

Genetický algoritmus jako prostředek pro optimalizaci v mechanice

Diplomová práce

Studijní program:

M2301 Strojní inženýrství

Studijní obor:

Aplikovaná mechanika – inženýrská mechanika

Autor práce:

Martin Hermann

Vedoucí práce:

Dr. Ing. Tomáš Hruš

Katedra mechaniky, pružnosti a pevnosti





Zadání diplomové práce

Genetický algoritmus jako prostředek pro optimalizaci v mechanice

Jméno a příjmení: **Martin Hermann**
Osobní číslo: S16000024
Studijní program: M2301 Strojní inženýrství
Studijní obor: Aplikovaná mechanika – inženýrská mechanika
Zadávací katedra: Katedra mechaniky, pružnosti a pevnosti
Akademický rok: **2020/2021**

Zásady pro vypracování:

- Popište a implementujte genetický optimalizační algoritmus
- Demonstrujte funkčnost a efektivitu genetického algoritmu na jednoduchých případech
- Formulujte optimalizační úlohu (kritérium a omezující podmínky) pro optimalizaci žebrování segmentu nosné konstrukce z hlediska její tuhosti a určete optimální žebrování
- Najděte optimální polohy magnetů tlumících kmitání sendvičového nosníku (viskoelastický magnetoreologický elastomer + krycí vrstvy z uhlíkového kompozitu) daných vlastností tak, aby byla první vlastní frekvence nosníku co nejvyšší
- Navrhněte experiment pro ověření výsledků optimalizace sendvičového nosníku
- U výše uvedených úloh diskutujte výsledky a možnosti využití jiných způsobů optimalizace

​

Rozsah grafických prací:
Rozsah pracovní zprávy: 40
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština



Seznam odborné literatury:

- [1] Randy L. Haupt, Sue Ellen Haupt: Practical Genetic Algorithms
- [2] Thomas Weise: Global Optimization Algorithms –Theory and Application –
- [3] R. Brepta, L. Půst, F. Turek: Mechanické kmitání
- [4] D.J. Mead, S. Markus: The Forced Vibration of a Three-Layer, Damped Sandwich Beam With Arbitrary Boundary Conditions

Vedoucí práce: Dr. Ing. Tomáš Hruš
Katedra mechaniky, pružnosti a pevnosti

Datum zadání práce: 31. října 2020
Předpokládaný termín odevzdání: 30. dubna 2022

prof. Dr. Ing. Petr Lenfeld
děkan

L.S.

doc. Ing. Iva Petříková, Ph.D.
vedoucí katedry

V Liberci dne 31. října 2020

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

7. června 2021

Martin Hermann

Genetický algoritmus jako prostředek pro optimalizaci v mechanice

Abstrakt

Diplomová práce demonstruje využití genetického algoritmu pro optimalizaci v mechanice. V práci je ukázaná optimalizace tvaru vrubu a vliv rozložení hmoty na tuhost součásti. V experimentální části je ukázaný vliv magnetického pole na vlastní frekvence sendvičového nosníku.

Klíčová slova: genetický algoritmus, optimalizace vrubu, rozložení hmoty, metoda konečných prvků, magnetické pole, vlastní frekvence

Genetic Algorithm as a Tool for Optimization in Mechanics

Abstract

Thesis demonstrates the use of a genetic algorithm for optimization in mechanics. This work shows the optimization of shape of the notch and the effect of mass distribution on the stiffness of the part. The experimental part shows the influence of the magnetic field on the eigenfrequency of the sandwich beam.

Keywords: genetic algorithm, optimization of notch, mass distribution, finite element method, magnetic field, eigenfrequency

Obsah

1	Teoretická část	8
1.1	Genetický algoritmus obecně	8
1.2	Inicializace počáteční generace a koloběh GA	9
1.3	Kriteriální funkce	9
1.4	Křížení	10
1.5	Mutace	11
1.5.1	Náhodná mutace	12
1.5.2	Řízená mutace	12
1.6	Selekce	12
2	Praktická část	13
2.1	Kriteriální funkce pomocí MKP	13
2.2	Optimalizace vrubu	13
2.2.1	Geometrie vrubu	13
2.2.2	Kriteriální funkce pro optimalizaci vrubu	14
2.2.3	Označení vrubů podle rozměrů	15
2.2.4	Počáteční generace pro vrub w40d10	15
2.2.5	Křížení	17
2.2.6	Mutace	19
2.2.7	Limity	20
2.2.8	Selekce	21
2.2.9	Průběh zdatnosti nejlepšího jedince	21
2.2.10	Poslední generace a nejlepší jedinec	22
2.2.11	Analýza křížení	23
2.2.12	Výsledky optimalizace vrubu	24
2.3	MKP v Matlabu	26
2.4	Vetknutý nosník zatížený spojitým zatížením	26
2.4.1	Geometrie nosníku	26
2.4.2	Síť a okrajové podmínky	27
2.4.3	Počáteční generace	28
2.4.4	Kriteriální funkce pro vetknutý nosník	28
2.4.5	Limity	29
2.4.6	Křížení	30
2.4.7	Mutace	30
2.4.8	Průběh svislého posuvu uzlu A	32
2.4.9	Výsledky optimalizace vetknutého nosníku	33
2.5	Nosník uložený v dolních rozích	34
2.5.1	Síť a okrajové podmínky	34
2.5.2	Kriteriální funkce pro nosník upevněný v dolních rozích	34
2.5.3	Průběh svislého posuvu uzlu A	35
2.5.4	Výsledky optimalizace pro nosník upevněný v dolních rozích	36
2.6	Předepsání funkčních posuvů	38
2.6.1	Síť a okrajové podmínky	38

2.6.2	Kriteriální funkce pro svorku	38
2.6.3	Výsledky optimalizace svorky	39
3	Experimentální část	41
3.1	Optimální polohy magnetů tlumících kmitání sendvičového nosníku	41
3.1.1	Výpočet vlastní frekvence nosníku bez působení magnetů	41
3.1.2	Síly od magnetů působící na nosník	43
3.1.3	Měření vlastních frekvencí	44
	Závěr	48
	Seznam použitých zdrojů	49

Úvod

Genetické algoritmy (dále jen GA) se dají použít pro optimalizaci v řadě oborů. Vhodné jsou pro určitý typ úloh, které mohou být charakterizovány i velkým počtem parametrů určujících řešenou úlohu. Tak jako pro ostatní optimalizační metody, je i pro GA nutno znát kritériální funkci, jejíž hodnotu se podle typu řešené úlohy snažíme minimalizovat nebo maximalizovat. Tato práce pojednává o použití GA při optimalizaci v mechanice. Pomocí metody konečných prvků (dále jen MKP) se získá hodnota kritériální funkce, která je odvozená od posuvů uzlů a napětí v prvcích.

Prvním příkladem, na kterém je ukázaná aplikace GA, je optimalizace tvaru vrubu za účelem minimalizace součinitele koncentrace napětí. Tvar vrubu byl popsán Bézierovou křivkou a lomenou čarou.

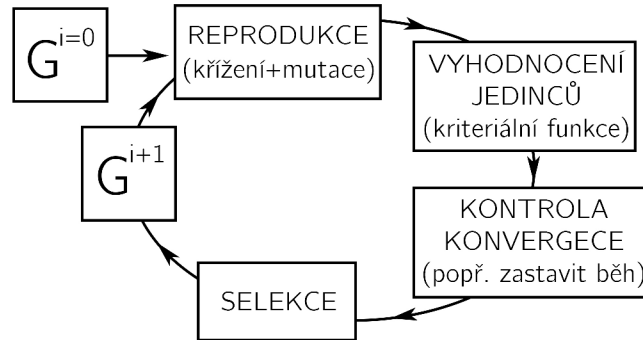
V práci je dále uvedena optimalizace rozložení hmoty za účelem zvýšení tuhosti žebrovaných segmentů (minimalizace posuvů) a předepsání funkčních posuvů (maximalizace posuvů).

V poslední části je ukázaný vliv magnetického pole na vlastní frekvence sendvičového nosníku (viskoelastický magnetoreologický elastomer + krycí vrstvy z uhlíkového kompozitu). Cílem bylo nalezení optimální polohy magnetů za účelem zvýšení první vlastní frekvence nosníku.

1 Teoretická část

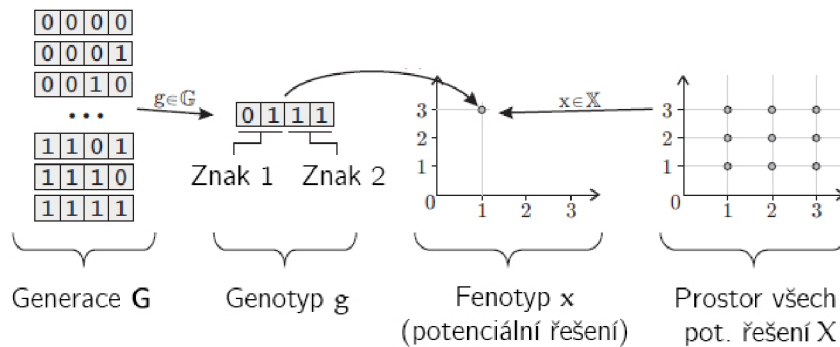
1.1 Genetický algoritmus obecně

Genetický algoritmus patří do skupiny evolučních algoritmů. Tyto optimalizační metody jsou inspirovány evoluční biologii - využívají procesy jako křížení, mutaci, přirozený výběr a přežití nejzdatnějších jedinců za účelem iterativního zlepšování sady možných řešení dané úlohy [1]. Obecné schéma GA je znázorněno na obrázku 1.



Obrázek 1: Koloběh genetického algoritmu. V každé iteraci vznikne nová generace G^i .

Každá generace G je tvořena jedinci. Každý jedinec je charakterizovaný uspořádanou n -ticí znaků (genotyp). Znaky představují hledané parametry (fenotyp). Vztah mezi genotypem a fenotypem je vždy známý. Například na obrázku 3 je vztah mezi genotypem a fenotypem daný převodem z dvojkové do desítkové soustavy.



Obrázek 2: Vztah mezi genotypem a fenotypem, převzato z [2].

Každému jedinci je pomocí kriteriální funkce přiřazena zdatnost (fitness). Fitness je hodnota, podle které porovnáváme jedince mezi sebou. Zvolený počet jedinců s nejlepší zdatností vstoupí do nové generace a stanou se potenciálními rodiči.

Křížením a mutací rodičů vzniknou potomci, kteří představují nové potenciální řešení úlohy.

Preferování lepších jedinců při křížení a přežití nejlepšího z nich zaručí směr k optimu. Není ale zaručené, že se nalezne globální optimum. Pokud si začnou být jedinci příliš podobní, hrozí nalezení pouze lokálního optima. Je vhodné vytvářet mutací atypické jedince, kteří pomohou lépe zmapovat prostor všech řešení. To pomáhá k nalezení globálního optima.

1.2 Inicializace počáteční generace a koloběh GA

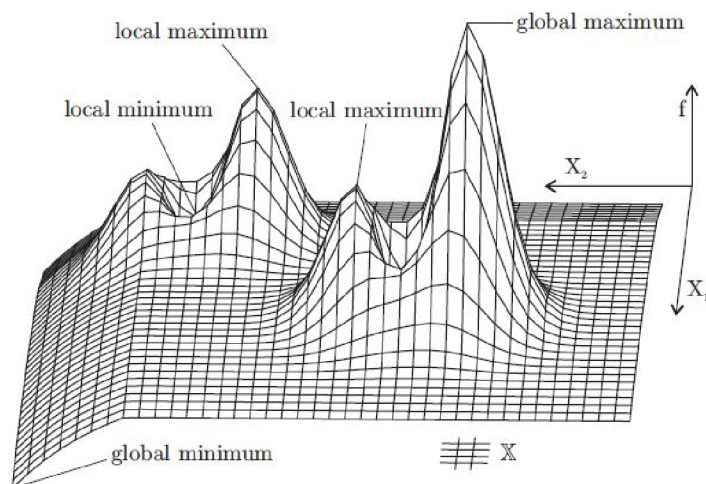
Před spuštěním GA je potřeba vytvořit počáteční generaci, která do algoritmu vstupuje. Tu vytvoříme z jedinců, jejichž znaky se buď náhodně vygenerují nebo zvolí na základě předchozích výpočtů a zkušeností. Počáteční generaci reprezentuje matice \mathbf{G}^0 . Každý řádek je tvořen znaky, které představují právě jednoho jedince. Jedinci reprodukci vytvoří potomky. Ti jsou vyhodnoceni kritériální funkcí, každému je přiřazen fitness. Na základě fitness dojde k selekci nejlepších, kteří vytvoří novou generaci \mathbf{G}^{i+1} , která vstupuje do další iterace.

V ideálním případě tento koloběh zastaví uspokojivá hodnota kritériální funkce. V opačném případě se cyklus ukončí při dosažení nastaveného počtu generací.

1.3 Kritériální funkce

Vstupem (proměnnou) kritériální funkce jsou parametry, které jsou reprezentovány znaky jedince. Hodnota kritériální funkce je informace o zdatnosti jedince - fitness jedince. Pomocí kritériální funkce zjistíme fitness jednotlivých jedinců. Podle hodnoty kritériální funkce selektujeme jedince a ovlivňujeme jejich reprodukci. Reprodukujeme nejlepší jedince a naopak redukujeme nejhorší (viz kapitola Selektce).

Cílem optimalizace je najít optimum. Optimum je globální minimum nebo maximum kritériální funkce (podle toho, čeho chceme optimalizací docílit).

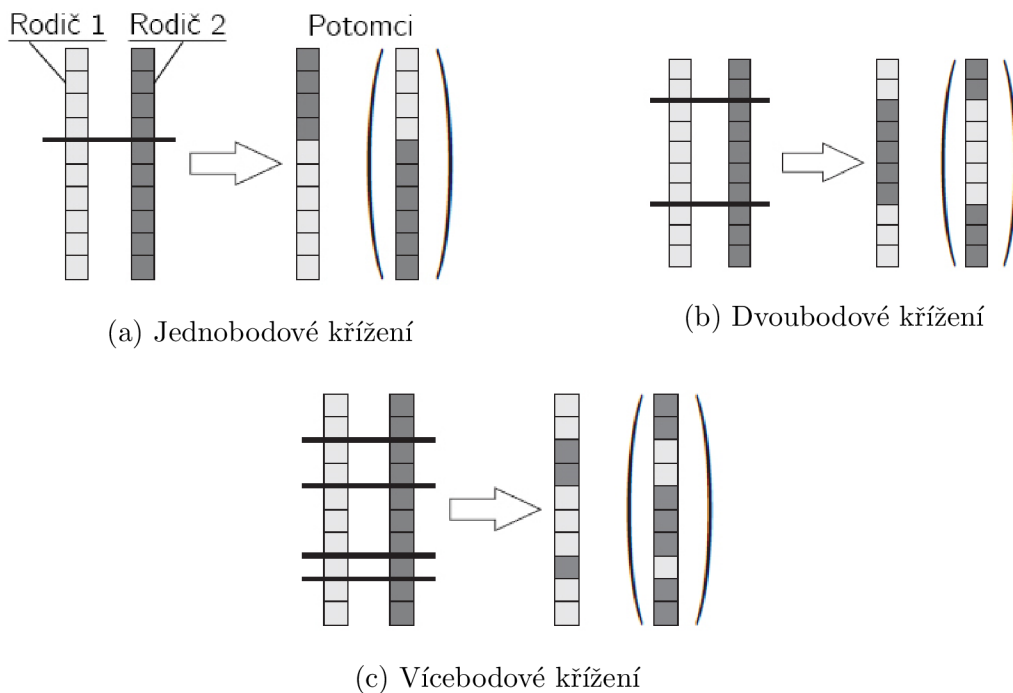


Obrázek 3: Globální a lokální optima pro dvoudimenzionální kritériální funkci, převzato z [3].

Je potřeba brát v potaz, že genetický algoritmus může nalézt lokální optimum. Celá generace se může vyskytovat v okolí lokálního optima a svojí reprodukcí se z něj nedostane. Je tedy potřeba dobře zmapovat celý prostor, na kterém je kritériální funkce definovaná. Toho lze docílit různými počátečními generacemi. Ve zmapování nám také pomůže mutování jedinců [4].

1.4 Křížení

Křížení je jedním z operátorů tvořících nové jedince. Jako operátor operuje s jedinci a tvoří z nich potomky. Křížení je inspirováno biologickými procesy. Vybraní jedinci si mezi sebou vymění znaky. Potomek je vytvořený ze znaků jedinců, mezi kterými proběhlo křížení. Tři základní způsoby křížení jsou ilustrovány na obrázku 4.



Obrázek 4: Křížení jedinců se stejným počtem znaků, převzato z [5].

