sa se găsească cea mai ușoară structură care poate purta o încărcare dată (variabile - dimensiunea barelor, cost - greutatea totală).
- aceeași problemă, dar costul include numărul barelor utilizate.
- găsirea topologiei optime.
- găsirea celei mai ușoare configurații care poate purta încărcările date.

Problema de analiză: pentru o structură dată, să se găsească încărcarea limită.

Caracteristicile materialului:



$u_i \in R$ este forța în bara i (u_i>0: întindere, u_i<0: compresiune)

$s_i \in R$ este deformația barei i (s_i>0: alungire, s_i<0: scurtare)

$s_i = 0$ dacă $-\alpha < u_i/x_i < \alpha$

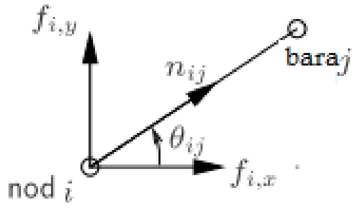
$u_i/x_i = \alpha$ dacă $s_i > 0$

$u_i/x_i = -\alpha$ dacă $s_i < 0$ (α este o constantă de material)

Structura de greutate minimă în condițiile de încărcare date:

- relațiile de echilibru pentru nodul liber i:

$$\sum_{j=1}^{m} u_j \begin{bmatrix} n_{ij,x} \\ n_{ij,y} \end{bmatrix} + \begin{bmatrix} f_{i,x} \\ f_{i,y} \end{bmatrix} = 0$$



n_{ij} depinde de topologie

$n_{ij} = 0$ dacă bara j nu este legată de nodul i

$n_{ij} = (cos\theta_{ij}, sin\theta_{ij})$ altfel

- design de greutate minimă utilizând programare liniară (LP):

  să se minimizeze $\sum_{i=1}^{m} l_i x_i$

în condițiile în care $\sum_{j=1}^{m} u_j n_{ij} + f_i = 0, \quad i = 1, \dots, n$

$-\propto x_j \le u_j \le\propto x_j, \quad j = 1, \dots, m$

(variabile x_j, u_j)

Exemplu:

să se minimizeze $l_1 x_1 + l_2 x_2 + l_3 x_3$

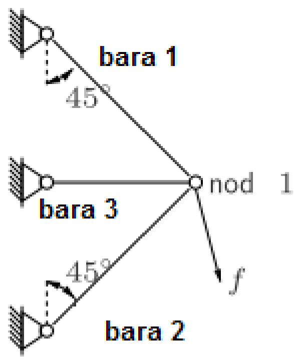
în condițiile în care $-u_1/\sqrt{2} - u_2/\sqrt{2} - u_3 + f_x = 0$

$u_1/\sqrt{2} - u_2/\sqrt{2} + f_y = 0$

$-\propto x_1 \le u_1 \le\propto x_1$

$-\propto x_2 \le u_2 \le\propto x_2$

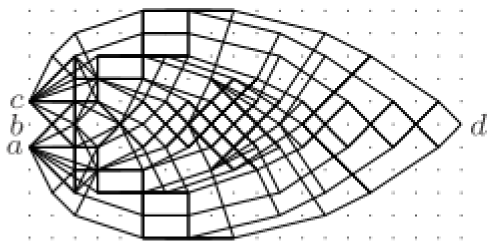
$-\propto x_3 \le u_3 \le\propto x_3$



Modelarea topologică:

- Se creează un grid de noduri unit cu bare între fiecare pereche de noduri
- Se caută designul structurii de greutate minimă: $u_i = 0$ pentru majoritatea barelor
- Topologia optimă: se utilizează doar barele cu $u_i \neq 0$

Exemplu:



Grid de 20 x 11, adică 220 de noduri potențiale și 24.090 bare potențiale.

Nodurile a, b, c sunt fixate iar în nodul d este aplicată o forță verticală unitară.

Topologia optimă are 289 de bare.

Scenarii de încărcare multiple

Structura de greutate minimă care poate rezista la M posibile încărcări $f_i^1, \dots, f_i^M$:

Sa se minimizeze $\sum_{i=1}^{m} l_i x_i$

Considerand restricțiile $\sum_{j=1}^{m} u_j^k n_{ij} + f_i^k = 0, \quad i = 1, \dots, n, \quad k = 1, \dots, M$

$$-\propto x_j \leq u_j^k \leq \propto x_j\,,\ j = 1, \dots, m, \quad k = 1, \dots, M$$

(variabile $x_j, u_j^1, \dots, u_j^M$)

Pentru a asigura robustețea designului: structura rezistă la orice încărcare

$$f_i = \lambda_1 f_i^1 + \cdots + \lambda_M f_i^M$$

Unde $\lambda_k \geq 0,\ \sum_k \lambda_k \leq 1$

Analiza limitei

Structura are geometria dată (inclusiv ariile secțiunilor x_i).

Încărcarea f_i este data de $f_i = \gamma g_i$, cu valorile $g_i \in R^2$ și $\gamma > 0$.

Să se gasească cea mai mare valoare a încărcării capabile:

Să se maximizeze γ

Considerând restricțiile $\sum_{j=1}^{m} u_j n_{ij} + \gamma g_i = 0 \quad i = 1, \dots, n$

$$-\propto x_j \leq u_j \leq \propto x_j\,,\ j = 1, \dots, m$$

Are forma unei probleme de programare liniară (LP) în variabilele γ, u_j. Pentru valoarea maximă admisa γ se numește factor de siguranță.

Design cu număr minim de bare

Formulare LP cu numere întregi (considerând $w \log x_i \leq 1$)

Sa se minimizeze $\sum_{j=1}^{m} z_j$

Considerand restricțiile $\sum_{j=1}^{m} u_j n_{ij} + f_i = 0, \quad i - 1, \dots, n$

$$-\propto x_j \leq u_j \leq \propto x_j\,, \quad j = 1, \dots, m$$

$$x_j \leq z_j, \quad j = 1, \dots, m$$

$$z_j \in \{0, 1\}, \quad j = 1, \dots, m$$

Variabile z_j, x_j, u_j

Extrem de greu de rezolvat; uneori trebuie enumerate toate cele 2^m valori posibile ale lui z.

În formularea euristică se înlocuiește $z_j \in \{0, 1\}$ cu $0 \leq z_j \leq 1$.

5.1.2 Optimizarea structurală evolutivă (ESO)

Tehnica a fost iniţial propusă în 1992 de către Mike Xie şi Steven Grant. Ei au propus dezvoltarea unei tehnici foarte simple, dar versatile pentru a găsi modele structurale optime. ESO se bazează pe conceptul de eliminare treptată a materialelor ineficiente dintr-o structură, astfel încât structura rezultată să evolueze spre forma optimă.

Metoda ESO se dovedeşte a fi capabilă să rezolve probleme de optimizare structurală pentru dimensiuni, formă şi topologie, pentru încărcări statice, stabilitate dinamică şi probleme de transfer de căldură sau combinaţii ale acestora. Metoda ESO are aplicabilitate în practică în special datorită simplităţii și eficienței. Orice persoană care are cunoştinţe de bază de analiză cu element finit (FEA), poate înţelege cu uşurinţă şi aplica metoda ESO. Un alt avantaj al metodei ESO este că poate fi uşor de implementat și legat de pachetele comerciale FEA, cum ar fi ABAQUS, ANSYS și NASTRAN.

Pentru structuri aflate numai în tensiune sau numai în compresiune, metoda tradiţională ESO elimină materialul de la o structură de baza pe criterii definite pentru tensiuni von Mises sau energia de deformare a fiecărui element. Anumite materiale de construcții, cum ar fi betonul și cablurile de oțel, sunt potrivite numai pentru supunerea la tensiuni exclusiv de compresiune sau întindere.

Pentru a realiza o structură optimă compusă din elemente solicitate doar la întindere, elementele cu cele mai mari tensiuni de compresiune vor fi înlăturate în prima etapă. Apoi, elementele puțin solicitate la întindere vor fi şterse din structură de asemenea. În mod similar, pentru a obține o structură optimă formată doar din elemente aflate în compresiune, barele cu cele mai mari tensiuni de întindere vor fi înlăturate în prima etapă. Apoi, elementele mai puțin comprimate vor fi şterse din structură la pasul următor.

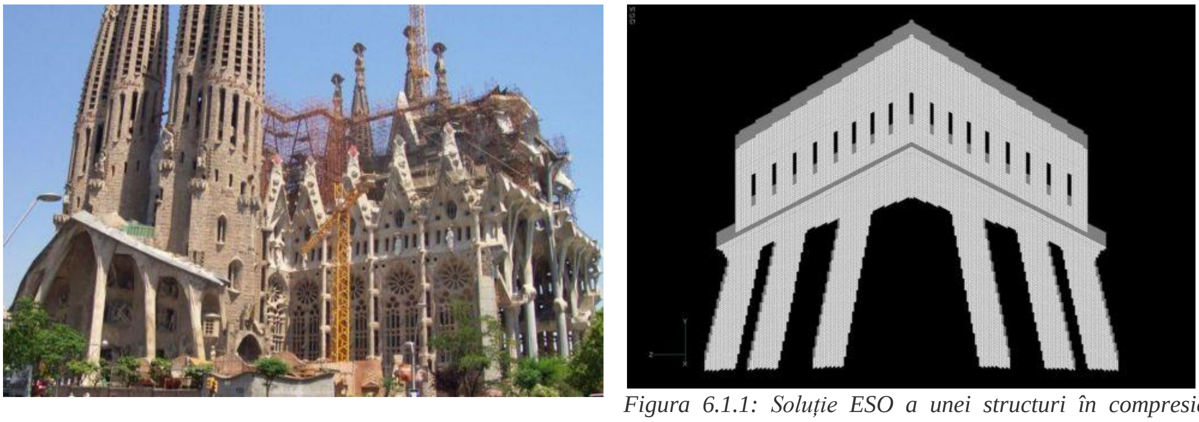


Figura 6.1.1: Soluție ESO a unei structuri în compresie exclusiv (simularea faţadei Patimilor a bisericii Sagrada Familia din Barcelona, Prof. Mark Burry al SIAL Sursa: http://isg.rmit.edu.au/research_ESO.html)

Pentru a obţine un design ESO neliniar, modele de elemente finite sunt analizate prin luarea în considerare a neliniarităţii materiale şi/sau a neliniarităţii geometrice. Două criterii pentru îndepărtarea materialului au fost experimentate. Unul se bazează pe ştergerea elementelor cu tensiuni von Mises mici, cealaltă se bazează pe eliminarea elementelor cu energie de deformare scăzută. Exemplul de mai jos se bazează pe criteriul energiei de deformare.

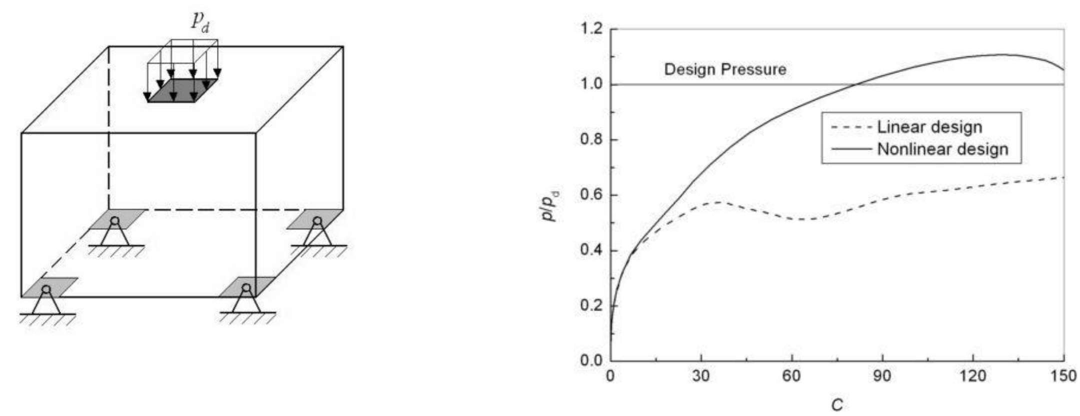


Figura 6.1.2: Comparaţie a capacităţii de transmitere a sarcinii-intre modele ESO liniar și ne-linear (soluție ESO neliniară presupunând că σ =ε0.2)

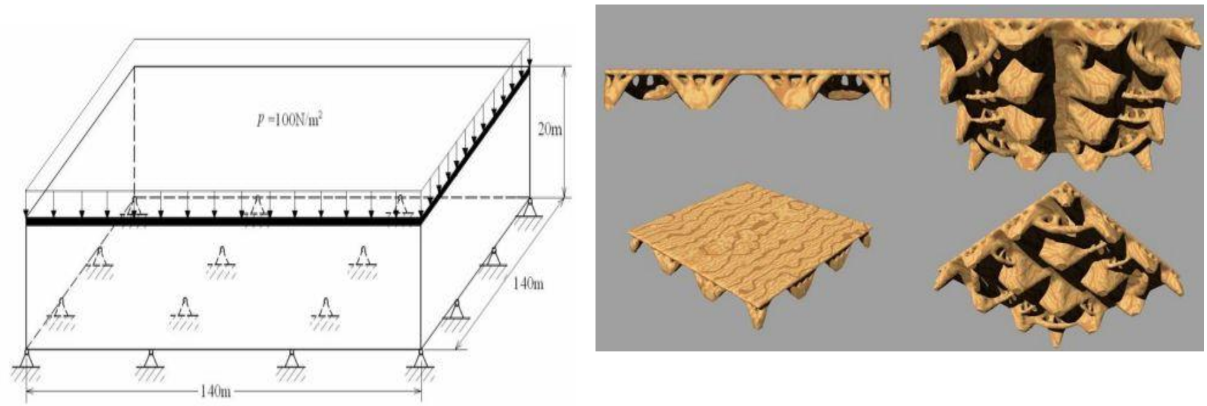


Figura 6.13: Modelarea unui spatiu subteran sustinut de 16 stalpi.

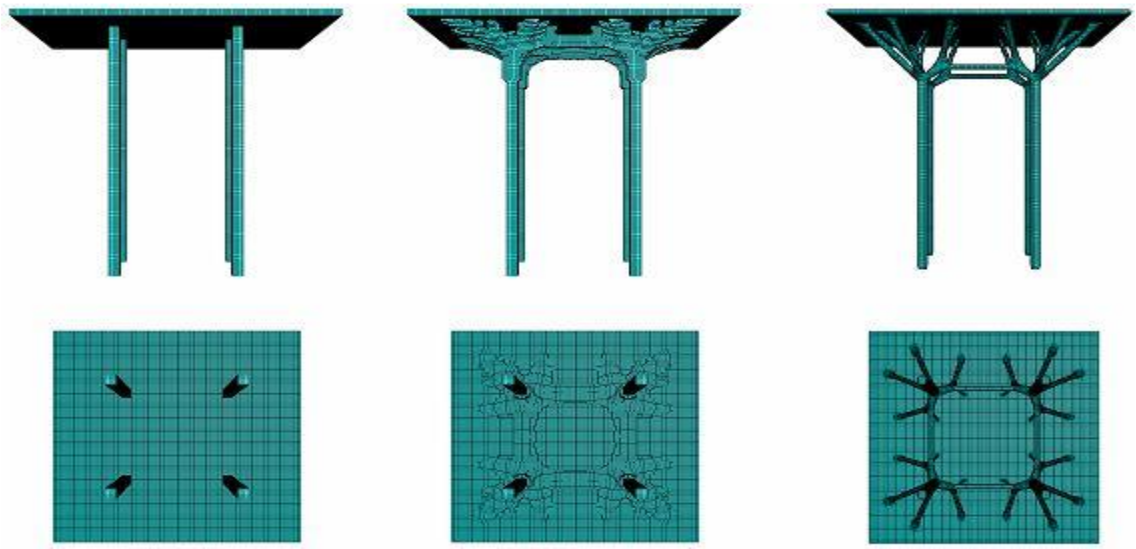


Figura 6.14: Sistem de transfer al încărcărilor pentru un atrium al unei cladiri multietajate.

5.1.3 Optimizare de tip PSO (particle swarm optimization)

PSO a fost formulată de către Edward și Kennedy în 1995. Algoritmul a fost inspirat din comportamentul social al animalelor, cum ar fi comportamentul păsărilor sau a peștilor. Asemănarea între PSO și AG este dată de operarea cu ajutorul unei populații aleatoare (exprimată sub formulare matriceală). Spre deosebire de AG, PSO nu folosește procesele de mutație sau încrucișare. Elementele matricei sunt numite particule (la fel ca cromozomii pentru AG). Fiecare particulă se mişcă pe suprafaţa de răspuns cu o anumită viteză [Otten, 1989].

5.1.4 Optimizare de tip Simulated Annealing

A fost introdusă de Kirkpatrick (1983). Această teorie se bazează pe procesul termic prin care se modifică microstructura materialului cu implicații directe asupra proprietăților fizice. Algoritmul analog acestui proces începe prin atribuirea de valori aleatoare variabilelor, iar procesul de încălzire este reprezentat prin modificarea aleatoare a variabilelor [Otten, 1989].

Procesul începe cu un algoritm care are ca efect selectarea aleatorie a variabilelor funcției obiectiv, iar încălzirea însemnând modificarea aleatorie a acestor valori, o valoare mai mare a încălzirii atrage după sine o mai mare fluctuație a valorii factorului aleator care modifică respectivele valori. Funcția obiectiv returnează rezultatul, *f*, cu un set cunoscut de variabile. Dacă rezultatul scade de-a lungul procesului, atunci rezultatul nou îl înlocuieşte pe cel vechi. Dacă rezultatul creşte, atunci rezultatul acceptat are ca variabile un număr aleator "r" și un T care este o variabilă, altfel setul nou de variabile este respins. Astfel, chiar dacă una dintre diferitele seturi de variabile duce spre o funcție obiectiv mai slabă, ea poate fi eligibilă cu o anume probabilitate. Setul nou de variabile este calculat aplicându-i un pas aleator setului vechi de variabile [Otten, 1989].

Există anumite probleme de optimizare care devin prea complexe pentru metodele clasice, odată cu creșterea numărului de variabile considerate. Pentru acest tip de probleme algoritmul de tip călire simulată (numit astfel deoarece mimează procesul la care sunt supuși atomii unui metal când acesta este încălzit și apoi supus unei răciri lente) se dovedește foarte eficient. Chiar dacă aceasta tehnica are puține șanse sa găsească soluția optimă, poate adesea găsi soluții foarte bune, chiar și în cazul unui domeniu de valori cu zgomot. Călirea simulată este o strategie care se bazează pe doua trucuri. Primul este așa numitul „algoritm Metropolis" (Metropolis et al. 1953), care permite selectarea unor soluții care nu reduc funcția fitness dar care permit explorarea mai vasta a spațiului de soluții posibile. Acestea sunt permise prin utilizarea criteriului ca:

$e^{-\Delta D/T} > R\,(0,\,1)$,

unde ΔD este distanța dintre soluțiile interschimbate (negativă în cazul unui schimb „bun", pozitivă în cazul unei schimbări nefavorabile), T este „temperatura sintetică" (synthetic temperature), iar $R\,(0,\,1)$ este o valoare aleatorie în intervalul $[0,\,1]$. D se numește "funcție de cost," și îi corespunde energiei libere din cazul călirii unui metal (caz în care parametrul de temperatură ar fi chiar kT, unde k este constanta lui Boltzmann iar T este valoarea temperaturii pe scara Kelvin. Dacă T este mare, sunt acceptate multe schimbări nefavorabile, și astfel o mare parte a spațiului de căutare este accesat. Valorile schimbate sunt alese, în mod normal, aleatoriu, dar există și tehnici mai sofisticate de generare a soluțiilor. Al doilea truc este, de asemenea prin analogie la călirea metalului, scăderea „temperaturii". După realizarea multor schimburi de soluții și observarea scăderii lente a funcției de cost, va fi scăzută temperatura, și limitat astfel numărul de schimburi nefavorabile. După scăderea repetată a temperaturii înspre o valoare scăzută, poate fi „stins" procesul prin acceptarea exclusivă a schimburilor bune, spre a găsi minimul local al funcției de cost. Există numeroase scheme de temperatură de călire, însă rezultatele nu sunt de obicei foarte sensibile la detalii.

Mai existăa o altă strategie rapidă numită prag de acceptare (threshold acceptance, Dueck and Scheuer 1990). Acestă strategie accepăa toate schimburile bune, și pe cele nefavorabile dar care cresc valoarea funcției cost cu o valoare aflată sub un anumit prag stabilit. Acest prag de acceptare este scăzut progresiv, asemenea temperaturii sistemului în cazul călirii simulate. Astfel este eliminat exponentul și generarea de numere aleatorii din cazul criteriului Boltzmann. Din acest motiv metoda poate fi mai rapidă în simulările computaționale.

6 STRATEGII DE OPTIMIZARE

Prin strategie de optimizare înțelegem ansamblul de proceduri, descrise în metode și algoritmi de optimizare matematică sau non-matematică, care pot fi combinate pentru obținerea rezultatului căutat (Turda, 2003).

6.1 DIRECȚII DE CERCETARE

Diferitele direcții de cercetare din domeniul optimizării structurale evoluționiste au fost împărțite în trei categorii: **design structural specializat, îmbunătățiri ale GA, și obiective ale optimizării.**

1. Designul structural specializat implică algoritmii genetici care sunt adaptați unor tipuri specifice de structuri. Aceste tipuri includ grinzi cu zăbrele, cadre plane, cadre spațiale, turnuri tip latice, ferme pentru acoperișuri.

2. Îmbunătățiri ale algoritmilor genetici - includ cercetarea condusă spre îmbunătățirea robusteții programelor de optimizare (timp de execuție, tehnici crossover, comunicare binivel, operatori fuzzy, selecție).

3. Obiective de optimizare - atenția este acordată algoritmilor genetici creați în mod specific pentru îmbunătățirea unuia sau mai multor obiective (dimensiunile, forma elementelor, topologia, detectarea daunelor, controlul vibrațiilor).

Multe modele inginereşti implică folosirea unor algoritmi speciali de optimizare pentru a încerca să minimizeze sau să maximizeze funcția merit sau obiectiv. Deseori, obiectivul este o sumă ponderată de mai multe sub-obiective. Cele mai simple probleme de optimizare au vectorul variabilelor definit prin secțiunile elementelor. Adăugând la vectorul variabilelor alți parametrii de proiectare problema crește în dificultate (Petrina, 1982).

Un astfel de software impune constrângeri privind soluțiile optime. Ele constau în restricții regionale (de exemplu restricții care limitează o dimensiune) și restricțiile funcții care limitează o funcție calculată (cum ar fi o valoare de tensiune). În cele mai multe probleme există multe variabile de analizat. Acestea includ atât dimensiuni fizice cât și proprietăți materiale.

6.2 OPTIMIZARE MULTI-MODALĂ

Funcțiile multi-modale pot avea un număr mare de puncte de optim local. Poate fi interesantă găsirea mai multor dintre aceste puncte optime locale, alături de optimul global, pentru a afla mai multe informații despre funcția obiectiv sau pentru a analiza aceste puncte după terminarea procesului de optimizare din perspectiva altor criterii.

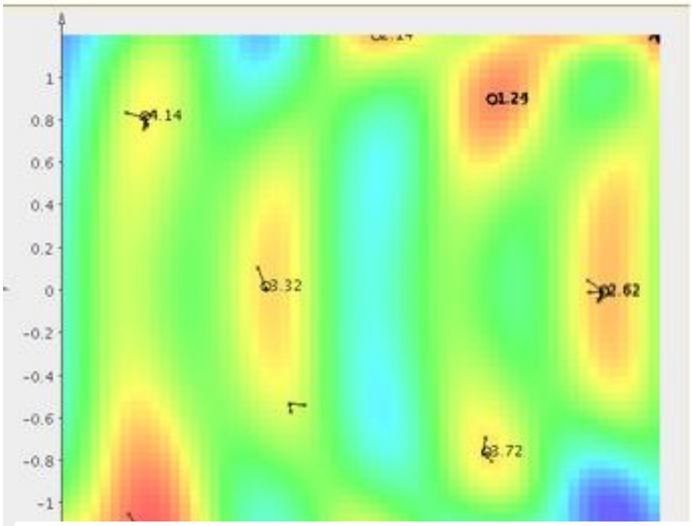


Figura 6.1 Prezența mai multor puncte de optim local pe suprafața domeniului de soluții.

Utilizatorul trebuie să selecteze un sub-set de variabile de proiectare, dintre toate variabilele analizate, care va fi variat de algoritmul de optimizare. Variabilele de analiză sunt utilizate pentru a calcula funcţii de analiză, cum ar fi: tensiuni, deformații, deplasări, flux termic, frecvenţele naturale, etc Unele dintre multiplele funcții de analiză sunt combinate pentru a defini obiectivul, sau funcția merit. Software-ul de optimizare, de obicei, utilizează un alt set de programe pentru a calcula funcțiile obiectiv şi pentru a analiza o structură. SolidWorks și ANSYS sunt exemple tipice, alături de subrutine scrise de utilizator. Cele mai multe modele reale au mai multe locaţii ale punctelor de minim local și, de obicei, nu putem fi siguri dacă a fost găsit cel mai bun design (minim global). Suprafeţele funcției merit arată de multe ori ca în figura prezentată mai jos, care are un minim global la (0, 0), dar mai multe minime locale în jurul acestuia.

Aplicarea pe scară largă a algoritmilor evoluționiști în probleme inginerești a fost împiedicată de cerințele computaționale restrictive. Algoritmii genetici necesită multe (de obicei de ordinul sutelor) soluții candidat să fie create și analizate pe durata execuției. Astfel, dacă evaluarea valorii funcției fitness pentru o structură (incluzând analiza structurii) durează 5 minute, pentru o structură de dimensiuni mari, evaluarea repetată necesară pentru o populație de 100 de structuri pe durata a 50 de generații ar fi estimată să se extindă pe 25.000 de minute (417 ore). O metodă tradițională de optimizare ar putea să necesite mai puțin timp pentru rezolvarea problemei, acesta fiind un argument pentru utilizarea acestor metode în cercetarea designului structural optim. Însă metodele tradiționale adesea au nevoie de premise restrictive și soluțiile găsite au nevoie să fie transformate pentru a echivala restricțiile practice de design, ceea ce necesită îndepărtarea lor de la soluția optimă din punct de vedere matematic. Deoarece GA utilizează variabile de design discrete, pot adesea găsi soluții practice mai bune decât metodele tradiționale.

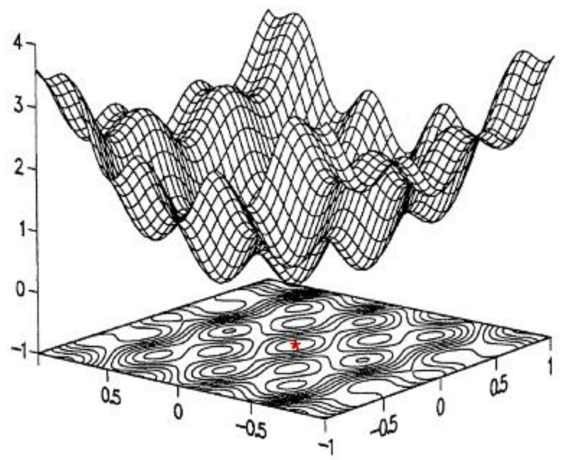


Figura 6.2 Exemplu de funcție obiectiv dificilă, cu mai multe puncte de optim local

Chiar dacă soluțiile obținute de algoritmii genetici nu pot fi dovedite a fi "optime", majoritatea aplicațiilor inginerești nu cer design "optim", ci mai degrabă soluții "foarte bune". Ținând cont de felul în care, în inginerie, sunt definite proprietățile materialelor, magnitudinile încărcărilor din vânt, accelerația terenului, etc., există prea multe incertitudini în definirea problemei pentru a justifica o insistență pe găsirea optimului absolut pentru valori aproximate ale problemei de calcul. Așadar algoritmii genetici sunt o metodă foarte atractivă pentru obținerea unor soluții foarte bune la problemele de design structural.

6.3 EFICIENȚĂ ȘI INCERTITUDINE ÎN PROCESUL DE OPTIMIZARE

Problemele pot ajunge în mod curent sa aibă mii sau zeci de mii de soluții, fiecare necesitând a fi analizată pentru a i se determina valoarea fitness. Ca rezultat, nivelul de complexitate a structurii este limitat de numărul de variabile necesare descrierii acesteia, și a nivelului de efort de calcul necesar găsirii unor soluții bune (în termen de zile de rulare).

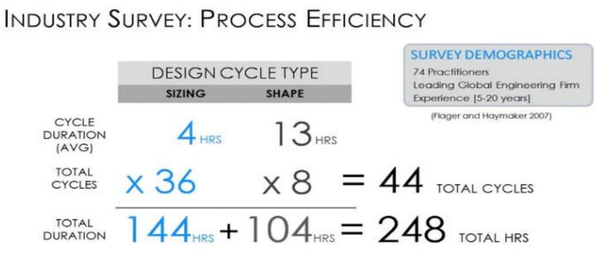


Figura 6.3 Numărul de ore mediu dedicat unui proiect.

Din punct de vedere statistic, erorile umane în domeniul proiectării și construcției tind să crească considerabil atunci când inovația este fragmentată și bruscă și atunci când nu există o evoluție graduală bazată pe cunoaștere științifică. Morfologia structurală libera (free-form design, FFD) care își are originea în dezvoltarea FFD, a avut o adevărată explozie în știința și tehnologia construcțiilor, care sunt în mod tradițional ancorate în tipologii și geometrii convenționale (cadre, grinzi, plăci, etc). Acest fapt a generat o modificare radicală în metodologia ingineriei structurale, mai ales referitor la controlul interpretativ al rezultatelor, a stării de tensiuni și deformații a structurilor supuse la încărcări gravitaționale, vânt, cutremur, obținute prin analiza cu element finit.

În EN 1990:2002 se încearcă garantarea nivelului de siguranță și performanță printr-o strategie de asigurarea calității - Quality Assurance (QA) strategy și proceduri de controlul calității procesului de design (punctul 2), în încercarea de minimizare a erorilor umane (punctul 8).

Estimarea **fiabilității structurale** depinde de calitatea cunoștințelor disponibile proiectantului. Cu cât acestea sunt completate cu cunoștințe noi despre structură, estimările devin mai complexe și, în general, gradul de incertitudine este redus – în mod deosebit acest lucru este vizibil în faza de design conceptual, când informațiile referitoare la rezistența materialelor, tipologia structurii, etc. devin disponibile și înlocuiesc prezumțiile bazate pe performanțe trecute sau experiența cu structuri similare. În cazul FFD nu există încă un feedback util disponibil în literatura tehnică.

Conform Majowiecki (1998, 1990), reducerea incertitudinilor în designul structurilor speciale poate fi realizată dacă sunt luate în considerare următoarele:

- evitarea colapsului progresiv al sistemului structural datorat cedării locale al elementelor structurale secundare;
- compatibilitatea restricțiilor și detaliilor de design cu ipotezele de modelare și răspunsul real al structurii;
- senzitivitatea parametrică a structurii, care depinde de tipul și gradul de nedeterminare statică.

Este de asemenea util accesul la un feedback sistematic asupra răspunsului structurii și monitorizarea performanțelor unor astfel de structuri pentru ca eficiența pe termen lung a designului să poată fi evaluată.

În cazul structurilor deplasabile, baza cunoașterii se referă în principal la sistemul de macarale iar procesul de design conceptual legat de acestea trebuie să ia în considerare observațiile existente, teste și specificații legate de comportarea unor structuri similare. Pentru a acoperi lacunele în acest domeniu, IASS a realizat un raport cu "state of the art" al acestor sisteme de acoperișuri, care include recomandări pentru design bazate pe observații ale unor cedări și colapsuri (IASS, 2000).

Incertitudinile fizice sunt legate de încărcări și de caracteristicile materialului. În cazul unor suprafețe construite extinse sau al clădirilor înalte cu morfologii neobișnuite, incertitudinile legate de încărcări pot fi reduse dacă se iau în considerare:

- distribuția și acumularea zăpezii în relație cu intensitatea și direcția vântului corelată statistic;
- distribuția presiunii vântului luând în considerare valori time-history sau puteri spectrale teoretice și experimentale corelate;
- efectul în timp al acțiunilor indirecte co-active ca și tensiuni inițiale, curgere lentă și efectele temperaturii.

Designul asistat de experimente (Eurocode 3 - punctul 8), cum sunt investigarea experimentală în tuneluri de vânt a modelelor la scară, și monitorizarea structurilor construite, au un rol important în designul sistemelor structurale atipice. Incertitudinile legate de material, asociate cu raporturi foarte mari de încărcări utile/greutate proprie, care sunt o caracteristică evidentă a structurilor ușoare, cresc considerabil aceste incertitudini statistice.

6.4 UTILIZAREA UNOR FORME NATURALE

Formele din natură au parte de aceleași limitări ca și mediul construit și de aceea au o relevanță majoră în a servi drept inspirație. Economia de cost, a forței de muncă, aplicarea de metode noi, simplificate, utilizarea de materiale inovative și tehnologii ecologice, forme și design remarcabile; toate acestea sunt trăsături fundamentale ale optimizării structurale. Noile tendințe și cercetarea în acest domeniu au fost dezvoltate în ultimele decenii de aplicarea cunoștințelor și observațiilor obținute din studiul proceselor naturale, a organismelor, a structurilor și materialelor, de la nivelul particulelor subatomice la comportamentul insectelor și animalelor, a anatomiei, a relațiilor ecologice din habitate naturale, și apoi aplicarea acestor cunoștințe la designul structurilor și mediului construit. Rezultatele sunt extrase din analiza atentă și sistematică a modurilor în care natura a proiectat structuri. Pe această bază putem dezvolta criterii și strategii pentru a evolua structurile într-o manieră asemănătoare, în mod eficient și sustenabil, găsind resurse noi, și răspunzând la mediul dinamic în care structurile sunt plasate.

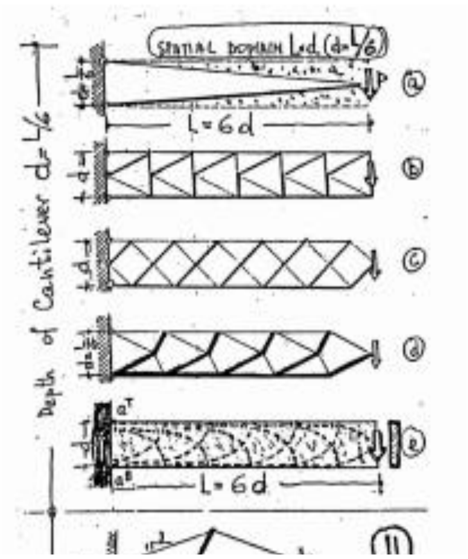


Figura 6.4 Acest set de schițe (Zalewski, 2002) ilustrează derivarea unei forme structurale eficiente care să preia forțele la care e supusă o grindă în consolă. Odată cu alinierea progresivă a elementelor structurale cu direcțiile vectorilor de tensiuni în grinda în consola conceptuală (schițată în primele 5 diagrame), cantitatea de material utilizată în structură descrește. Dacă formei îi este permisă extinderea nerestricționată, acesta devine și mai eficientă.

Ca formă structurală, forma de ramuri este capabilă să fragmenteze forțele prin distribuția mai uniformă a încărcării în structură și transferarea lor fundației construcției. Inspirația acestor forme a dus la dezvoltarea de soluții pentru unele din cele mai înalte clădiri din lume. Amalgamarea ingineriei cu biologia permite structurilor să fie eficiente și durabile, iar beneficiile utilizării unor precedente organice are un impact major asupra designului, îmbunătățind calitatea generală a proiectelor.

Dar dacă formele din natură asigură o sursă abundentă de inspirație, Tsui avertizează că nu putem alege o formă și să încercăm să o aplicăm la o scară mai mica sau mai mare fără consecințe dezastruoase (Tsui, 1999). Teoreticienii structurali Edward Allen și Waclaw Zalewski (1998) au explorat prin diagrame o serie de soluții pentru grinzi în consolă, analizând modul în care designul elementului evoluează ca răspuns la vectorii forțelor interioare. Aceștia au observat faptul că, cu cât elementele structurale sunt modificate pentru a urma mai bine aceste linii de forțe, cu atât structura este mai rezistentă, înspre punctul în care din setul de configurații, cea mai expresivă ca formă, cea mai eficientă structural este de asemenea și cea mai eficientă ca și consum de material. Forma evoluată devine „potrivită intrinsec" pentru mediul exterior dat, și conectează obiectul final cu conceptul formării acestuia.

Zalewski subliniază că structurile nu sunt artă, ele existând pentru un singur scop, a satisface nevoile umane. Însă pot fi elegante și estetice, nu datorită faptului că imită o formă naturală, copaci, oase, etc. – criteriile de frumusețe pentru flori nu sunt aceleași cu cele pentru structuri. Aceste forme sunt la o scară diferită, prea mică pentru a fi transpusă direct în structuri de dimensiunile la care avem nevoie. Structurile trebuie sa găsească propria formă naturală, care să reiasă din curbe funiculare, diagrame de momente, curgerea internă a forțelor.

Aliniat cu metodologia designului (planul de lucru), proiectarea conceptuală poate fi definită ca o abordare bazată pe cunoștințe și intuiție care permite identificarea tipologiei structurii, elaborarea unui model numeric preliminar și apoi aplicarea analizei structurale și verificării fiabilității. Aceste concepte sunt incluse în unele coduri de construcție naționale, care sunt în mod normal îndreptate doar spre sisteme structurale convenționale. În ceea ce privește designul inovativ, cum este cazul majorității construcțiilor recente de tip free-form, există puține recomandări.

Tiparele de creștere

Forma iconică a multor turnuri a rezultat din sintetizarea biomimetismului (derivat din cuvintele bios, însemnând viață, și mimică, imitație) și a designului structural. Influența biologică a început prin investigarea secțiunilor transversale a unor structuri naturale care prezintă un tipar de creștere exponențial, cu segmentare predictibilă matematic, care au rol de contravântuiri. Luând ca exemplu tiparele de creștere ale tulpinii de bambus și geometria fractală a unei cochilii de nautilus se pot deduce anumite criterii de

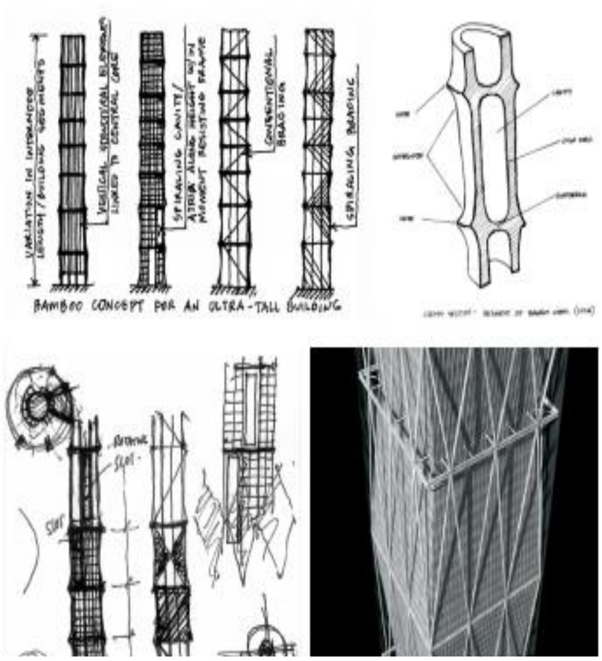


Figura 6.5 Evolutia formei adoptate pentru proiectul China World Trade Center Tower

formare a rezistenței naturale. Aplicațiile acestor principii împrumută secvențe specifice ale structurilor organice pentru a reproduce caracteristicile cele mai atractive ale acestora (o valoare mare a raportului rezistență-greutate, comportament elastic, anduranța pe termen lung, și o formă eficientă care rezistă încărcărilor și maximizează stabilitatea).

Tulpina de bambus are caracteristici structurale unice. Aceste tulpini lungi și subțiri asigură suportul unui frunziș extins, pe durata vieții plantei, și este un material de construcție rezistent atunci când este folosit la structuri construite. Chiar și atunci când este supus la tsunami, bambusul răspunde eficient la încărcările laterale, datorită proporțiilor geometrice. Nodurile, sau diafragmele, în formă de inele pe lungimea tulpinii, nu sunt distribuite uniform – sunt mai apropiate între ele la baza tulpinii, și distanțate progresiv înspre vârf. Localizarea acestor diafragme nu este aleatorie și poate fi determinată matematic. Acestea sunt distribuite astfel pentru a preveni flambajul pereților subțiri ai bambusului, sub încărcările gravitaționale și laterale. Grosimea pereților și diametrul pot fi calculate în mod similar. Toate ecuațiile care definesc localizarea diafragmelor, diametrul și grosimea pereților se bazează pe o formulare pătratică. Dacă se face un grafic cu diametrul necesar vs. înălțimea tulpinii (cu relația dintre diafragme și grosimea pereților similară), acesta arată ca și diagrama de încovoiere a unei grinzi în consolă sub încărcări laterale – teoria structurală este aceeași pentru bambus și alte structuri în consolă. Bambusul e format dintr-o tulpină, compusă din noduri și internoduri. Nodurile marchează localizarea diafragmelor unde există o ușoară modificare în diametru. Internodurile sunt goale, formând o cavitate înconjurată de pereții tulpinii. Materialul tulpinii este localizat la cea mai mare distanță de axa neutră a tulpinii, asigurând cea mai mare rezistență la încovoiere posibilă, și permițând încărcărilor gravitaționale să existe doar în anvelopă și minimizând astfel greutatea totală. Caracteristicile geometrice ale bambusului sunt aplicate la sistemul structural al proiectului China World Trade Center Tower. Turnul este divizat în opt segmente pe înălțime. Cerințele structurale datorate încărcării laterale au cele mai mari valori la baza tulpinii (sau turnului) și de aceea distanțele internodale sunt mai mici comparate cu jumătatea superioară. Distanța mai mică crește momentul capabil și rezistența la flambaj. În jumătatea superioară a turnului, distanțele internodale scad proporțional cu diametrul diafragmelor. Astfel forma tulpinii (turnului) răspunde la încărcările exterioare. Proiectul foarte eficient pentru China World Trade Center a fost realizat cu un nucleu intern conectat la un tub perimetral, aceste conexiuni fiind definite matematic pentru a contravântui cadrul împotriva flambajului în concordanță cu tiparele creșterii bambusului.

Diagrid

Gridul, care formează cadrul structural perimetral este modelat după forma de mesh a structurilor celulare biologice. Acest cadru e caracterizat de o membrană de diagonale cu spațiu mic între ele, care este mult mai deasă decât forma unui diagrid convențional.

Diafragmele de nivel concentrice sunt rotite pe înălțimea clădirii pentru a spori stabilitatea. Spațiul dintre cele patru elemente masive devine mai mic la jumătatea înălțimii apoi converge spre bază, astfel încât etajele inferioare asigură rigiditate și echilibru în timp ce etajele superioare sunt mai flexibile, acolo unde forțele de încărcare din vânt sunt mai mari.

Pentru a rezista încărcărilor laterale, elementele verticale și orizontale ale sistemului lateral sunt combinate pentru a crea un mesh diagonal, unde fiecare element este în întindere sau compresiune, fără forțe de încovoiere, rezultând o structură optimă cu rezistență și rigiditate potrivite pentru înălțimea clădirii. Compoziția gridului structural s-a bazat pe derivarea matematică a grinzii în consolă optime create original de Anthony Mitchell în 1904, oferind o geometrie eficientă. Forma optimă a unei grinzi în consolă este rotunjită, compusă din liniile de curgere a forțelor. Bazat pe seria Fibonacci și proporțional cu tiparul spiralelor unei cochilii de nautilus, gridul structural a fost realizat cu un factor de scalare care concentrează diagonalele într-un punct și apoi devine progresiv mai puțin concentrat înspre exterior.

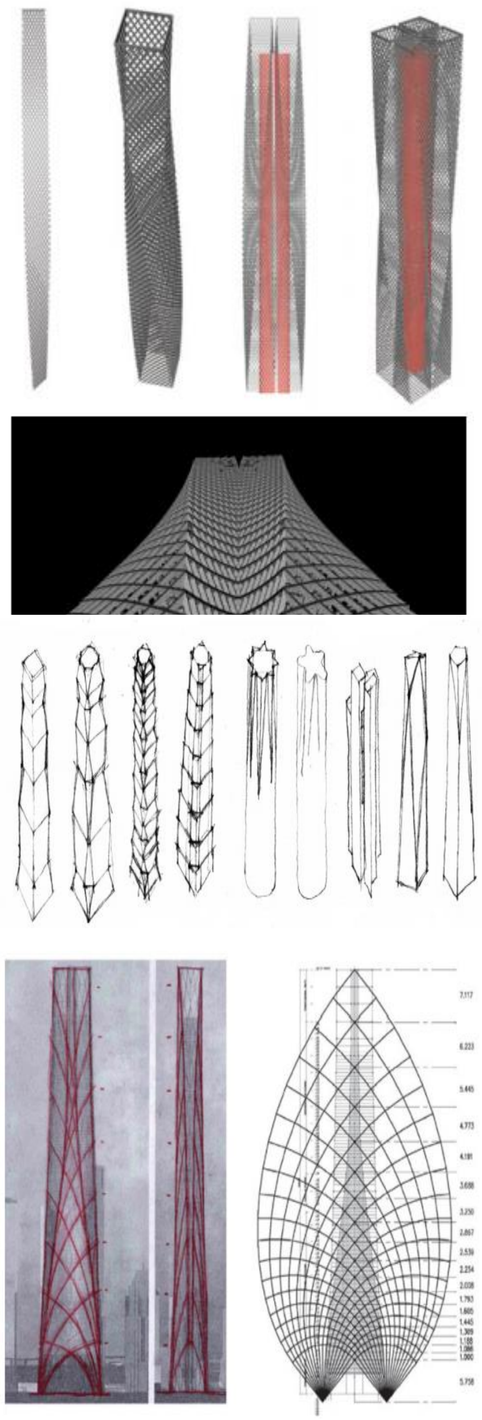


Figura 6.6 Conceptul structural al Jinling Hotel Tower în Nanjing, China a condus la reduceri substanțiale în cantitatea de material structural și dimensiunea elementelor structurii necesare cadrului perimetral.

7 VARIABILE DE PROIECTARE ȘI VARIABILE DE OPTIMIZARE

Sunt cantități numerice reale care trebuie determinate în urma proiectării unei structuri. În cadrul variabilelor de proiectare pot să apară și cantități cunoscute (determinate din condiții de funcționare a sistemului), care poartă numele de parametri.

În funcție de natura variabilelor de proiectare există două tipuri de aplicații de optimizare: optimizare dimensională și optimizare configurativă. Optimizarea configurației se referă la acea clasă de probleme la care orice schimbare a variabilelor de proiectare produce modificări în geometria problemei sau a discretizării. În afara problemelor tipice de optimizare dimensională și de formă mai există o clasă specială de probleme la care atât parametrii dimensionali cât și cei de formă se definesc ca variabile de proiectare.

Într-o etapă timpurie a procesului de proiectare (conceptuală și faza de definire a proiectului), este de o mare importanță găsirea celei mai bune topologii structurale posibile, în contextul obiectivelor de proiectare și constrângerilor. Astfel, în ultimul deceniu, eforturi de cercetare substanțiale au fost dedicate dezvoltării unor metode computaționale de optimizare structurală eficiente și de încredere cum sunt optimizarea formei structurale [Bletzinger, Ramm, 2001] și optimizarea topologiei (metode de optimizare structurale evolutive, ESO).

Cele mai utilizate criterii care stau la baza modelelor de calcul pentru optimizarea structurilor sunt: greutate minimă, tensiuni minime (rezistență maximă), energie potențială de deformație minimă, rigiditate maximă, deplasări minime, rigiditate maximă pentru o greutate dată, formă de egală rezistență, cost minim etc. Relația dintre tensiuni (uneori eforturi) și forma structurii este factorul fundamental atât în proiectarea curentă, cât și în cea optimală, această dependență folosindu-se fie pentru determinarea tensiunilor când se cunoaște configurația structurii, fie pentru determinarea formei structurii când se cunosc (sau se impun) valorile maxime ale tensiunilor. Criteriul de alegere a formei structurii depinde de condițiile care trebuie satisfăcute de structură, fiecare criteriu având o importanță decisivă asupra rezultatului optimizării. Criterii "absolute" de optimizare nu există și nici nu par a fi de dorit. Cea mai simplă procedură de „optimizare" este "optimizarea intuitivă", care constă în realizarea de modele ale unor soluții alternative ale structurii și - prin încercări repetate – de a obține o variantă optimă a acesteia. Procesul este empiric și nu duce cu certitudine la cea mai bună soluție posibilă. Aceste instrumente de calcul în general nu sunt utilizate de arhitecți sau ingineri care ar avea nevoie de metode simplificate de calcul al formei și optimizării structurale în stadiile incipiente critice ale procesului de proiectare. "Principiile structurilor ușoare din natură" introduc metode de legătură între teoria de optimizare structurală și aplicarea acesteia în practica de proiectare structurală, care utilizează metode simplificate și programe pentru form-finding și optimizare bazate pe procesele din natură.

Integrarea morfologiei structurale în practica curentă (form-finding) ar simplifica considerabil procesul de optimizare. Aplicarea unor metode robuste de analiză structurală, cum este metoda elementelor finite (FEM), reduce timpul necesar unui ciclu de optimizare prin determinarea caracteristicilor de performanță a structurii și informarea directă a proiectantului despre modalitățile în care structura poate fi modificată pentru a i se îmbunătăți proprietățile în funcție de cerințe sau obiective. Procedurile complet automatizate de proiectare permit participarea eficientă, activă și creativă la procesul de dezvoltare a unui design. Acestea reduc timpul necesar acestui proces și găsesc cele mai bune soluții în mod sistematic [Kress, Keller, 2007].

Diferiți algoritmi de optimizare au puncte forte și puncte slabe diferite. Unii funcționează bine pentru anumite clase de probleme în timp ce pentru altele sunt ineficienți. Nu doar au performanțe diferite pentru clase diferite de probleme, ci se și comportă diferit în funcție de diverse instanțe ale aceleiași probleme. Conform teoremei „No free lunch”, în categoria algoritmilor euristici nu există un algoritm care să aibă în general o performanță mai bună ca a celorlalți.

În procesul rezolvării unei probleme concrete, primul pas este alegerea unui algoritm. Însă, deoarece majoritatea algoritmilor au un set de parametri care le controlează comportarea, această alegere devine și mai dificilă. Alegerile referitoare la valorile acestor parametrii pot avea un impact major asupra performanței algoritmului. În cazul GA, acest lucru a fost exemplificat în (Bäck, 1993) și (Ochoa, 2000). Alegerea parametrilor optimi pentru un singur algoritm pentru rezolvarea unei singure probleme de optimizare este deja o problemă non-trivială. DeJong a încercat să găsească parametri optimi pentru GA care să funcționeze pentru orice problemă, însă, deși unele dintre aceste valori sunt în prezent utilizate ca valori implicite pentru GA, în cele mai multe cazuri, acestea necesită a fi adaptate pentru fiecare instanță a problemei vizate, pentru a obține o performanță optimă a algoritmului (De Jong, 1975).

Modificarea acestor parametri se poate face în două moduri: prin *tuning* si prin *control*. Tuningul parametrilor este cea mai des întâlnită opțiune, care caută valorile cele mai potrivite ale acestor parametrii *înaintea* începerii optimizării. Aceste valori rămân fixe pe întreaga durată a rulării algoritmului.

Controlul parametrilor

Când valorile parametrilor sunt schimbate în timpul execuției unui algoritm, trebuie să fie definite unele reguli care guvernează această schimbare.

Există mai multe abordări diferite pentru aceasta:

• Control determinist: O regulă deterministă este folosită pentru a se adapta valorile parametrilor de comportament fără să reacționeze la vreun feedback de la procesul de căutare. Aceste reguli se bazează de obicei pe timp sau pe numărul de iterații.

• Control adaptiv al parametrilor: Când o regulă de adaptare este implementată, feedback-ul de la procesul de căutare este utilizat pentru a controla modul în care valorile parametrilor se schimbă. De exemplu, dacă algoritmul detectează că diversitatea populației devine prea scăzută, aceasta ar putea crește valoarea operatorului de mutație și viceversa. Regula 1/5 in ES este un exemplu de control al parametrilor de adaptare.

• Controlul auto-adaptiv al parametrilor: Auto-adaptarea este inspirată de ideea de evoluție. Parametrii sunt codați în cromozomii indivizilor și se supun acelorași mecanisme ca indivizii, și anume mutația, recombinarea, și selecția. Indivizii cu fitness mare răspândesc cromozomii lor în rândul populației cu o probabilitate mai mare, astfel încât valorile bune ale parametrilor sunt folosite mai des. Ele se pot schimba, de asemenea, în timp datorită mutației. Totuși, această abordare este posibilă numai pentru parametrii la nivel de individ, cum ar fi valoarea mutației. La nivel de populație, parametrii cum ar fi mărimea populației sau operatorul de selecție nu pot fi adaptați în acest fel.

Tuningul parametrilor

Chiar dacă controlul parametrilor are avantajele sale, este mult mai complexă problema găsirii unor reguli și metode care să poată adapta o valoare-parametru într-un mod care îmbunătățește performanța unui algoritm, decât determinarea în avans a unei valori a unui parametru. Metodele de control al parametrilor sunt, de asemenea, în cea mai mare parte adaptate pentru parametrii cu valori reale. Când vine vorba de alegerea operatorilor potriviți pentru un algoritm, tuningul parametrilor este frecvent utilizat. Cu toate acestea, alegerea valorilor parametrilor rămâne o sarcină complexă, care necesită ca un utilizator sa aibă experiență și o înțelegere profundă în interdependențele parametrilor și a impactului acestora.

Metodele de ajustare a parametrilor pot fi clasificate după cum urmează:

• Ad-Hoc: Pentru alegerea valorilor parametrilor, mulți utilizatori se bazează pe convenții și valori implicite (Smit, 2009). Bazat pe valorile implicite, valorile parametrilor sunt variate până când se atinge o performanță acceptabilă. Problema este că uneori valorile optime ale parametrilor diferă foarte mult de la valorile implicite. În plus, numărul de repetiții pentru o parametrizare testată este adesea prea scăzut. Cât de bine funcționează această abordare depinde de experiența utilizatorului, însă aceasta este, de obicei, o abordare foarte rapidă.

• Experimental: Efectuarea de experimente sistematice cu diferite setări de parametri este o abordare mai științifică. Cu toate acestea, încercarea tuturor valorilor diferite ale parametrilor și combinațiilor acestora este imposibilă în cele mai multe cazuri. Acest lucru conduce la reducerea combinațiilor sau doar la variația parametrilor individuali. Optimizarea parametrilor, unul câte unul, mai mult ca sigur nu conduce la setările optime, deoarece majoritatea parametrilor tind să aibă efecte asupra altor parametri. Bartz-Beielstein a dedicat o carte întreagă pentru domeniul de testare și cercetare experimentala în calcul evolutiv (Bartz-Beielstein, 2005).

• Meta-optimizarea: O altă abordare de control al parametrilor este de a vedea căutarea pentru parametrii optimi ca o problemă de optimizare în sine. Din această perspectivă, căutarea pentru setări optime are mult în comun cu problemele tradiționale de optimizare. Valorile pentru un set de variabile de intrare (valorile parametrilor), trebuie să fie găsit pentru a maximiza calitatea atunci când e evaluat (în timpul execuției algoritmului). Interdependențele variabilelor de intrare și efectele lor asupra calității soluțiilor sunt cunoscute și spațiul de căutare este foarte mare. Pentru a rezolva această problemă de optimizare, un algoritm de optimizare este aplicat ca un optimizator meta-nivel. Acest concept este numit meta-optimizare.

8 RESTRICȚIILE DE PROIECTARE ȘI FUNCȚIA OBIECTIV

Un set de valori atribuit variabilelor de proiectare reprezintă o soluție de proiectare care definește o structură. Dacă structura respectivă îndeplinește condițiile pentru care a fost proiectată, aceasta este o structură fezabila. Condițiile care trebuie să le îndeplinească o structură ca să fie fezabilă poartă numele de restricții de proiectare. Numărul de restricții al unei probleme nu este obligatoriu să fie egal cu numărul variabilelor de proiectare. În majoritatea cazurilor, numărul restricțiilor de proiectare este mai mare decât numărul variabilelor de proiectare.

În proiectarea structurală se întâlnesc două tipuri de restricții: restricții de comportament și restricții de mărginire. Restricțiile de comportament sunt date de condițiile de rezistență și rigiditate impuse structurii, care permit acesteia să-și îndeplinească rolul pentru care a fost proiectată. Tensiunile echivalente von Mises reprezintă un exemplu tipic de condiții de comportament în proiectarea structurală ($\sigma_{ech} \leq \sigma_a$).

Restricțiile de mărginire provin din condițiile de limitare a unor variabile de proiectare. Restricțiile de proiectare se notează cu r_k (k=1...K) și se pot exprima explicit în funcție de variabilele de proiectare x.

Restricțiile de comportament determină domeniul în care se face proiectarea. Ele pot fi formulate pentru o comportare a structurii în domeniul elastic și în acest caz proiectarea se face în domeniul elastic. După cum este cunoscut, în cazul solicitării în domeniul elastic, structurile au o importantă rezervă de capacitate portantă pe care proiectarea în acest domeniu nu o poate utiliza. Dacă restricțiile de proiectare sunt formulate prin intermediul criteriilor de plasticitate, proiectarea optimală se face în domeniul plastic.

Funcția obiectiv este o funcție f(x), definită ca o funcție de variabile de proiectare (ce figurează și în restricțiile de proiectare) care este extremizată în cadrul procesului de optimizare.

Greutatea (sau volumul) unei structuri este un exemplu tipic de funcție obiectiv. **Alegerea funcției obiectiv reprezintă unul din cele mai importante aspecte ale procesului de optimizare.** Construirea modelului matematic al unei probleme de optimizare impune o cunoaștere temeinică a comportării sistemului studiat. Scrierea incorectă a unei condiții sau omiterea unor condiții importante pot conduce la obținerea unor rezultate inaplicabile în proiectare, deși din punct de vedere matematic acestea pot fi juste.

Există situații în care funcția obiectiv este constituită din două sau mai multe valori cantitative. În asemenea situații se definește o funcție obiectiv compusă. Astfel dacă $f_1(x)$ și $f_2(x)$ sunt două funcții obiectiv ale unei probleme, se poate defini o funcție obiectiv compusă de forma

$$f(x) = a_1 f_1(x) + a_2 f_2(x)$$

unde a_1 și a_2 sunt constante de pondere.

Restricțiile r_k (k=1...K) și funcția obiectiv f(x) reprezintă modelul problemei de optimizare formulată. În cadrul unui proces de optimizare, funcția obiectiv este extremizată în vederea găsirii combinațiilor de variabile de proiectare pentru care aceasta capătă valori maxime sau minime.