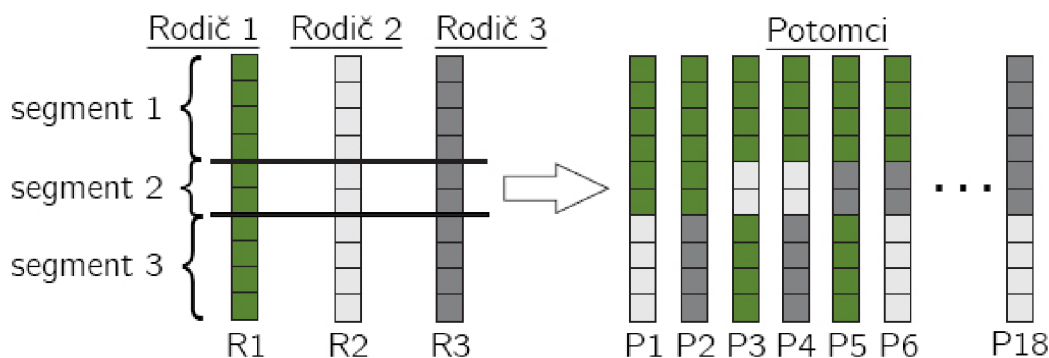
Při jednobodovém křížení se každý z rodičů rozdělí na dvě části. Místo rozdělení je náhodně vybráno. Potomek vznikne spojením první části z prvního rodiče a druhé části z druhého rodiče, popř. opačně viz obr. 4a.

Dvoubodové křížení rozdělí znaky vybraných jedinců na tři části. Je tedy zapotřebí náhodně vybrat dvě místa rozdělení. Potomek vznikne spojením první a třetí části z prvního rodiče a prostřední části z druhého rodiče viz obr. 4b.

Zobecněním výše uvedených metod křížení vznikne vícebodové křížení. Náhodně se vybere n -míst rozdělení. Počet rozdělení se může měnit, ale pro dva sobě příslušné rodiče musí být rozdělení identické [5].

Párování jedinců probíhá buď náhodně, a nebo na základě zdatnosti jedince. Jedinci s lepší zdatností mají větší pravděpodobnost stát se potencionálním rodičem (více v kapitole Selekcce).

Výše uvedené metody křížení vyžadují dva rodiče se stejným počtem znaků. Z dvou rodičů lze při daném místě rozdělení vytvořit pouze dva unikátní jedince. Pro jednobodové křížení je počet rodičů vždy dva. Pro dvou a vícebodové křížení lze metody modifikovat pro větší počet rodičů. Takové křížení jsme nazvali kombinované n-bodové křížení. Tím se zvýší počet unikátních jedinců, které lze vytvořit z daného počtu rodičů. Na obr. 5 je znázorněno kombinované dvoubodové křížení pro tři rodiče.



Obrázek 5: Ukázka kombinovaného dvoubodového křížení.

Rodiče jsou rozděleni na tři segmenty. Potomek musí obsahovat segmenty alespoň ze dvou rodičů. Tím se docílí kombinovaného křížení. Jednak vznikají jedinci, kteří by vznikli ze dvou rodičů jednobodovým křížením (např. potomci P1, P2, P18...) nebo dvoubodovým křížením (např. potomci P3, P5). Navíc vznikají jedinci ze tří rodičů (např. potomci P4, P6).

1.5 Mutace

Mutace je dalším operátorem důležitým při reprodukci jedinců. Mutace následuje po křížení. Kdybychom tvořili potomky pouze křížením, byly by znaky potomků omezeny pouze znaky jedinců z počáteční generace.

Znaky mění zásluhou mutace svou hodnotu. To napomáhá k mapování stavového prostoru kritériální funkce. Dává to prostor k objevování nových potenciálních řešení.

Do následující generace zpravidla vstupuje nemutovaný nejlepší jedinec. Nemutovaný vstupuje do další generace jako dosavadní nejlepší řešení úlohy. Ukazuje se ale výhodné, pokud se přidá nemutovaný nejlepší jedinec, přidat do následující generace i zvolený počet mutací nejlepšího jedince. Mutace nejlepšího jedince prozkoumávají jeho blízké okolí.

Mutace může probíhat náhodně, nebo řízeně na základě zdatnosti jedinců vytvořených za tímto účelem.

1.5.1 Náhodná mutace

Při náhodné mutaci se znaky potomka vytvořeného křížením změni o náhodně vygenerované procento své původní hodnoty. Mutování je určeno procentem mutovaných znaků a dovoleným rozsahem mutace. Rozsah mutace je daný minimální a maximální možnou velikostí mutace.

Dovolený rozsah mutace může být po celou dobu běhu genetického algoritmu neměnný, nebo se může měnit podle fáze algoritmu. V konečné fázi, kdy je generace v okolí optima, je vhodné rozsah mutace zmenšit. Tím se zrychlí nalezení optima.

Existuje ale více metod, jak mutaci provádět. Hodnotu zmutovaného znaku můžeme získat například jako lineární kombinaci znaků od rodičů.

1.5.2 Řízená mutace

Mutace může být řízená na základě zdatnosti jedinců z předešlé generace. Dá se tedy použít až od druhé generace. Pro řízenou mutaci je potřeba uměle vytvořit testovací jedince, pomocí kterých určíme váhu jednotlivých znaků. Každý z testovacích jedinců odpovídá právě jednomu znaku.

Testovací jedinci vychází z nejlepšího jedince předešlé generace. Tomu se změni hodnota určeného znaku. První testovací jedinec je vytvořený z nejlepšího jedince předešlé generace, kterému se změnila hodnota prvního znaku. Počet testovacích jedinců je roven počtu znaků jednoho jedince. Každému z testovacích jedinců se změni hodnota příslušného znaku o stejnou hodnotu. Porovnáním zdatnosti testovacích jedinců zjistíme váhu jednotlivých znaků. To znamená, že zjistíme, jaký znak při zvolené změně nejvíce ovlivní hodnotu kritériální funkce.

Zvolený počet znaků je řízeně mutovaný na základě vah znaků. Hodnota znaku se zvětší nebo zmenší v závislosti na tvorbě testovacích jedinců (podle toho, jestli se znaky testovacích jedinců měnily o kladnou nebo zápornou hodnotu).

Vhodnost použití řízené mutace závisí na druhu řešené úlohy. Doba, za kterou uběhne jedna iterace, se při použití řízené mutace prodlouží, protože je třeba vyhodnotit sadu testovacích jedinců. Tím ale získáme informaci o váze jednotlivých znaků.

1.6 Selekcce

Selekcce se provádí po křížení, mutování a přiřazení fitness jedincům. Jedinci se seřadí od nejlepších po nejhorší. Zvolený počet nejlepších postupuje do další generace. Jedinci, kteří nepostoupí do další generace jsou nahrazeni potomky.

Při selekci se k sobě přiřazují rodiče, kteří vytvoří potomky. Párování rodičů může probíhat náhodně nebo na základě zdatnosti jedinců. Zdatnější jedinci mohou mít více potomků. Při zohlednění zdatnosti mají jedinci s větší zdatností větší pravděpodobnost vytvořit potomka.

2 Praktická část

V následujících úlohách je ukázána aplikace GA při minimalizaci napětí (koncentrace napětí ve vrubu), minimalizace posuvů (žebrování ohýbaných segmentů) a předsání funkčních posuvů (sevření čelistí - svorka)

2.1 Kriteriaální funkce pomocí MKP

Pro minimalizaci deformace a napětí nebo předsání funkční deformace je metoda konečných prvků pro výpočet hodnoty kriteriaální funkce vhodným nástrojem. Kriteriaální funkci lze definovat jako funkci posuvů v uzlech a napětí v prvcích vynásobenými příslušnými vahami:

$$F = f(\vec{u}, \vec{\sigma}, \vec{V}^u, \vec{V}^p) = \sum u_i V_i^u + \sum \sigma_k V_k^p, \quad (1)$$

kde \vec{u} - vektor posuvů uzlů,
 $\vec{\sigma}$ - vektor napětí v prvcích,
 \vec{V}^u - vektor vah pro jednotlivé uzly,
 \vec{V}^p - vektor vah pro jednotlivé prvky.

2.2 Optimalizace vrubu

Vrub funguje na součásti jako koncentrátor napětí, tedy ve vrubu a jeho blízkém okolí vzniká vysoké napětí (násobně vyšší oproti nominálnímu napětí), které snižuje únostnost a životnost součásti. Velikost maximálního napětí ve vrubu závisí na jeho tvaru. Vypovídající informace o vrubu je součinitel koncentrace napětí α , který je dán poměrem maximálního a nominálního napětí

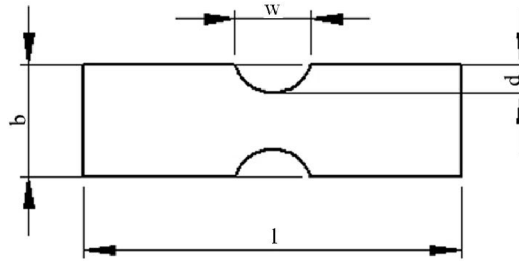
$$\alpha = \frac{\sigma_{max}}{\sigma_{nom}}. \quad (2)$$

Cílem je minimalizovat tento součinitel při zachování hloubky a šířky vrubu. Pro posouzení kvality vrubu porovnáváme hodnotu součinitele koncentrace napětí optimalizovaného vrubu s hodnotou součinitele vrubu referenčního. Jako referenční vrub je brán vrub kruhový. V této úloze je použit dvojí způsob popisu geometrie vrubu - Bézierova křivka a lomená čára.

Genetický algoritmus pro optimalizaci vrubu byl napsán v Matlabu, kriteriaální funkce byla implementovaná ve FreeFemu++ [6].

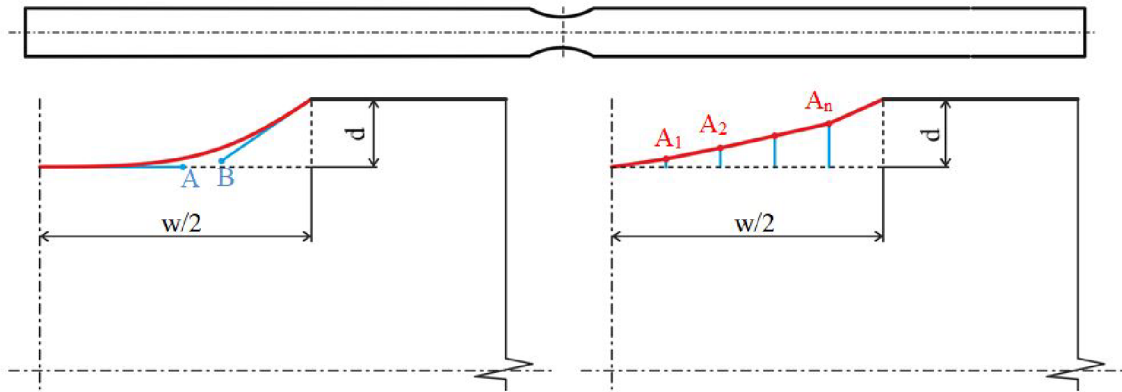
2.2.1 Geometrie vrubu

Před spuštěním optimalizačního algoritmu je potřeba určit parametry, které udávají tvar vrubu, který chceme měnit. Úloha je určena čtyřmi rozměry (plochá tyč s vrubem viz obr. 6). Tyč je určena svojí délkou a šířkou. Vrub určuje jeho šířka a hloubka.



Obrázek 6: Rozměry ploché tyče s vrubem.

Geometrii samotného vrubu jsme popsali dvěma způsoby. První variantou je Bézierova křivka, která díky využití symetrie definuje tvar vrubu pouze třemi parametry A_x, B_x, B_y . Bod A leží na tečném ke kořeni vrubu, souřadnice A_y je určena hloubkou vrubu (viz. obr. 7). Druhá varianta je lomená čára, při které počet zalomení odpovídá počtu parametrů A_n .



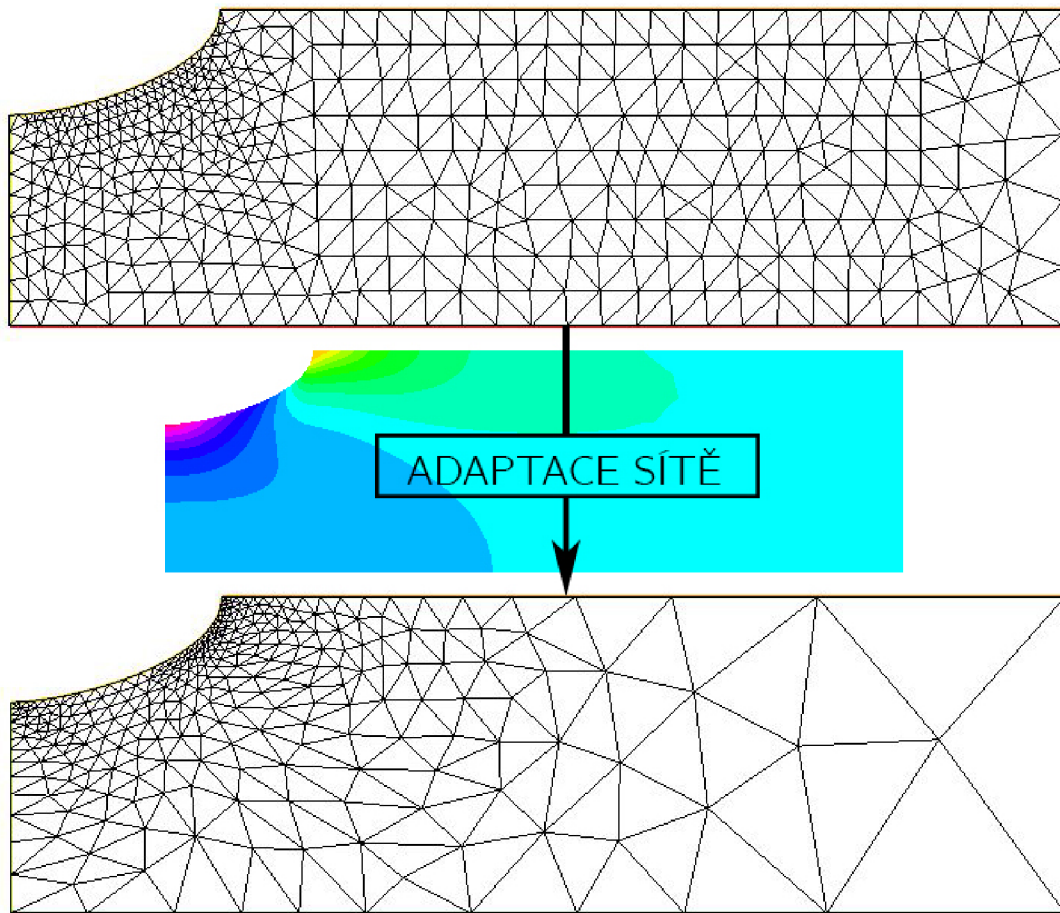
Obrázek 7: Geometrie vrubu.

2.2.2 Kriteriaální funkce pro optimalizaci vrubu

Hodnota kriteriaální funkce odpovídá součiniteli koncentrace napětí (2). Ekvivalentní napětí je určeno podle energetické hypotézy HMM.

Pro tuto úlohu byla kriteriaální funkce implementována v softwaru FreeFem++. Jedná se o programovací jazyk a software napsaný v jazyce C++, který je zaměřený na řešení parciálních diferenciálních rovnic metodou konečných prvků.

Pro přesnější výsledky (na úkor delší doby výpočtu) umí software FreeFem++ adaptovat síť. Pomocí prvotní sítě se spočítají požadované veličiny a na základě gradientu stanovené veličiny se síť adaptuje. V místech s velkým gradientem se síť zjemní. V místech malé změny síť zhrubne. Adaptace sítě podle gradientu ekvivalentního napětí HMM je znázorněna na obr. 8.



Obrázek 8: Adaptace sítě na základě gradientu napětí.

2.2.3 Označení vrubů podle rozměrů

V následující tabulce je přehled označení vrubů na základě jejich rozměrů. Vrub je charakterizován svojí hloubkou a šířkou.

	Hloubka $d = 5$ mm	Hloubka $d = 10$ mm	Hloubka $d = 15$ mm
Šířka $w = 40$ mm	w40d5	w40d10	w40d15
Šířka $w = 50$ mm	w50d5	w50d10	w50d15
Šířka $w = 60$ mm	w60d5	w60d10	w60d15

Tabulka 1: Označení vrubů podle rozměrů

2.2.4 Počáteční generace pro vrub w40d10

Na následujících stránkách jsou ukázány parametry a běh GA pro vrub w40d10 popsaný Bézierovou křivkou. Šířka vrubu $w = 40\text{mm}$ a jeho hloubka $d = 10\text{mm}$.

Matice počáteční generace pro tento výpočet obsahuje pouze 8 jedinců. Každý řádek představuje jedince tvořeného třemi uspořádanými znaky, které odpovídají normalizovaným souřadnicím B_{xn}, B_{yn}, A_{xn} . Tyto souřadnice určují polohu řídicích bodů Bézierovy křivky. Použitím normalizovaných souřadnic se zajistí, že je každý znak omezen intervalem $(0,1)$ nezávisle na rozměrech vrubu a tyče.

Vztahy mezi skutečnými souřadnicemi řídicích bodů Bézierovy křivky a normalizovanými souřadnicemi (hodnotou znaků) jsou dány rozměry vrubu a tyče:

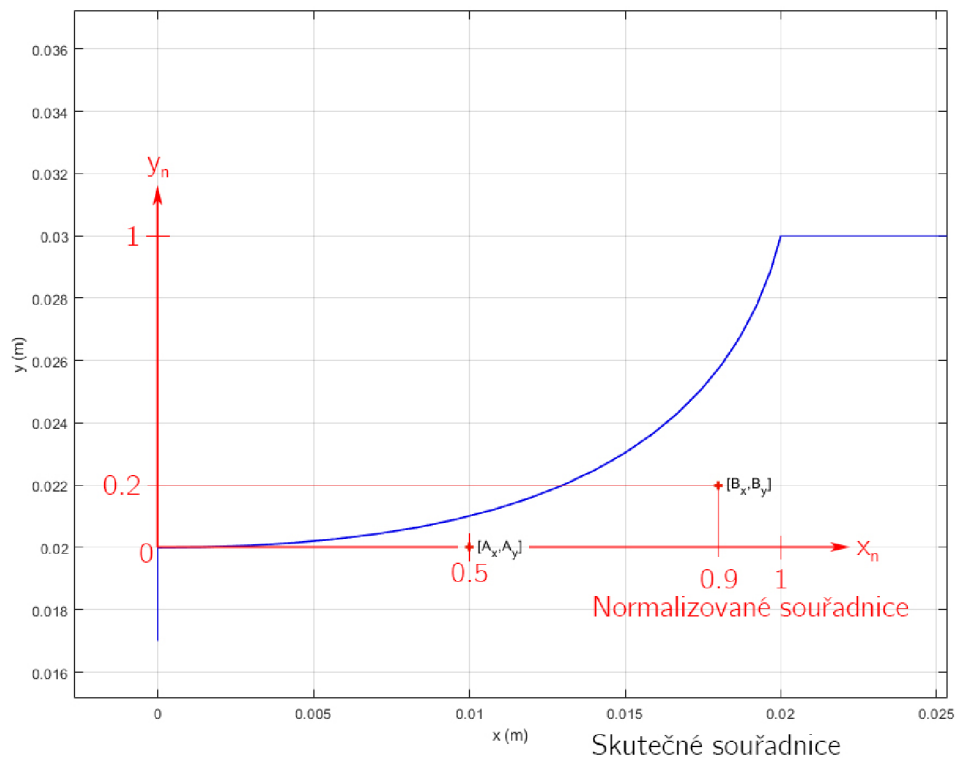
$$B_x = B_{xn} \cdot \frac{w}{2}, \quad (3)$$

$$B_y = \left(\frac{b}{2} - d\right) + B_{yn} \cdot d, \quad (4)$$

$$A_x = A_{xn} \cdot \frac{w}{2}, \quad (5)$$

w - šířka vrubu,
kde b - šířka tyče,
 d - hloubka vrubu.

Například jedinec $[B_{xn} = 0.9 \quad B_{yn} = 0.2 \quad A_{xn} = 0.5]$ představuje skutečné souřadnice řídicích bodů $[B_x = 0.018 \quad B_y = 0.022 \quad A_x = 0.01]$. Tvar vrubu určen těmito souřadnicemi je znázorněn na obr. 9.



Obrázek 9: Vrub reprezentovaný jedincem $[0.9 \ 0.2 \ 0.5]$. Skutečné souřadnice jsou $[0.018 \ 0.022 \ 0.01]$.

Znaky počáteční generace se náhodně vygenerují z intervalu (0,1). Ukázka náhodně vygenerované generace obsahující osm jedinců:

$$\mathbf{G}^0 = \begin{bmatrix} 0.8147 & 0.9575 & 0.4218 \\ 0.9058 & 0.9649 & 0.9157 \\ 0.1270 & 0.1576 & 0.7922 \\ 0.9134 & 0.9706 & 0.9595 \\ 0.6324 & 0.9572 & 0.6557 \\ 0.0975 & 0.4854 & 0.0357 \\ 0.2785 & 0.8003 & 0.8491 \\ 0.5469 & 0.1419 & 0.9340 \end{bmatrix}.$$

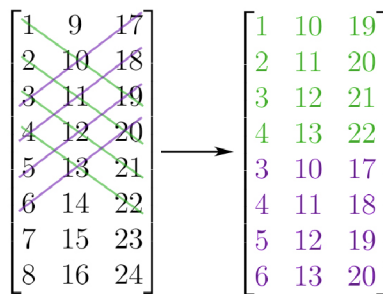
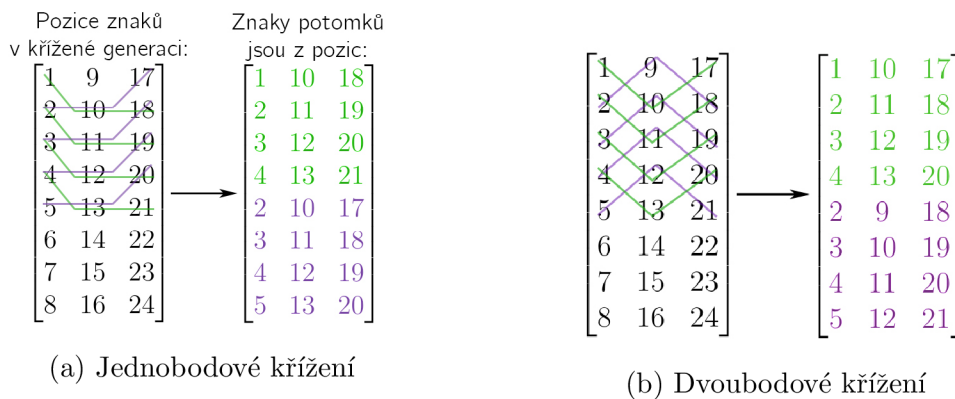
Výhodou použití normalizovaných souřadnic je možnost uplatnění počáteční generace pro jakýkoliv rozměry vrubu a tyče. Toho se využije, pokud počáteční generace není náhodně vygenerovaná, ale je vytvořena na základě předchozích výpočtů a zkušeností. Možnost využít stejnou matici počáteční generace pro libovolné rozměry vrubu je hlavním důvodem pro zavedení normalizovaných souřadnic.

Po optimalizaci jednoho vrubu mohou být využiti nejlepší jedinci pro tvorbu počáteční generace, která může být využita při optimalizaci vrubů jiných rozměrů.

2.2.5 Křížení

V této úloze bylo použito jednobodové, dvoubodové a kombinované křížení popsané na str. 10. Kombinované křížení v tomto případě vytváří potomky spojením znaků od třech odlišných jedinců. Každá z metod křížení vytvoří osm potomků. Díky malému počtu jedinců a znaků lze schéma křížení napsat ručně.

Způsob tvorby potomků různými metodami křížení je ukázáno na obr. 10.



(c) Kombinované křížení

Obrázek 10: Schéma křížení různými metodami. Schéma znázorňuje, z jakých pozic pochází znaky tvořící potomka. Potomci se tvoří skládáním znaků podle schémat křížení. Zelené a fialové čáry spojují pozice znaků, z kterých jsou potomci složeni. Například první jedinec vytvořený jednobodovým křížením se skládá ze znaků, které se v matici křížené generace vyskytují na pozicích 1, 10, 18.

V následujícím skriptu je ukázána funkce provádějící křížení. Jejím vstupem je matice generace, výstupem je nová nezmutovaná generace.

```

1 function new_gen = krizeni(G)
2 sp=[1 10 18; 2 11 19;3 12 20;4 13 21;2 10 17;3 11 18;4 12
   19;5 13 20];
3 dp=[1 10 17;2 11 18;3 12 19;4 13 20;2 9 18;3 10 19;4 11 20;5
   12 21];
4 komb=[1 10 19;2 11 20;3 12 21;4 13 22;3 10 17;4 11 18;5 12
   19;6 13 20];
5 p=[sp ; dp ; komb ];
6 for i=1:size(p,1)
7     rada=p(i, :);
8     a=G(rada(1));
9     b=G(rada(2));
10    c=G(rada(3));
11    new_gen(i, :)=[a b c];
12 end
13 end

```

2.2.6 Mutace

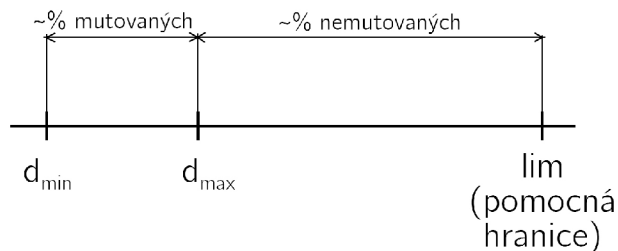
V této úloze byla využita pouze náhodná mutace. Mutace byla provedena pomocí matice mutace \mathbf{M}_{mut} . Rozměr této matice je totožný s rozměrem matice generace po křížení a její prvky m_{ij} představují hodnotu, kterou je znak na příslušné pozici vynásoben. Hodnota m_{ij} je náhodně vygenerovaná (s ohledem na dolní a horní mez). Znak na i -tém řádku a j -tým sloupci je vynásoben hodnotou m_{ij} .

Před tvorbou matice mutace si musíme zvolit dolní a horní mez mutace, tedy minimální a maximální možnou mutaci (d_{min} , d_{max}). Dále zvolíme, kolik procent znaků bude mutováno ($proc_mut$). Pomocí těchto parametrů mutace vypočítáme hodnotu pomocné hranice (lim), která slouží pouze pro dodržení procenta mutovaných znaků. Z obr. 11 se dá vyjádřit procento mutovaných:

$$proc_mut = \frac{d_{max} - d_{min}}{lim - d_{min}}. \quad (6)$$

Kolik procent znaků se bude mutovat si zvolíme my, $proc_mut$ je pro nás tedy známá a ze vztahu 6 odvodíme hodnotu pomocné hranice:

$$lim = \frac{d_{max} - d_{min}}{proc_mut} + d_{min}. \quad (7)$$



Obrázek 11: Parametry mutace znázorněné na číselné ose.

Nejdříve jsou hodnoty generovány v intervalu od minimální dovolené mutace do hodnoty pomocné hranice. Hodnota pomocné hranice vyplývá z rozsahu mutace a procenta mutovaných znaků podle vztahu 7. Po vygenerování matice se ve *for* cyklu projedou všechny prvky a ty z větší hodnotou než je maximální dovolená mutace se přepíší na jedničku. Počet jedniček odpovídá počtu nemutovaných znaků. Pro každou generaci se generuje jiná matice mutace.

Funkce napsaná v Matlabu generující matici mutace je ukázána v následujícím skriptu.

```
1 function M = mutace(rozmer1 , rozmer2)
2 dmin = 0.96; %Minimalni dovolena mutace
3 dmax = 1.04; %Maximalni dovolena mutace
4 proc_mut=0.4; %Procento znaku, ktere budou mutovany
5 lim=(dmax-dmin)/proc_mut+dmin; %Pomocna hranice
```



```

6 M = (lim-dmin) .* rand(rozmer1 , rozmer2)+dmin; %Generovani
   hodnot v rozsahu (dmin, lim)
7 for i=1: numel(M) %Projede vsechny prvky matice M
8     if M(i)>dmax
9         M(i)=1;
10        end
11 end
12 end

```

Parametry mutace ve skriptu odpovídají použitým parametrům při výpočtu této úlohy. Mutuje v průměru 40% znaků, maximální změna o 4% původní hodnoty znaku.

Pro ilustraci je uvedena část jedné z vygenerovaných matic mutace:

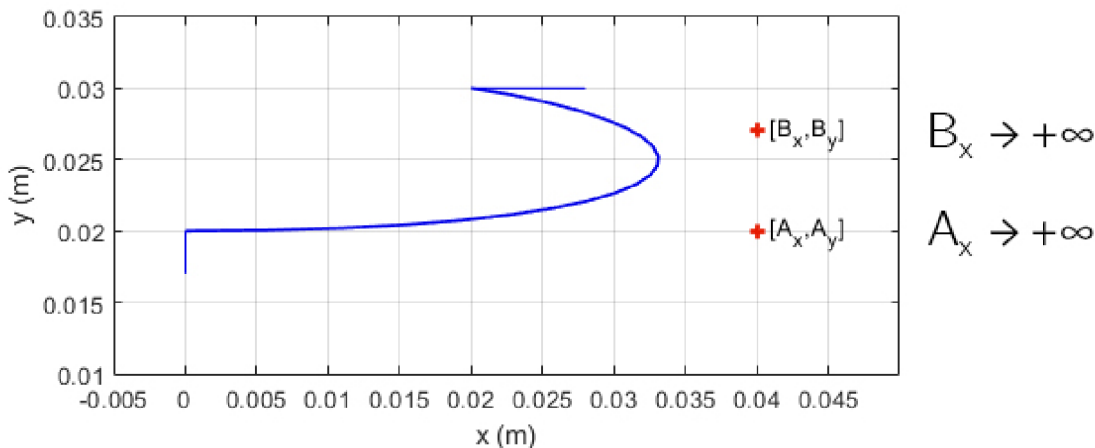
$$\mathbf{M}_{\text{mut}} = \begin{bmatrix} 1.0000 & 1.0384 & 0.9838 \\ 1.0000 & 1.0000 & 1.0000 \\ 0.9854 & 0.9942 & 1.0000 \\ 1.0000 & 1.0000 & 1.0281 \\ 1.0000 & 0.9664 & 1.0000 \\ 0.9795 & 1.0154 & 1.0048 \\ \vdots & \vdots & \vdots \\ 1.0000 & 0.9794 & 1.0110 \end{bmatrix} .$$

Do nové matice generace se po křížení a mutaci ještě přidávají tři mutace nejlepšího jedince z předešlé generace a nemutovaný nejlepší jedinec z předešlé generace.

2.2.7 Limity

Po křížení a mutaci je zapotřebí zkontrolovat, jestli se některý ze znaků neocitl mimo interval (0,1). Tento interval je dán použitím normalizovaných souřadnic. Pokud se hodnota znaku ocitne mimo dovolený interval, vrátí se hodnota znaku na dovolenou hranici, kterou překročila.

Rostoucí hodnota souřadnic A_x a B_x v průběhu optimalizace je názorným příkladem toho, co se stane při nedodržení dovoleného intervalu (0,1). Pokud by nebyly hodnoty znaků omezeny, rostly by do nekonečna. Tím by se vrub začal "odstraňovat". Již po pár generacích se začne vrub vytrácet. Začne vznikat tyč konstantního průřezu, který odpovídá nejužšímu místu. Vrub optimalizovaný bez ohledu na dovolený interval je ukázán na obr. 12.



Obrázek 12: Nežádoucí tvar vrubu způsobený nezohledněním limitů. Souřadnice A_x a B_x by bez limitů rostly do nekonečna.

Takováto optimalizace bez zohlednění omezujících podmínek ztrácí smysl. Je potřeba dbát na dovolené limity, aby se zachoval vrub požadovaných rozměrů.

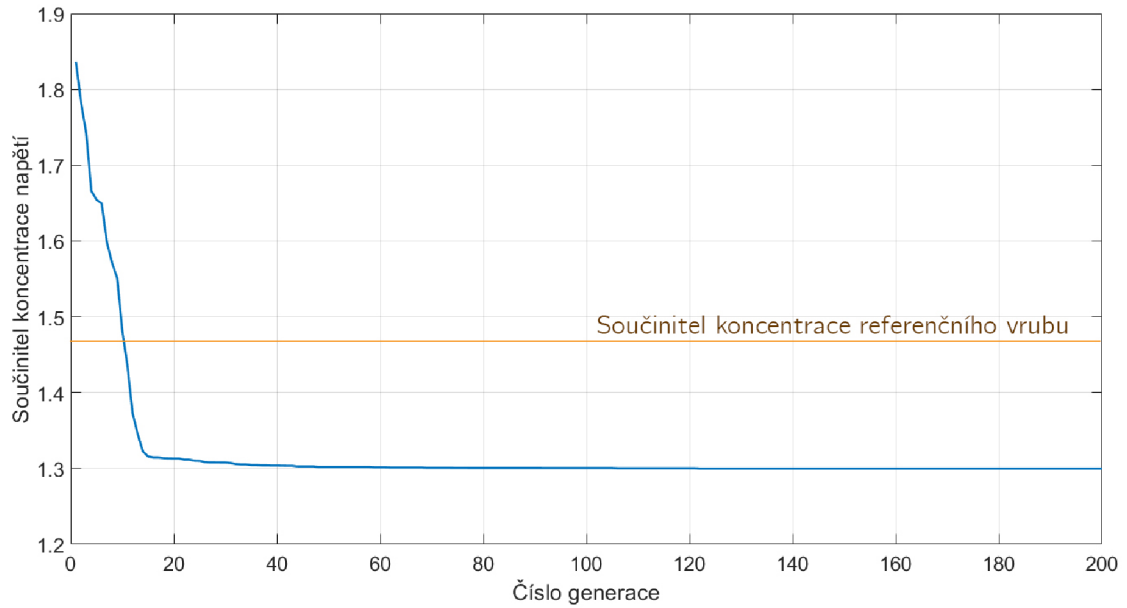
2.2.8 Selektce

Křížením a mutací vytvoříme novou generaci, která obsahuje 28 jedinců. Jedním z jedinců je nejlepší jedinec z předešlé generace. 24 jedinců jsme vytvořili křížením a tři jedince jsme vytvořili mutací nejlepšího jedince z předešlé generace.

Po přiřazení zdatnosti jedincům se jedinci seřadí od nejlepšího. Osm nejlepších postupuje do další iterace.

2.2.9 Průběh zdatnosti nejlepšího jedince

Fitness nejlepšího jedince je daný nejnižším součinitelem koncentrace napětí v dané generaci. Průběh zdatnosti nejlepšího jedince pro vrub w40d10 s výše uvedenými parametry GA je uveden na obr. 13.