Dacă se consideră o funcție obiectiv

$$f = f(x_1, x_2)$$

dependentă de două variabile, graficul acesteia este dat de suprafața Σ.

Intersecția acestei suprafețe cu plane paralele cu planul orizontal ($x_1 0 x_2$), dă naștere unor contururi închise, care proiectate în planul orizontal formează o familie de curbe circumscrise Γ_i. Curbele rezultate din secționarea suprafeței Γ cu plane mai apropiate de planul orizontal sunt situate în interiorul curbelor corespunzătoare unor înălțimi de secționare mai mari, dacă suprafața Σ admite un punct de minim. Dacă suprafața Σ ar admite un punct de maxim, dispunerea acestor curbe ar fi inversă.

8.1 FUNCȚII-OBIECTIV GLOBALE

O funcție obiectiv globală definește modul în care obiectivul global depinde de parametrii de design și cum va fi influențat de modificările valorilor variabilelor de design. Însă, ca proprietate globală, va fi independentă de coordonate spațiale. Funcția obiectiv globală este formulată astfel încât valoarea sa minimă absolută să corespundă obiectivului, sau celei mai bune soluții referitor la obiectiv,

$$\min\{f(x)\},$$

iar dacă obiectivul este maximizarea unei proprietăți cum ar fi volumul V, funcția obiectiv va lua fie forma f(x)=-V(x) ori f(x)=1/V(x). În contextul utilizării tehnicilor algoritmilor evoluționiști, funcția obiectiv este numită funcție fitness.

Restricțiile sunt exprimate prin funcții de restricție, care sunt împărțite în egalități g și inegalități h:

$$g_i(x) \leq 0, \quad i = 1, \dots, m$$

$$h_j(x) = 0, \quad j = 1, \dots, n$$

Inegalitățile formează hiperplanuri în spațiul de căutare împărțindu-l în regiuni fezabile și regiuni nefezabile, unde restricțiile sunt încălcate. Atât timp cât vectorul variabilelor de optimizare este îndreptat spre o regiune fezabilă, inegalitățile sunt inactive și nu restricționează căutarea. Inegalitățile devin active când restricțiile sunt încălcate sau e atinsă starea limită

$$g_i(x) \geq 0.$$

Restricțiile tip egalitate sunt întotdeauna active.

Problemele de optimizare multicriterială pot fi reduse la probleme de optimizare cu o funcție obiectiv scalară prin formularea unei probleme substitutive cu o funcție de preferință p astfel încât

$$\min_{X \in R^n} p[f(x)],$$

Astfel încât

$$p[f(\tilde{x})] = \min_{X \in R^\backslash} p[f(x)].$$

[Eshenauer] citează variate formulări ale funcțiilor de preferință dintre care menționez suma ponderată a obiectivelor:

$$p[f(x)] := \sum_{j=1}^{m} w_j = 1.$$

Dacă sunt considerate două sau mai multe criterii de design, respectivele funcții obiectiv pot fi minimizate simultan. Aceste proceduri se numesc optimizare multicriterială, optimizare vectorială sau optimizare multiobiectiv. În practică, mai multe tipuri de răspuns structural sau moduri de cedare trebuie considerate în procesul de design, iar aici intervine relevanța acestui tip de optimizare. Forma care o ia problema este:

$$\min_{X \in R^n} \{f(x) | h(x) = 0, g(x) \leq 0\},$$

Unde $f(x)$ este numit vectorul funcțiilor obiectiv ale variabilelor de design

$$f(x) \coloneqq \begin{Bmatrix} f_1(x) \\ \dots \\ f_m(x) \end{Bmatrix}.$$

La un anumit stagiu al procesului de optimizare se observă că o continuare a minimizării uneia dintre funcțiile obiectiv determină creșterea valorii unei alte funcții. Situația descrisă se numește conflict de obiective deoarece niciuna dintre soluții nu permite optimizarea simultană a tuturor obiectivelor.

Un vector se numește Pareto-optim dacă și numai dacă nu există nici un vector $x \in X$ pentru care

$$f_j(x) \leq f_j(x^*) \quad pentru\ oricare\ \ j \in \{1, \dots, m\}$$

$$si\ f_j(x) < f_j(x)\ pentru\ cel\ puțin\ o\ valoare\ \ j \in \{1, \dots, m\}.$$

Dacă toate obiectivele sunt convexe, un set de soluții Pareto-optimale poate fi generat prin rularea unei secvențe de probleme scalare substitut unde funcțiile de preferință acoperă un interval de valori potrivit pentru factorii de ponderare w. Dacă unul sau mai multe obiective nu sunt convexe, setul Pareto-optim poate fi greu de generat.

8.2 METODE DE TRANSFORMARE ȘI PSEUDO-OBIECTIVE

Găsirea minimului unei funcții obiectiv cu restricții complică și mai mult procesul de menținere a soluțiilor candidat în zona fezabilă. Algoritmii existenți pentru căutare fără restricții pot fi utilizați pentru rezolvarea problemelor cu restricții care sunt supuse unor anumite transformări. Printre acestea se numără metoda penalizărilor și metoda multiplicatorilor.

Metodele de penalizare transformă funcția obiectiv $f(x)$ și funcțiile de restrictie h și g într-o funcție obiectiv transformată p fără restricții explicite. Funcția devine un pseudo-obiectiv și este obținută prin adăugarea la f a funcției de penalizare Ω care este compusă din restricții și parametrii de penalizare R.

$$p(x, R) = f(x) + \Omega\big(R, g(x), h(x)\big).$$

Funcția Ω poate fi definită ori prin **metoda punctului exterior** ori **metoda punctului interior**.

Un exemplu de aplicare a metodei punctului exterior este utilizarea penalizării pătratice astfel încât încălcarea restricțiilor este penalizată:

$$\Omega(x,R) = R\sum_{j=1}^{m}\left\{\max[0,g_j(x)]\right\}^2 + R\sum_{k=1}^{l}\{h_k(x)\}^2.$$

Astfel, dintre restricțiile $g(x)$, doar cele active sunt considerate în formulare. Prin adăugarea acesteia la $f(x)$ rezultă o funcție fără restricții al cărei punct de minim se află în afara regiunii fezabile, de unde se trage și denumirea metodei. Prin creșterea valorilor parametrului de penalizare R minimul se deplasează tot mai aproape de zona fezabilă din spațiul de căutare dar nu poate să o atingă.

O metodă de penalizare de punct interior rezultă din selectarea unei forme pentru Ω care să forțeze punctele staționare ale $P(x,R)$ să fie fezabile. Acestea se mai numesc metode de barieră, deoarece penalizarea formează o barieră de valori P la limita regiunii fezabile. Deoarece păstrarea restricțiilor poate fi esențială pentru obținerea de soluții sigure de design, este preferată această metodă de găsire a soluțiilor îmbunătățite în interiorul domeniului fezabil. Pentru restricțiile sub formă de inegalități, este penalizată apropierea de zona nefezabilă, înainte chiar de încălcarea restricțiilor. Aceasta se poate obține, de exemplu, prin factorul de penalitate invers:

$$\Omega(x.R) = R'\sum_{j=1}^{m}\frac{-1}{g_j(x)} + R'\sum_{k=1}^{l}\{h_k(x)\}^2.$$

Prin reducerea valorii lui R' punctul de minim al obiectivului transformat se deplasează mai aproape de zona nefezabilă sau de minimul restricționat.

Soluțiile aflate aproape de minimul restricționat pot fi obținute prin utilizarea de valori foarte mari respectiv foarte mici pentru factorii de penalizare R sau R', și minimizarea pseudo-obictivelor pentru aceste valori, însă pseudo-obiectivele sunt distorsionate în comparație cu obiectivele originale. Metodele de căutare ale pseudo-obiectivelor sunt în general create să funcționeze pentru funcții obiectiv care se comportă asemănător funcțiilor cuadratice. Acestea pot să eșueze când sunt aplicate funcțiilor pseudo-obiectiv foarte distorsionate rezultate în urma aplicării parametrilor de penalizare când minimul este foarte aproape de punctul minim restricționat. De aceea, problema de optimizare cu restricții este rezolvată printr-un sir de subprobleme fără restricții, în care parametrii de penalizare sunt adaptați la fiecare pas. Prin metoda punctului exterior, parametrului R îi sunt atribuite valori mici, uneori zero la primul pas, și majorate succesiv în pașii următori. Pentru metoda punctului interior, se începe cu o valoare mare a lui R' care este progresiv scăzută. Este însă inevitabil ca subproblemele generate să devină progresiv rău-condiționate, astfel încât, la un moment dat, iterația să fie încheiată nu datorită găsirii unei aproximații suficient de bune pentru punctul de minim restricționat, ci datorită eșuării algoritmului de căutare.

9 MODELARE PARAMETRICĂ

Problema de optimizare include restricții, criterii de calitate și funcția obiectiv care trebuie minimizată sau maximizată în funcție de variabilele de proiectare. În general, variabilele determină direct geometria și proprietățile unei structuri (Valery, 1999).

Modelarea parametrică poate oferi o soluție, în contextul descris, la problema numărului mare de variabile necesare descrierii unei structuri. Modelele parametrice sunt capabile să descrie geometrii complexe utilizând un număr relativ redus de variabile, lăsând totodată loc pentru o marjă mare de variație. Software-uri care permit acest tip de manipulare a datelor structurale au fost dezvoltate - Generative Components (Bentley Systems), Grasshopper (Robert NcNeel), Digital Project (Gehry Technologies - Dassault Systemes) – special pentru tehnici de modelare parametrică la îndemâna inginerilor și arhitecților.

Aceste soluții care pot fi explorate cu ajutorul modelării parametrice pot fi însă în număr foarte mare, iar problema devine găsirea în rândul acestora a celor mai bune din punct de vedere al performanțelor dorite. Pentru acest tip de căutare, algoritmii genetici sunt foarte potriviți, datorită capacității modelului parametric de a utiliza un număr relativ mic de variabile.

Termenul *parametric* își are originea în matematică, dar există opinii divergente despre data exactă când a început să fie utilizat de către proiectanți. David Gerber [2007], în teza de doctorat intitulată *Parametric practice*, citează pe Maurice Ruiter ca fiind primul care utilizează termenul într-o lucrare din anul 1988 cu titlul *Parametric Design*.

În sensul utilizat astăzi de matematicieni, *parametric* semnifică un set de ecuații care exprimă un set de cantități ca funcții explicite de un număr de variabile independente, cunoscute ca „parametri” [Weisstein, 2003]. Această definiție subliniază două criterii importante: o ecuație parametrică exprimă un set de cantități cu un număr de parametri; și rezultatele (setul de cantități) sunt legate de parametri prin funcții explicite. Un exemplu de ecuație parametrică este formula ce descrie o curbă catenară:

$$
\begin{aligned}
x(a,t) &= t \\
y(a,t) &= a \cosh\left(\tfrac{t}{a}\right)
\end{aligned}
$$

Aceste două formule întrunesc criteriile menționate deoarece exprimă un set de cantități (x și y) în termeni de un număr de parametri (a, care controlează forma curbei, și t, care controlează locul de pe lungimea curbei unde se află punctul). În același timp rezultatele (x și y) sunt legate de parametrii (a și t) prin funcții explicite (nu există ambiguitate în relația dintre aceste variabile). Aceasta este originea termenului parametric: un set de cantități exprimate ca funcție explicită de un anumit număr de parametri.

Utilizarea ecuațiilor parametrice în modelarea structurilor este cel mai bine cunoscută prin proiectele lui Gaudi, dar este cel mai bine ilustrată de modelul lănțișorului [Burry, 2011].

Gaudi folosește acest principiu la designul capelei Colònia Güell prin crearea unui model răsturnat al capelei folosindu-se de sfori de care atârna greutăți. Datorită principiului lui Hooke, sforile vor găsi pozițiile de echilibru în forma care, inversată, va fi în compresiune pură. Modelul lănțișorului are toate componentele unei ecuații parametrice. Comparat cu utilizarea anterioară a ecuațiilor parametrice, de matematicieni, inovația modelului lui Gaudi constă în calcularea automată a rezultatelor parametrice (în schimbul calculului manual al soluțiilor formulei parametrice a curbei catenare, Gaudi a derivat automat forma acestora prin utilizarea încărcărilor gravitaționale asupra sforilor). Această metodă a calculului analog a fost mai departe dezvoltată de Frei Otto, care a inclus, printre altele, suprafețe minime obținute cu ajutorul baloanelor de săpun și căile minime obținute cu lână udă.

Modelarea parametrică trebuie să permită proiectantului să exploreze „o varietate de soluții” [Teresko, 1993]. Acest lucru este posibil prin manipularea directă a parametrilor și prin modificarea relațiilor ce compun modelul. Modelele parametrice permit ca deciziile să fie luate târziu în procesul de design, acest fapt fiind unul dintre cele mai interesante caracteristici ale modelării parametrice.

Ipek Dino [2012] argumentează că script-urile sunt inerent parametrice, notând faptul că sistemele parametrice se bazează pe principii algoritmice deoarece un algoritm ia un set de valori de intrare, execută o serie de pași computaționali care transformă datele de intrare, și produce un alt set de valori ca ieșiri. Interfețele de codare nu au observat o dezvoltare semnificativă de la introducerea AutoCAD în proiectare, însă în ultima decadă s-a observat emergența unei noi forme de interfețe de codare, **interfața vizuală**.

Programarea vizuală se bazează pe reprezentarea programelor nu ca text ci ca diagrame.

În arhitectură a fost introdus primul limbaj de programare vizuală când Robert Aish a început testarea variantei beta a Generative Components cu câteva firme de arhitectură în 2003. Rhinoceros (Rhino) este un program comercial de modelare 3D pe bază de geometrii NURBS, dezvoltat de Robert McNeel & Associates.

Ca multe alte aplicații de modelare, Rhino utilizează limbaje de scripting, bazate pe Visual Basic iar Grasshopper permite modelarea de componente în C#,Visual Basic și Python. David Rutten a creat pentru McNeel versiunea proprie a acestui mediu, în 2007, numit Explicit History, nume schimbat mai apoi în Grasshopper.

Ambele medii de programare sunt bazate pe grafuri care reprezintă relațiile dintre parametri, prin funcții definite de utilizator. Modificările aduse parametrilor sau relațiilor modelului determina propagarea schimbărilor prin funcțiile explicite și actualizarea automată a rezultatelor finale. Astfel, acestea reprezintă o metodă eficientă de creare a modelelor parametrice.

9.1 MODELAREA DATELOR ÎN GRASSHOPPER

Grasshopper nu folosește, spre deosebire de alte medii de programare, nici un *nume de obiect* pentru a defini un obiect. Acest lucru poate suna banal, dar este una dintre diferențele cele mai fundamentale față de un mediu de programare tradițional. În Grasshopper obiectul sau obiectele sunt plasate într-o listă. Diferitele liste de date sunt organizate într-o structură de date arbore în care fiecare ramură și conținut de date a ramurei să aibă un număr de index. Accesarea obiectul este astfel mai problematică decât într-un mediu obișnuit de scripting. Grasshopper dispune de diferite instrumente pentru a remedia această problemă. Aceste instrumente sprijină editarea și selectarea conținutului listei și editarea structurii de arbore de date. Cunoașterea acestor tehnici este esențială pentru utilizarea eficientă a Grasshopper. Parametrizarea designului înseamnă definirea variabilelor de design și a parametrilor de design ficși. Variabilele de design pot descrie configurația structurii, caracteristici cantitative cum sunt caracteristicile secționale, grosimi ale pereților, forme și proprietățile materialului.

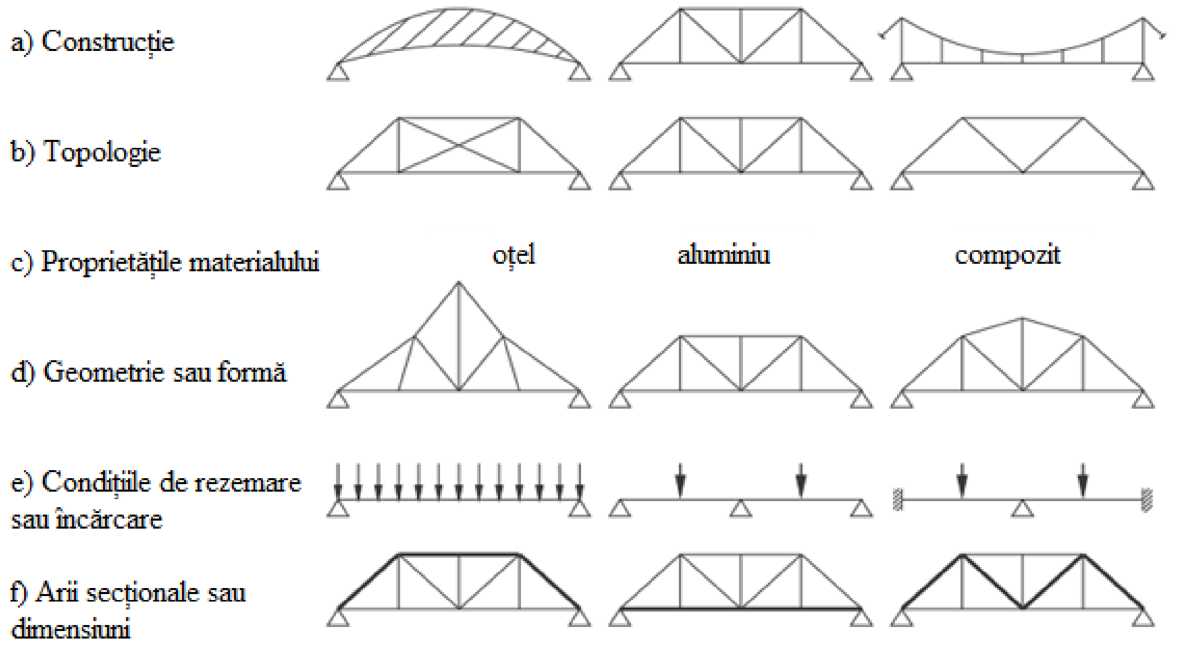


Figura 9.1 Clasificarea problemelor de optimizare pentru structuri articulate plane în termeni de variabile de design, conform Eschenauer.

Așa cum am prezentat în capitolul 4 – Forme de optimizare structurală, optimizarea structurilor poate fi clasificată în funcție de tipul variabilelor de design. În cadrul modelării parametrice, aceste variabile de design iau forma de parametrii, cu valori cuprinse între o limită inferioară și una superioară, ce trebuie cunoscute înaintea începerii procesului de optimizare. Luând în considerare o structură articulată plană, și pornind de la [Schmit, 1963] [Olhoff, 1983], variabilele de design pot fi împărțite în următoarele clase indicate în Fig. 9.1.

a. Forma constructivă. Determinarea celei mai bune forme presupune optimizarea fiecărui model luat în considerare și compararea soluțiilor optime rezultate.

b. Topologia. Topologia sau aranjarea elementelor în structură este în cele mei multe cazuri descrisă de parametri care pot fi modificați doar în pași de valori discrete. Topologii diferite mai pot fi obținute prin eliminarea de noduri și a elementelor de legatură. Este notabilă metoda de optimizare topologică introdusă de Bendsoe și Kikuchi [Bendsœ, 1988].

c. Proprietățile materialului. Acestea, în cazul materialelor de construcție cum sunt oțelul sau aluminiul descriu rigiditatea cu ajutorul modulului lui Young și a coeficientului lui Poisson, rezistența prin tensiunea maximă admisibilă sau alte limite de tensiuni, sau greutatea în funcție de greutate specifică sau densitatea materialului. Designerul poate adesea selecta cel mai potrivit material dintr-o selecție de aliaje și uneori poate decide dacă oțelul, aluminiul sau alt tip de aliaj este cel mai potrivit pentru a satisface cerințele obiectivului de design. Toate aceste alegeri sunt de natură discretă, ceea ce duce la un set discret de variabile de design cuprins în baza de date. Excepție o fac laminatele din materiale compozite anizotropice care pot avea variabile continue în funcție de orientarea fibrelor.

d. Geometria și forma. Geometria cadrelor sau fermelor este descrisă de coordonatele nodale. Forma corpurilor solide e dată de suprafețele care le definesc.

e. Condiții de rezemare și încărcări. Designul poate fi îmbunătățit prin modificarea condițiilor limită.

f. Dimensiuni. Pentru structuri de tip bare, ferme, grinzi, plăci sau membrane și modelele lor FEM, pot fi folosite ca variabile de design grosimea, aria secțiunii, momentul de inerție, etc. Este importanta distincția între variabile de design independente și variabile dependente. Când geometria secțiunii este exprimată printr-o variabilă, proprietățile enumerate mai sus sunt dependente de aceasta. Optimizarea dimensională conduce de obicei la probleme de optimizare discrete datorită valorilor dimensionale discrete ale secțiunilor comerciale disponibile.

9.1.1 Formularea obiectuală în procesul de calcul

Există două metode principale de utilizare a arborilor de către componentele create și folosite în studiul de față. Pentru definirea geometriei, legăturilor, proprietăților fizice ale materialelor, încărcărilor, etc., sunt utilizate componente, care funcționează ca un schelet pentru arhitectura programului. Componentele pot lucra pe seturi de obiecte (denumite ramuri), sau pot să combine elemente din mai multe intrări. Desigur, combinații din mai multe componente avansate sunt de asemenea posibile, dar, pentru claritate, voi începe cu cele două opțiuni principale:

- *Componente care lucrează pe ramuri de la o singură intrare.* În primul caz, componenta va funcționa numai pe datele de pe aceeași ramură din același arbore. De exemplu, o polilinie utilizează datele unei serii de puncte de pe o singură ramură pentru a defini polilinia. Deci, pornind de la conținutul unei ramuri este generată o polilinie și astfel pentru fiecare ramură a arborelui va fi creată o polilinie. Există un singur canal de intrare pentru punctele necesare pentru a defini polilinia.
- *Componente care lucrează pe ramuri de la mai multe intrări.* Exemplul cel mai simplu este componenta care generează linii utilizând input de la două puncte. Diferența

fundamentală față de prima opțiune este că avem nevoie întotdeauna de două intrări și două structuri de date care vor interacționa la același nivel de ramură.

Pentru a explica felul în care este definită o problemă în mediul de programare vizuală Grasshopper, de aici înainte numit GH, voi utiliza un exemplu de problemă structurală simplă.

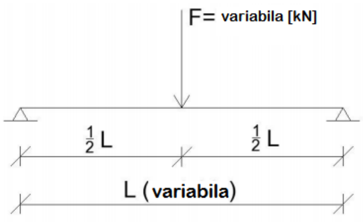
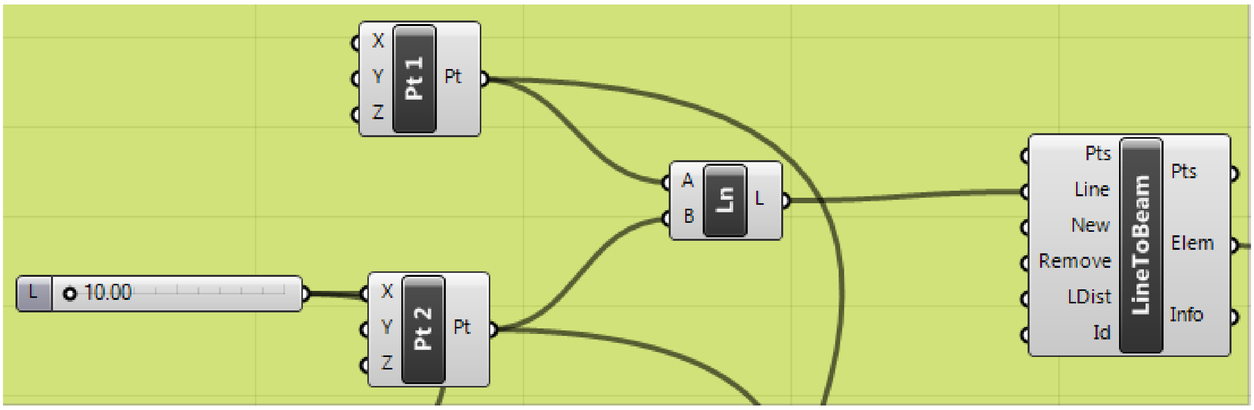


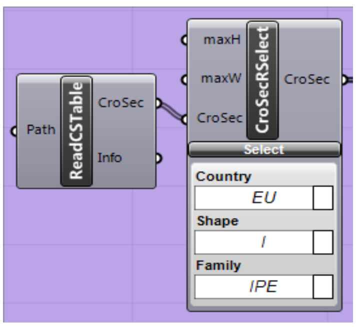
Figura prezintă configurația structurii:

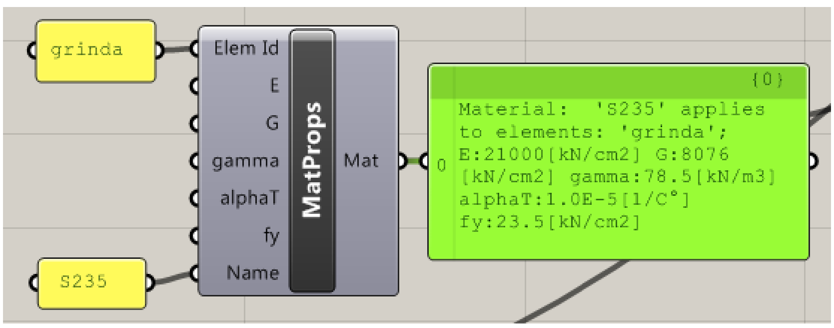
Grinda este articulată la ambele capete, încărcată la mijloc, și având o dimensiune variabilă în intervalul $10 \leq L \leq 20$ [m].

Modelul parametric în GH poate varia prin valorile introduse și astfel se obțin diferite modele structurale. Avem două variabile: lungimea grinzii și încărcarea, astfel modelul este simplu. Utilizând variabile de tip <slider> se introduce dimensiunea în metri și valoarea încărcării în kN. Geometria și topologia structurii sunt acum definite și lungimea grinzii poate fi controlată. Este definit punctul de încărcare la jumătatea barei apoi toate aceste variabile intră în componenta de asamblare a structurii.

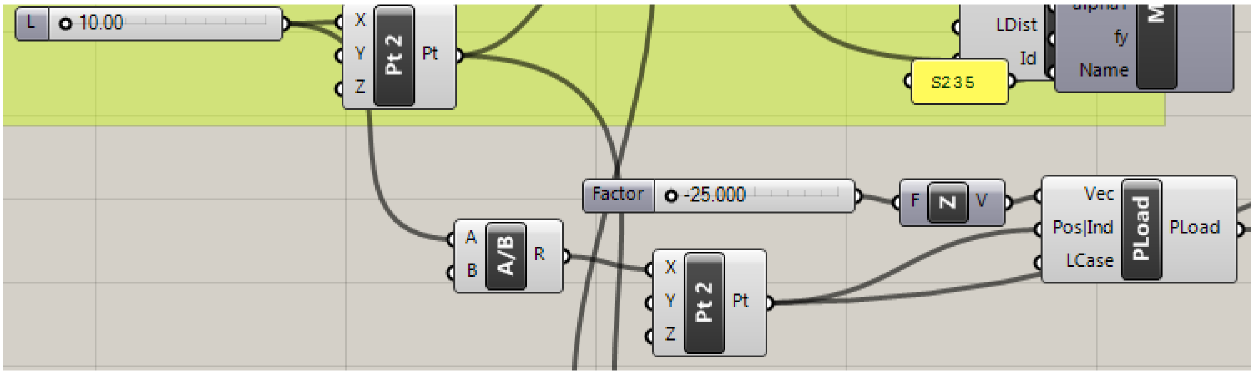


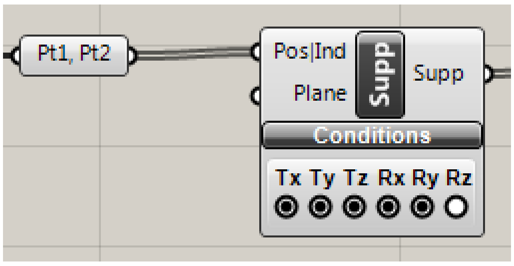
Este definit apoi tipul de material (in cazul de față oțel s235), iar din baza de date cu secțiuni este aleasă o valoare.



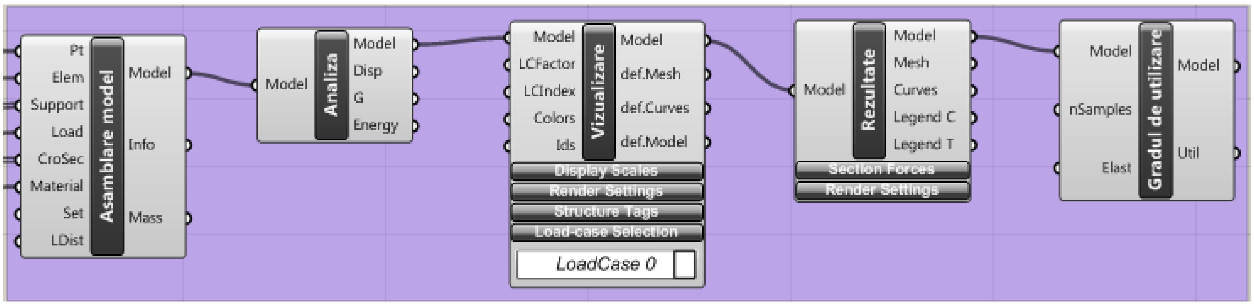


Sunt definite tipul de sprijiniri la capetele barei și valoarea încărcării la mijlocul barei (pentru care am setat limite [25,50] kN).