Obrázek 13: Průběh zdatnosti nejlepšího jedince. Z průběhu je patrné, že k výraznému snížení součinitele koncentrace napětí došlo v prvních 20 generacích. Oranžová čára značí hodnotu součinitele koncentrace referenčního vrubu. Algoritmus byl ukončen generací číslo 200.

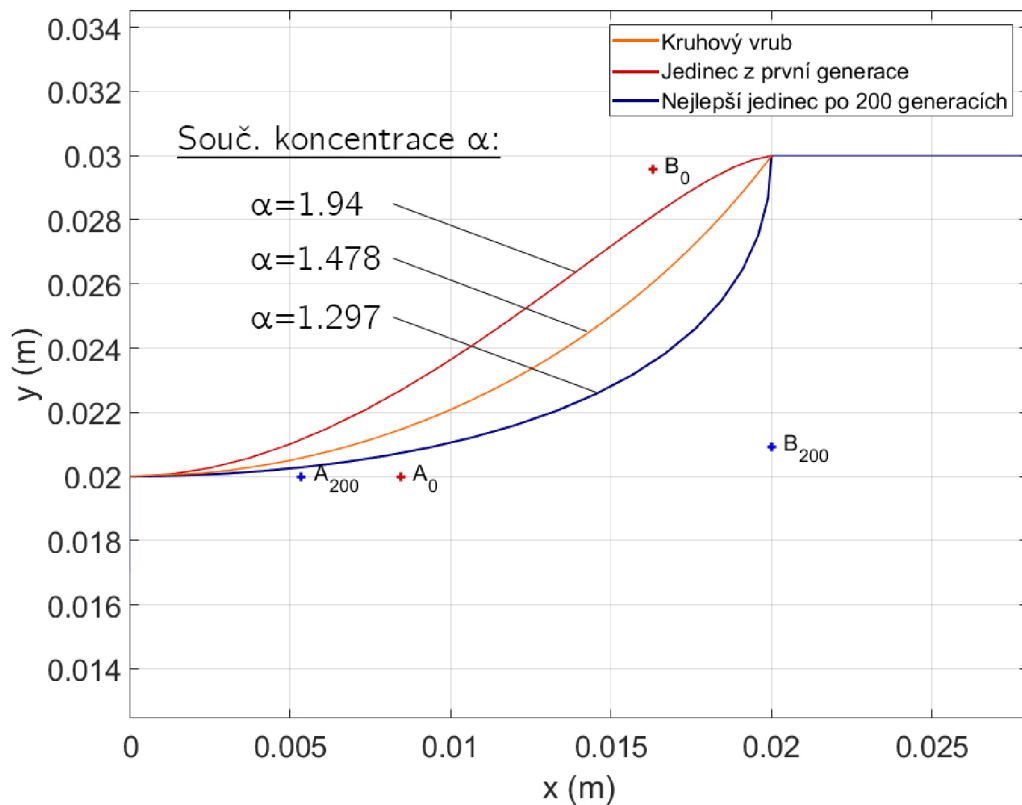
2.2.10 Poslední generace a nejlepší jedinec

Po ukončení algoritmu získáme matici poslední generace, která je po selekci seřazena podle fitness:

$$G^{200} = \begin{bmatrix} 1 & 0.0924 & 0.2671 \\ 1 & 0.0910 & 0.2650 \\ 1 & 0.0945 & 0.2778 \\ 1 & 0.0734 & 0.2336 \\ 0.9902 & 0.0731 & 0.2388 \\ 1 & 0.0960 & 0.2696 \\ 1 & 0.0903 & 0.2618 \\ 1 & 0.0954 & 0.2675 \end{bmatrix}.$$

Jedinci jsou v generaci seřazení podle své zdatnosti. Ti s lepší zdatností jsou výše. Nejlepší jedinec je na prvním řádku a je charakterizován znaky $[1 \ 0.0924 \ 0.2671]$.

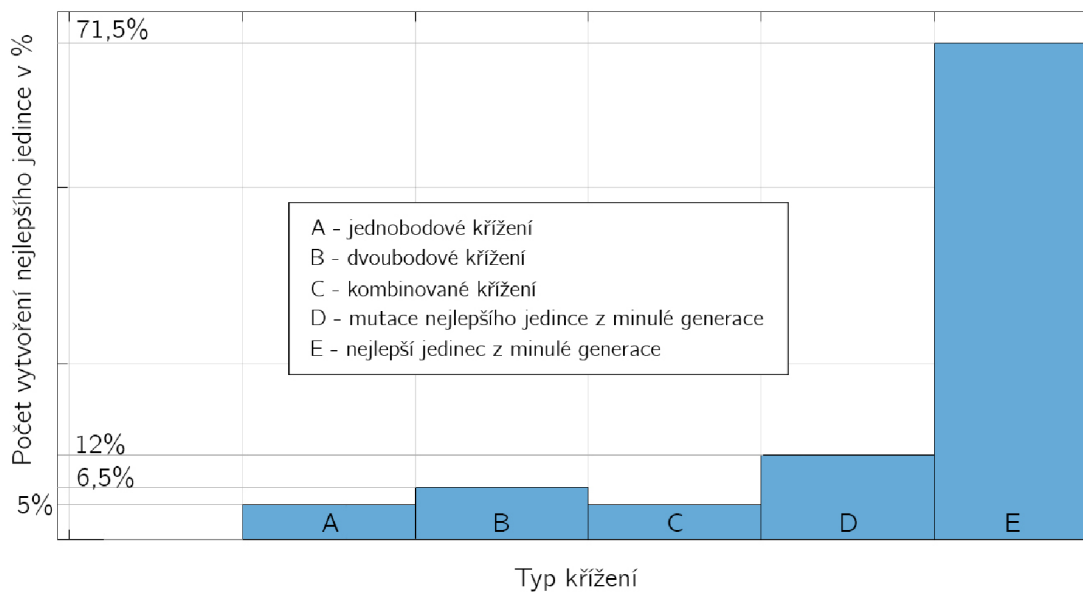
Na obr. 14 je ukázané porovnání tvaru vrubu popsané nejlepším jedincem s referenčním vrubem a vrubem popsáném prvním jedincem z první generace.



Obrázek 14: Porovnání tvarů vrubů. Referenční vrub tvořen kružnicí je znázorněn oranžovou barvou. Červenou barvou je vyznačen vrub reprezentován prvním jedincem z první generace [0.8147 0.9575 0.4218]. Modrou barvou je vyznačen vrub reprezentován nejlepším jedincem z poslední generace [1 0.0924 0.2671].

2.2.11 Analýza křížení

Křížením vznikne nová generace, v který jedinci nejsou seřazeni podle svého fitness. Z pozice jedince vyplývá, jakým typem křížení byl vytvořen. V každé iteraci se po přiřazení zdatnosti k jedincům uloží pozice toho nejlepšího (před tím, než se jedinci v generaci srovnají podle své zdatnosti). Podle pozice se poté zjistí, z jakého typu křížení nejlepší jedinec vyšel. Analýza potomků nám umožňuje posoudit efektivitu jednotlivých typů křížení. Z histogramu na obr. 15 lze vyčíst efektivnost jednotlivých křížení.

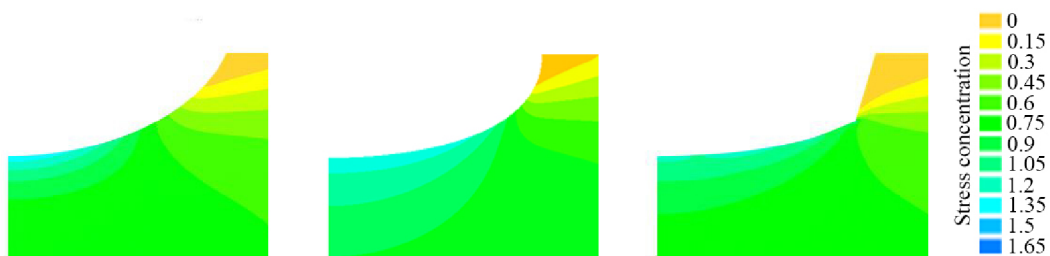


Obrázek 15: Efektivnost křížení pro konkrétní případ. Typ křížení, který vytvořil největší počet nejlepších jedinců je nejefektivnější. Vyjma sloupce E. Pravý sloupec E ukazuje kolikrát se stalo, že žádný z jedinců nové generace nepřekonal zdatnost nejlepšího jedince z předešlé generace.

Z histogramu je patrné, že jednotlivé metody křížení jsou v tomto konkrétním případě rovnocenné. Sloupec D reprezentuje mutace nejlepšího jedince z minulé generace (viz kapitola Mutace). Z histogramu plyne, že se vyplatí mutovat nejlepšího jedince z předešlé generace. Poslední pozice v nové generaci je obsazena nejlepším jedincem z minulé generace. Ten zůstává nejlepším jedincem v případě, když křížením a mutací nevznikne nový lepší jedinec.

2.2.12 Výsledky optimalizace vrubu

Počítali jsme s izotropním materiálem definovaným Youngovým modulem $E = 210$ GPa a Poissonovou konstantou $\mu = 0.3$. Na obr. 16 jsou porovnány koncentrace napětí ve vrubech o různé geometrii.



Obrázek 16: Koncentrace napětí ve vrubech (rozměr w40d10). Zleva: referenční(kruhový), Bézierova křivka a lomená čára s 10 parametry.

V Tab. 2 jsou výsledky pro různé rozměry tyče namáhané čistým tahem. Tab. 3 obsahuje výsledky tyče namáhané ohybem.

	d = 5 mm	d = 10 mm	d = 15 mm	
Kruhový	1.3727	1.4780	1.4913	w = 40 mm
Bézier	1.2198	1.2972	1.2846	
Lomená čára	1.2926	1.3561	1.2887	
Kruhový	1.2643	1.3869	1.3653	w = 50 mm
Bézier	1.1535	1.2035	1.1863	
Lomená čára	1.2365	1.3360	1.2115	
Kruhový	1.1943	1.2837	1.2727	w = 60 mm
Bézier	1.1110	1.1441	1.1256	
Lomená čára	1.1656	1.2098	1.1297	

Tabulka 2: Koncentrace napětí ve vrubech - TAH

	d = 5 mm	d = 10 mm	d = 15 mm	
Kruhový	1.2168	1.2858	1.2615	w = 40 mm
Bézier	1.1130	1.1356	1.1041	
Lomená čára	1.1242	1.1443	1.0812	
Kruhový	1.1463	1.2048	1.1984	w = 50 mm
Bézier	1.0718	1.0791	1.0607	
Lomená čára	1.0765	1.0799	1.0382	
Kruhový	1.1053	1.1524	1.1536	w = 60 mm
Bézier	1.0441	1.0475	1.0368	
Lomená čára	1.0496	1.0526	1.0366	

Tabulka 3: Koncentrace napětí ve vrubech - OHYB

Z tabulek je patrné, že se vždy povedlo nalézt vrub s nižším součinitelem koncentrace napětí v porovnání s referenčním vrubem. Na tomto jednoduchém příkladu byla ukázaná funkčnost genetického algoritmu.

Doba výpočtu byla v desítkách minut. Běh genetického algoritmu byl zpomalován opakovaným voláním externího řešiče pro výpočet hodnoty kritériální funkce.

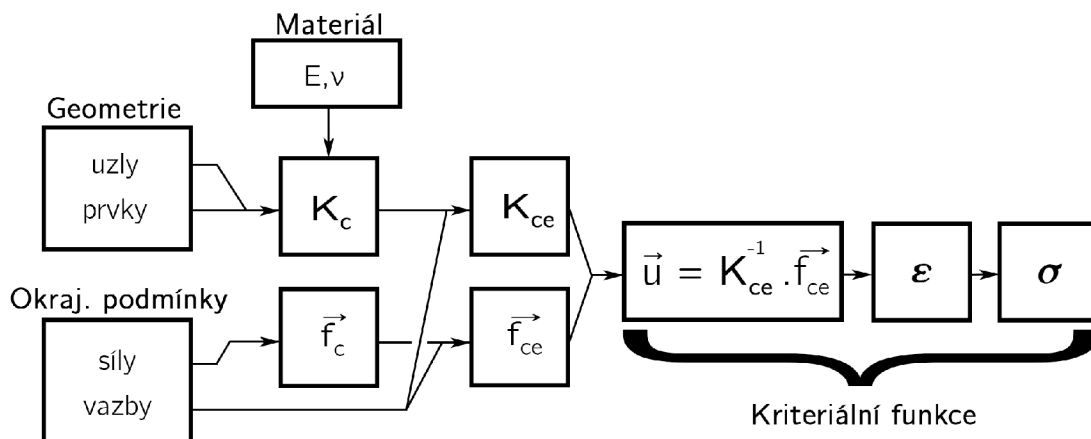
2.3 MKP v Matlabu

Vlastní metoda konečných prvků v Matlabu umožní výpočet hodnoty kritériální funkce bez volání externího řešiče. Tím se značně zrychlí algoritmus, protože volání externího řešiče stojí čas. Pro výpočet hodnoty kritériální funkce je potřeba znát posuvy uzlů, z těch lze dopočítat napětí v prvcích. Předpokládáme malé posuvy a lineární chování materiálu. Posuvy vypočítáme ze základní rovnice MKP:

$$\mathbf{K}_{ce} \cdot \vec{u} = \vec{f}_{ce}, \quad (8)$$

\mathbf{K}_{ce} - celková eliminovaná matice tuhosti,
 kde \vec{u} - vektor posuvů uzlů,
 \vec{f}_{ce} - celkový eliminovaný vektor zatížení.

Pro výpočet posuvů uzlů se musí provést inverze \mathbf{K}_{ce}^{-1} . Celková neeliminovaná matice tuhosti \mathbf{K}_c je singulární, nelze provést její inverzi - úloha není jednoznačná a soustava nemá řešení. Proto se musí matice tuhosti eliminovat zohledněním okrajových podmínek - vazeb. Vazby se promítnou i na vektoru zatížení a vedou k vytvoření celkového eliminovaného vektoru zatížení \vec{f}_{ce} .



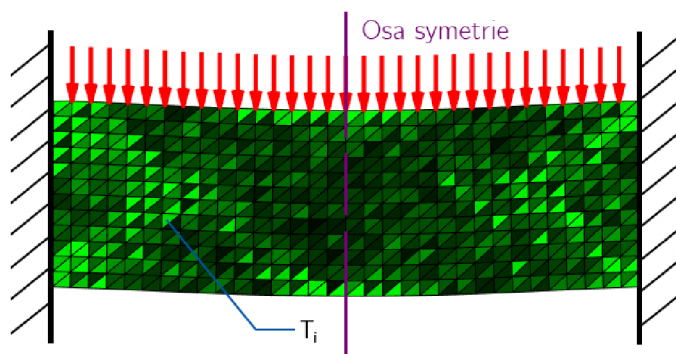
Obrázek 17: Schéma MKP

2.4 Vetknutý nosník zatížený spojitým zatížením

Tato část pojednává o optimalizaci rozložení hmoty za účelem zvýšení tuhosti při zachování objemu. Řešenou úlohou je vetknutý nosník zatížený spojitým zatížením na vrchní straně (viz obr. 18).

2.4.1 Geometrie nosníku

Distribuce hmoty je popsána tloušťkami prvků T_i (na obr. 18 znázorňuje světlejší barva větší tloušťku). Vhodné rozložení hmoty vede k minimalizaci posuvů. Nosník je řešen ve 2D, třetí dimenze je daná tloušťkou jednotlivých prvků.

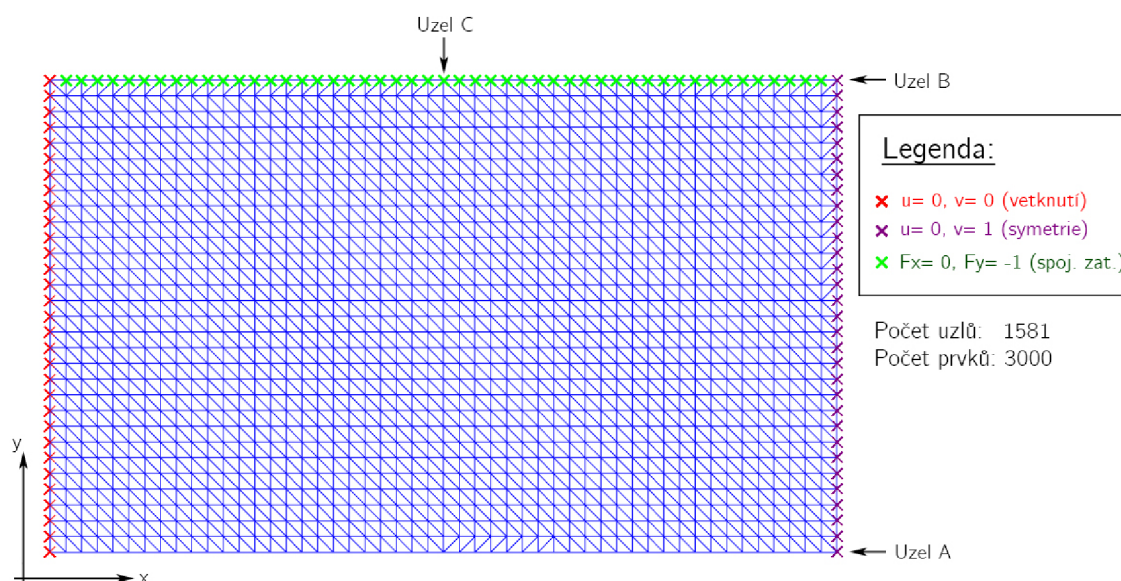


Obrázek 18: Ilustrační obrázek vetknutého nosníku. Světlejší barva znamená větší tloušťku prvku.

Celá tato úloha je řešena v Matlabu, včetně metody konečných prvků, která je potřebná pro výpočet hodnoty kritériální funkce (stanovení fitness jedinců). Díky tomu se nemusí volat externí řešič, což má za následek ušetření času a možnost snadnějších operací s vypočtenými daty.

2.4.2 Síť a okrajové podmínky

Při tvorbě sítě je žádoucí využít symetrie, díky kterým se sníží počet prvků sítě při zachování jejich velikosti. Tím se významně zrychlí běh výpočtu. Síť a okrajové podmínky jsou ukázaný na obr. 19.



Obrázek 19: Síť na levé polovině vetknutém nosníku. Hodnoty u posuvů v legendě znamenají, zda je pohyb uzlu dovolen. Hodnota 0 odpovídá zamezení posuvu, hodnota 1 znamená dovolení posuvu. Uzly A, B a C jsou vyznačeny, protože jsou středem zájmu při optimalizaci.

Symetrie se zohlední okrajovými podmínkami pro uzly ležící na ose symetrie. V místě vetknutí se uzlům zamezí posuvy ve směru x (posuv u) i y (posuv v). Uzlům ležícím na vrchní straně nosníku se předejde síla ve směru proti ose y .

2.4.3 Počáteční generace

Jedinec je v tomto případě tvořen 3000 znaky. Počet znaků odpovídá počtu prvků sítě. Každý ze znaků reprezentuje jednu tloušťku prvku.

Matice počáteční generace nemůže být v této úloze vygenerovaná zcela náhodně. Úkolem této optimalizace je zvýšit tuhost při zachování objemu.

Jedinci z počáteční generace se vytvoří nejprve na základě počáteční tloušťky nosníku, která definuje objem. Poté se každá tloušťka prvku změní o náhodně vygenerovanou hodnotu z určeného intervalu. Tím se sice nedocílí stejného objemu pro všechny jedince, ale rozdíl oproti počátečnímu objemu nebude velký. Tento rozdíl objemu se srovná následným modifikováním tlouštěk (viz následující kapitola Limity na str. 29).

Na následujícím skriptu je ukázaná tvorba matice počáteční generace pro počáteční tloušťku 4mm, změna tloušťky maximálně o 2mm.

```

1 G0size=100; %pocet jedincu v pocatecni generaci
2 poc_tloustka=0.004; %m
3
4 for cislo_jedince=1:G0size
5     for i=1:prvky_n %prvky_n ... pocet prvku
6         spodni_mez = -0.002;
7         vrchni_mez = 0.002;
8         zmena_tl = (vrchni_mez-spodni_mez).*rand+spodni_mez
9         ;
10        G0(cislo_jedince , i)=poc_tloustka+zmena_tl;
11    end
12 end

```

2.4.4 Kriteriaální funkce pro vetknutý nosník

Úkolem této optimalizace je maximalizace tuhosti nosníku. V této úloze je pro dané zatížení a okrajové podmínky optimalizovaná tuhost v různých místech minimalizováním posuvů zvolených uzlů. Kriteriaální funkce je odvozená od posuvů uzlů.

Minimalizace svislého posuvu uzlu A

Uzel A leží na ose symetrie a na dolní hraně sítě viz obr. 19. Hodnota kriteriaální funkce bude v tomto případě odvozená od svislého posuvu uzlu A. Vzhledem k orientaci spojitého zatížení bude svislý posuv záporný. Jelikož budeme hodnotu kriteriaální funkce minimalizovat, bude mít funkce podobu:

$$F_A = f(\vec{u}) = -v_A. \quad (9)$$

Minimalizace svislého posuvu uzlu B

Uzel B leží také na ose symetrie, ale na horní hraně sítě. Při této optimalizaci bude mít kritériální funkce podobu:

$$F_B = f(\vec{u}) = -v_B. \quad (10)$$

Minimalizace svislého posuvu uzlu B a C

Třetí použitá kritériální funkce je kombinací svislých posuvů uzlů B a C. Uzlu B je daná větší váha, protože leží uprostřed nosníku v místě největšího průhybu. Kritériální funkce má tvar:

$$F_{BC} = f(\vec{u}) = -3.v_B - v_C. \quad (11)$$

2.4.5 Limity

Tloušťky jsou omezeny objemem nosníku a dovolenou mezní hodnotou T_{min}, T_{max} .

Všechny prvky sítě mají stejnou plochu. Z toho plyne, že součet tloušťek prvků (při zachování objemu nosníku) je konstantní.

Při tvorbě počáteční generace a nebo po křížení a mutování vznikají jedinci, jejichž suma tloušťek je různá. Proto je zapotřebí provést modifikaci tloušťek, aby jejich suma byla rovna sumě počátečních tloušťek. Suma počátečních tloušťek je daná součinem počáteční tloušťky a počtu prvků $T_0.prvky_n$. Každá tloušťka se tedy modifikuje vztahem:

$$T_i = T_i + \frac{T_0.prvky_n - \sum(T_i)}{prvky_n}, \quad (12)$$

po úpravě:

$$T_i = T_i + T_0 - \frac{\sum(T_i)}{prvky_n}. \quad (13)$$

Ke každé tloušťce se tedy přičte taková hodnota, aby suma tloušťek odpovídala počáteční hodnotě. Hodnota, o kterou se tloušťky popsané jedním jedincem změní, je stejná. To znamená, že tvar plochy vytvořené tloušťkami zůstane stejný, pouze se posune tak, aby objem pod plochou odpovídal počátečnímu objemu.

Skript modifikující tloušťky:

```
1 for i=1:G0size
2     modifikace=(sum(G0(i, :)) / prvky_n) - poc_tloustka ;
3     G0(i, :) = G0(i, :) - modifikace ;
4 end
```

Takováto modifikace tloušťek lze uplatnit pouze v případech, kdy žádná z tloušťek nenabývá svojí minimální hodnotu T_{min} . Pokud některé z prvků mají tloušťku T_{min} , nelze je více ztenčovat. Proto se budou modifikovat pouze tloušťky větší než T_{min} . Nejdříve se musí zjistit, kolik je prvků větších než T_{min} . Tento počet odpovídá počtu

modifikovaných tloušťek *poc_mod_tl*. Poté je potřeba znát jejich pozice. Tloušťky větší než T_{min} se modifikují dle vztahu:

$$T_i = T_i - \frac{\sum T_i - T_0 \cdot prvky_n}{poc_mod_tl}. \quad (14)$$

Tvar plochy vytvořené tloušťkami se v tomto případě změní, k posunu dojde jen u modifikovaných tloušťek.

Skript modifikující tloušťky zohledňující T_{min} :

```

1 for i=1:size(G,1)
2     idx_mod=find(G(i,:)>T_min);
3     modifikace=(sum(G(i,:))-prvky_n*poc_tloustka)/length(
4         idx_mod);
5     for ii=1:length(idx_mod)
6         G(i,idx_mod(ii))=G(i,idx_mod(ii))-modifikace;
7     end
end

```

2.4.6 Křížení

Pro tuto úlohu bylo použito jednobodové křížení (viz str. 10). Níže je ukázaná funkce v Matlabu, která křížení provádí. Tato funkce má tři vstupy. Prvním parametrem x je matice generace. Druhý parametr *pocet_deti* určuje, kolik se křížením vytvoří nových jedinců. Parametr *pocet_plodnych* určuje počet nejlepších jedinců, kteří budou párováni pro křížení.

```

1 function new_gen = krizeni_sp(x, pocet_deti, pocet_plodnych)
2
3 for i=1:pocet_deti
4     misto_rozdeleni=randi([2 size(x,2)-1]);
5     par=[randi([1 pocet_plodnych]),randi([1 pocet_plodnych])
6         ];
7     G(i,:)=[x(par(1),1:misto_rozdeleni),x(par(2),
8         misto_rozdeleni+1:size(x,2))] ];
9 end
new_gen=G;
end

```

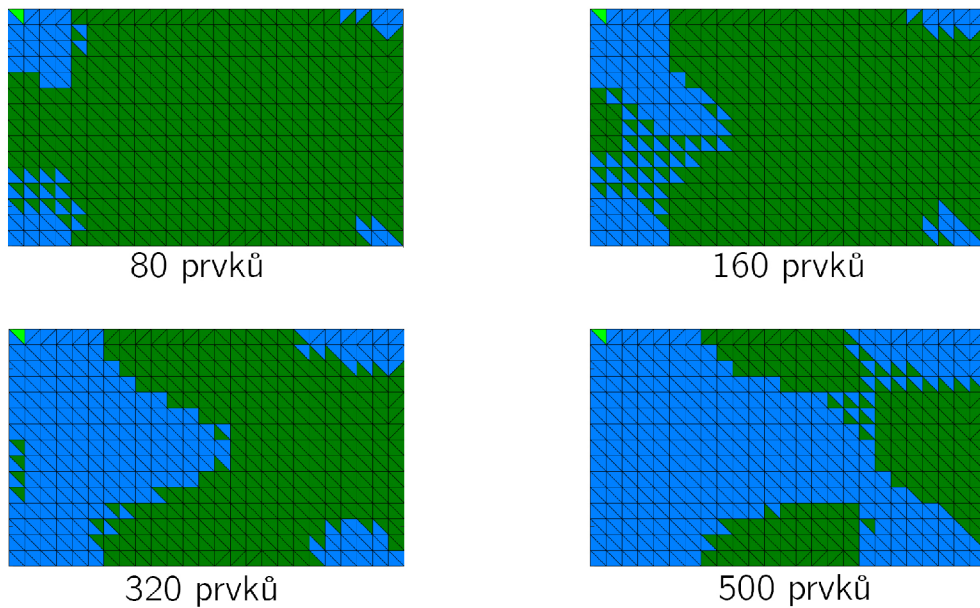
2.4.7 Mutace

V této úloze byla použita náhodná i řízená mutace. Náhodná mutace pomocí matice mutace \mathbf{M}_{mut} byla popsána v předešlé úloze viz str. 19 (optimalizace tvaru vrubu). Z toho důvodu je zde popsána pouze řízená mutace.

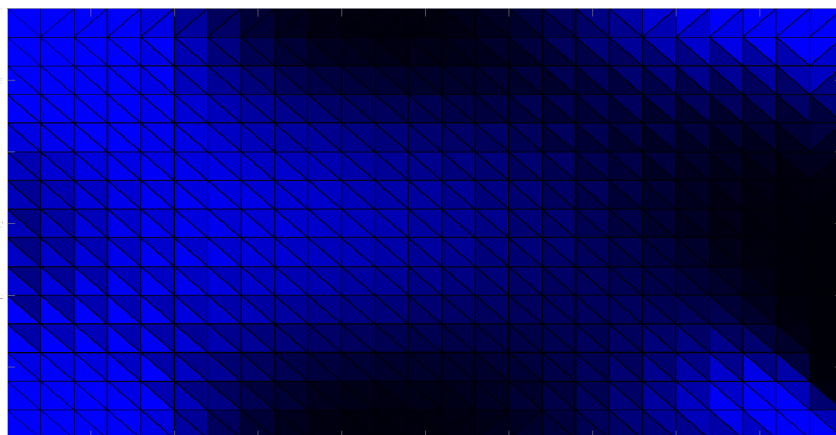
Při řízené mutaci se zjistí váhy jednotlivých znaků pomocí testovacích jedinců. Ti vzniknou úpravou nejlepšího jedince z předešlé generace. Znaky v tomto případě

představují tloušťky prvku. Nejlepšímu jedinci se postupně zvětší tloušťka jednotlivých prvků na maximální dovolenou tloušťku a sleduje se, jaký vliv měla změna tloušťky na hodnotu kriteriální funkce. Podle toho se jednotlivým znakům přidělí váha znaku. Zvolenému počtu znaků s největší vahou se mutací zvětšuje hodnota znaku.

Na obr. 20 jsou modře zbarveny různé počty prvků s největší vahou po první generaci.



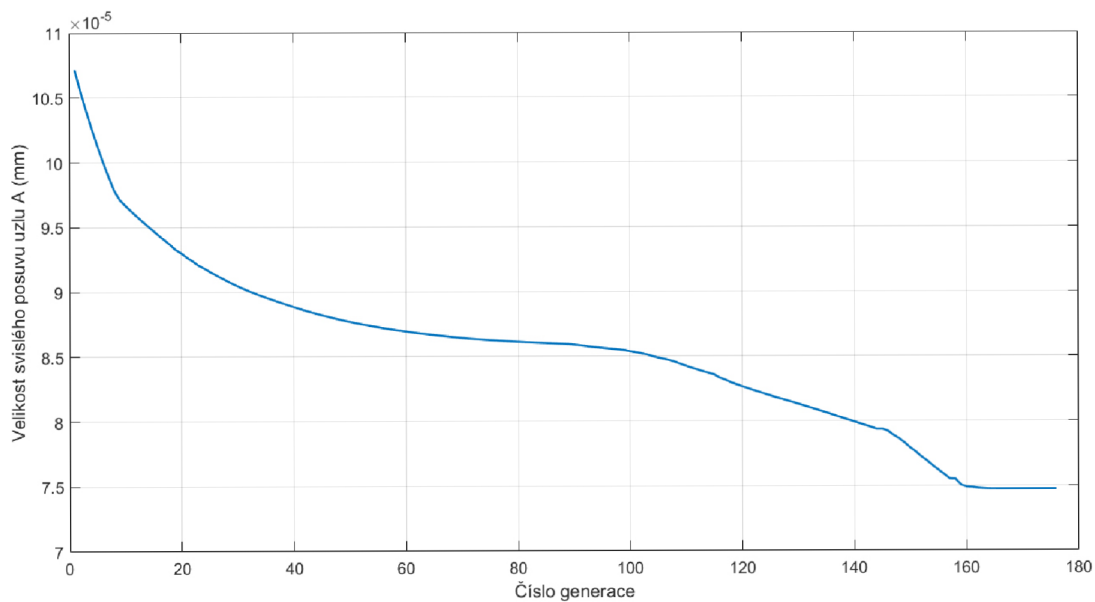
Obrázek 20: Vyznačené prvky s největší vahou. Modrou barvou jsou znázorněny ty prvky, které zvětšením své tloušťky nejvíce ovlivní hodnotu kriteriální funkce (9).



Obrázek 21: Porovnání vah jednotlivých prvků. Čím je prvek světlejší, tím má větší vliv na hodnotu kriteriální funkce (9).

2.4.8 Průběh svislého posuvu uzlu A

Na obr. 22 je ukázaný průběh minimalizace posuvu uzlu A při výpočtu s kriteriální funkcí (9).

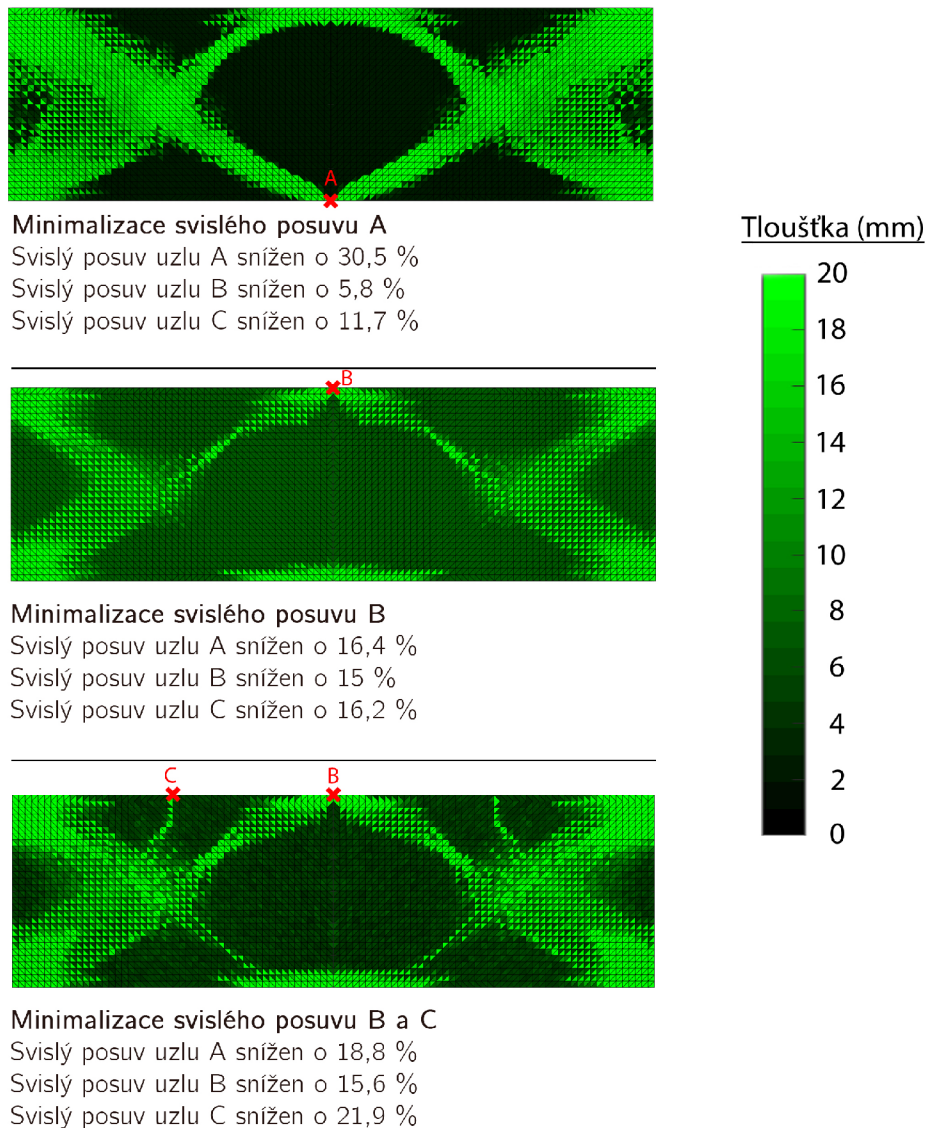


Obrázek 22: Minimalizace svislého posuvu uzlu A. Zlom kolem generace 100 a 150 je způsobený spojením žeber.