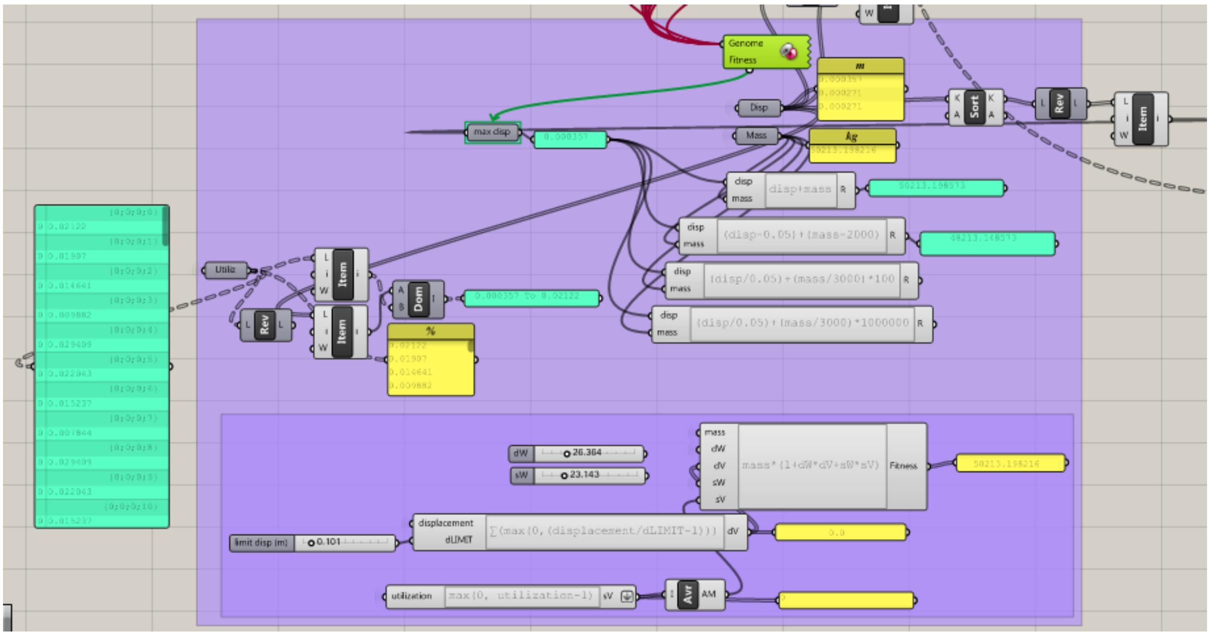




Următorul pas, după definirea completă a structurii, este analizarea acesteia. Este aleasă o analiză statică pentru configurația curentă a modelului parametric.

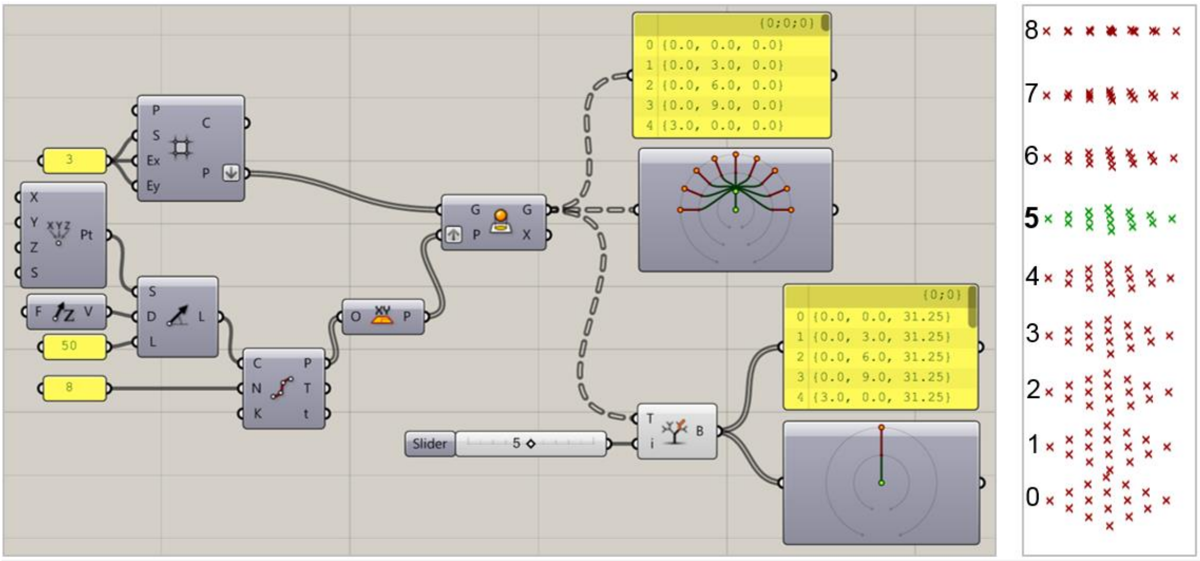


Rezultatele sunt apoi extrase și vizualizate, iar pe baza acestora se poate trece la următoarea etapă – configurarea modulului de optimizare.



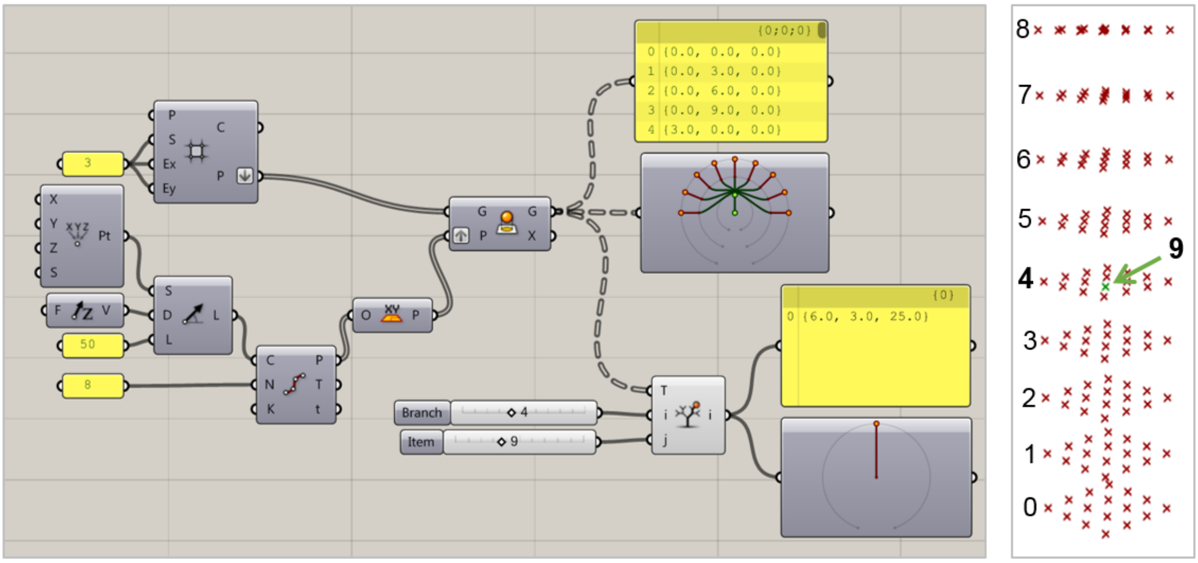
În cazul structurilor mai complexe au fost folosite urmatoarele funcții, pentru identificarea punctelor de interes (sprijin, încărcare):

Tree Branch (TB) – identificarea elementelor incluse într-o ramură anume a arborelui de date



Această componentă va prelua toate elementele dintr-o ramură de index specificat dintr-un arbore de date. Acest lucru este util în stadiile inițiale ale unui proiect pentru debugging și / sau atunci când este necesară vizualizarea de date care aparțin unei ramuri anume și pentru a vedea cum ramurile sunt ordonate în 3D.

Tree Item (Ti) – identificarea unui singur element dintr-o ramură specificată din arborele de date



Această componentă va prelua un anumit element dintr-o anumită ramură de date dintr-un arbore. Se comportă similar cu componenta de ramură în modul în care se accesează ramurile unui arbore de date, dar preia numai elementul solicitat (cu ajutorul unui indice de bază zero) de la arborele solicitat. Din nou, acest lucru este util pentru debugging și vizualizarea datelor 3D.

9.1.2 Crearea funcției fitness

Despre importanța funcției fitness

Adesea, partea cea mai dificilă în utilizarea unui algoritm evoluționist în probleme de optimizare este definirea funcției fitness. Problemele cel mai adesea analizate au un număr relativ mare de variabile ce trebuie determinate. Uneori aceste variabile lucrează împreună și prin îmbunătățirea uneia rezultă îmbunătățirea celorlalte, alteori sunt complet independente unele de altele sau chiar funcționează în mod competitiv. Putem avea adesea funcții care le dorim minimizate respectiv maximizate sau optimizate în jurul unei valori absolute. Pentru formularea aceasta nu există nici o regulă. Trebuie doar gasită o funcție fitness care combină cele trei valori. Forma generală poate lua următoarea formă:

F = -X + Y – Abs(Z – const)

Unde X este valoarea ce se dorește minimizată, Y este valoarea ce se dorește maximizată, iar Z este valoarea ce tinde spre valoarea absolută constantă. La aceasta poate fi necesară adăugarea de factori de penalizare pentru a evita ca una dintre variabile să fie mult mai puternică decât celelalte.

Se dorește o normalizare a componentelor funcției fitness, adică asigurarea valorii fitness cea mai bună și valorii fitness cea mai slabă pentru fiecare componentă să fie aceeași.

Dacă utilizăm exemplul anterior, având funcția compusă din valorile X, Y si Z. X va fi maximizată, Y minimizată și Z se dorește optimizată la valoarea absolută de 15. Mai știm că valoarea lui X poate varia între 10 și 500, Y poate varia între 0.1 și 0.8 iar Z între 5 și 60.

În aceste condiții, cea mai bună valoare fitness va fi pentru *{X=10, Y=0.8, Z=15}* și cea mai slabă va fi *{X=500, Y=0.1, Z=60}*.

Aceste proprietăți ale problemei pot fi exprimate în forma tabelară:

X {min = 10; max = 500; interval = 490; valoare tinta = 10}

Y {min = 0.1; max = 0.8; interval = 0.7; valoare tinta = 0.8}

Z {min = 5; max = 60; interval = 55; valoare tinta = 15}

Intervalul de variație e important deoarece oferă informații legate de influența variabilei în cadrul întregii funcții. În mod tipic se urmărește ca toate variabilele să aibă un grad de influență egală. Asta înseamnă că funcția fitness va avea nevoie de introducerea unor factori de penalizare pentru a aduce toate valorile ce intră în componența ei în intervalul {0.0, 1.0}.

Noua funcție va lua următoarea formă:

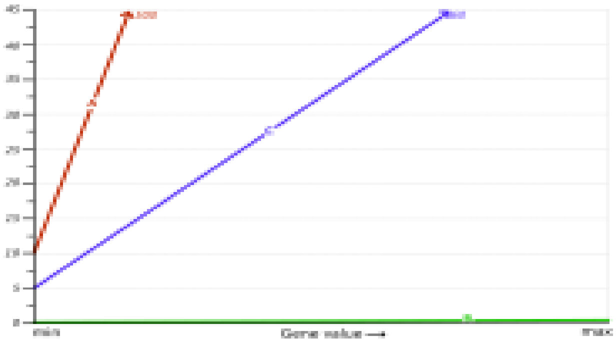
f = -((X-10) / 490) + ((Y-0.1) / 0.7) - Abs((Z-15) / 55)

Regulile acestor transformări pot fi exprimate în modul următor:

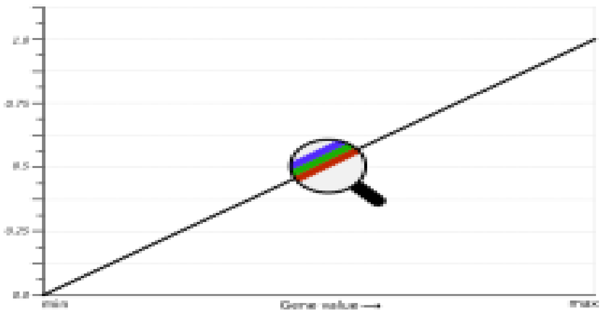
- Semnul din fața fiecarei variabile indică strategia ce va fi aplicată: (+) maximizare, (-) minimizare sau optimizare – asta în cazul în care algoritmul utilizat nu are alte setări de tip min/max.
- Daca dorim optimizarea unei variabile, atunci funcția fitness este definită ca *Abs(x - c),* unde x este variabila iar c este constanta țintă. Altfel spus, optimizarea este echivalentul minimizării diferenței dintre variabilă și valoarea țintă.
- Este necesară o centrare a variabilelor la valoarea zero (sau o altă constantă numerică, însă spre zero este cel mai simplu), prin scăderea valorii minime pe care o pot lua.

▪ Variabilele trebuie normalizate pe domeniul {0,1} sau un alt domeniu constant, prin împărțirea variabilei centrate la intervalul domeniului.

În cazul unor funcții liniare (caz rar întâlnit), valorile înainte de normalizare arată în felul următor:



Prin normalizare, factorii de penalizare ai acestei valori se grupează astfel:



De obicei, valoarea minimă respectiv maximă a unei componente a funcției fitness nu poate fi dedusă din formularea problemei. În acest caz, o metodă de a le găsi este prin încercarea a cât mai multor combinații și înregistrarea modificărilor intervalului.

În cazul ideal există o întelegere aprofundată a problemei ce se dorește rezolvată, și aceste valori pot fi cel puțin aproximate.

Când încercăm să modelăm o structură, în general luăm în considerare urmatoarele variabile:

(V) Volumul,

(M) tipul de Material și implicit greutatea și proprietățile de rezistența și comportare elastică,

(U) Gradul de utilizare al elementului (se dorește evitarea irosirii de material),

(D) Limitarea deplasărilor și deformațiilor structurii, (C) Minimizarea costurilor - care pot depinde, pe lângă consumul de material, de tipul conexiunilor, complexitatea structurii (topologia) care poate duce la o durată mai lungă a procesului de construcție și a necesității utilizării unor unelte și utilaje mai costisitoare.

Am numit aceste cinci proprietăți V, M, U, D, și respectiv C. V va fi codat ca o singură valoare scalară. Unele dintre aceste proprietăți sunt, cel puțin parțial, interdependente. O creștere în volum va rezulta într-o creștere a greutății (G), și respectiv a consumului de material și costului total. Însă e probabil ca acest fapt să determine o reducere în deplasări și deformații. Practic ne confruntăm cu cinci forțe care trag de soluție în direcții diferite.

Cea mai directă formulare a funcției fitness (pe care dorim să o minimizăm) poate arăta în felul următor:

$F = G(V,M) - U + D + C$,

Unde semnul din fața variabilelor determină dacă variabila respectivă se dorește a fi minimizată sau maximizată. Deși funcția poate arăta simplu, progresia algoritmului înspre o soluție poate fi dificilă. Ceea ce nu rezultă din formularea de mai sus este relația dintre variabilele incluse în funcția fitness și acele variabile de intrare (genele) cu care îi permitem algoritmului să lucreze. Aceste relații vor fi în cele mai multe cazuri complicate și interdependente. În cazul prezentei formulări generale se pot trage următoarele concluzii, bazate pe unele cunoștințe preliminare despre formularea problemei:

- Toate aceste variabile sunt tratate în mod egal, ceea ce este foarte probabil în dezavantajul nostru, deoarece toate aceste variabile au unități de măsură diferite. Volumul poate fi exprimat în m^3, greutatea în tone sau kN, deplasările în cm, gradul de utilizare sub formă de procent iar costul fără unitate de măsură.
- Funcția este lineară. Asta însemnă că un volum mic poate fi răsplătit cu un fitness foarte bun iar deplasările pot fi inacceptabile.

Pentru a rezolva aceste neajunsuri pot fi introduși factori în funcția fitness care să aducă variabilele pe un teren comun. Dacă luăm de exemplu volumul și gradul de utilizare, V va lua valori între 0.01 m^3 și volumul pentru structura luată în calcul, dat de secțiunea cea mai mare – zeci de mii de m^3, iar gradul de utilizare variază între 0 și 1 (pentru ipoteza utilizării la limită).

Această diferență mare poate duce la ignorarea completă a criteriului de utilizare a elementului. Pentru a preveni acest lucru se poate introduce un factor de multiplicare.

Descrierea algoritmului genetic

Selecția. Selecția naturală determină direcția fondului genetic prin alegerea indivizilor cărora li se permite să se reproducă. În cazul rezolvării problemelor cu algoritmi evoluționiști se utilizează o anumită formă de selecție artificială. Nu sunt utilizate noțiuni de gen (feminin sau masculin) în forma utilizată de calculator. Există doar un set mic de operatori de selecție, însă aceștia par să fie eficienți. Selecția isotropică, sau uniformă este cea mai simplă formă a operatorului, și este de fapt o lipsă a selecției, tuturor indivizilor permițându-li-se să se reproducă.

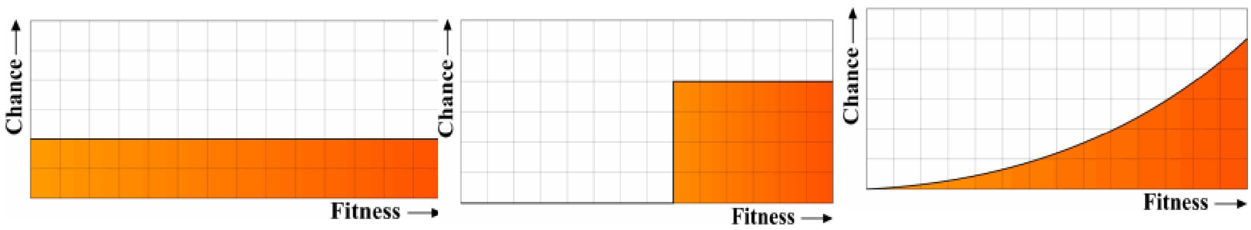
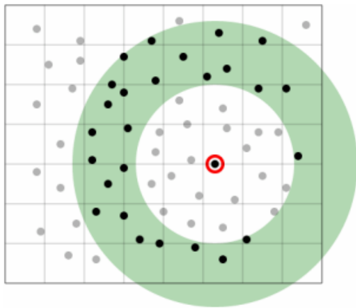


Figura 9.2 Selecție de tip uniform, exclusivistă și pǎrtinitoare.

Deși pare o strategie de selecție ineficientă, deoarece nu face nimic spre a spori evoluția fondului genetic, ea are precedent în natură (polenizarea prin vânt, înmulțirea coralilor). *Selecția uniformă* are rolul de a reduce viteza de convergență a populației, și astfel protejează împotriva unei colonizări premature a unui punct de optim local posibil inferior. *Selecția exclusivistă*, unde doar cei mai buni N% indivizi sunt copiați în fondul de reproducere. Acei indivizi selecționați au probabilitate mare sa aibă urmași multipli. Un alt tipar de selecție întâlnit frecvent în natură este *Selecția pǎrtinitoare*, unde șansa de a se reproduce a unui individ e proporțională cu valoarea sa fitness comparată cu a celorlalți. Aceasta poate fi amplificată prin utilizarea unor funcții de putere pentru a aplatiza sau exagera curba.

Cuplarea. S-ar părea că cea mai bună opțiune ar fi un echilibru între reproducerea în mediul genetic familial și cea cu indivizi foarte diferiți din punct de vedere genetic prin selectarea indivizilor nici prea îndepărtați nici prea aproape unii de alții. În Galapagos acest factor variază intre (-100% și +100%, în totalitate cei mai apropiați/îndepărtați).

Genomul este reprezentat pe o hartă. Aceasta conține toți indivizii dintr-o anumită populație ca puncte pe un grid. Distanta dintre două genoame pe grid este aproximativ analoagă cu distanța dintre genoame în spațiul de gene, în măsura în care se poate aproxima distanța pe o hartă. Un singur genom este definit de un număr de gene, iar în cazul de față toate genoamele speciei au același număr de gene. Astfel distanța dintre două genoame este o valoare n-dimensională, unde n este numărul de gene. Este imposibilă reprezentarea precisă a norului de puncte n-dimensionale pe o suprafață bidimensionala, astfel încât harta genomului este doar o aproximare grosieră, care reprezintă aproximativ cât de similare/apropiate sunt două puncte.



Împerechere. Genele în Galapagos sunt stocate ca și variabile floating point, care pot lua valori între doua extreme definite.

Pentru generarea urmașilor, la nivel genetic, procesul biologic este extrem de complicat și el însuși supus evoluției (de ex. genele evoluează să afecteze procesul de meioză și iși îmbunătățesc astfel șansele să fie transmise urmașilor). Varianta digitală este mult mai simplă, parțial datorită faptului că genele algoritmilor evoluționiști nu sunt foarte similare genelor biologice. În mod ironic, genele biologice sunt mult mai digitale decât cele programatice. Așa cum Mendel a descoperit în 1860, genele nu sunt cantități variabile continue. Acestea se comportă ca un întrerupator electric (mazărea obținută de Mendel prin încrucișarea populațiilor diferite a rezultat în procente diferite de indivizi cu trăsături moștenite de la părinți, nu cu mazăre parțial netedă și parțial striată). Algoritmii evoluționiști, însă fac exact acest lucru prin interpolarea genelor. Galapagos are mai multe mecanisme de transmitere parțială a genelor.

Dacă sunt considerate două genoame a câte patru gene fiecare, într-un proces simetric (fără determinarea genului individului) există următoarele posibilități de încrucișare: Crossover – unde progenitura moștenește un număr de gene de la mamă și restul de la tată. Această metodă funcționează bine când părinții sunt similari. Dacă se consideră că genele din fiecare dintre genoame trebuie sa coopereze pentru a da o valoare fitness bună, nu are logică combinarea aleatoare a acestora. Interpolarea calculează valori noi pentru gene, făcând o medie între valorile părinților în locul copierii lor. O altă variantă a acestui operator este interpolarea preferențială, unde valorile genelor părintelui cu valoarea fitness mai mare sunt mai dominante.

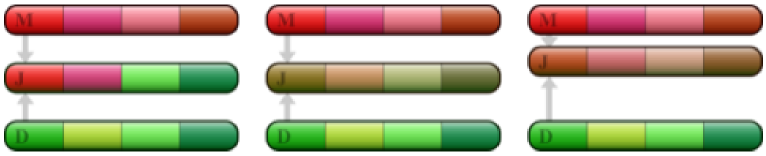


Figura 9.3 Crossover, interpolare genetică și interpolare preferențială

Mutația. O metodă des utilizată de reprezentare a punctelor multi-dimensionale pe un mediu bidimensional este desenarea lor ca o serie de linii care conectează diferite valori pe un set de abscise verticale. Fiecare abscisă reprezintă o singură dimensiune. În acest fel pot fi reprezentate ușor nu doar puncte cu orice număr de dimensiuni, ci și puncte cu număr diferit de dimensiuni pe același grafic. În exemplu avem un genom format din cinci gene (un punct în spațiul cu 5 dimensiuni) care definește această specie anume. Poziționarea lui G0 la o treime înseamnă că valoarea este la o treime între limita inferioară și cea superioară pentru gena respectivă. Printre beneficiile acestui tip de grafic se numără ușurința identificării unor sub-specii într-o populație, și a unor indivizi izolați. Când este aplicată mutația asupra genomului, se observă o modificare în grafic, deoarece fiecare genom are un grafic unic.

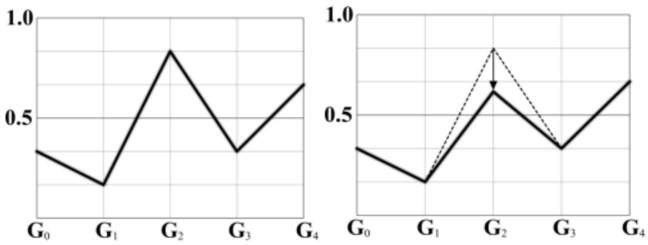


Figura 9.4 Reprezentare grafică a modificării genomului prim mutația unei singure gene.

10 IMPLEMENTAREA STRATEGIILOR DE OPTIMIZARE STRUCTURALĂ CU ALGORITMI GENETICI

10.1 PROGRAMUL ELABORAT ÎN PLATFORMA MATLAB

A fost dezvoltat pentru a rezolva problemele de optimizare folosind algoritmii genetici simpli. Pentru soluționarea restricțiilor s-au folosit programe de element finit, elaborate în MATLAB. Obiectivul optimizării este reducerea greutății structurilor metalice plane și spațiale, alcătuite din bare articulate și supuse la restricții de deplasări și tensiuni. Optimizarea structurilor spațiale alcătuite din bare articulate spuse la restricții de deplasări și tensiuni, variind coordonatele nodurilor și secțiunile barelor.

Pentru analiza rezultatului obținut după rezolvarea problemei, sunt generate două grafice care prezintă soluția optimă, valoarea soluției optime, timpul de rulare, numărul de generații și viteza de convergență. Primul grafic reprezintă valoarea diversității pentru fiecare generație, utilizând ca marker distanța medie între indivizi. În al doilea grafic sunt reprezentate mediile valorilor fitness și soluția optimă pentru fiecare generație.

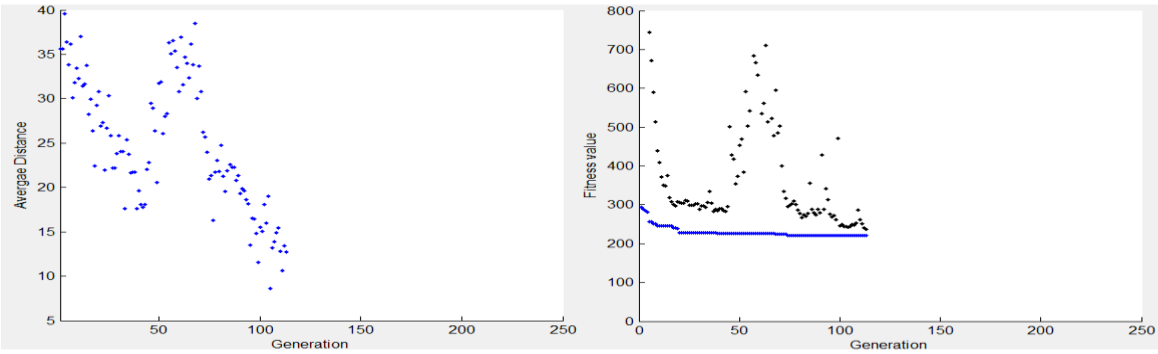


Figura 10.1 *Evoluția valorilor funcției fitness și a diversității populației în timpul unei rulări tipice.*

10.1.1 Probleme testate

- **Cadrul plan și structura articulată plană**

Problema a fost formulată după cum urmează, pentru o structură definită de M noduri și N bare cu ariile secțiunilor: Ai, i=1,2,…,N. Toate ariile Ai compun vectorul x al parametrilor de optimizare:

$$\mathbf{x} = [A_1 \quad A_2 \quad \dots \quad A_N]^T \qquad (10\text{-}1)$$

Problema consta în calcularea vectorului x pentru care greutatea minimă W a cadrului sau grinzii cu zăbrele este obținută:

$$W = \sum_{i=1}^{N} \rho A_i L_i \qquad (10\text{-}2)$$

Pentru restricțiile de tensiuni și deplasări date:

$$\begin{aligned} \sigma_i &\le \sigma_{0i} \qquad 1 \le i \le N \\ u_i &\le u_{0i} \qquad 1 \le i \le M \end{aligned} \qquad (10\text{-}3)$$

În ecuația (10.3), σ_{0i} și u_{0i} sunt limitele superioare pentru valorile admisibile ale tensiunilor și deplasărilor, respectiv.

Restricțiile legate de rezistență și stabilitate sunt luate din “EN 1993: Design of steel structures” și implementate în algoritm. Rezistența de proiectare pentru secțiuni este calculată:

$$N_{c,Rd} = \frac{A f_y}{\gamma_{M0}} \qquad (10\text{-}4)$$

Si încărcările aplicate structurii:

$$E_d = E \cdot (\gamma_{G,j} \cdot G_{k.j}; \gamma_{Q,1} \cdot Q_{k.1}; \gamma_{Q,i} \cdot \psi_{0,i} \cdot Q_{k.i}) \qquad (10\text{-}5)$$

Unde:

E_d – design load – încărcare de calcul

γ – Partial coefficient – coeficient parțial

G – Permanent action – acțiuni permanente

Q – Variable action - acțiuni variabile

ψ – Coefficient variable action – coeficient al acțiunilor variabile

Algoritmul de optimizare

Deși există o mulțime de implementări diferite ale algorimilor evoluționiști (EA), conceptul de bază al așa-numitului “EA canonic” reprezintă modelul tuturor acestora. Procesul de selecție utilizat favorizează indivizii cu cele mai mari valori ale funcției fitness față de candidații cu valori sub medie. Indivizii noi sunt creați prin copiere și aplicarea operatorilor genetici de variație – mutație și recombinare. Acești pași sunt repetați până la îndeplinirea condiției de terminare a algoritmului – numărul maxim de generații este atins sau variația soluției candidat celei mai bune ramâne neschimbată pentru un anumit număr de iterații.

Population size	*300*
Crossover parameter	0.7
Mutation parameter	0.1
Maximum number of iterations	100

Tabel 10-1: Valorile parametrilor algoritmului genetic utilizat.

Problema analizată are un singur obiectiv însă este supusă unui număr arbitrar de restricții. Numărul parametrilor de optimizare determină dimensiunea genotipului și a spațiului de căutare. O funcție fitness care atribuie rigidității, rezistenței și greutății evaluate din modelul FE o valoare fitness unică este definită și testată pe un cadru cu 8 elemente, o structură articulată plană cu 9 bare și una spațială cu 120 elemente. Din 20 de rulări au fost alese cele mai bune 5 și valorile sunt prezentate în tabel.

Prima problemă: cadru 8 bare 8 noduri

Problema are 8 variabile reprezentate de secțiunile elementelor.

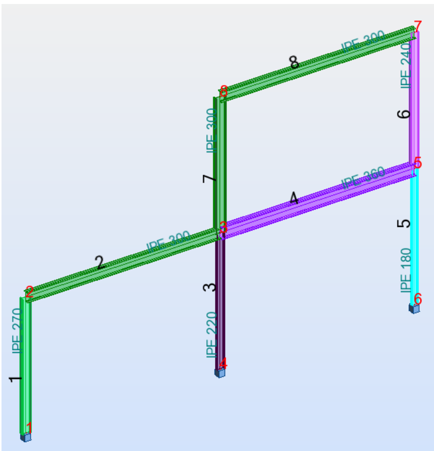


Figura 10.2 Structură cu 8 bare.

Încărcările pe structură:

Bar	*Uniformly distributed load*							
	Permanent loads		Live Loads		S now Loads		Wind Loads	
	X	Z	X	Z	X	Z	X	Z
1	0	0	0	0	0	0	-2.5	0

2	0	-10	0	-15	0	-6	0	0
3	0	0	0	0	0	0	0	0
4	0	-10	0	-15	0	0	0	0
5	0	0	0	0	0	0	-5	0
6	0	0	0	0	0	0	-5	0
7	0	0	0	0	0	0	-2.5	0
8	0	-10	0	-15	0	-6	0	0

Tabel 10-2 Încărcări problema 1.

Bar	*Profiles*				
	Run 1	Run 2	Run 3	Run 4	Run 5
1	IPE 120	IPE 140	IPE 120	**IPE 140**	IPE 120
2	IPE 180	IPE 200	IPE 180	**IPE 200**	IPE 160
3	IPE 330	IPE 270	IPE 330	**IPE 300**	IPE 360
4	IPE 140	IPE 180	IPE 180	**IPE 140**	IPE 140
5	IPE 330	IPE 330	IPE 330	**IPE 330**	IPE 330
6	IPE 140	IPE 140	IPE 140	**IPE 120**	IPE 120
7	IPE 200	IPE 200	IPE 200	**IPE 180**	IPE 160
8	IPE 330	IPE 330	IPE 330	**IPE 330**	IPE 390
Total weight	1193.9	1200.4	1217.5	**1152.5**	1240.7

Tabel 10-3 Profile alese de algoritm

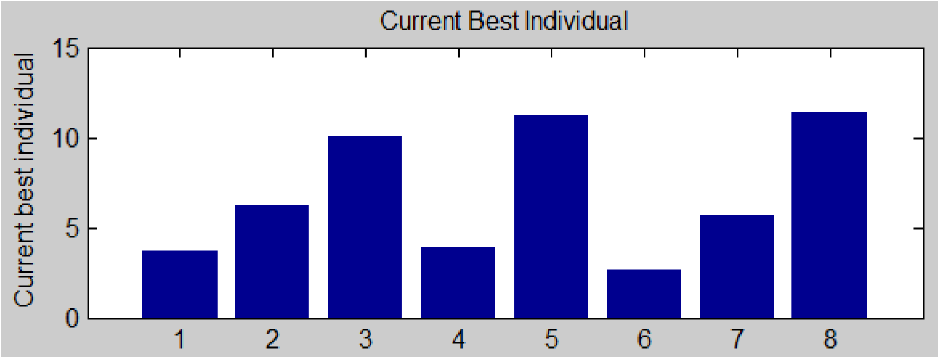


Figura 10.3 Cel mai bun individ.

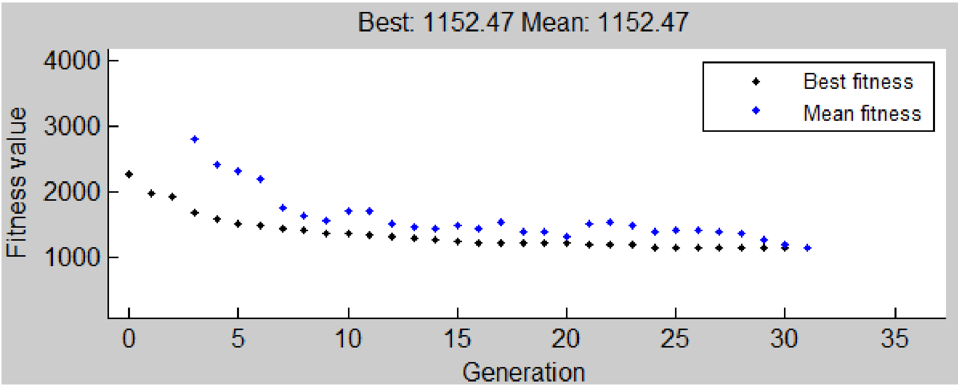


Figura 10.4 Convergența algoritmului.

Structura a fost verificată cu ajutorul programului Autodesk Robot Structural Analysis. O greutate totală de 1466 kg a fost obținută, în comparație cu soluția de 1152 kg obținută utilizând programul creat în MATLAB.

A doua problemă: grinda cu zăbrele 9 bare 6 noduri

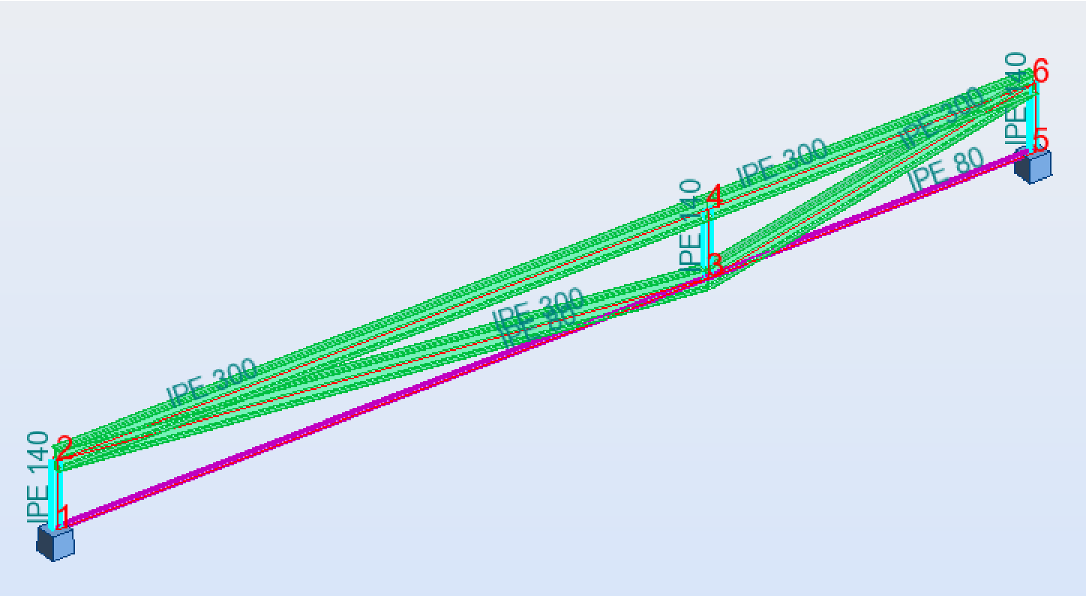


Figura 10.5 Structura nr. 2 – 9 bare.

Problema are 9 variabile reprezentate de secțiunile elementelor.

Nodes			
	Permanent Loads	Live loads	Snow loads
	Z	Z	Z
1	0	0	0
2	-50	-75	-30
3	0	0	0
4	-75	-112.50	-45
5	0	0	0
6	-25	-37.5	-15

Tabel 10-4 Încărcări pe structură.

Bar	*Profiles*				
	Run 1	Run 2	Run 3	Run 4	Run 5
1	**IPE 80**	IPE 80	IPE 80	IPE 80	IPE 80
2	**IPE 80**	IPE 80	IPE 80	IPE 80	IPE 80
3	**IPE 300**	IPE 300	IPE 300	IPE 300	IPE 300

4	**IPE 300**	IPE 330	IPE 300	IPE 300	IPE 330
5	**IPE 180**	IPE 180	IPE 140	IPE 140	IPE 140
6	**IPE 300**	IPE 300	IPE 300	IPE 300	IPE 300
7	**IPE 180**	IPE 140	IPE 140	IPE 200	IPE 140
8	**IPE 300**	IPE 300	IPE 300	IPE 300	IPE 300
9	**IPE 180**	IPE 220	IPE 140	IPE 160	IPE 140
Total weight	**1402.25**	1456.0	1402.25	1414.6	1436.8

Tabel 10-5 Profile alese de algoritm.

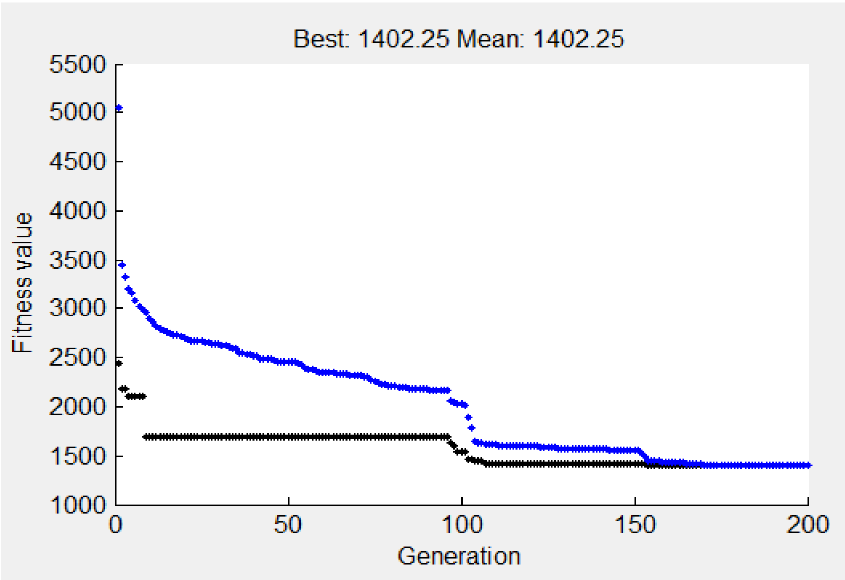


Figura 10.6 Convergența algoritmului.

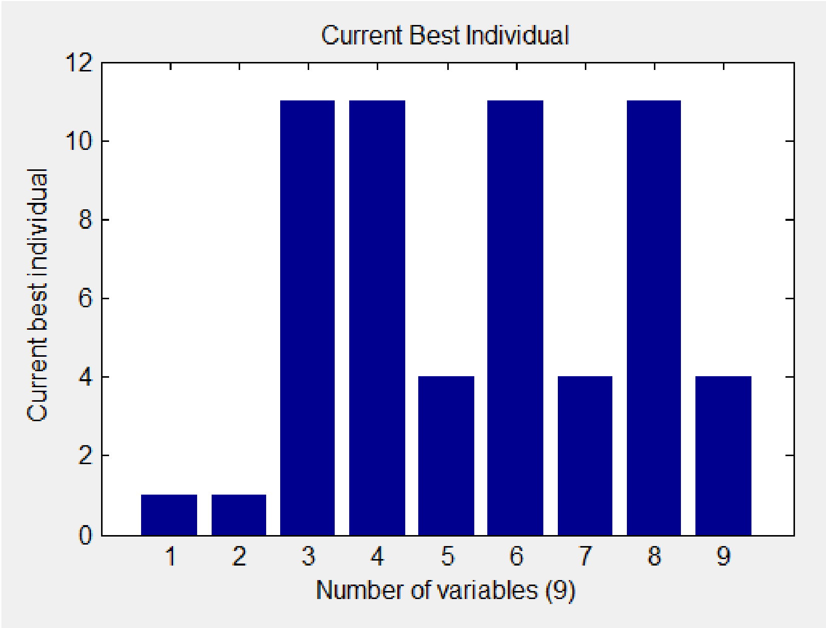


Figura 10.7 Cel mai bun individ.

Utilizând Autodesk Robot Structural Analysis s-a obținut o structură cu o greutate totală de 1403 kg apropiată de valoarea de 1402 kg a soluției obținute utilizând algoritmul implementat în MATLAB.

Problema a 3-a: structură articulată spațială 120 bare 49 noduri

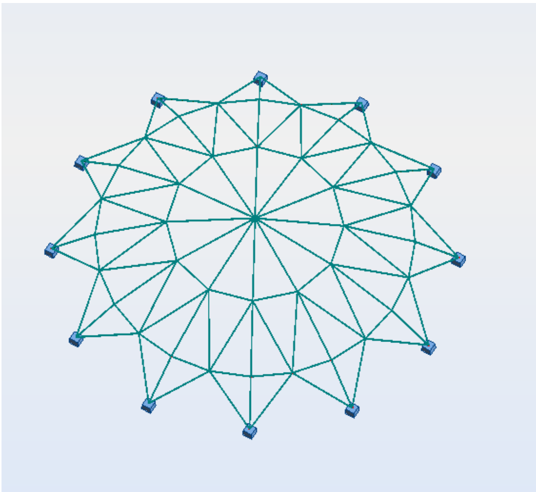


Figura 10.8 Structura nr. 3 – 120 bare.

Problema are 120 de variabile reprezentate de secțiunile elementelor.

Bar	*Profiles*				
	Run 1	Run 2	Run 3	Run 4	Run 5
Total weight	14008.1	15080.4	14941.6	14466.9	14953.9

Tabel 10-6 Greutatea soluțiilor obținute.

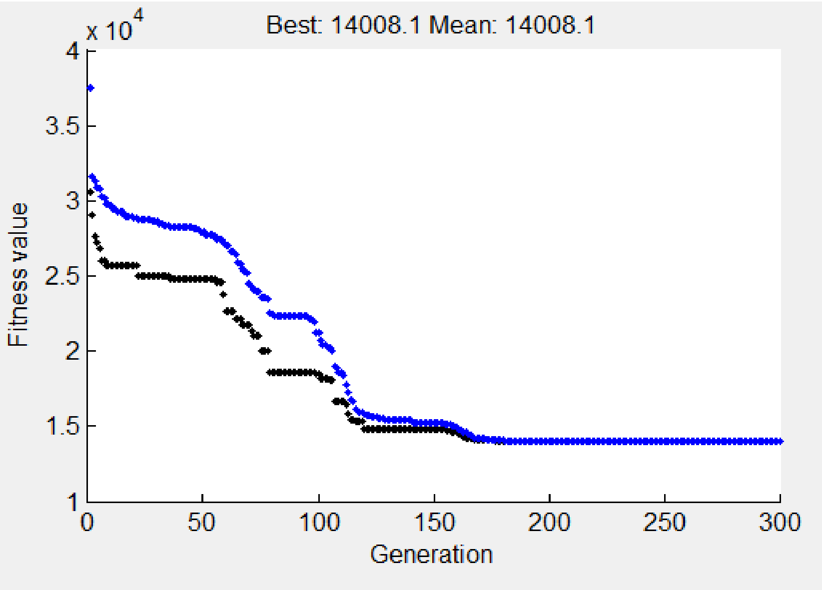


Figura 10.9 Convergența algoritmului.

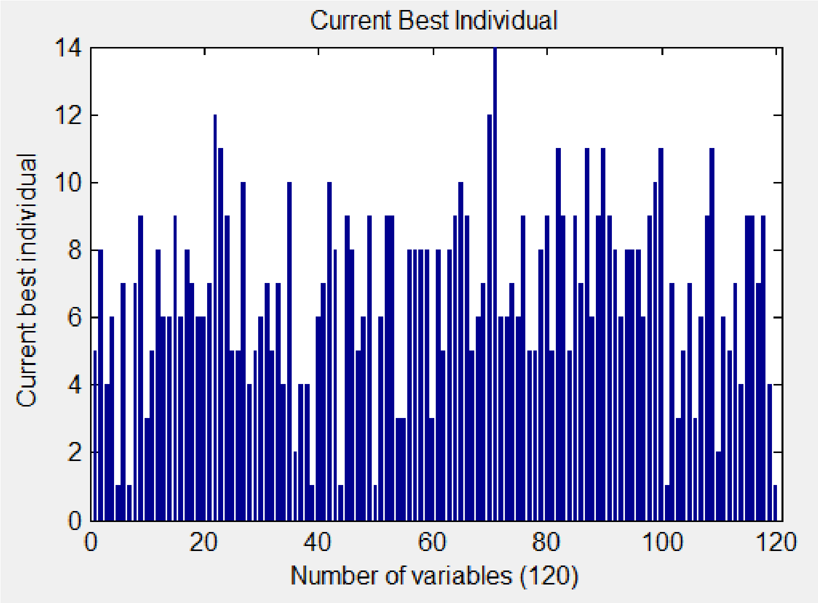


Figura 10.10 Cel mai bun individ.

MATLAB dispune și de “Genetic Algorithm and Direct Search Toolbox” care permite utilizarea algoritmilor genetici într-un domeniu amplu de probleme. Acest toolbox include multiple opțiuni în ceea ce privește parametrii programului, ca de exemplu operatori variați de selecție, crossover și mutație, o interfață grafică interactivă, reprezentarea grafică a curbei de convergență și a indivizilor. De asemenea, datorită faptului că algoritmii sunt scriși în limbajul MATLAB, utilizatorul poate inspecta și modifica aceste fișiere și poate crea funcții personalizate.

Pentru a putea aplica acest GA toolbox unei probleme de optimizare, funcția MATLAB trebuie sa fie implementată cu o reprezentare specifică problemei, să aibă un sistem propriu de codare genotip/fenotip, o metodă de evaluare a funcției fitness și a funcției de penalizare.

10.2 PROGRAME CU FORMULARE PARAMETRICĂ

În formularea problemelor uni- și multi-obiectiv de optimizare a topologiei, datorită minimelor locale care apar, metodele iterative de căutare locală nu sunt foarte eficiente. Pe de altă parte algoritmi de optimizare globală pot deveni prea scumpi, datorită numărului mare de variabile de proiectare. Pentru e rezolva acest neajuns sunt propuse în literatura de specialitate un număr de metode hibrid. Un exemplu tipic este metoda hibrid propusă în (Kaminakis, 2012), care este bazată pe algoritmi de optimizare globală, cum este Particle Swarm Optimization (PSO) și Differential Evolution (DE), și folosind o metodă iterativă de căutare locală ca instrument de evaluare. În lucrarea de față, metoda iterativă de căutare locală se bazează pe discretizarea spațiului de căutare cu secțiuni de oțel din industrie.

Funcția fitness descrisă mai jos a fost implementată în Grasshopper, un plug-in pentru Rhino.

Structura a fost analizată cu ajutorul unor componente din Karamba, o bibliotecă FEM. Optimizarea s-a bazat pe utilizarea de algoritmi evoluționiști din Galapagos Evolutionary Solver și Force Flow Finder, ambele fiind biblioteci de algoritmi pentru Gragghopper.

Karamba este încorporat în mediul parametric al Grasshopper. Acest fapt face ușoară combinarea modelelor parametrice, calculului cu element finit și a algoritmilor de optimizare, comunicarea între aceste componente și biblioteci fiind făcută prin programare vizuală.

Valoarea fitness a structurii a fost calculată pe baza greutății. Optimizarea tensiunilor a fost realizată prin adoptarea de tensiuni admisibile maxime pentru material, iar deplasările au fost limitate prin penalizare. Funcția fitness adaptată din [Hayalioglu,2001], este definită ca:

$$FITNESS = MASS * (1 + dW * dV + sW * Sv) \qquad (\ 10\text{-}6)$$

Unde (dW) și (sW) sunt constante de penalizare pentru deplasari și tensiuni și sunt înmulțite cu valorile încălcărilor restricțiilor de tensiuni și deplasări (dV) și (sV).
Valorile acestora sunt calculate conform (6.2) și (6.3).

$$dV = \sum_{l=1}^{NL} \sum_{n=1}^{NN} D_{l,n} \qquad (\ 10\text{-}7)$$

$$sV = \sum_{l=1}^{NL} \sum_{e=1}^{NE} S_{l,e} \qquad (\ 10\text{-}8)$$

Magnitudinile încălcării restricțiilor de deplasare (Dl,n) respectiv tensiune (sl,n) sunt însumate pe toate cazurile de încărcare (NL), noduri (NN), și elemente (NE) unde este aplicabil. Pentru a asigura că doar încălcările limită contribuie la presiunea mediului de selecție, au fost utilizate formulele (10-9) și (10-10).

$$D_{l,n} = max\left\{0, \left(\frac{displacement_{l,n}}{dLIM_n} - 1\right)\right\} \qquad (\ 10\text{-}9)$$

$$S_{l,e} = max\left\{0, \left(\frac{stress_{l,e}}{sLIM_e} - 1\right)\right\} \qquad (\ 10\text{-}10)$$

Valoarea limită a deplasării pentru fiecare nod este (dLIM=5 cm) iar limita tensiunilor (sLIM) pentru fiecare element este definită ca rezistența ultimă a materialului (S235). Setul de funcții fitness necesită patru valori obținute din analiza structurală: greutate, tensiuni, deplasări și nivelul de utilizare a elementelor.
O valoare de penalizare este folosită pentru determinarea importanței condițiilor limită. Am folosit valoarea 15.5 și au rezultat structuri care nu depășesc restricțiile impuse.

Dacă valoarea acestor coeficienți de penalizare este prea mică structurile rezultate pot frecvent depăși limitele domeniului fezabil. Însă, dacă acestea sunt prea mari, structura converge spre o formă suboptimă datorită presiunii selective crescute. Valoarea fitness ajustată (scalată) este utilizată pentru a ierarhiza indivizii din populație, care este apoi folosită la selecția exclusivistă pe bază de rang. Aceste seturi de ecuații rezultă într-o valoare pozitivă ce depinde de magnitudinea încălcărilor. Nu se vor răsplăti valori ale modelelor care nu încalcă restricțiile deoarece deplasările și tensiunile sunt invers proporționale cu greutatea, astfel încât răsplătirea acestor valori va determina generarea de structuri grele și rigide datorită valorilor fitness mari.
În general, valoarea fitness a unei structuri este egală cu greutatea structurii (în cazul ideal), sau mai mare decât greutatea acesteia.

10.2.1 Probleme testate

GA codat în Galapagos pare să nu dea rezultate optime chiar și atunci când problema are o formă simplă. Se observă o convergență prea rapidă sau lipsa unei convergențe (găsește o soluție admisibilă dar mai apoi nu gasește optimul). Pentru a găsi soluția optimă algoritmul a fost rulat de mai multe ori, pentru aproximativ 300 de generații. Populația utilizată a fost de 50 de indivizi, cu o populație inițială dublă, și o valoare mică pentru recombinare pentru a preveni convergența prematură. Rezultatele obținute au fost îmbunătățite prin utilizarea SA (călirii simulate) pentru găsirea unei soluții suficient de bune, care mai apoi este preluată ca punct de plecare pentru GA. Folosirea algoritmilor evoluționiști în tandem a dat rezultate bune și în cazul altor tipuri de probleme de optimizare dificile, întâlnite în literatură. Amintesc aici folosirea a DE (Differential evolution), PSO si GA pentru optimizarea de controllere fuzzy pentru structuri inteligente (smart structures) cu rezultate superioare față de metodele fuzzy clasice, sau metode fuzzy utilizând doar unul dintre algoritmii menționați (M. Marinaki, 2012).

10.2.1.1 Problema 1: Structură articulată cu 10 bare

Este utilizată des în literatură pentru testarea algoritmilor - valorile variabilelor sunt luate din cataloage de profile, incluzând 1407 tipuri de profile (IPE, HEA, HEB, FRQ, RR). Structura este încărcată în nodurile 0 și 2 cu o forță de 444.5 kN. Modulul de elsticitate utilizat E=21000 kN/cm2, fy=23.5 kN/cm2, densitate gamma=78.5kN/m3, corespund materialului S235.

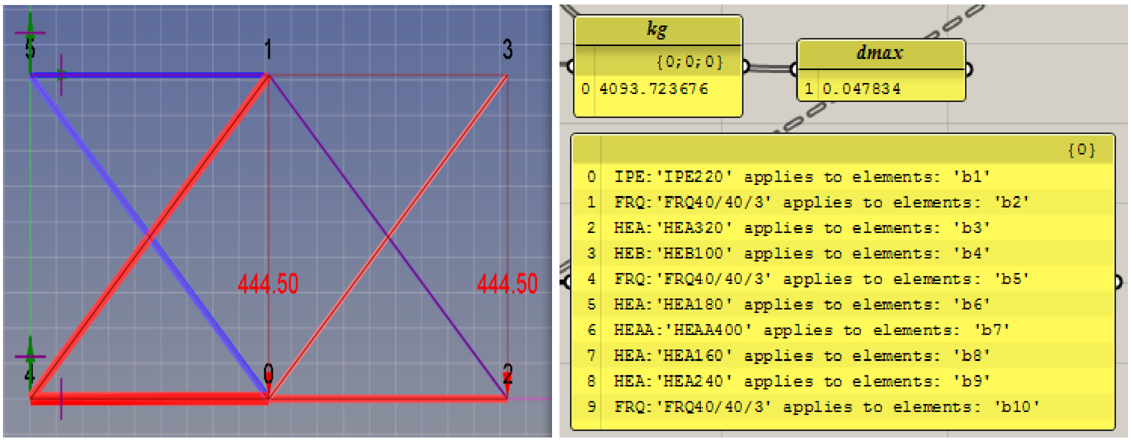


Figura 10.11 Structura cu 10 bare benchmark și rezultatele obținute.

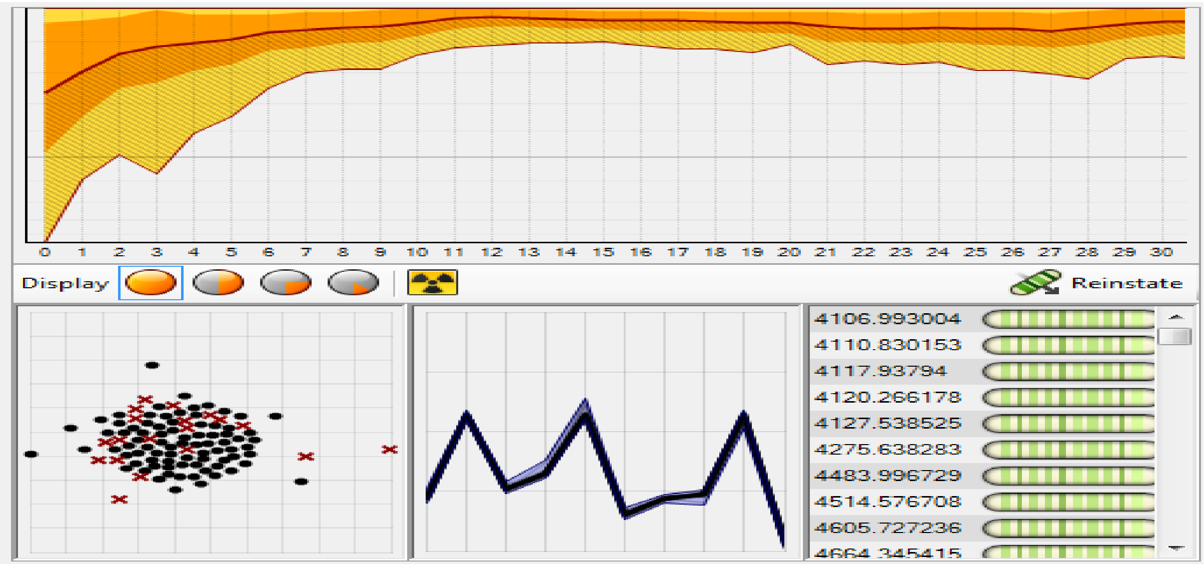


Figura 10.13 Rezultatele obținute și convergența ambilor algoritmi.