Výpočet byl zastaven při generaci číslo 178. Reprodukci se v každé iteraci vytvořilo 3200 jedinců. Vyhodnocení jedné generace trvalo přibližně 4 minuty. Celkový čas výpočtu byl okolo 12 hodin.

2.4.9 Výsledky optimalizace vetknutého nosníku

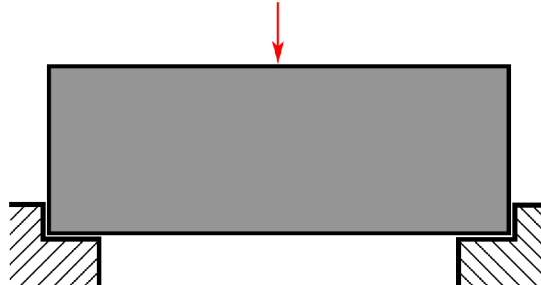
Na následujícím obrázku jsou ukázány výsledky optimalizace pro různé kriteriální funkce.



Obrázek 23: Porovnání výsledků pro různé kriteriální funkce. Výsledek pro kriteriální funkci F_{BC} zmenšil svislý posuv uzlu B více než pro kriteriální funkci F_B . To znamená, že při optimalizaci s kriteriální funkcí F_B se našlo pouze lokální minimum. Rozložením hmoty se povedlo nejvíce minimalizovat svislý posuv uzlu A při použití kriteriální funkce F_A .

2.5 Nosník uložený v dolních rozích

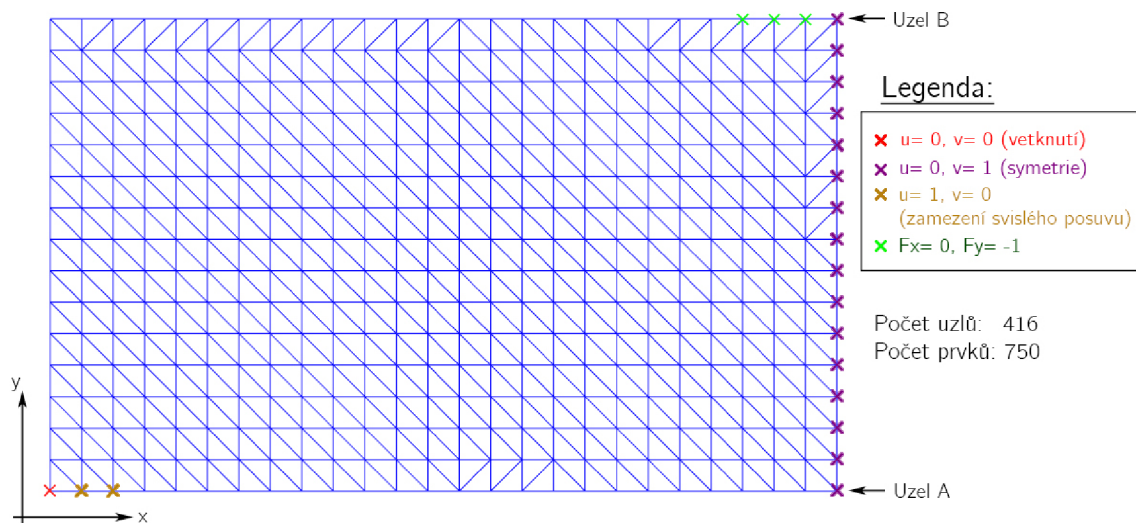
V této kapitole jsou výsledky pro nosník uložený pouze v dolních rozích (schéma na obr. 24). Použitý genetický algoritmus pro tuto úlohu má stejné parametry jako v předešlé kapitole o vetknutém nosníku. To znamená, že tvorba počáteční generace, limity, křížení a mutace jsou stejné jako v minulém příkladu a nejsou zde popsány.



Obrázek 24: Schéma nosníku uloženého v dolních rozích

2.5.1 Síť a okrajové podmínky

Pro tento příklad byla použita síť s menším počtem prvků. Na obrázku 25 jsou znázorněné okrajové podmínky



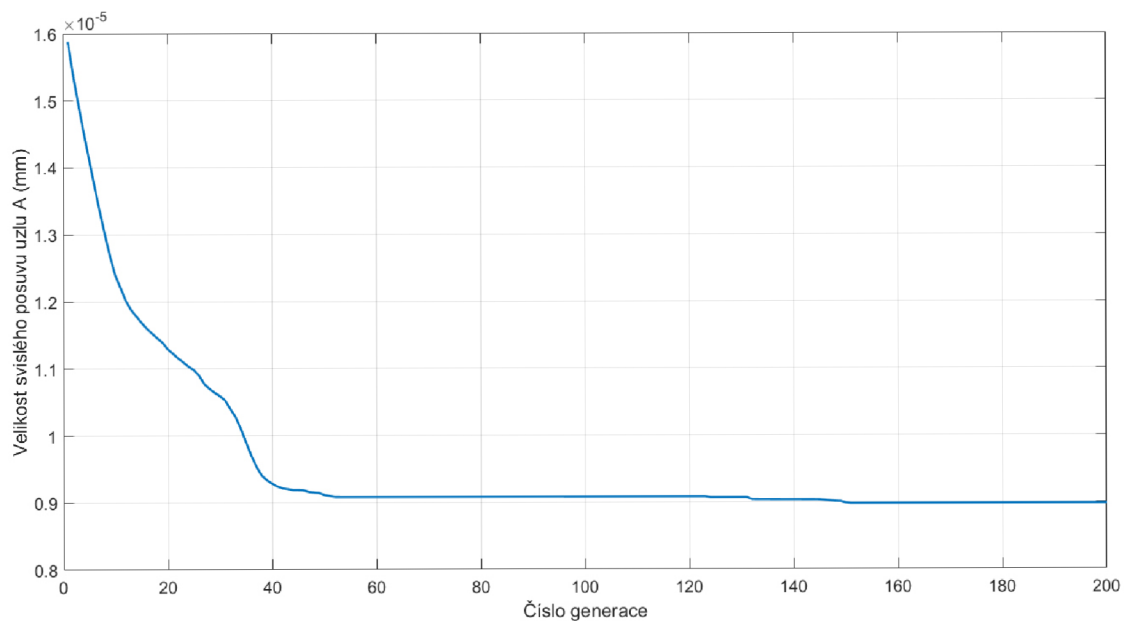
Obrázek 25: Síť na nosníku

2.5.2 Kriteriaální funkce pro nosník upevněný v dolních rozích

Byly použity pouze dvě kriteriaální funkce. První pro minimalizaci svislého posuvu uzlu A, druhá pro minimalizaci svislého posuvu uzlu B.

2.5.3 Průběh svislého posuvu uzlu A

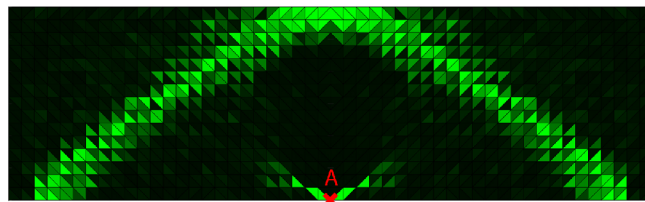
Na obr. 26 je ukázaný průběh minimalizace posuvu uzlu A.



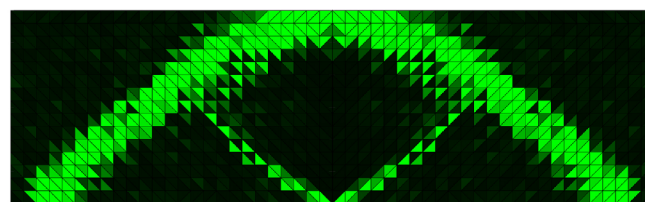
Obrázek 26: Minimalizace svislého posuvu uzlu A.

Výpočet byl zastaven při generaci číslo 200. Reprodukci se v každé iteraci vytvořilo 950 jedinců a síť měla v tomto případě pouze 750 prvků, proto vyhodnocení jedné generace trvalo přibližně jen 15 vteřin. Celkový čas výpočtu byl okolo 50 minut.

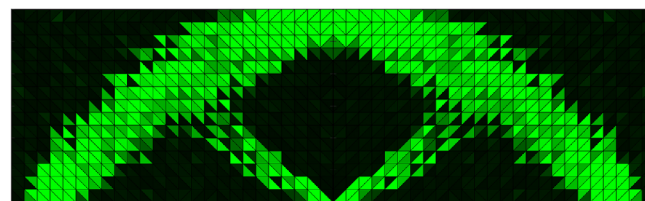
2.5.4 Výsledky optimalizace pro nosník upevněný v dolních rozích



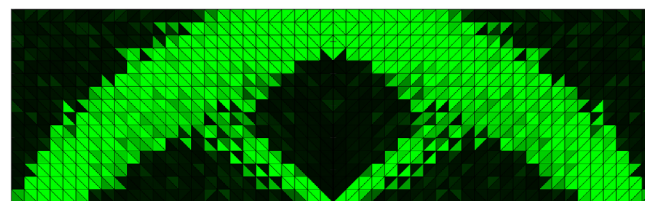
Počáteční tloušťka: 4 mm
Objem: 48 cm³
Svislý posuv uzlu A snížen o 48,2%
Svislý posuv uzlu B snížen o 38,4%



Počáteční tloušťka: 6 mm
Objem: 72 cm³
Svislý posuv uzlu A snížen o 50,7%
Svislý posuv uzlu B snížen o 33,8%

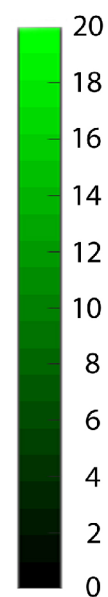


Počáteční tloušťka: 8 mm
Objem: 96 cm³
Svislý posuv uzlu A snížen o 49,2%
Svislý posuv uzlu B snížen o 29,63%

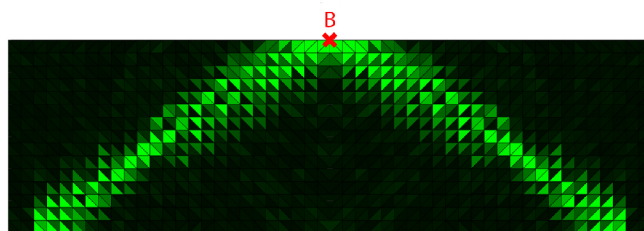


Počáteční tloušťka: 10 mm
Objem: 120 cm³
Svislý posuv uzlu A snížen o 45,9%
Svislý posuv uzlu B snížen o 25,6%

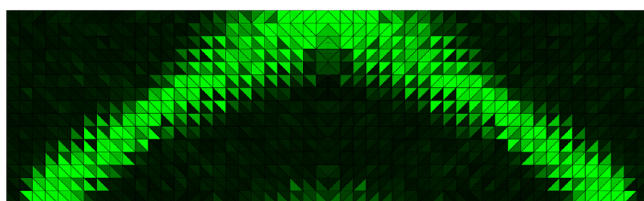
Tloušťka (mm)



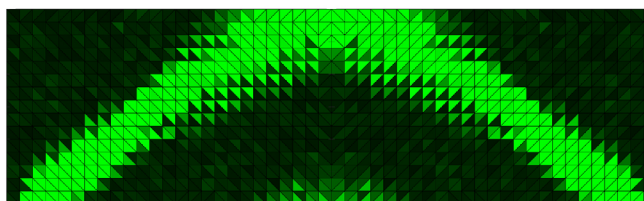
Obrázek 27: Žebra na nosníku po minimalizaci svislého posuvu uzlu A. Největší zvýšení tuhosti uzlu A nastalo pro objem 72cm³



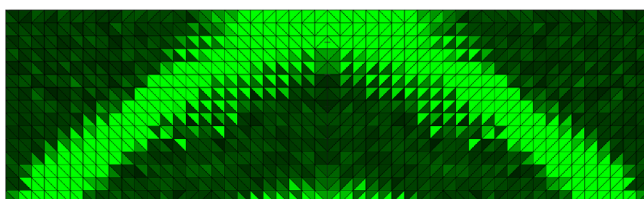
Počáteční tloušťka: 4 mm
 Objem: 48 cm³
 Svislý posuv uzlu A snížen o 42,3%
 Svislý posuv uzlu B snížen o 41,9%



Počáteční tloušťka: 6 mm
 Objem: 72 cm³
 Svislý posuv uzlu A snížen o 40,4%
 Svislý posuv uzlu B snížen o 38,2%

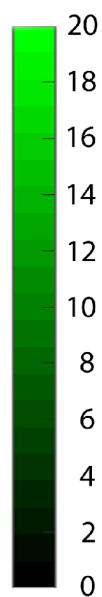


Počáteční tloušťka: 8 mm
 Objem: 96 cm³
 Svislý posuv uzlu A snížen o 36,9%
 Svislý posuv uzlu B snížen o 34,6%



Počáteční tloušťka: 10 mm
 Objem: 120 cm³
 Svislý posuv uzlu A snížen o 30,8%
 Svislý posuv uzlu B snížen o 29,2%

Tloušťka (mm)



Obrázek 28: Žebra na nosníku po minimalizaci svislého posuvu uzlu B. Největší zvýšení tuhosti uzlu B nastalo pro objem 48cm³

Narozdíl od předešlého příkladu jsme tento nosník optimalizovali i pro různé objemy. Podmínka pro konstantní objem zůstává, ale počítalo se s různými počátečními objemy. Výsledky optimalizace nosníku pro kritériální funkci minimalizující posuv uzlu A jsou ukázaný na obrázku 27. Výsledky pro kritériální funkci minimalizující posuv uzlu B jsou na obrázku 28.

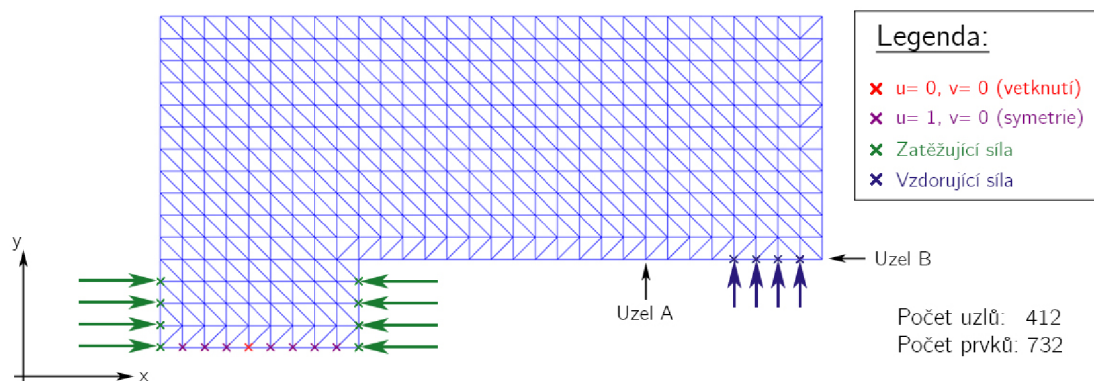
Procentuální zlepšení posuvů je počítáno vůči posuvům na nosníku s konstantní počáteční tloušťkou.

2.6 Předepsání funkčních posuvů

Další aplikací genetického algoritmu je předepsání funkčních posuvů. Kritériální funkce je stejně jako v předešlých příkladech odvozena od posuvů. V tomto příkladu se ale nesnažíme posuvy minimalizovat, ale naopak maximalizovat za účelem požadované funkce součásti.

Příkladem je zde ukázaná svorka. Chceme, aby se čelisti svorky co nejvíce přibližovaly k sobě pro předepsané silovém působení.

2.6.1 Síť a okrajové podmínky



Obrázek 29: Síť a okrajové podmínky pro svorku. Zelenou barvou je znázorněna zatěžující silou. Modrou barvou je reakční síla.

2.6.2 Kritériální funkce pro svorku

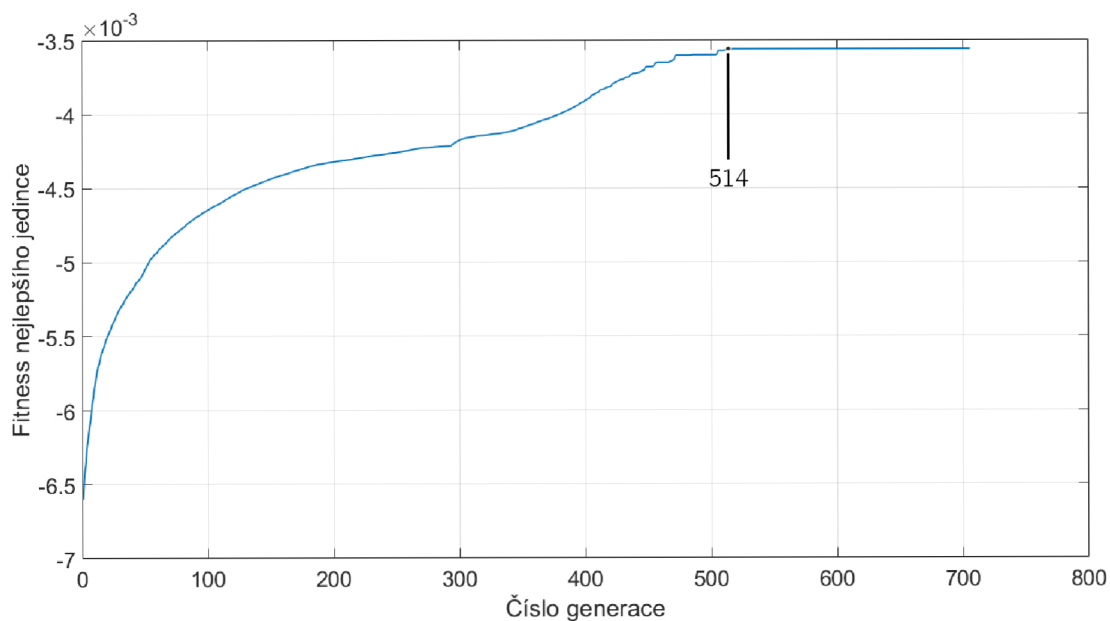
Hodnota kritériální funkce pro tento případ je odvozena od svislých posuvů uzlů A a B a má tvar:

$$F = f(\vec{u}) = 2 \cdot v_A + v_B. \quad (15)$$

Svislý posuv uzlu A má větší váhu než svislý posuv uzlu B. Účel je snaha udržet čelisti rovnoběžné.

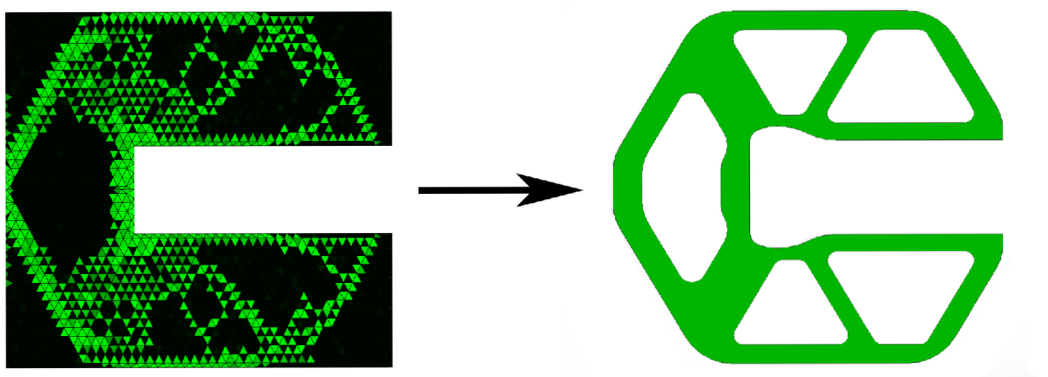
2.6.3 Výsledky optimalizace svorky

Zdatnost nejlepšího jedince v závislosti na generaci je ukázaná na obrázku 30.



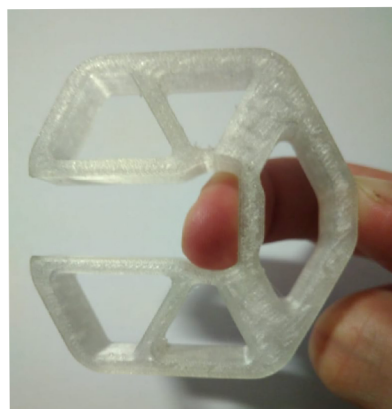
Obrázek 30: Průběh zdatnosti nejlepšího jedince. Od generace číslo 514 se nevytvořil potomek s větší zdatností.

Reprodukcí se v každé iteraci vytvořilo 950 jedinců. Vyhodnocení jedné generace trvalo opět přibližně 15 vteřin. Výpočet běžel necelé tři hodiny.

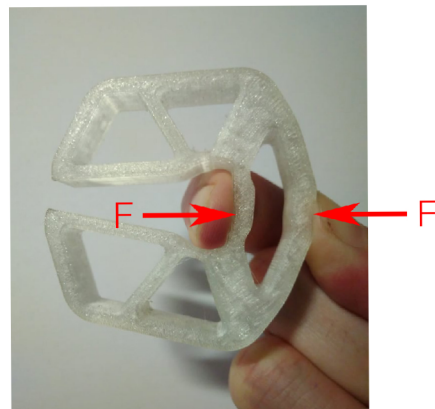


Obrázek 31: Výsledné rozložení hmoty reprezentované nejlepším jedincem z generace číslo 514.

Na základě nejlepšího jedince z generace číslo 514 byl vytvořený model a svorka byla vytištěná pomocí 3D tiskárny z elastického materiálu FLEX viz obr. 32. Na vytištěné svorce byla vyzkoušena funkčnost.



Nezatížený stav



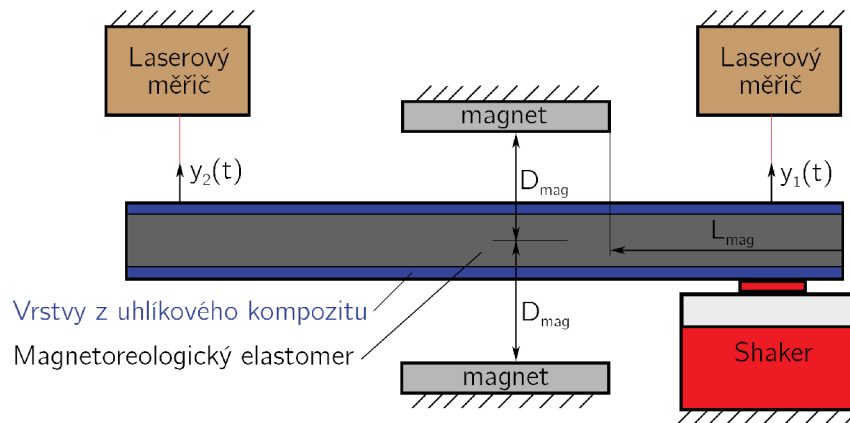
Zatížený stav

Obrázek 32: Vytisknutý model svorky na 3D tiskárně.

3 Experimentální část

3.1 Optimální polohy magnetů tlumících kmitání sendvičového nosníku

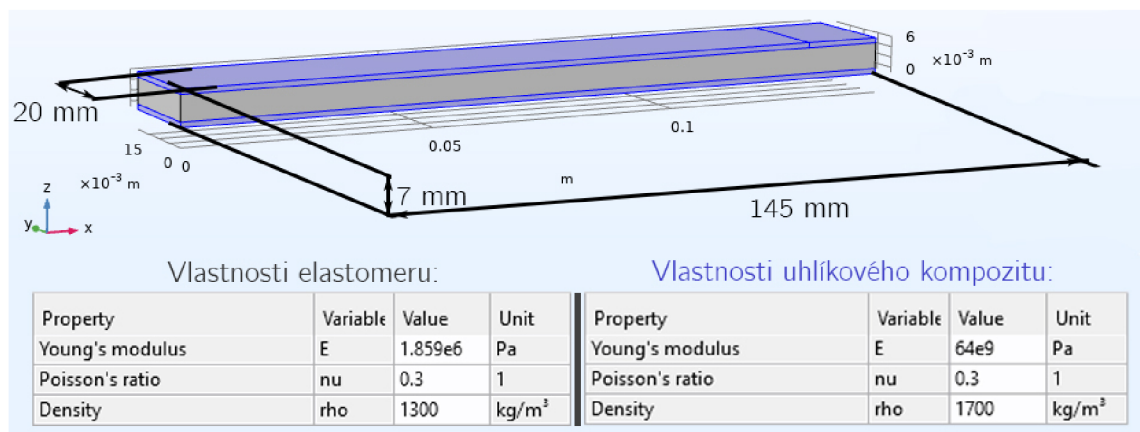
Zadanou úlohou pro optimalizaci bylo nalezení optimálních poloh magnetů tak, aby byla první vlastní frekvence nosníku co nejvyšší. Experiment byl prováděný se sendvičovým nosníkem (viskoelastický magnetoreologický elastomer + krycí vrstvy z uhlíkového kompozitu).



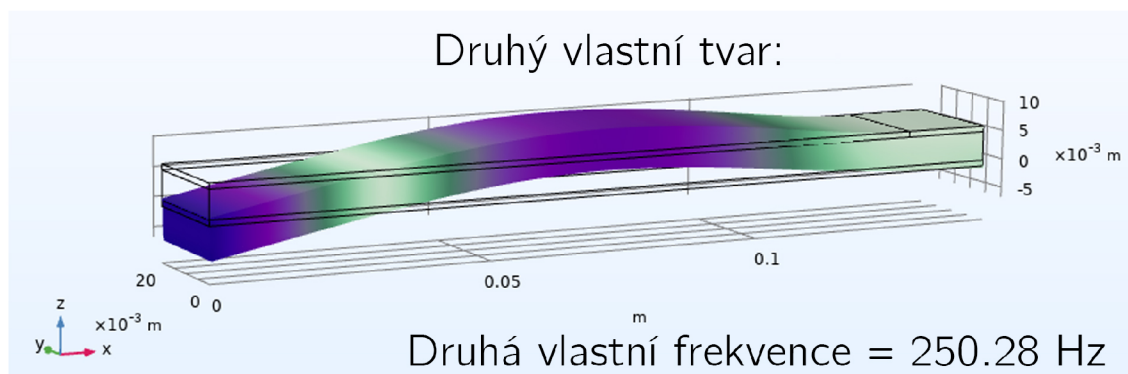
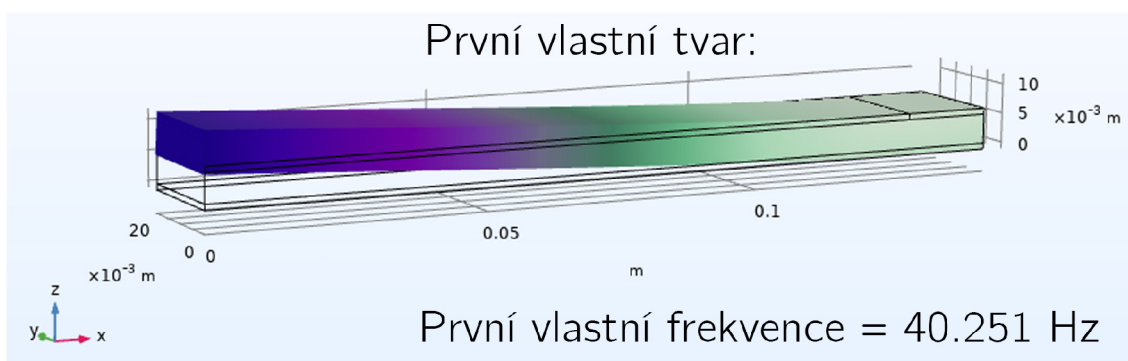
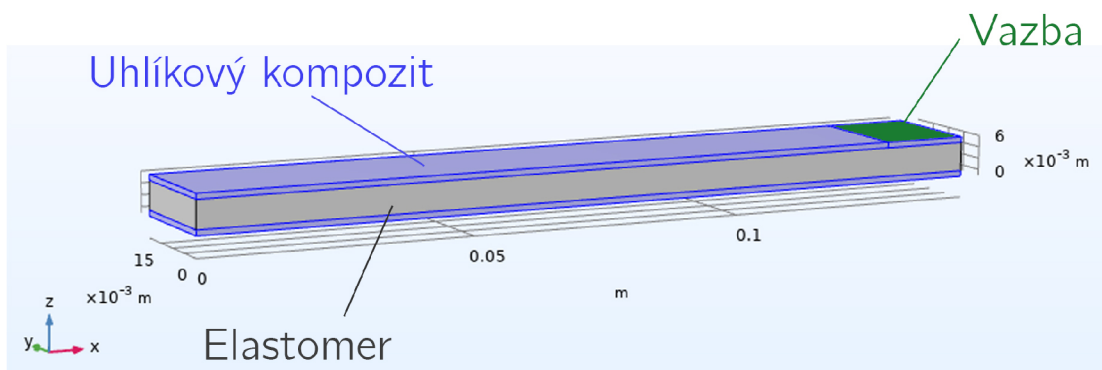
Obrázek 33: Schéma experimentu kmitajícího nosníku. Polohu magnetů určují dva parametry. Podélné umístění magnetů L_{mag} a vzdálenost magnetů od středu nosníku D_{mag} . Magnety jsou umístěny symetricky vůči středu nosníku.

3.1.1 Výpočet vlastní frekvence nosníku bez působení magnetů

Roměry sendvičového nosníku a vlastnosti materiálů použité pro simulaci v COMSOLu jsou ukázané na obr. 34.[7]



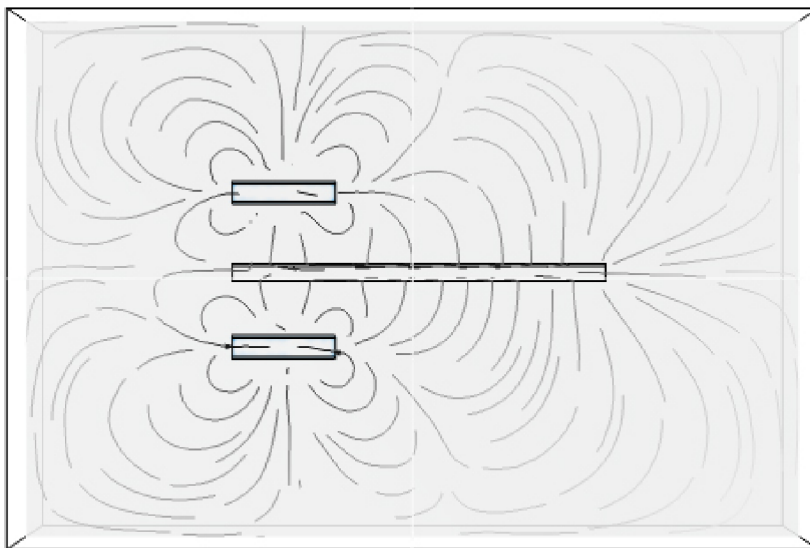
Obrázek 34: Rozměry a vlastnosti zadané do softwaru COMSOLu.



Obrázek 35: První dvě vlastní frekvence a tvary bez působení magnetů.

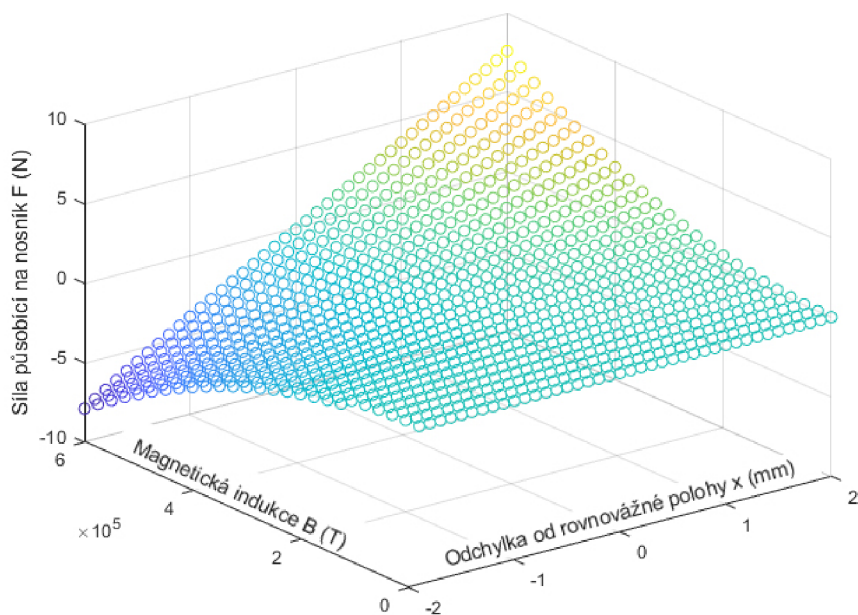
3.1.2 Síly od magnetů působící na nosník

Kolem nosníku vzniká magnetické pole viz obr. 36. Toto pole bylo vykresleno v programu COMSOL.