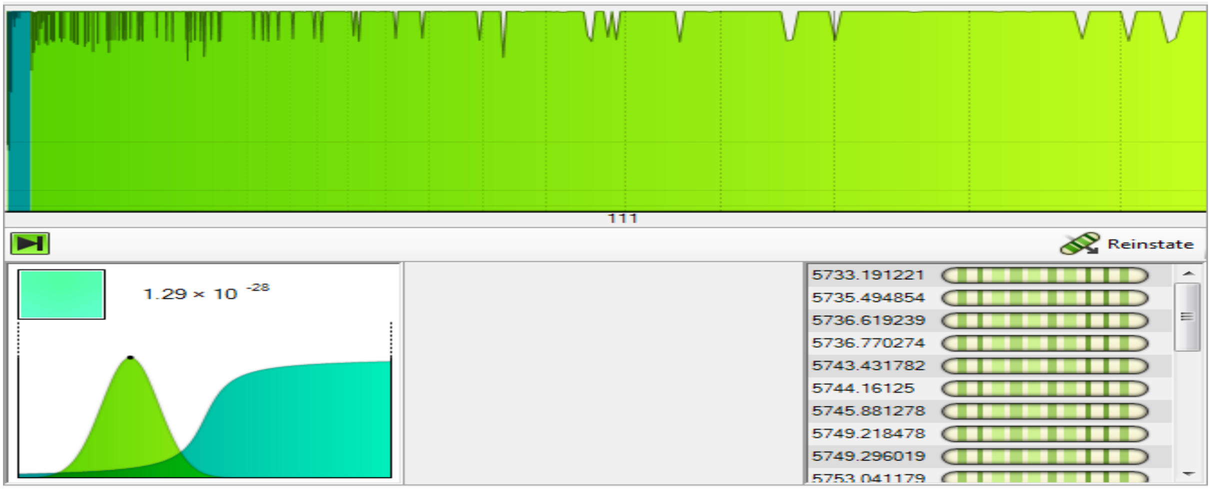
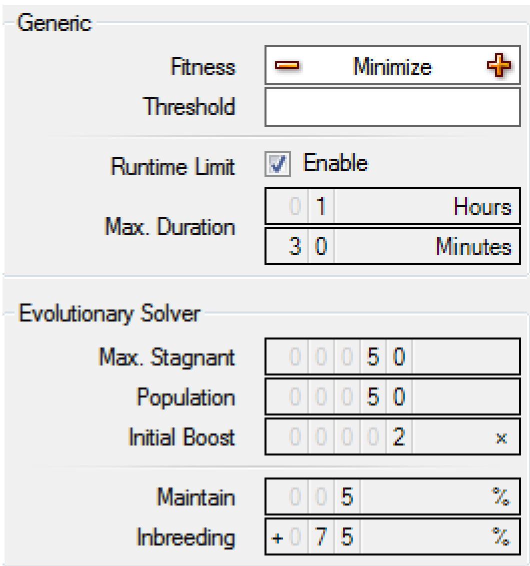


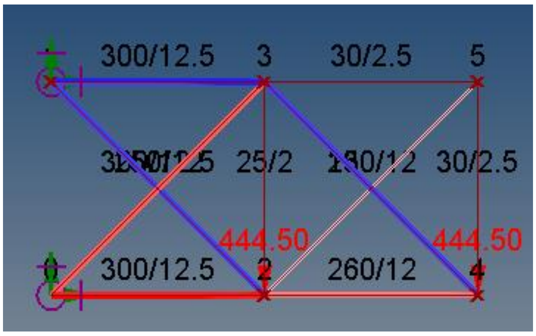
Figura 10.12 Figura arată rezultatele obținute cu algoritmi genetici și călire simulată și indivizii cei mai buni din ultima rulare.

Am folosit următoarele setări pentru algoritm:



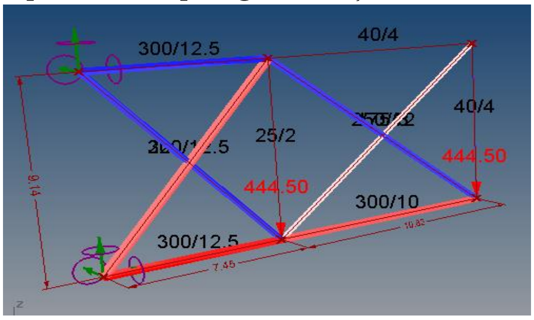
Pentru a putea compara rezultatele cu cele din literatură am folosit caracteristicile materialului pentru a corespunde cu cele ale aluminiului și am obținut următoarele rezultate:

- Optimizare secțiuni



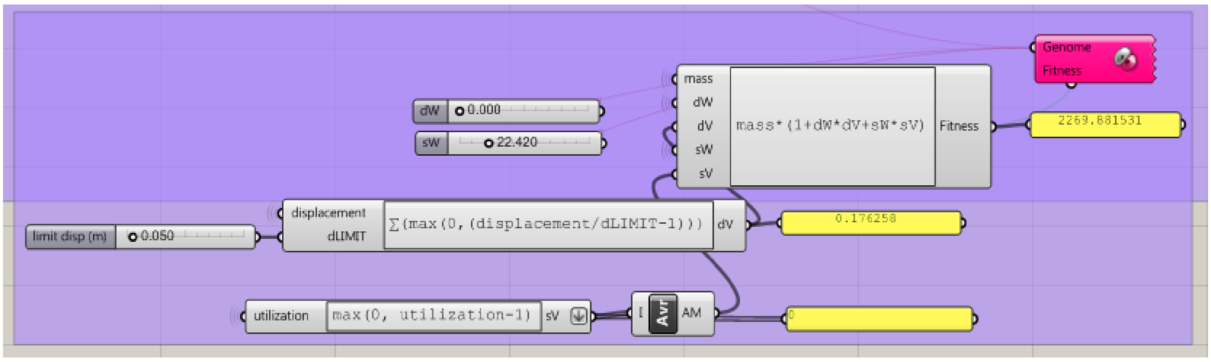
Nr de secțiuni:	161	Material:	aluminiu
Tip sectiune:	FRQ - Germany		
Greutate	[]	name	A
kg	Gradul de utilizare	-	cm2
2381.370974	0.513997	250/12	108
	0.259248	25/2	1.74
Deplasare maximă	0.832915	300/12.5	137
m	0.472309	250/12	108
0.058265	0.470402	30/2.5	2.59
	0.470574	130/12	50.5
Tensiune maximă	0.673392	300/12.5	137
[kN/cm2]	0.603933	300/12.5	137
-7.53E+00	0.467466	260/12	113
	0.472036	30/2.5	2.59

- Optimizare topologie si secțiuni

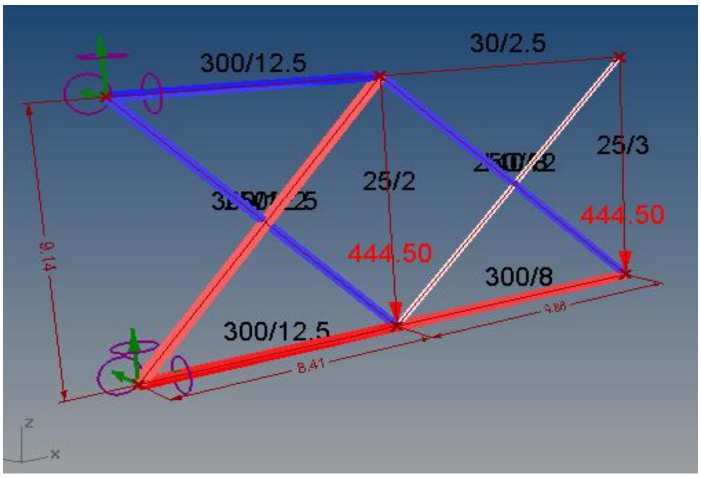


Dupa 10rulări:

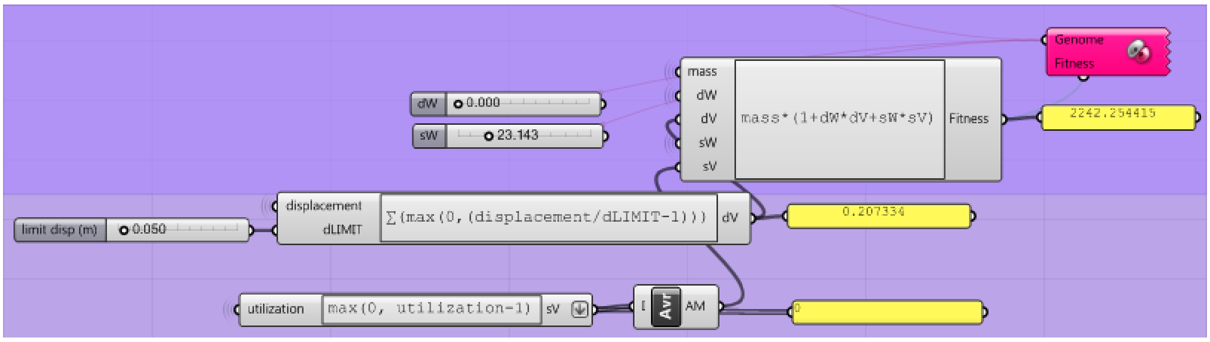
Nr de secțiuni:	161	Material: aluminiu	
Tip sectiune:	FRQ - Germany		
Greutate	[]	name	A
kg	Gradul de utilizare	-	cm2
2269.881531	0.542538	220/12.5	97
111.489443	0.330339	25/2	1.74
Deplasare maximă	0.617334	300/12.5	137
m	0.508859	250/12	108
0.058813	0.456701	40/4	5.35
	0.571945	175/8	51.2
Tensiune maximă	0.6953	300/12.5	137
[kN/cm2]	0.602216	300/12.5	137
-7.79E+00	0.53988	300/10	113
	0.542164	40/4	5.35



Dupa 50 de rulări:



Nr de secțiuni:	161	Material:	aluminiu
Tip sectiune:	FRQ - Germany		
Greutate	[]	name	A
kg	Gradul de utilizare	-	cm2
2242.254415	0.501367	250/12	108
	0.238805	25/2	1.74
Deplasare maximă	0.734319	300/12.5	137
m	0.494745	250/12	108
0.060367	0.55887	25/3	2.41
	0.557218	140/8	40
Tensiune maximă	0.675176	300/12.5	137
[kN/cm2]	0.604441	300/12.5	137
-7.55E+00	0.519979	300/8	91.2
	0.563083	30/2.5	2.59



- Optimizare topologie, secțiuni și eliminare elemente subutilizate (BESO)

O variație a acestei structuri este obținută prin plasarea de elemente pe liniile de curgere a forțelor în structură, iar GA este utilizat concomitent pentru optimizarea secțiunilor. Valorile obținute pentru greutatea structurii variază între 4100 kg și 6196 kg pentru condițiile impuse.

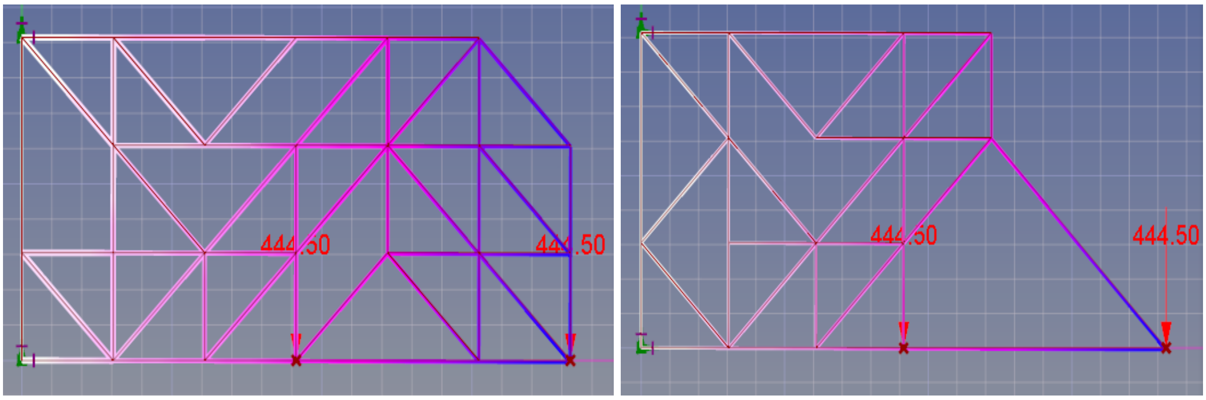
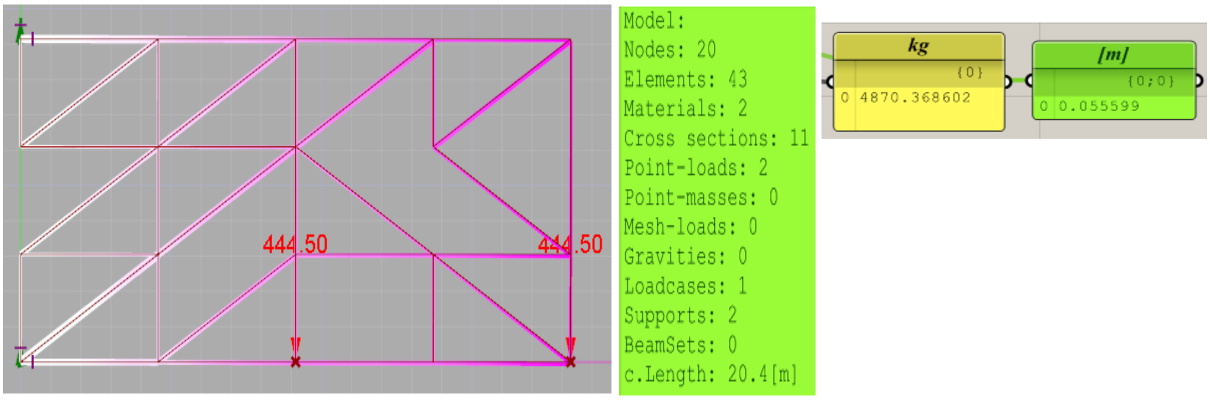
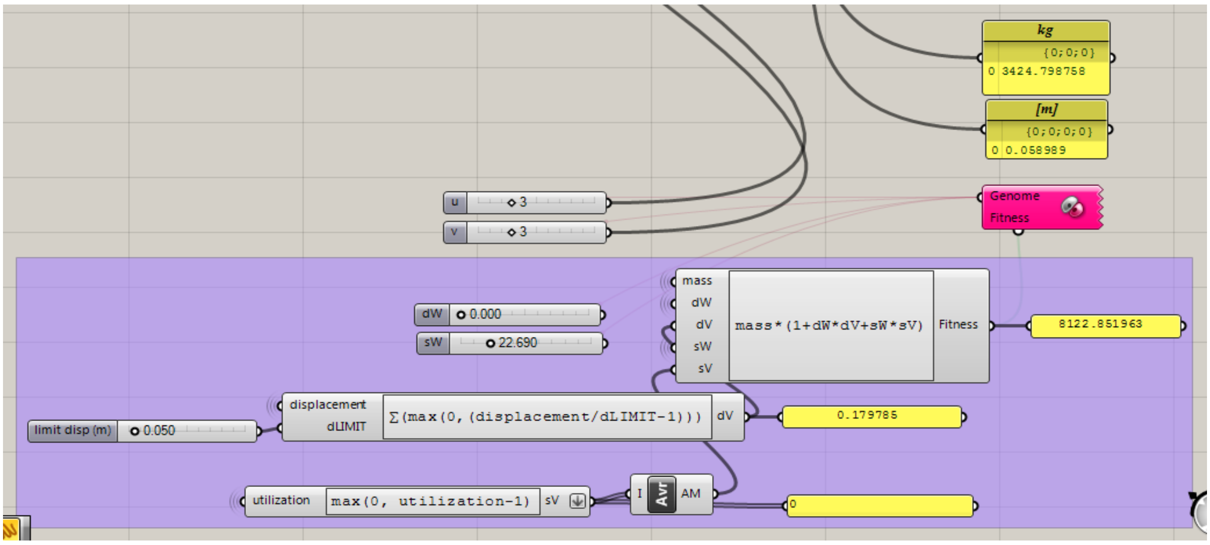
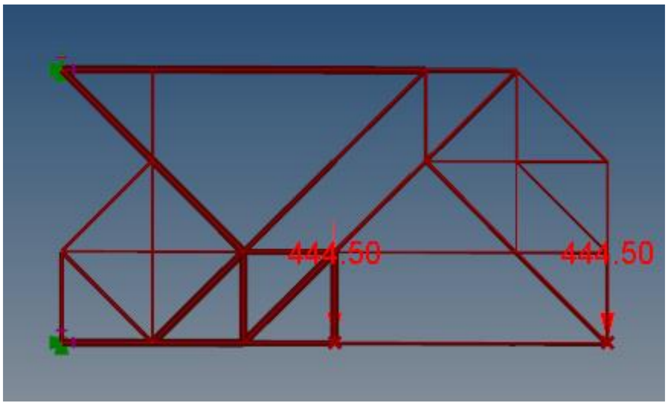


Figura 10.14 *Rezultatele optimizării cu Force Flow Finder (BESO) și GA.*

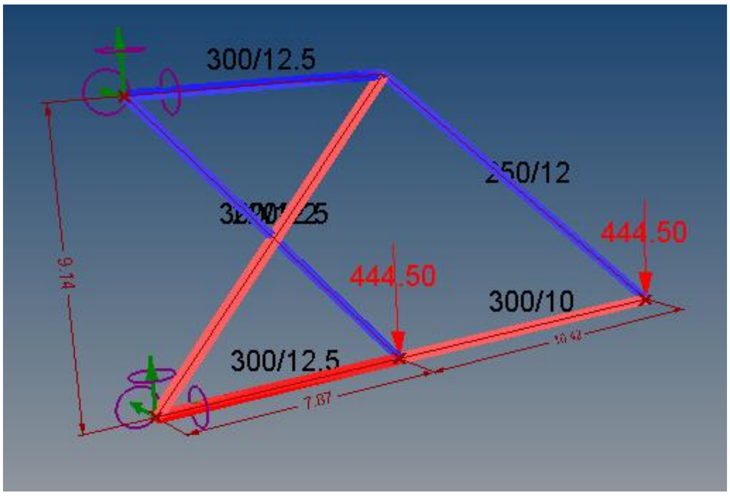


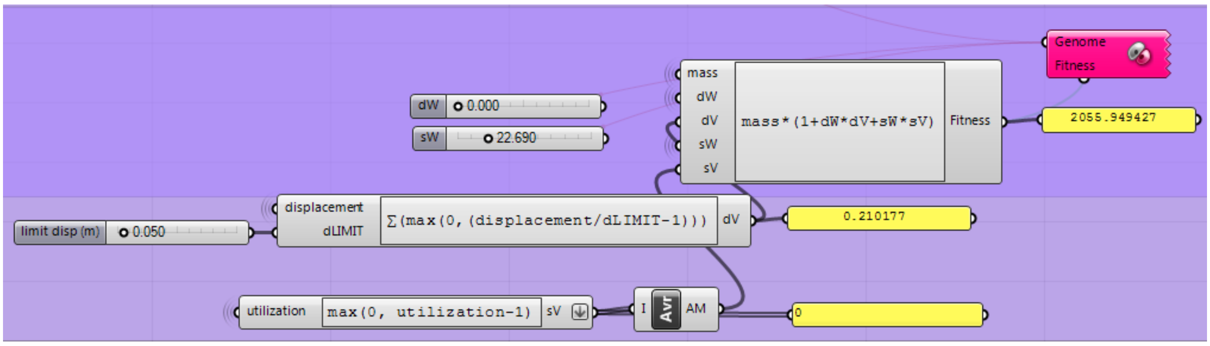




Rezultat intermediar 3424 kg. De la valorile parametrilor acestei structuri am continuat spre obținerea formei optime si greutății minime:

Nr de secțiuni:	161	Material:	aluminiu
Tip sectiune:	FRQ - Germany		
Greutate	[]	name	A
kg	Gradul de utilizare	-	cm2
2055.949427	0.546683	220/12	93.7
	0.692004	300/12.5	137
Deplasare maximă	0.529324	250/12	108
m	0.677305	300/12.5	137
0.060509	0.614507	300/12.5	137
	0.514085	300/10	113
Tensiune maximă			
[kN/cm2]			
-7.58E+00			





Tabel 10-7 Cel mai bun rezultat găsit pentru problema benchmark cu 10 bare din diferite surse, pentru optimizarea secțiunilor.

Sursa	Greutate in lbs	Greutate in kg
(Rajeev & Krishnamoorthy, 1992)	5613.84	2546.39
(Coello Coello, 1994)	5586.59	2534.03
(Turkkan, 2003)	5491.71	2490.99
(Kripka, 2004)	5490.74	2490.55
(Sousa, 2003)	5525.04	2506.11
Studiul prezent	5250.022	2381.37

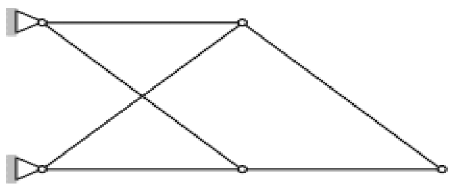


Figura 10.15 Forma optimă a structurii conform literaturii (aceeași formă a fost gasită pentru valoarea minimă a greutății in studiul prezent).

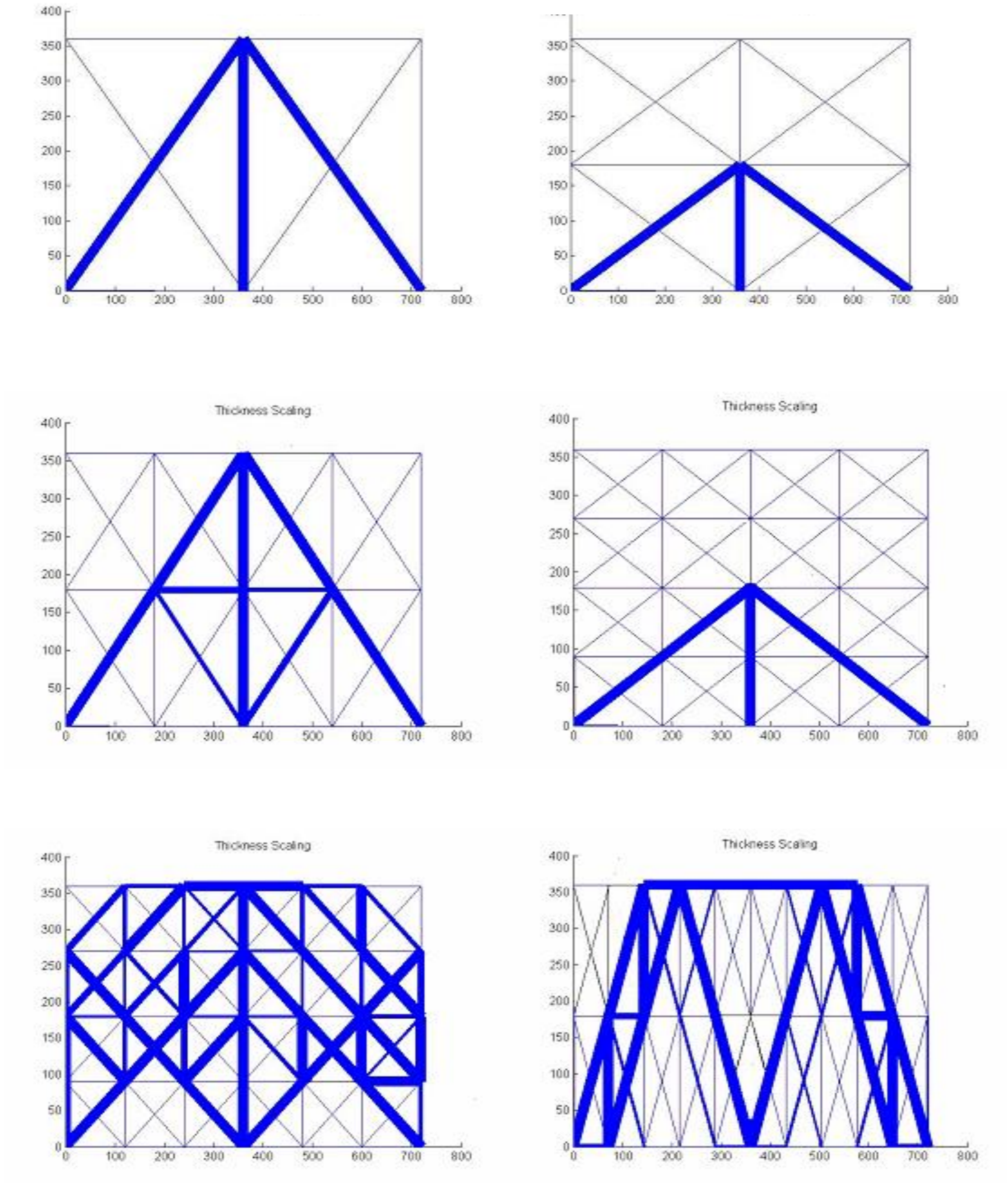
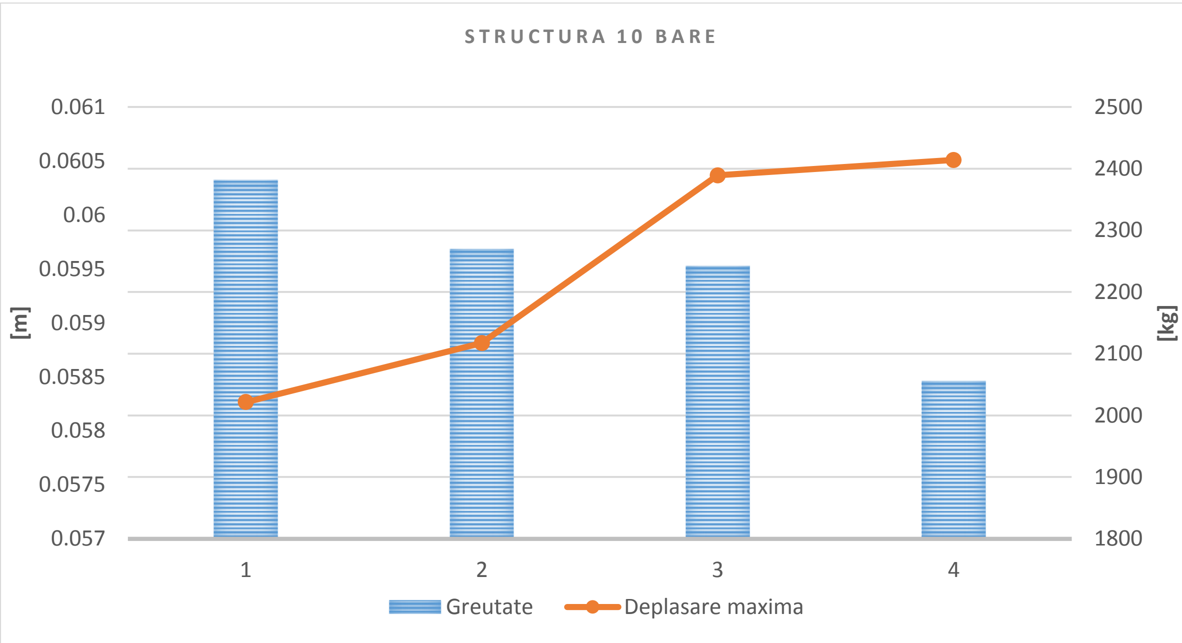


Figura 10.16 Rezultate obținute pentru niveluri diferite de discretizare.

10 bare			
Strategia	**Greutate**	**Deplasare maximă**	**Gradul de utilizare mediu**
Optimizare secțiuni	**2381.370974**	0.058265	0.5236272
Optimizare topologie si secțiuni	**2269.881531**	0.058813	0.5407276
Optimizare topologie si secțiuni (50 rulări)	**2242.254415**	0.060367	0.5448003
Optimizare topologie si secțiuni + BESO	**2055.949427**	0.060509	0.595651333



10.2.1.2 Problema 2: Grinda cu zăbrele cu 9 bare

Structura plană cu zăbrele alcătuită din 6 noduri și 9 bare, care a fost optimizată în programul în platforma Matlab este acum optimizată în programul conceput în Grasshopper. Metalul folosit este oțel S235. Structura este încărcată cu o forță verticală distribuită având valoarea variabilă 10-50 kN și greutatea proprie. Pentru optimizarea acestei structuri, din punct de vedere al volumului, se impun restricții asupra tensiunilor din bare, deplasării maxime și gradului de utilizare a elementelor.

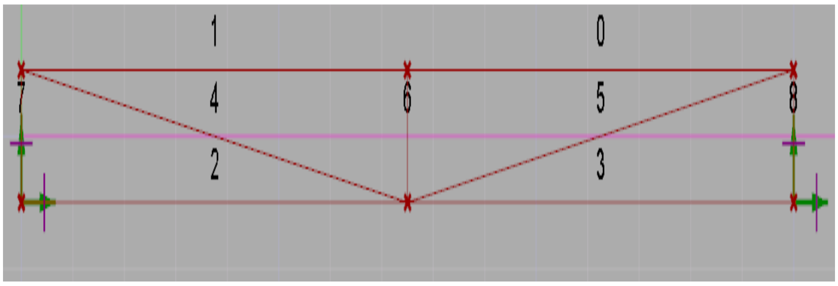


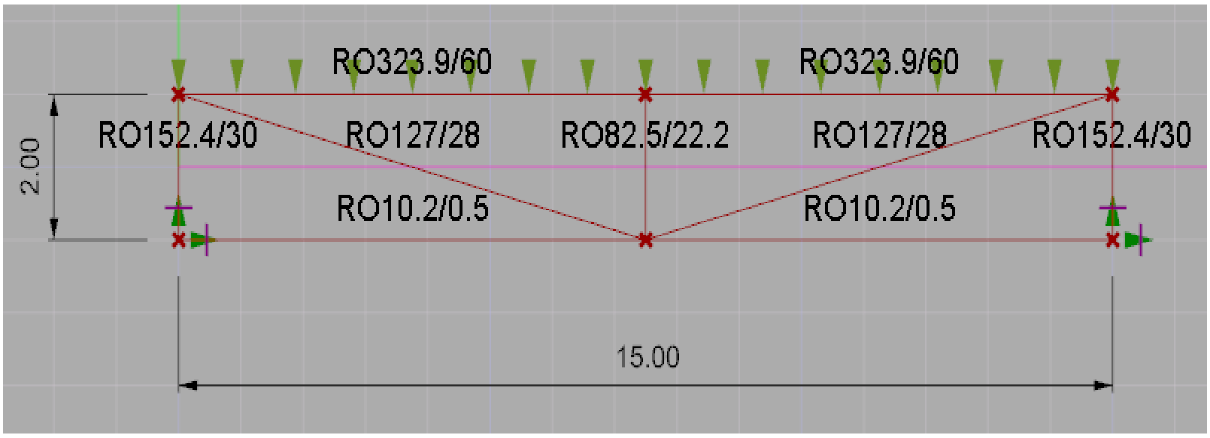
Figura 10.17 Structura articulată plană cu 9 bare.