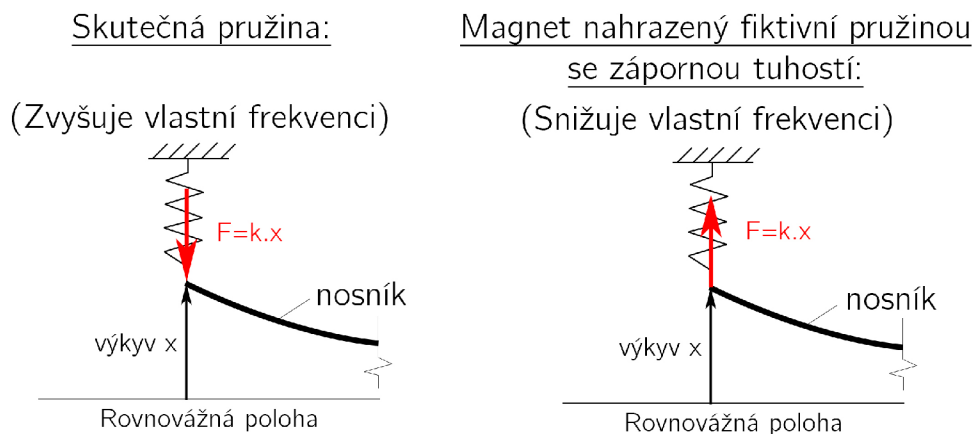
Obrázek 36: Magnetické indukční čáry kolem nosníku.

Síly od magnetů působící na nosník jsou vyneseny do grafu v závislosti na magnetické indukci a vzdálenosti nosníku od magnetů viz obr. 37. Hodnoty pro graf byly vypočteny v COMSOLu, graf byl vykreslen v MATLABu.



Obrázek 37: Síla působící na nosník v závislosti na magnetické indukci a výkyvu nosníku. Pro $B = 0T$ na nosník nepůsobí žádná síla nehlédě na výkyv nosníku.

Magnety působí na nosník jako pružina se zápornou tuhostí (směr síly směřuje narozdíl od skutečné pružiny opačným směrem). Pružina s kladnou tuhostí zvyšuje soustavě tuhost a tím zvyšuje vlastní frekvenci. Magnet, který se chová jako pružina se zápornou tuhostí naopak snižuje vlastní frekvenci.



Obrázek 38: Vliv skutečné a fiktivní pružiny nahrazující magnet na velikost vlastní frekvence.

Tak jako silnější pružina zvyšuje vlastní frekvenci, tak silnější magnet snižuje vlastní frekvenci nosníku. Čím je magnet nosníku blíže (na obr. 33 rozměr D_{mag}), tím posune vlastní frekvenci nosníku níž. Magnet na větší páce (dál od vetknutí nosníku) více snižuje vlastní frekvenci.

Z toho plyne, že nejvyšší vlastní frekvence bude v případě bez magnetů. Nejnižší frekvence nastane, když bude magnet co nejdál od vetknutí na konci nosníku a co nejbližší nosníku. Tato úloha má jednoznačné jedno globální minimum a proto nebyl použitý genetický algoritmus.

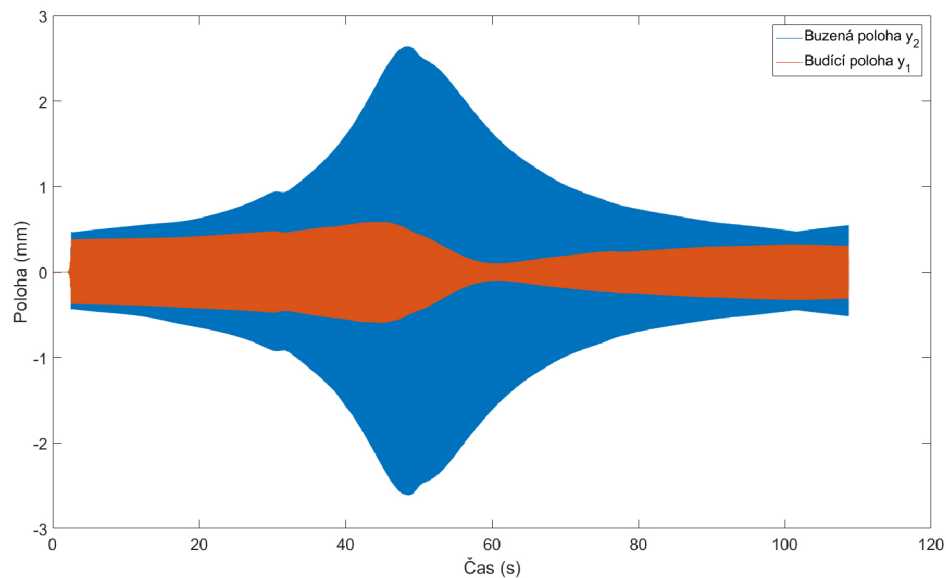
Vliv magnetů na nosník popsany výše je potvrzen experimentem níže.

3.1.3 Měření vlastních frekvencí

Cílem experimentu bylo změření první vlastní frekvence sendvičového nosníku v závislosti na poloze magnetů. Schéma experimentu je na obr. 33.

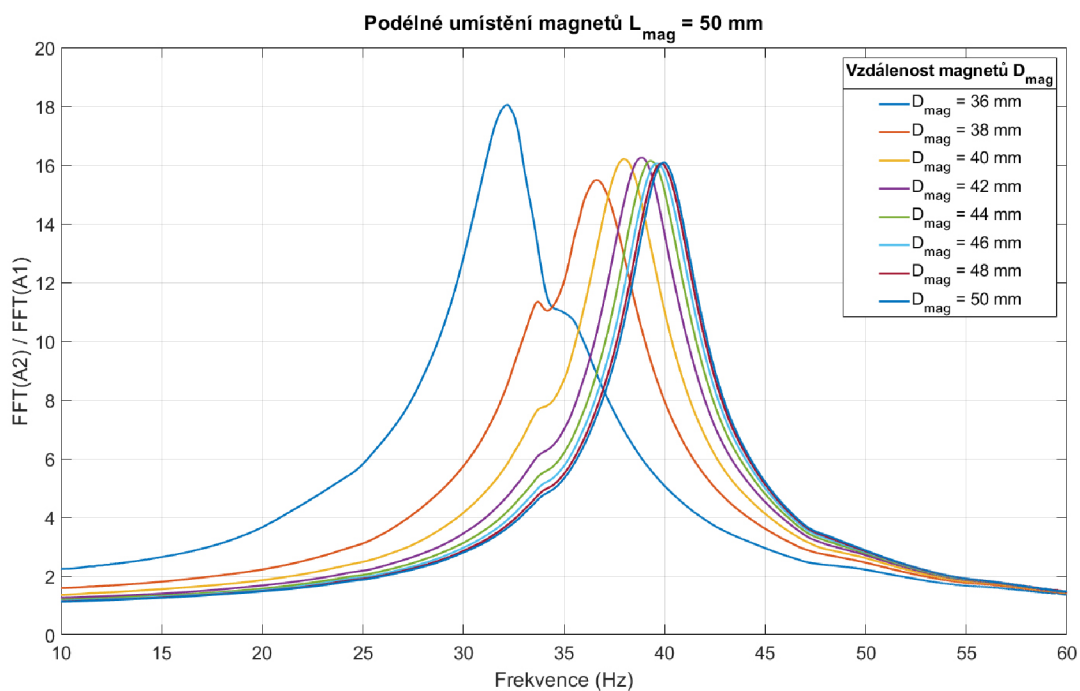
Shaker vybuduje nosník. Budící funkce byla amplitudově-frekvenční rozmítaný sinus. S měnící se frekvencí se mění i amplituda tak, aby zrychlení bylo konstantní. Budící polohu $y_1(t)$ i buzenou polohu $y_2(t)$ měříme laserem.

Měření bylo provedeno pro různé umístění magnetů. Příklad průběhu budící a buzené polohy je na obr. 39.

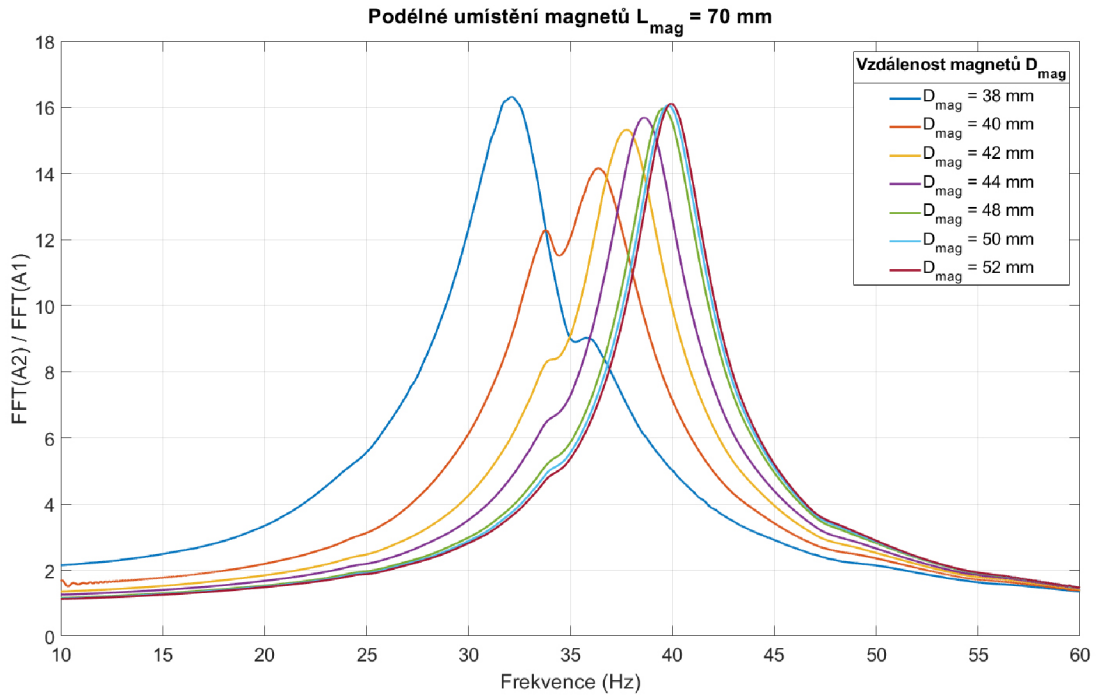


Obrázek 39: Průběh budící a buzené polohy.

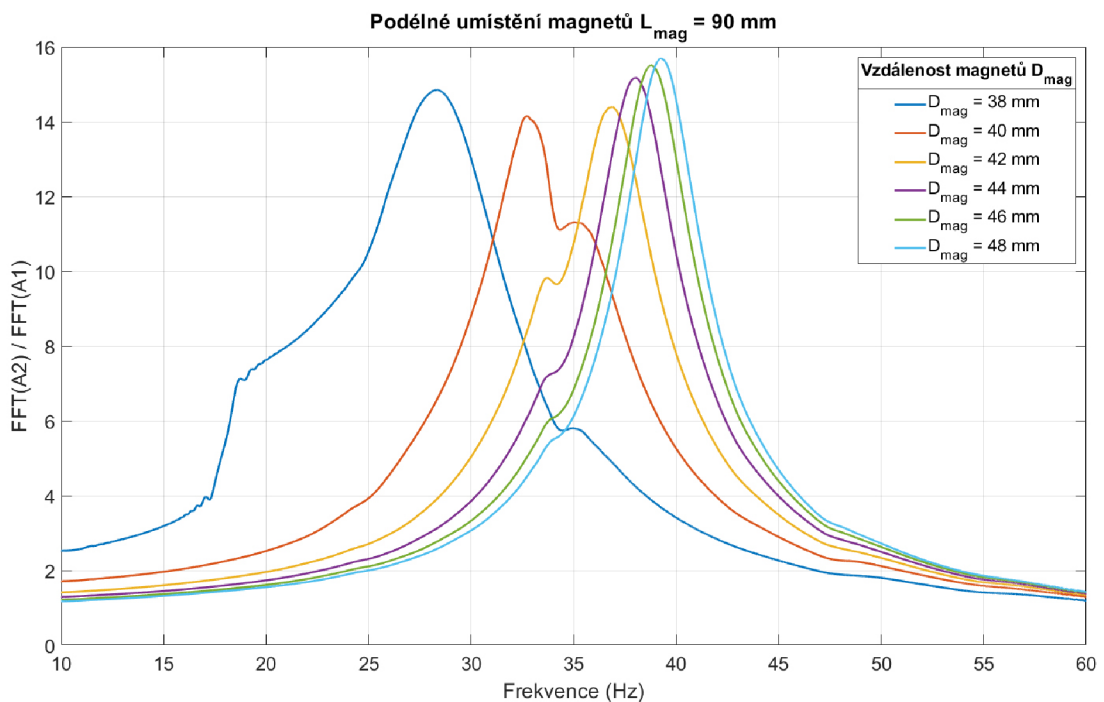
Pro naměřené polohy se provedla rychlá Fourierova transformace (FFT). Podílem FFT buzené polohy a FFT budící polohy se zjistila vlastní frekvence nosníku.



Obrázek 40: Rezonanční oblasti pro různé D_{mag} , podélné umístění magnetů $L_{mag} = 50$ mm.

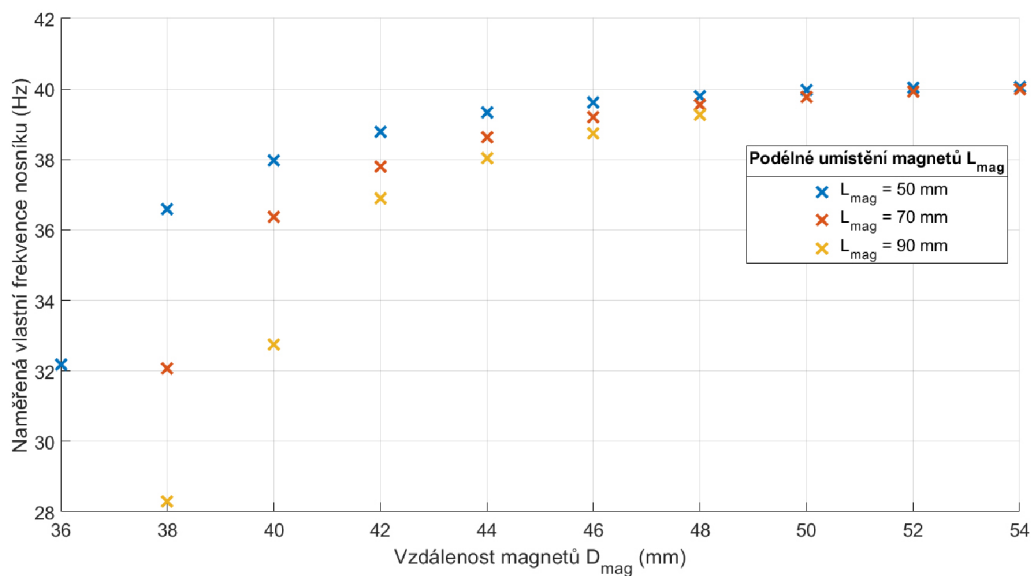


Obrázek 41: Rezonanční oblasti pro různé D_{mag} , podélné umístění magnetů $L_{mag} = 70$ mm.



Obrázek 42: Rezonanční oblasti pro různé D_{mag} , podélné umístění magnetů $L_{mag} = 90$ mm.

Z předchozích grafů se vlastní frekvence pro různé umístění magnetů vynesly do následujícího grafu:



Obrázek 43: Naměřená vlastní frekvence v závislosti na umístění magnetů. Se zmenšující vzdáleností magnetů D_{mag} klesá vlastní frekvence. Se zvětšující podélnou vzdáleností magnetů L_{mag} klesá vlastní frekvence.

Experiment potvrdil, že působením magnetů dochází ke snižování vlastní frekvence nosníku. Nejnížší vlastní frekvence bude pro $\min(D_{mag})$ a $\max(L_{mag})$.

Závěr

Cílem této práce bylo ukázat možné aplikace genetického algoritmu v mechanice. Aplikace GA byla úspěšně ukázaná pro optimalizaci tvaru vrubu, zvýšení tuhosti ohýbaných segmentů pomocí žebrování a pro předsání funkčních posuvů.

Tvar vrubu byl popsán dvěma způsoby: Bézierovou křivkou a lomenou čarou. Oběma popisy tvaru vrubu se povedlo snížit součinitele koncentrace napětí vzhledem k referenčnímu vrubu. Pro optimalizaci vrubu byla kritériální funkce implementovaná pomocí softwaru FreeFem++.

Pro optimalizaci tuhosti ohýbaných segmentu a předsání funkčních posuvů byla hodnota kritériální funkce zjištěna pomocí Matlabu, což zahrnovalo naprogramování vlastní metody konečných prvků. Tím se zrychlil běh GA. Na uvedených příkladech se ukázalo, že optimálním rozložením hmoty se docílí zvýšení tuhosti až o desítky procent.

Pro nalezení optimálních poloh magnetů nebyl použit genetický algoritmus kvůli jednoznačnosti optima. Magnety svým působením na nosník snižují vlastní frekvence nosníku. Tento závěr byl experimentálně potvrzen.

Seznam použitých zdrojů

- [1] T. Weise, *Global Optimization Algorithms - Theory and Application*, p. 95. Self-Published, 2009.
- [2] T. Weise, *Global Optimization Algorithms - Theory and Application*, p. 45. Self-Published, 2009.
- [3] T. Weise, *Global Optimization Algorithms - Theory and Application*, p. 26. Self-Published, 2009.
- [4] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, p. 60. John Wiley & Sons Inc, 2003.
- [5] T. Weise, *Global Optimization Algorithms - Theory and Application*, p. 148. Self-Published, 2009.
- [6] F. Hecht, “New development in freefem++,” *J. Numer. Math.*, vol. 20, no. 3-4, pp. 251–265, 2012.
- [7] www.comsol.com, “Comsol multiphysics® v. 5.6. comsol ab,” *Stockholm, Sweden*.