Încărcare	#	Volum [m3]	Masa [kg]
Gravitațională	LC1	0.008579	67.344245
Distribuita 50kN + Gravitațională	LC2	1.011998	7944.181896
Distribuita 25kN + Gravitațională	LC3	0.522089	4098.396603
Distribuita 10kN + Gravitațională	LC4	0.328378	2577.764055

LC1	Bara	Gradul de utilizare	Secțiunea aleasă		Gradul de utilizare	Secțiunea aleasă
	0	0.985894	RO31.8/6.3		0.984722	RO244.5/65
	1	0.985894	RO31.8/6.3		0.989233	RO244.5/65
	2	0.16512	RO10.2/0.5		0.964279	RO152.4/36
	3	0.16512	RO10.2/0.5		0.99037	RO152.4/30
	4	0.250521	RO10.2/0.5		0.987724	RO159/40
	5	0.250521	RO10.2/0.5		0.984333	RO159/40
	6	0.850783	RO10.2/0.5		0.980363	RO48.3/12.5
	7	0.967245	RO22/5		0.966162	RO219.1/45
	8	0.967245	RO22/5		0.964038	RO219.1/45
LC3	0	0.970314	RO219.1/30	LC4	0.972069	RO168.3/28
	1	0.974894	RO219.1/30		0.977929	RO168.3/28
	2	0.956374	RO127/20		0.978857	RO88.9/16
	3	0.984627	RO127/17.5		0.97853	RO88.9/14.2
	4	0.998711	RO127/25		0.970128	RO88.9/22.2
	5	0.995325	RO127/25		0.964775	RO88.9/22.2
	6	0.950214	RO42.4/7.1		0.94819	RO33.7/6.3
	7	0.993732	RO168.3/50		0.974032	RO127/36
	8	0.99145	RO168.3/50		0.994532	RO127/32

Optimizare geometrie și secțiuni ale barelor: sunt introduse deschiderea grinzii și înălțimea ca variabile de design.

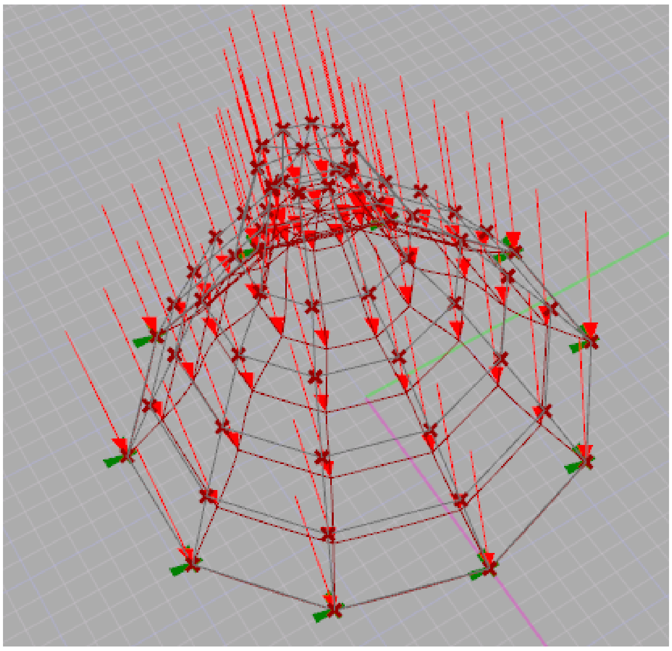
- dimensiuni 2.00m pe 15.00m, LC1:

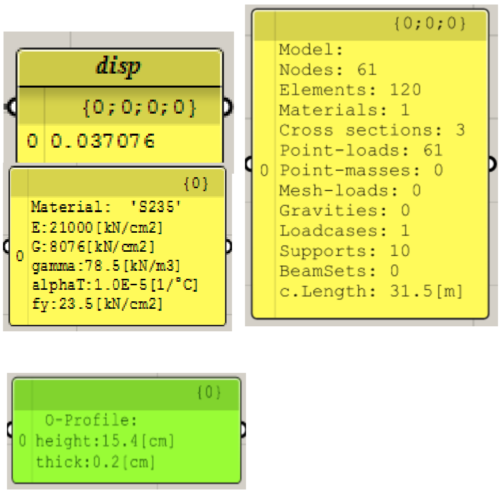


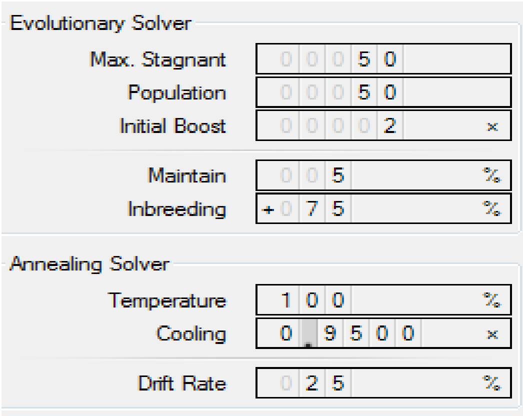
10.2.1.3 Problema 3: Structura cu 120 de bare

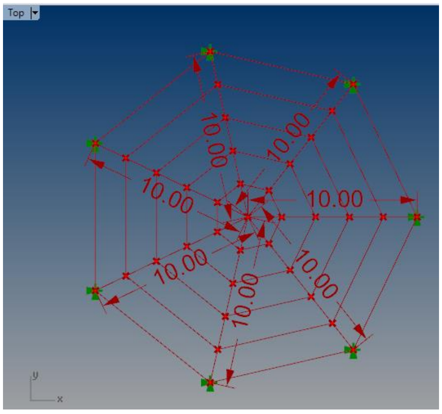
Structura articulată cu 120 de elemente optimizată cu Galapagos. Secțiunea este fixată și forma este optimizată sub restricții de deplasare. Structura este supusă la încărcări în 61 de noduri cu valoarea de 100 kN.

Am căutat o soluție aproximativă cu SA apoi aceasta este folosită ca punct de plecare pentru algoritmul evoluționist.

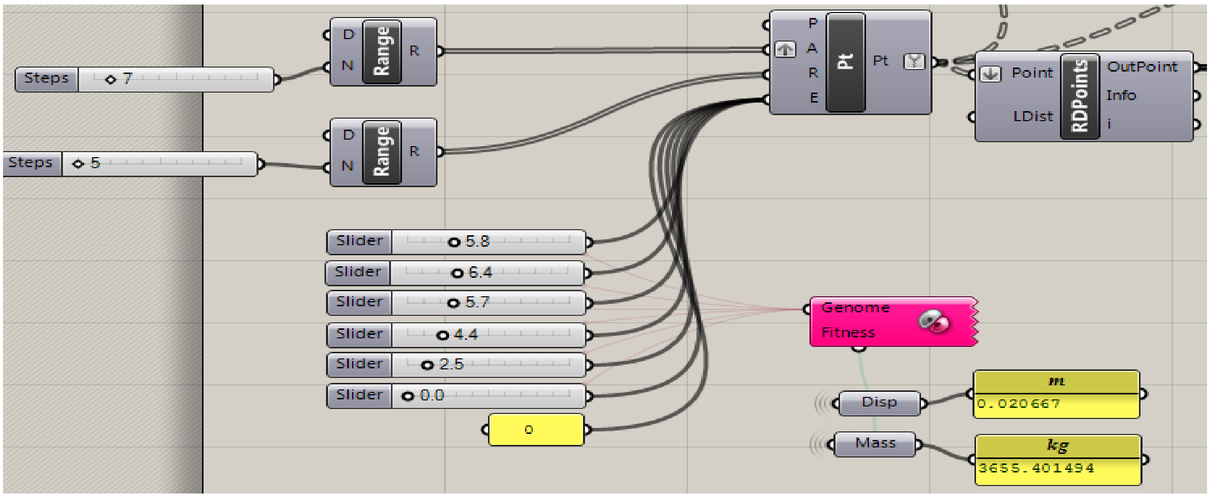




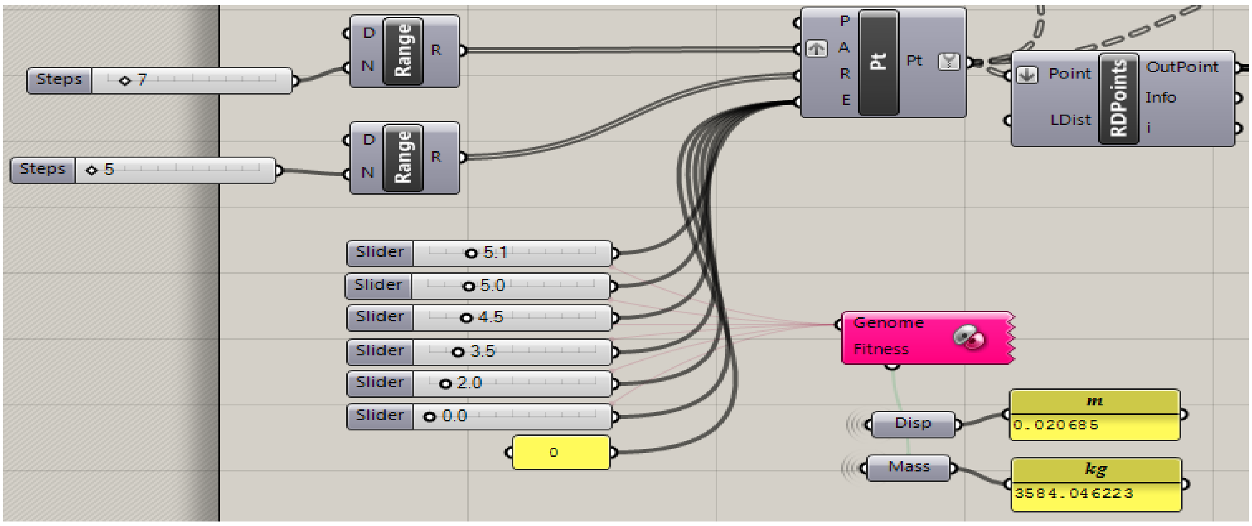




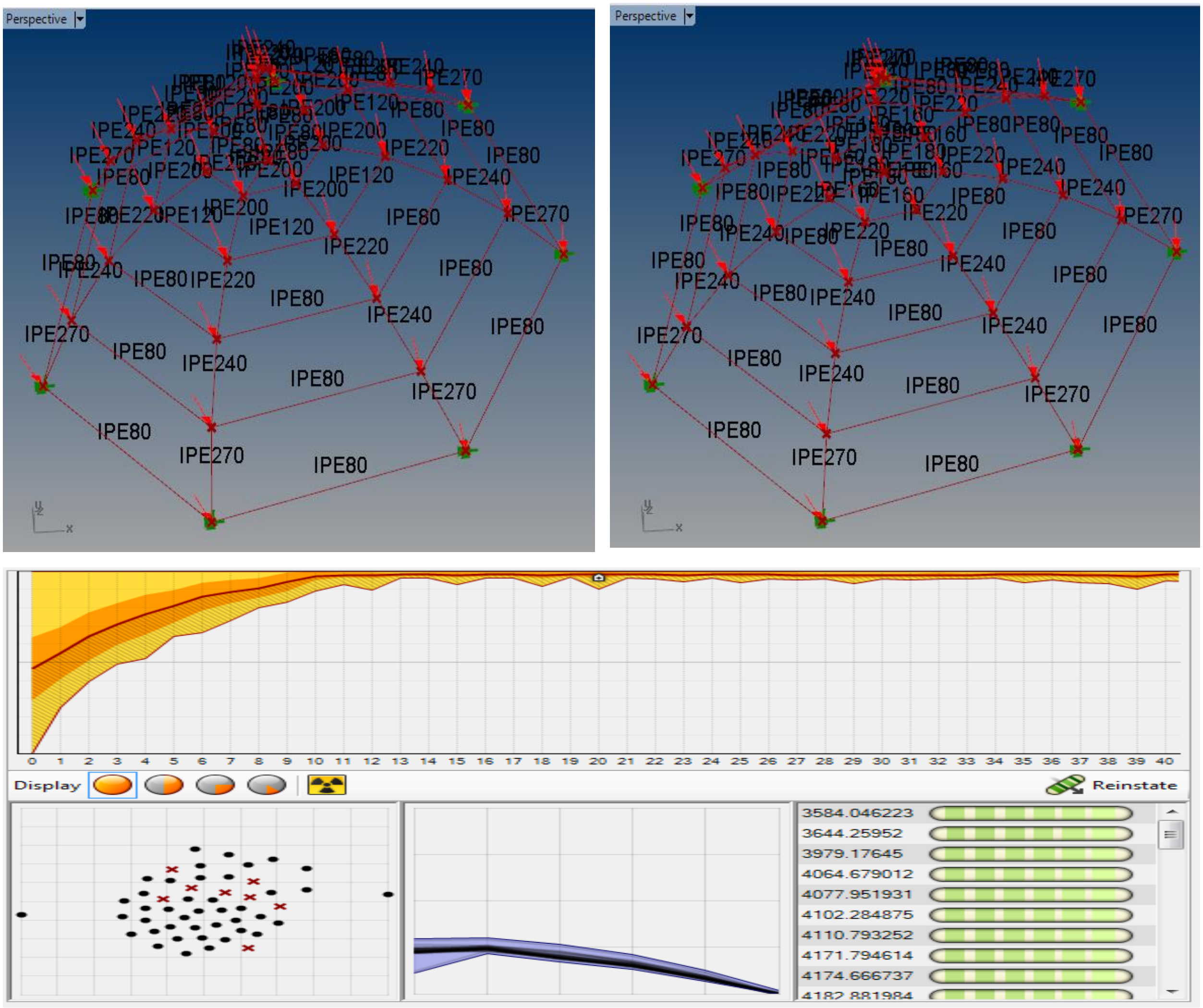
- Optimizare secțiuni și topologie



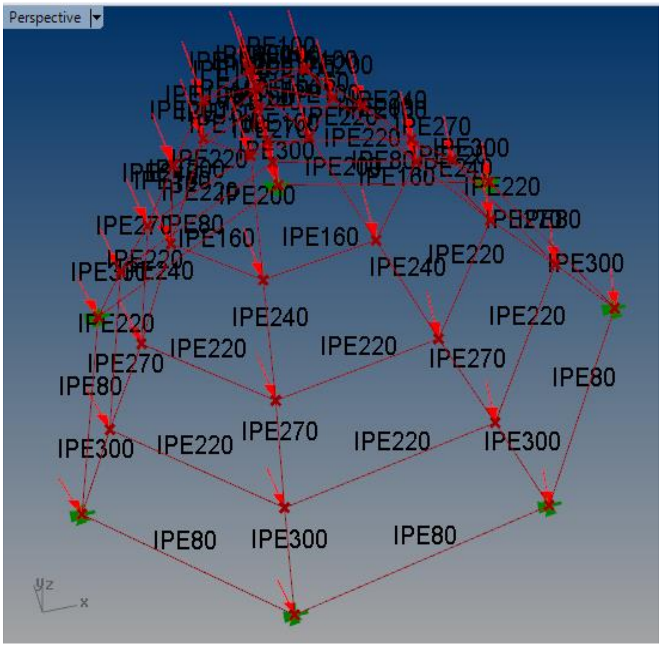
Pozițiile nodurilor sunt introduse ca variabile în coordonate cilindrice.



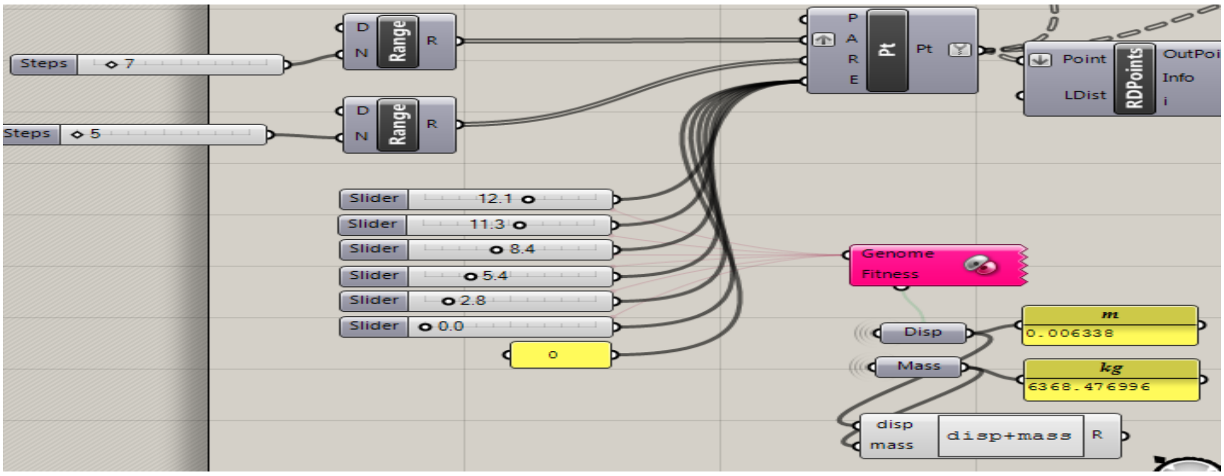
Rezultat obținut dupa 10 rulări, funcția obiectiv incluzând doar masa structurii. Mai jos este prezentată o rulare tipică, valorile obținute și convergența.



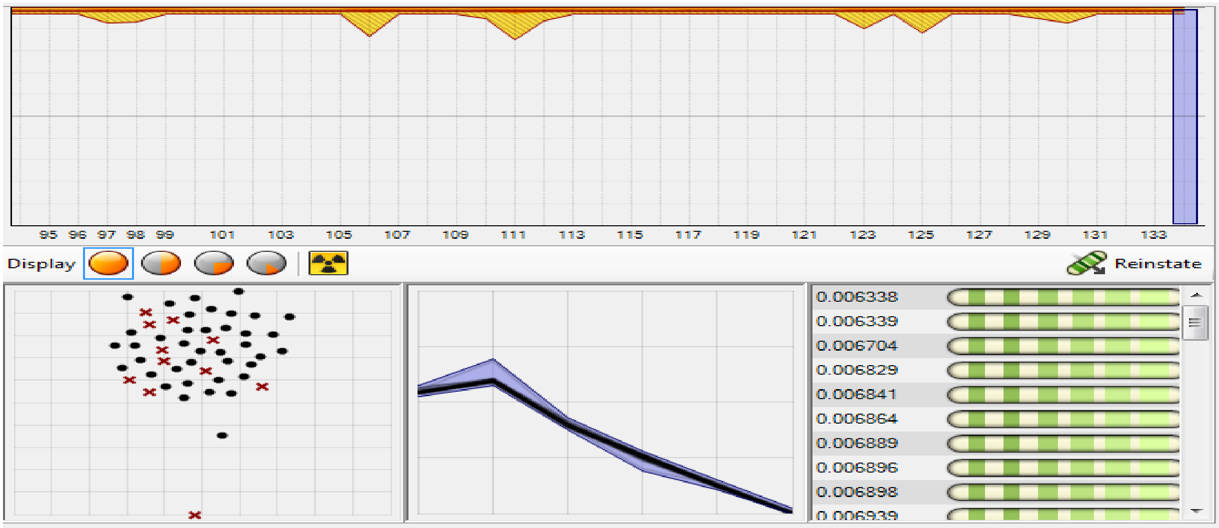
Am folosit apoi doar deplasarea ca obiectiv pentru a observa influența în cadrul funcției obiectiv a acestei valori. Din nou sunt prezentate valorile obținute pentru variabile, valoarea obiectiv și convergența.



Perspective
IPE160
IPE160
IPE240
IPE220
IPE300
IPE220
IPE240
IPE220
IPE220
IPE220
IPE270
IPE80
IPE270
IPE80
IPE300
IPE220
IPE270
IPE220
IPE300
IPE80
IPE300
IPE80

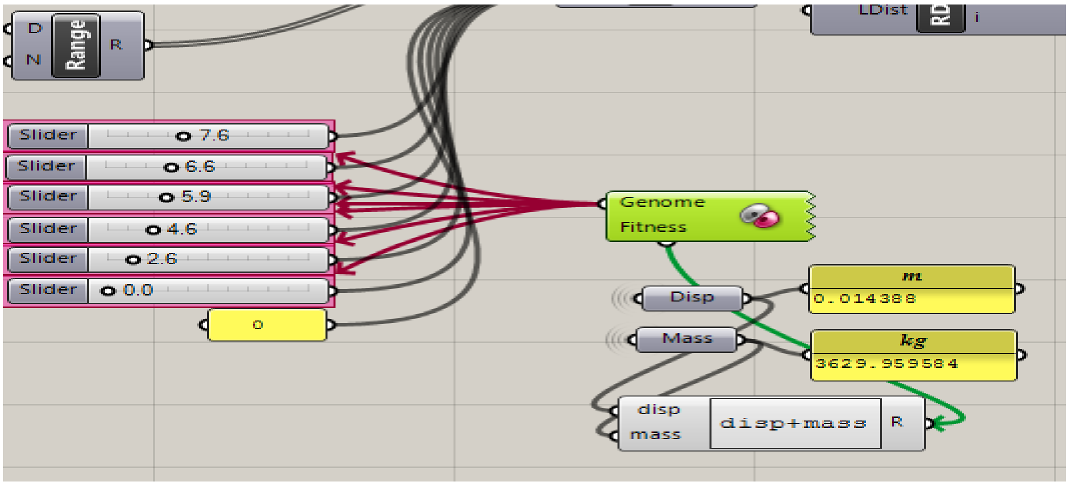


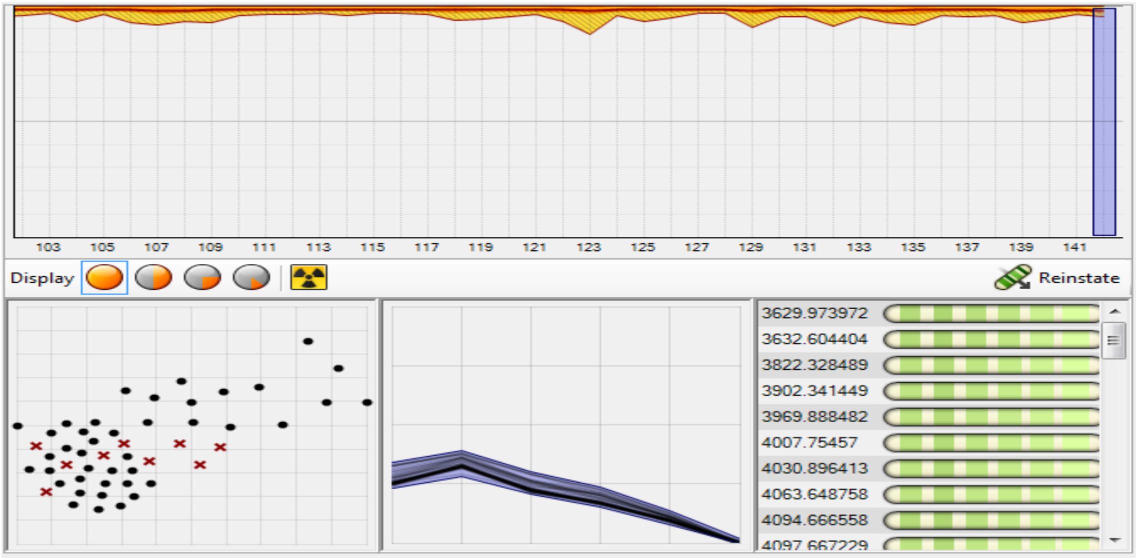
Steps
7
Steps
5
D
N
Range
R
P
A
R
E
Pt
Point
LDist
RDPoints
OutPoi
Info
Slider
12.1
11.3
8.4
5.4
2.8
0.0
Genome
Fitness
Disp
Mass
m
0.006338
kg
6368.476996
disp
mass
disp+mass

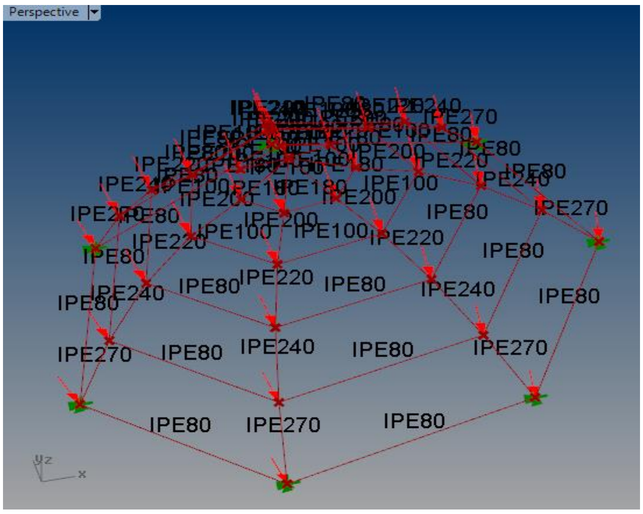


95 96 97 98 99 101 103 105 107 109 111 113 115 117 119 121 123 125 127 129 131 133
Display
Reinstate
0.006338
0.006339
0.006704
0.006829
0.006841
0.006864
0.006889
0.006896
0.006898
0.006939

Se observă că masa structurii crește de la 3584 kg la 6368.47 kg pentru o reducere a deplasării maxime de la 0.02 m la 0.0016 m. Plecând de la cele două extreme posibile, am introdus ambele obiective în funcția fitness, fără penalizări.



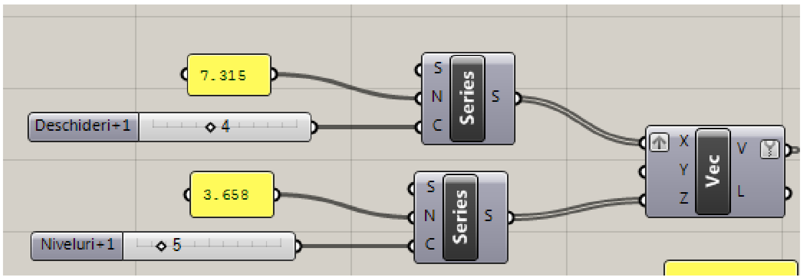


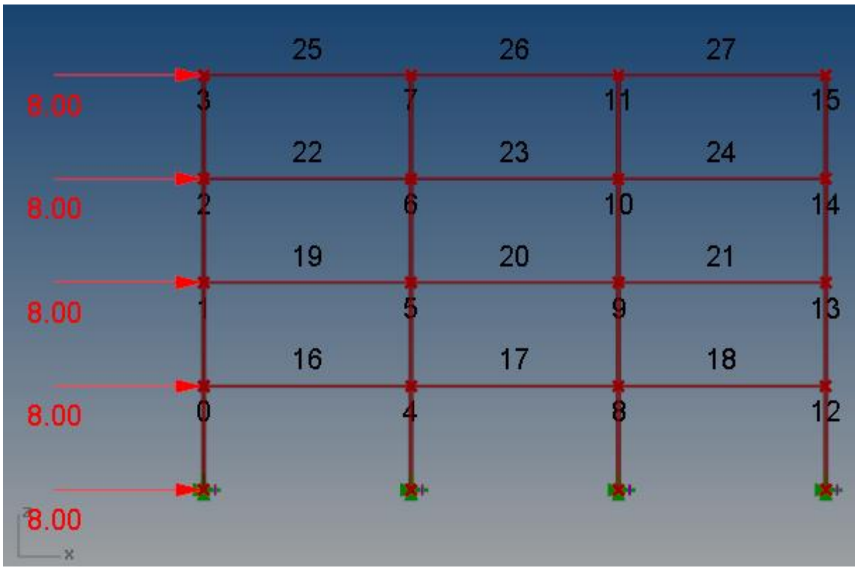


Valori comparabile	SGA	Kaveh	Lee and Geem
Greutate [kg]	6233	8840	8898

10.2.1.4 Problema 4: Cadru multi-nivel

Varianta 1: 3 deschideri @ 7.315m si 4 niveluri @3.658 m

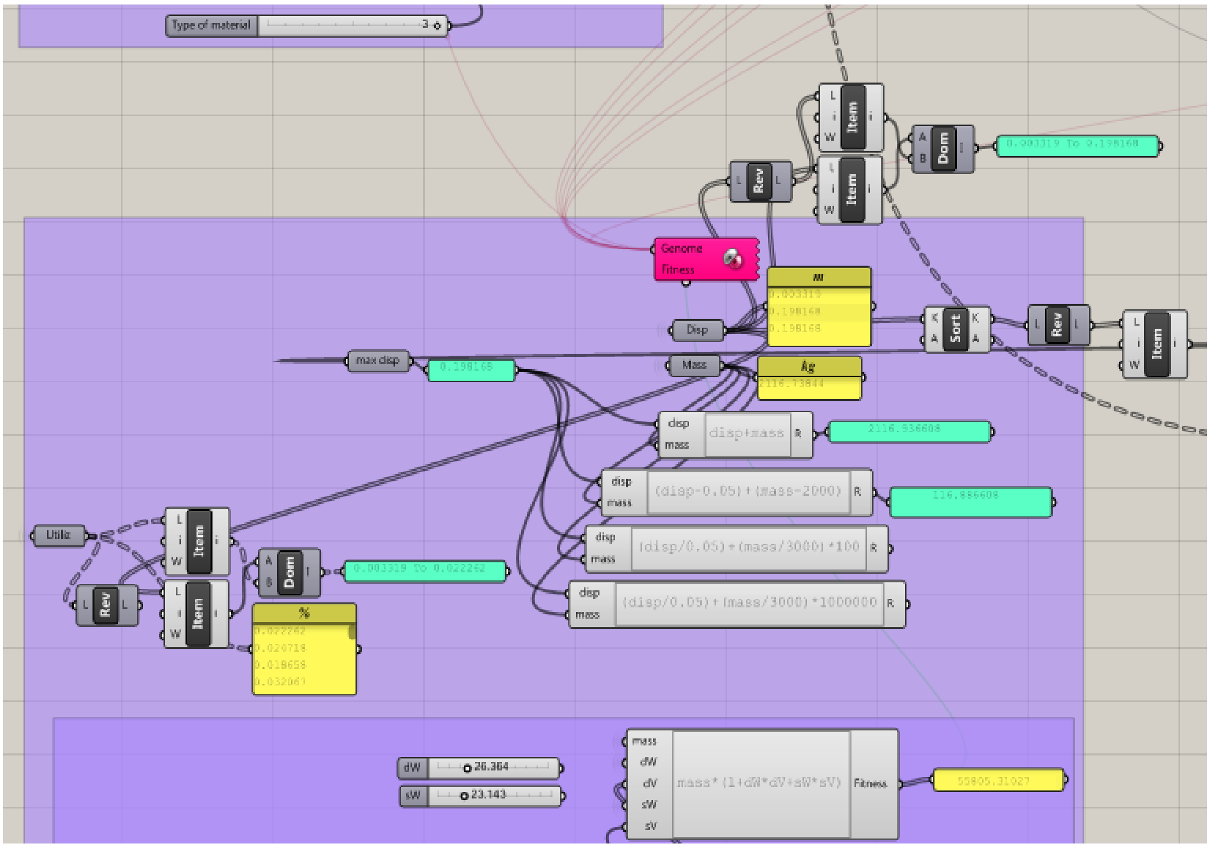


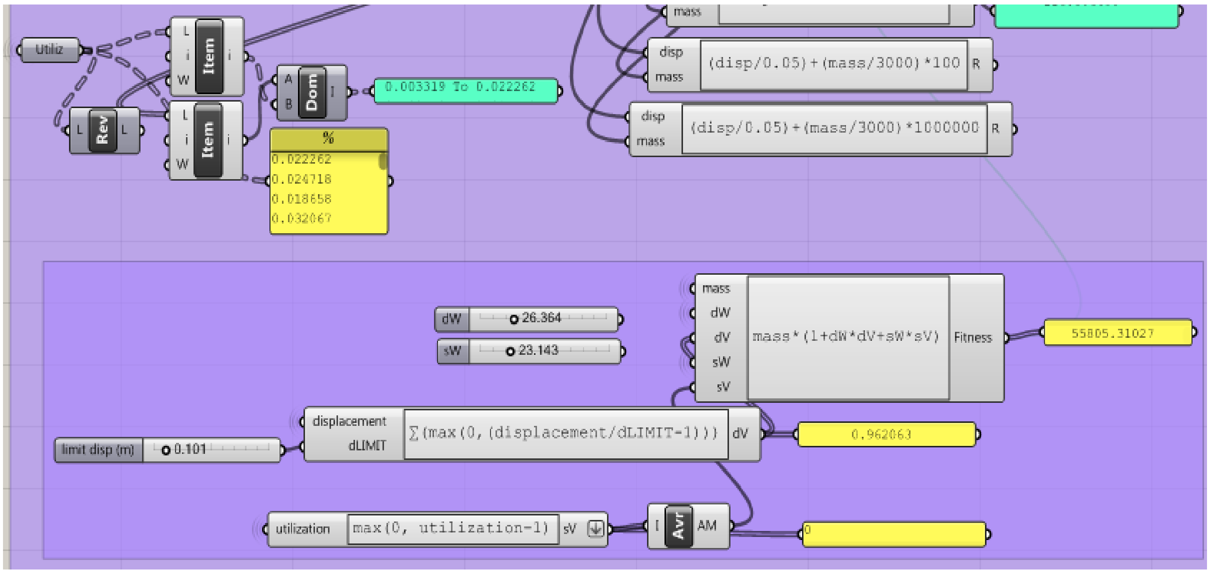


Încărcările gravitaționale cu coeficient de multiplicare 2 și încărcarea din vânt distribuită uniform pe direcția x sunt luate în calcul. Materiale: Steel 'S235' E:21000[kN/cm2] G:8076[kN/cm2] gamma:78.5[kN/m3] alphaT:1.2E-5[1/C°] fy:23.5[kN/cm2], Steel 'S275' E:21000[kN/cm2] G:8076[kN/cm2] gamma:78.5[kN/m3] alphaT:1.2E-5[1/C°] fy:27.5[kN/cm2], Steel 'S355' E:21000[kN/cm2] G:8076[kN/cm2] gamma:78.5[kN/m3] alphaT:1.2E-5[1/C°] fy:36[kN/cm2], 'Aluminum' E:7000[kN/cm2] G:2700[kN/cm2] gamma:27[kN/m3] alphaT:2.4E-5[1/C°] fy:12[kN/cm2].

La fel ca și în cazul structurilor studiate anterior, este formulată funcția fitness pentru problema de față cu coeficienți de penalizare adaptați la nivelul de influență al componentelor funcției fitness. Am rulat algoritmul SA pentru funcțiile obiectiv simple de greutate și deplasare, pentru a determina domeniul de variație al valorilor și apoi am calculat coeficienții de penalizare pentru funcția compusă care vizează minimizarea greutății și deplasărilor și maximizarea gradului de utilizare al elementelor.

Tipul de material și dimensiunile secțiunilor de la care se pleacă pentru căutarea locală sunt luate ca variabile în căutarea evoluționistă.





Utiliz
Item
Rev
Item
Dom
0.003319 To 0.022262
%
0.022262
0.024718
0.018658
0.032067
(disp/0.05)+(mass/3000)*100
(disp/0.05)+(mass/3000)*1000000
dW
26.364
sW
23.143
mass*(1+dW*dV+sW*sV)
Fitness
55805.31027
limit disp (m)
0.101
displacement
dLIMIT
∑(max(0,(displacement/dLIMIT-1)))
dV
0.962063
utilization
max(0, utilization-1)
sV
Avr
AM
0

CONCLUZII ȘI DIRECȚII DE CERCETARE VIITOARE

În studiul de față, motivația analizării comparate a celor două tipuri de algoritmi genetici (simplu, creat în Matlab pe baza bibliotecilor standard și hibrid creat în Grasshopper), a călirii simulate (care este o căutare euristică single-point) și a algoritmului BESO, alături de căutarea locală, a pornit de la o nevoie de a afla ce anume înseamnă, pentru problemele cele mai frecvent analizate în lucrările de optimizare structurală, compromisul no free lunch. Este în general acceptat faptul ca nici un algoritm nu are o comportare superioara celorlalți când este testat pe probleme variate. În lucrare am prezentat studii parametrice pentru structuri articulate plane, structuri articulate spațiale, cadre plane și spațiale întâlnite frecvent în practică și în literatura de specialitate. Sunt testați diverși algoritmi evoluționiști pe probleme cu formulare multi-obiectiv, și care au un număr de variabile între câteva zeci și câteva sute, în condiții diverse de încărcare. Am evaluat și am analizat critic formulările funcțiilor fitness și apoi am propus o strategie de formulare a acestora care să poată fi de ajutor în proiectarea optimală a structurilor.

Am constatat prin analizarea problemelor testate că adăugarea mai multor obiective la o problemă de optimizare sporește exponențial complexitatea acesteia, lucru amintit și de alți autori. În cazul designului structural, se dorește un rezultat care să fie, pe cât posibil, și ușor și rigid. Deoarece aceste două obiective sunt conflictuale, trebuie să existe un compromis. Va exista un design de greutate minimă, un design de rigiditate maximă și un număr infinit de soluții care sunt un compromis între greutate și rigiditate.

Aceste soluții care nu pot fi îmbunătățite din punctul de vedere al unuia dintre criterii fără a influența negativ rezultatul altuia formează setul Pareto, și curba descrisă de aceste soluții cu greutate și rigiditate pe axele de referință formează frontiera Pareto. Un design Pareto optim nu este dominat de nici un altul, adică nici unul dintre celelalte nu au valori mai bune pentru toate criteriile considerate. Alegerea aceasta este delegată decizionarului, care va alege soluția preferată. Cu alte cuvinte, a defini o problemă ca multi-obiectiv semnalează lipsa unor informații legate de aceasta. Obiectivele dorite sunt știute dar nu și detaliul combinării lor. În unele cazuri, această informație poate fi dedusă printr-un calcul interactiv.

Adesea problemele de optimizare sunt multi-modale, adică au un număr multiplu de soluții bune. Pot fi soluții bune global (aceeași funcție obiectiv) sau poate exista un amestec de soluții bune locale și globale. Obținerea cel puțin parțială a acestora este scopul optimizării multi-modale.

Tehnicile de optimizare clasică, datorită metodei iterative utilizate, nu au o comportare satisfăcătoare când sunt utilizate la obținerea de soluții multiple, nefiind garantată obținerea de soluții diferite chiar și cu pornirea din soluții inițiale diferite la fiecare rulare a algoritmilor. Algoritmii evoluționiști, pe de altă parte, sunt foarte populari în utilizarea lor la obținerea de soluții multiple la probleme multi-modale.

Contribuții personale

Pornind de la stadiul actual în domeniul optimizării structurilor, descrierea conceptului, a tipurilor de optimizare posibile în domeniu, din punct de vedere al variabilelor alese și al obiectivelor vizate, algoritmi utilizați în domeniu, metode de calcul și metode de modelare alternative, teza trece în revistă avantajele utilizării optimizării în proiectare.

Utilizarea tehnicilor de optimizare descrise duce la un proces de proiectare rapid, eficient, la obținerea unui consum proiectat redus al materialelor, avantaje care trebuie speculate în condițiile dezvoltării actuale și ritmului alert de pe piața construcțiilor .

Am descris o procedură de realizare a optimizării structurale în MATLAB, și a posibilităților oferite de Global Optimization Toolbox, și codarea operatorilor proprii.

Utilizând programele de optimizare în mediul de programare vizuală parametrică Grasshopper (McNeel Rhinoceros), am descris o strategie de aplicare a doi algoritmi evoluționiști care conduc la găsirea mai rapidă a soluțiilor bune. Am testat trei tipuri de optimizare structurală: topologică, geometrică, secțională.

Etapa de evaluare a indivizilor s-a realizat printr-o modelare numerică, folosind funcții fitness dar există posibilitatea alegerii unor soluții bune, chiar dacă nu optime global, pe criterii estetice, utilizatorul având astfel ultimul cuvânt de spus în alegerea soluției finale.

Au fost efectuate studii teoretice și calculul unor structuri benchmark cu rezultate comparabile cu cele din literatură.

Am studiat dimensionarea aceleiași grinzi cu zăbrele păstrând deschiderea și alternând parametrii care influențează dimensionarea și geometria.

Am realizat o analiză critică a performanțelor diverșilor algoritmi.

Direcții de cercetare viitoare

Optimizarea este o provocare pentru ingineri și arhitecți, și, deși s-au făcut un număr mare de pași înspre introducerea ei în practica curentă încă nu este adoptată de nespecialiști. Totuși accesul la algoritmi simpli este extrem de util atât în calculul structurilor clasice cât și în inovația structurilor free-form. Din punct de vedere structural, pentru a garanta nivelul necesar de fiabilitate, e nevoie de expertiza de specialitate în designul și construcția de morfologii free-form. Optimizarea este recomandată în procesul preliminar de design pentru a găsi cele mai potrivite soluții în concordanță cu funcția așteptată a structurii. Dar utilitatea strategiei de optimizare este limitată de includerea unei analize avansate în procedură. Autoarea își dorește să dezvolte în continuare soluțiile de formulare a problemelor în domeniul plastic și experimentarea cu formulări geometrice fractale (lucrare în curs de publicare).

BIBLIOGRAFIE

Anderson, S., 2004. Eladio Dieste: innovation în structural art. New York: Princeton Architectural Press.

Arora, J. S. ,1989. Introduction to optimum design. McGraw-Hill, New-York.

Arora, J. S. ,1997. Guide to structural optimization. ASCE Manuals and Reports on Engineering Practice No. 90.

Barakat SA, Altoubat S. Application of evolutionary global optimization techniques în the design of RC water tanks. Eng Struct 2009;31(2):332_44.

Bartz-Beielstein, T. C. L. a. M. P., 2005. *Sequential Parameter Optimization. IEEE,* s.l.: s.n.

Bäck, T., 1993. *Optimal mutation rates in genetic search. In Proceedings of the fifth International Conference on Genetic Algorithms, pp. 2–8. Morgan Kaufmann,* s.l.: s.n.

Bendsœ, M. & N., K., 1988. Generating optimal topologies în structural design using a homogenization method. Computer Methods în Applied Mechanics and Engineering, Volumul 71, p. 197–224.

Bletzinger. K-U și R. E., „Structural optimization and form finding of lightweight structures,” Computers & Structures, vol. 79, pp. 2053-2062, 2001.

Buelow,v. P. et.al., „Optimization of structural form using a genetic algorithm to search associative parametric geometry”.

Burry, Mark. 2007. “Innovative Aspects of the Colònia Güell Chapel Project.” În Gaudí Unseen: Completing the Sagrada Famíla, edited by Mark Burry, 59-61. Berlin: Jovis

Burry, Mark. 2011. Scripting Cultures. Chichester: Wiley.

Camp CV, Pezeshk S, Cao G. Optimized design of two dimensional structures using a genetic algorithm. J Struct Eng ASCE 1998;124(5):551_9.

Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Comput Methods Appl Mech Eng 2002;191(11_12):1245_87.

Coello Coello, C., 1994. *Discrete Optimization of Trusses Using Genetic Algorithms. In Expert Systems Applications and Artifficial Intelligence EXPERSYS-94.International Technology Transfer Series, pp. 331-336.,* s.l.: s.n.

Colin , R. R. & Jonathan, E. R., 2002. Genetic Algorithms Principles and Perspectives. New York: Kluwer Academic .

De Jong, K., 1975. *An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan,* s.l.: s.n.

Dino, Ipek. 2012. “Creațive Design Exploration by Parametric Generative Systems în Architecture.” METU Journal of Faculty of Architecture 29 (1): 207–224.

Dueck, G. and Scheuer, T. "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing." J. Comp. Phys. 90, 161-175, 1990.

EN 1990:2002 (English): Eurocode - Basis of structural design [Authority: The European Union Per Regulation 305/2011, Directive 98/34/EC, Directive 2004/18/EC]

Erbatur F, Hasancebi O, Tutuncil I, Kihc H. Optimal design of planar and space structures with genetic algorithms. Comput Struct 2000;75:209_24.

Eurocode 3: EN 1993-1-1 Design of steel structures – Part 1-1: general rules and rules for buildings. s.l.:s.n.

Gallagher, R. H.,Gellatly,R.A.,"A procedure for automated minimum weight structural design"-1.Theoretical basis, 2.Application, Aeronaut J.,1976.

Genetic Algorithm and Direct Search Toolbox 2.4.2. (2009). The MathWorks.. s.l.:s.n.

Gerber, David. 2007. “Parametric Practices: Models for Design Exploration în Architecture.” PhD dissertation, Harvard University.

Gutkowski, W.,Bauer,J.,º.a.”Explicit formulation of Kuhn-Tucker necessary conditions în structural optimization”, Computers & Structures, vol.37, 1990.

H., E., N., O. & W., S., 1997. Applied Structural Mechanics. s.l.:Springer.

Hajela, P. 1990: Genetic search - an approach to the nonconvex optimization problem.AIAA J. 26, 1205–1210. s.l.:s.n.

Hayalioglu, M., 2001. Optimum load and resistance factor design of steel space frames using genetic algorithm. Structural and Multidisciplinary Optimization, 21(4), pp. 292-299.

Heyman, Jacques. 1995. The Stone Skeleton: Structural Engineering of Masonry Architecture. Cambridge: Cambridge University Press.

Hoenderkamp, J.C.D. şi Snijder, H.H., Preliminary Analysis of High-Rise Braced Frames with Facade Riggers, Journal of Structural Engineering, vol.129, no.5, May 2003, p.640–647.

Hooke, Robert. 1675. A Description of Helioscopes, and Some Other Instruments. London: Royal Society.

Hristache, P. & Veturia, C., 1981. *Calculul Structurilor Optimăle.* Bucuresti: s.n.

Hulea, R., 2011. Cercetari Privind optimizarea structurilor la stadioane medii si mari, Departamentul de Mecanica Construcțiilor, Universitatea Tehnica Cluj-Napoca.

Hulea, R., Nicoreac, M., Pârv, B., & Petrina, B., 2011. Optimum design of outrigger and belt truss systems using genetic algorithm, Structural Engineers World Congress, Como, Italy.

Ingber, L. "Simulated Annealing: Practice Versus Theory." Math. Comput. Modelling 18, 29-57, 1993.

Iordăchescu, A. 2001: *Construcţii inteligente*, Editura Reg. Arcadia.

K., D. & S., G., 2001. Design of truss-structures for minimum weight using genetic algorithms. Finite Elements în Analysis and Design, Volumul 37, pp. 447-465.

Kaminakis, N, G. E. Stavroulakis, 2012. *Topology optimization for compliant mechanisms, using evolutionary-hybrid algorithms and application to the design of auxetic materials,* s.l.: Composites Part B Engineering (Impact Factor: 2.14).; 43(6):2655–2668. DOI: 10.1016/j.compositesb.2012.03.018.

Kaveh A, Abditehrani A. Design of frames using genetic algorithm, force method and graph theory. Int J Numer Methods Eng 2004;61:2555_65.

Kaveh A, Rahami H. Analysis, design and optimization of structures using force method and genetic algorithm. Int J Numer Methods Eng 2006;65(10): 1570_84.

Kaveh, A. & Talatahary, S., 2009. Particle Sworm Optimization, Ant Colony Strategy and Harmony Search Scheme Hybridized for Optimization of Truss Structures. Computers and Structures.

Kavlie,D.,Moe,J.,"Application of nonlinear programming to optimum grillage design with nonconvex sets of variables",J.Numerical Meth. 4,1979.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. "Optimization by Simulated Annealing." Science 220, 671-680, 1983.

Kress, G. & Keller, D., 2007. Structural optimization, Zurich: Swiss Federal Institute of Technology.

K-U, B. & E., R., 2001. Structural optimization and form finding of lightweight structures. Computers & Structures, Volumul 79, pp. 2053-2062.

Kripka, M., 2004. *Discrete Optimization of Trusses by Simulated Aneealing, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. XXVI, No. 2*, s.l.: s.n.

Kuhn, H. W.; Tucker, A. W. ,1951. "Nonlinear programming". Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press. pp. 481–492. MR47303

Lagaros ND, Psarras LD, Papadrakakis M, Panagiotou G. Optimum design of steel structures with web openings. Eng Struct 2008;30(9):2528_37.

Lee, K. S. & Geem , Z. W., 2004. A New Structural Optimization Method Based on the Harmony searth Algorithm. Computers and Structures.

Leite ,J.P.B., B.H.V. Topping, Improved genetic operators for structural engineering optimization, Advances în Engineering Software, Volume 29, Issues 7–9, August–November 1998, Pages 529-562, ISSN 0965-9978, http://dx.doi.org/10.1016/S0965-9978(98)00021-0.

Lightweight wide span coverings, International Association for Wind Engineering, ANIV, 1990.

Liu, T. & Deng, Z., 2006. Design optimization for truss structures under elasto-plastic loading condition. Acta Mechanica Solida Sinica, 19(3), pp. 264-274.

M. Majowiecki: Snow and wind experimental analysis în the design of long span subhorizontal structures, J. Wind Eng. Ind. Aerodynamics, 1998

M. Marinaki, G. Stavroulakis, 2012. *"A Differential Evolution Algorithm for Fuzzy Control of Smart Structures"*, Stirlingshire, UK, Paper 277, doi:10.4203/ccp.99.277: in B.H.V. Topping, (Editor), "Proceedings of the Eleventh International Conference on Computational Structures Technology", Civil-Comp Press.

M¨uller, S., 2002. Bio-inspired optimization algorithms for engineering applications. Dissertation, s.l.: Swiss Federal Institute of Technology Zurich.

McNeel B., Rhino, http://www.rhino3d.com

Melanie, M., 1996. An Introduction to Genetic Algorithms. London: Massachusetts Institute of Technology.

M.P., B. & N., K., 1988. Generating Optimal Topologies in Structural Design using a Homogenization Method. *Computer Methods in Applied Mechanics and Engineering,* Volumul 71, pp. 197-224.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M.; Teller, A. H.; and Teller, E. "Equation of State Calculations by Fast Computing Machines." J. Chem. Phys. 21, 1087-1092, 1953.

Michalewicz, Z . “Genetic Algorithms + Data Structures = Evolution Programs”, SpringerVerlag , 1996

Mitsuo, G., Runwei, C. & Lin, L., 2008. Network Models and Optimization Multiobjective Genetic Algorithm Approach. London: Springer-Verlag London Limited.

Moses,F., Onoda,S., "Minimum weight design of structures",J. Numerical Meth. 4,1979.

Moses,F.,"Optimum structural design using linear programming",J.Str.Div.ASCE 90,1968.

Murren, P. C., 2011. Development and implementation of a design-driven harmony search algorithm în steel frame optimization. Notre Dame, Indiana : University of Notre Dame.

M. Nicoreac, B. Pârv, M. Petrina: “Analysis of Tall Buildings with Braced Frames and Outrigger Trusses”, Acta Technica Napocensis: Civil Engineering & Architecture, ClujNapoca, Romania, ISSN 1221-5848, http://construcții.utcluj.ro/ActaCivilEng/

M. Nicoreac, B. Pârv, R. Hulea, B. Petrina: “An Approximate Method of Analysis for Tall Buildings Comprising Outrigger Braced Cores”, Conferința Internațională DEDUCON, 11 noiembrie 2011, Iași, ISSN 2248-0293.

Ochoa, G. I. H. a. H. B., 2000. *Optimal mutation rates and selection pressure in genetic algorithms. In Genetic And Evolutionary Computation Conference,* s.l.: s.n.

Olhoff, N. & J.E., T., 1983. On structural optimization.. J. Applied Mechanics, Volumul 50, p. 1139–1151.

Otten, R. H. J. M. and van Ginneken, L. P. P. P. The Annealing Algorithm. Boston, MA: Kluwer, 1989.

P., v. Buelow. & et.al., fără an *Optimization of structural form using a genetic algorithm to search associative parametric geometry,* s.l.: s.n.

Petrina, M., 1982. *Probleme ale Optimizării proiectării structurilor alcătuite din bare.* Cluj-Napoca: s.n.

Petrina, M., Cătărig, A., ș.a., 2007, Statica construcțiilor în formulare matriceală, Editura U.T. Press.

Petrina, M. , B. Pârv, M. Nicoreac, ș.a., Comparative Study of a Tall Building using Equivalent Column and FEM, Conference Proceedings of the IASS-IABSE "Taller, Longer, Lighter - Meeting Growing Demand with Limited Resources" 2011 Symposium, 20-23 Septembrie 2011, Londra, Marea Britanie, http://www.iabse-iass-2011.com/.

Petrina, M., Cătărig, A. şi S.a., Statica Construcțiilor în Formulare Matriceală, Editura U.T. Press, 2007.

B. Pârv, M. Nicoreac, M. Petrina, T. Petrina : “Results of the Romanian researchers from Cluj Napoca concerning high-rise structures”, Acta Technica Napocensis: Civil Engineering & Architecture, Vol. 53, No. 1, (2011),Cluj-Napoca, Romania, UTPRESS, ISSN 1221-5848, http://construcții.utcluj.ro/ActaCivilEng/

Pezeshk S, Camp CV, Chen D. Design of nonlinear framed structures using genetic algorithms. J Struct Eng ASCE 2000;126(3):382_8.

Popa, R. “Cercetari privind elaborarea unor noi metode de testare a structurilor numerice”, Teza de doctorat , Universitatea "Dunarea de Jos" din Galati , 1998 .

Poterasu, V. F. & Florea, N., 1984. *Practica Optimizatii Structurilor.* Iasi: Editura Junimea.

Rahami H, Kaveh A, Gholipour Y. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. Eng Struct 2008;30(9):2360_9.

Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms. J Struct Eng ASCE 1992;118(5):1233_50.

Randy, L. H. & Sue, E. H., 2004. Practical Genetic Algorithms. New Jersey: Publeshed by John Wiley & Sons.

Robert McNeel & Associates, 2012. Available at: http://www.rhino3d.com/features/.

Robot, usa.autodesk.com/robot-structural-analysis-professional/

Roe¨ sset, Jose M. , Hon.M., James T. P. Yao, State of the Art of Structural Engineering, 2003

Romstad, K.,Wang,C.,"Optimum design of framed structures", J.Str.Div.ASCE 94, 1978.

Rozvany, G.,"Structural optimization", Springer-Verlag, Heidelberg, 1999.

Rutten D., Grasshopper, www.grasshopper3d.com

Saka MP, Kameshki ES. Optimum design of multi-story sway steel frames to BS 5950 using a genetic algorithm. În: Topping BHV, editor. Advances în engineering computational technology. Civil-Camp press; 1998. p. 135_41.

SAP 2000, www.csiberkeley.com/sap2000.

Schmidt, L.A., Fox, R.L., "An integrated approach to structural synthesis and analysis", AIAA J. 6, 1975.

Schmit, L. & R.H., M., 1963. Structural synthesis and design parameters. Hierarchy. s.l., Journal of the Struct. Division, Proceedings of the ASCE, 89:269–299, 1963..

Sivanandam, S. N. & Deepa, S. N., 2008. Introduction to Genetic Algorithms. Berlin: Springer-Verlag Berlin Heidelberg.

Smit, S. a. A. E., 2009. *Comparing parameter tuning methods for evolutionary algorithms. In IEEE Congress on Evolutionary Computation, pp.399–406,* s.l.: s.n.

Sousa, F. V. V. R. F., 2003. *Generalized Extremal Optimization for Solving Complex Optimal design Problems. Presented in: Genetic and Evolutionary Computation Conference (GECCO-2003), Chicago, Illinois USA, pp. 1-11,* s.l.: s.n.

Structural Design Of Retractable Roof Structures, IASS working group n°16, WIT Press, 2000

Teresko, John. 1993. "Parametric Technology Corp.: Changing the way Products are Designed." Industry Week, December 20.

Thompson, D. W., 1917. On Growth and Form. s.l.:Cambridge University Press.

Tsui, E., 1999. Evolutionary architecture: nature as a basis for design. N.Y.: John Wiley.

Turda, Dan. 2003. *Aspecte priving optimizarea structurilor reticulate*. Cluj-Napoca : Teza de doctorat.

Turkkan, N., 2003. *Discrete Optimization of Structures Using a Floating Point Genetic Algorithm, Proceedings of the Annual Conference of the Canadian Society for Civil Engineering,Moncton, Nouveau-Brunswick, Canada, pp. GCM-134-1/8,* s.l.: s.n.

V. Fiacco/G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques. XIV + 210 S. m. Fig. New York/London/Sydney/Toronto 1969.

Valery, V. V., 1999. *Optimal Design Theory and Applications to Materials and Structures.* Lancaster: Technomic .

Vanderplaats, G., 1984. Numerical Optimization Techniques for Engineering Design: with Applications. s.l.:McGraw-Hill: Series în Mechanical Engineering.

Wang D. Optimal shape design of a frame structure for minimization of maximum bending moment. Eng Struct 2007;29(8):1824_32.

Weisstein, Eric. 2003. CRC Concise Encyclopedia of Mathematics. Second edition. Boca Raton: Chapman & Hall/CRC.

X. Huang and Y.M. Xie, Evolutionary Topology Optimization of Continuum Structures: Methods and Applications, John Wiley & Sons, Chichester, England, 2010, 235 pp., ISBN: 9780470746530.

Y.M. Xie and Grant .P. Steven, 'A simple evolutionary procedure for structural optimization', Computers & Structures, Vol. 49, No. 5, pp 885-896, 1993.

Y.M. Xie and Grant .P. Steven, Evolutionary Structural Optimization, Springer-Verlag, London, June, 1997, xii+188 pp., ISBN 3-540-76153-5.

Y.M. Xie, Z.H. Zuo, X. Huang, J.W. Tang, B. Zhao and P. Felicetti, 'Architecture and urban design through evolutionary structural optimisation algorithms', Keynote Lecture of International Symposium on Algorithmic Design for Architecture and Urban Design, Tokyo, Japan, March 14-16, 2011, 11pp.

Zawlewski, Waclaw, and Edward Allen. Shaping Structures: Statics. New York: John Wiley & Sons, Inc., 1998.

LUCRARI PUBLICATE

ing. Ioana Dorina BALEA

Ioana D. Balea , Radu Hulea, Georgios E. Stavroulakis. „Implementation of Eurocode Load Cases în Optimization Problems of Steel Frames, Based on Genetic Algorithms”, February, 2013, Applied Mechanics and Materials, 310, 609, DOI 10.4028/www.scientific.net/AMM.310.609

Ioana D. Balea, Adina M. Popescu, Georgios E. Stavroulakis. „Parametric design and optimization of steel roof trusses”, 10th HSTAM International Congress on MechanicsChania, Crete, Greece, 25 – 27 May, 2013

Ioana D. Balea, Radu Hulea, Georgios E. Stavroulakis. „Discrete optimization approach for steel frames and trusses, based on genetic algorithms ” SEECCM III 3rd South-East European Conference on Computational Mechanicsan ECCOMAS and IACM Special Interest Conference, Kos Island, Greece, 12–14 June 2013

Popescu A. M, **Balea I. D**. “Steel multi-story structures with added damping. A consumption approach” 13th International Scientific Conference VSU’ 2012, 6-7 June, 2013, Sofia, Bulgaria- Proceedings.

Popescu A. M, **Balea I. D**. „Optimal Steel Multi-Storey Structures. Added Damping vs. Stiffness.” C60 International Conference, 7-9 November,2013, Cluj-Napoca, Romania.

TABELUL FIGURILOR: