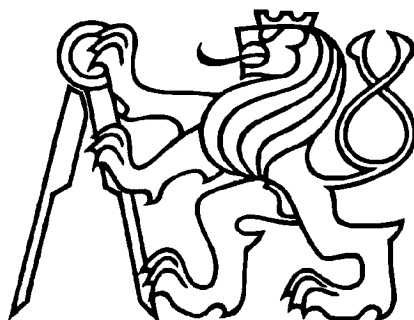


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

FAKULTA STAVEBNÍ



Rozšířené simulované žíhání
aplikované na cenovou optimalizaci návrhu
železobetonového nosníku

Matěj Lepš

Praha, prosinec 2000

Na tomto místě bych chtěl co nejsrdečněji poděkovat Prof. Ing. Zdeňkovi Bittnarovi, DrSc. a Ing. Michalovi Šejnohovi, Ph.D. za jejich podporu, nekonečnou trpělivost, za rady a inspirace nejen při vypracovávání této práce.

Dále bych chtěl poděkovat Ing. Janu Zemanovi a Ing. Karlu Matoušovi, Ph.D. za jejich pomoc a užitečné rady nejen z oblasti optimalizačních algoritmů.

Zpracování této práce bylo podpořeno projektem MŠMT č. VS 97 040.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
FAKULTA STAVEBNÍ

Abstrakt

Rozšířené simulované žíhání
aplikované na cenovou optimalizaci návrhu
železobetonového nosníku

Matěj Lepš

V této práci je představena aplikace optimalizačních strategií, založených na principech genetických algoritmů, spolu s problémem návrhu železobetonového prvku. V konkrétním případě se jedná o minimalizaci ceny konstrukce. Dosažený výsledek musí splňovat nejen požadavek nízké ceny, ale též musí vyhovět požadavkům únosnosti a použitelnosti pro zadané zatížení.

Nejprve jsou představeny jednotlivé parametry a podmínky ovlivňující cenu konstrukce. Je též popsán postup vytvoření optimalizované funkce spolu se začleněním omezení daných zvoleným normovým předpisem.

K řešení takto složitého problému s mnoha omezujícími podmínkami je nutný efektivní optimalizační algoritmus. V tomto případě je použito několik stochastických metod. Jejich porovnáním vyplynula převaha genetického algoritmu spojeného s principem simulovaného žíhání.

Za příklad pro tuto metodu je vzata analýza jednoduchého železobetonového spojitého nosníku. Výsledky ukazují možnosti úspory financí při navrhování tohoto typu konstrukcí.

Obsah

1	Úvod	1
1.1	Modelování a optimalizace železobetonu	2
1.2	Genetické algoritmy a simulované žíhání	3
1.3	Prezentovaná práce	5
2	Model železobetonového nosníku	6
2.1	Objektivní funkce	6
2.2	Výpočetní model	7
2.3	Návrh podélné výztuže	9
2.4	Posouzení podélné výztuže	10
2.5	Návrh a posudek smykové výztuže	10
2.6	Posouzení mezního stavu přetvoření	12
2.6.1	Průřez bez trhliny	12
2.6.2	Průřez s trhlinou	14
2.7	Penalizační funkce	15
2.8	Optimalizované parametry nosníku	17
3	Optimalizační postupy	20
3.1	Kódování dat	20
3.2	Stochastické optimalizační metody	22
3.2.1	Genetické algoritmy	22
3.2.2	Genetické operátory	22

3.2.3	Výběr jedinců	24
3.2.4	Genetický algoritmus I (GA I)	28
3.2.5	Genetický algoritmus II (GA II)	28
3.3	Rozšířené simulované žíhání	29
3.3.1	Simulované žíhání (SA)	29
3.3.2	Algoritmus AUSA	30
3.3.3	Návaznost penalizačních funkcí	32
4	Výsledky	33
4.1	Příklad nosníku	33
4.2	Výsledky optimalizace tvaru průřezu	34
4.3	Optimalizace tvaru průřezu a ohybové výztuže	35
4.4	Maximální optimalizační problém	36
4.4.1	Optimalizace s diskrétními hodnotami dimenzí	38
4.4.2	Zamyšlení nad volbou parametrů	40
4.4.3	Optimalizace se spojitými hodnotami dimenzí	42
5	Závěr a budoucí směry výzkumu	45
	Literatura	47

Kapitola 1

Úvod

Hledání efektivního návrhu betonových konstrukcí není nový problém. V současné době je ale tomuto problému kladen větší důraz, zejména z důvodu širokého rozšíření betonu ve stavebnictví. Mnohé výzkumy se zabývají zejména optimální hmotností konstrukce ve vztahu k zadaným podmínkám. Ve většině případů je řešen problém návrhu konstrukce s maximálním poměrem únosnost/hmotnost, např. [Mare and Holy, 2000] [Laníková, 1999], [Lampinen and Zelinka, 1999] je jen několik málo zástupců z mnoha v poslední době.

Z důvodu rozšíření a zobecnění optimalizovaného problému byl přidán faktor ceny. Tím se uvolnil prostor pro přirozenější pohled na celou problematiku optimalizace návrhu, což by mohla být v budoucnu výhoda pro nasazení v konstrukční praxi.

Za předmět optimalizace je zvolen ocelí vyztužený beton. Ocel je charakterizována vysokou pevností a duktilitou, zatímco u betonu je výhodný poměr ceny k pevnosti v tlaku. Kombinací těchto vlastností, vhodným poměrem obou materiálů a jejich vzájemným uspořádáním lze získat poměrně levnou konstrukci, která vyhovuje zadaným normovým podmínkám. Největší úspory lze dosáhnout vyváženým návrhem ohybové a smykové vyztuže v návaznosti na dimenze betonového průřezu.

1.1 Modelování a optimalizace železobetonu

Modelování objektů je v současnosti jedna z nejvíce aktivních oblastí výzkumu. Důvodem je požadavek z oblasti praxe na rychlý vývoj příslušného softwaru. Při analýze zkoumané konstrukce je potřeba vyřešit dva značně rozdílné problémy:

Idealizace zahrnuje abstrakci geometrických vlastností. Tyto požadavky jsou pokryty popisem jednotlivých prvků konstrukce spolu s popisem např. spojů nebo podpor, vše v návaznosti na konečné prvky.

Diskretizace je proces transformace idealizované konstrukce do souboru elementů, který je schopen konstrukci přesně reprezentovat. Při studiu železobetonové konstrukce může jít diskretizace až na úroveň jedné ocelové výztužné vložky. Takto komplexní problém by byl z pohledu návrhu zřejmě značně výpočtově náročný a i z pohledu praxe je zde nutné zjednodušení, které je v této práci charakterizováno shodnými profily jednotlivé výztuže po celé délce nosníku spolu se skokovými hodnotami rozměrů průřezu.

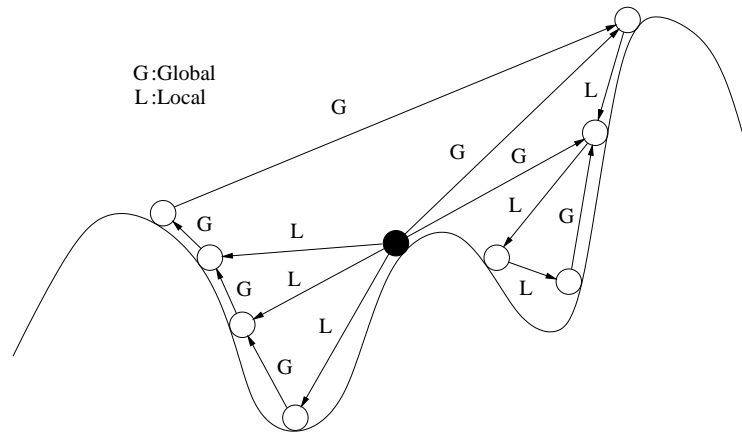
Proces optimalizace je prováděn na úrovni diskretizovaného modelu, ale z důvodu získání např. vnitřních sil musí s idealizovaným modelem spolupracovat. V současnosti je takto složitý problém často převáděn do lineární oblasti, kde je možné využít možnosti lineárního programování [Chudoba, 1995, Laníková, 1999]. Nevýhodami tohoto přístupu jsou nutnost přeformulování objektivní funkce do lineárního tvaru a nemožnost optimalizace nelineárních závislostí. Pokud se chceme vyhnout těmto problémům, je vhodné použít některý z genetických algoritmů nebo simulované žíhání. Tyto algoritmy vyžadují pouze hodnotu objektivní funkce (bez znalosti její formulace) a jsou schopny řešit i nelineární problémy.

1.2 Genetické algoritmy a simulované žíhání

Genetické algoritmy jsou velmi přizpůsobivé metody, které slouží k prohledávání zadaného prostoru a k řešení optimalizačních problémů. Základní princip je okopírován z genetických procesů živých organismů - jmenovitě z vývinu a samoorganizování přírodních systémů přes mnoho generací v závislosti na metodě přirozeného výběru a “přežití nejsilnějšího jedince”. Jednotlivá individua v populaci spolu soupeří o přírodní zdroje nebo o vliv nad ostatními. Jedinci, kteří jsou v tomto boji více úspěšní, budou mít více potomků, zatímco slabší individua obvykle zahynou bez následovníků. Tímto způsobem se vlastnosti “vítězů” dostávají do dalších generací a výsledkem je velice rychlá adaptace na okolní prostředí. Genetické algoritmy kopírují všechny uvedené procesy - pracují s *populací* jedinců, přičemž každý reprezentuje jedno řešení daného problému. Jedinec je v závislosti na úspěchu ohodnocen *váhou* nebo *silou*. U individuí s větší silou je větší pravděpodobnost, že budou vybráni do procesu *křížení* a díky tomu se jejich vlastnosti okopírují do jejich *potomků*. Jedince též může ovlivnit náhodnost prostředí v podobě *mutací*. Výborné vlastnosti se kombinují, dopňují a prorůstají celou populací až vývoj jedinců přes množství úspěšných populací nakonec dospěje do optimálního řešení.

Nejstarší prací na toto téma nejspíš bylo Darwinovo dílo *O původu druhů*, kde byly formulovány výše popsané principy. Základní teze genetických algoritmů byly poprvé odborně popsány v [Holland, 1975], jejich rozvedení a doplnění následovalo v dalších letech, uveďme např. [Goldberg, 1989, Michalewicz, 1992, Beasley et al., 1993a]. V klasické verzi pracují genetické algoritmy s binární reprezentací objektivních proměnných. Michalewicz v díle [Michalewicz, 1992] představil genetické algoritmy založené na reálném kódování proměnných a ukázal jejich přednosti při optimalizaci reálných funkcí. Z těchto základů nyní vychází i nejmladší optimalizační metoda nazývaná *diferenciální evoluce* [Storn, 1996].

Metoda *simulovaného žíhání* je založena více na fyzikálních než biologických principech - optimalizační algoritmus efektivně kopíruje proces žíhání v pevných látkách. Při žíhání je nejprve teplota nastavena na vysokou hodnotu a pak je postupně pomalu snižována tak,



Obr. 1.1: Princip metody simulovaného žíhání

aby jednotlivé částičky materiálu měly čas zaujmout stav s minimální energií na dané teplotní hladině. Jak teplota klesá dolů, klesá i celková energie materiálu až dosáhne své minimální hodnoty. Algoritmus simulovaného žíhání pracuje na stejném principu - je náhodně vytvořena počáteční populace, virtuální parametr nazývaný *teplota* je nastaven na startovací hodnotu a je náhodně vytvořeno další řešení. Pokud je z pohledu síly nové řešení lepší než staré, je automaticky nové přijato do populace. Pokud je horší, stále zůstává šance, že nahradí staré řešení - tato pravděpodobnost je závislá na stavu populace a aktuální teplotě, která umožní řešení uniknout z lokálního minima (Obr. 1.1). Proces je opakován několikrát při konstantní teplotě, která je postupně snižována do předepsaného minima. Tato verze algoritmu byla poprvé představena v [Kirkpatrick et al., 1983] a nezávisle v [Cerny, 1985]. Rozvedení základních principů a další aspekty této metody jsou uvedeny např. v [Ingber, 1993, Ingber, 1995, Kvasnička, 1993].

V díle [Mahfoud and Goldberg, 1992] byla představena efektivní kombinace genetických algoritmů a metody simulovaného žíhání. Využívá základy genetických algoritmů (populace chromozomů a genetické operátory) spolu s principem vyhledávání řešení směrem ke stavům s minimální energií. Poznámky a doporučení k implementaci jsou uvedeny v [Kvasnička, 1994].

1.3 Prezentovaná práce

Hlavním tématem této práce je aplikace výše uvedených algoritmů při analýze vlivu jednotlivých parametrů železobetonového nosníku na celkovou cenu konstrukce.

Kapitola 2 formuluje objektivní funkci se započítáním vlivu omezujících podmínek. Je představen prvek metody konečných prvků, který je použit pro výpočet vnitřních sil na zatížené konstrukci. Jsou podrobně popsány některé normové požadavky vztahující se na získané vnitřní síly a zjištěné deformace. Splnění či nesplnění těchto podmínek je následně zahrnuto do optimalizované funkce metodou penalt. Na závěr tohoto oddílu jsou představeny jednotlivé proměnné objektivní funkce.

V kapitole 3 je nejprve zmíněn princip převodu dat mezi proměnnými optimalizované funkce a binárním prostorem, na kterém je optimální řešení vyhledáváno. Genetické operátory a metody výběru v závislosti na relativní úspěšnosti jednotlivých řešení jsou základní kameny genetických algoritmů. Přehled tří vybraných algoritmů následuje spolu s popisem některých implementovaných úprav.

V kapitole 4 jsou prezentovány výsledky aplikace uvedených metod na některé optimalizační problémy spjaté s příkladem spojitého železobetonového nosníku. První problém se týká pouze optimalizace dimenzí průřezu, druhý započítává do ceny vliv ohybové výztuže a třetí příklad si bere na mušku všechny předchozí vlivy spolu se smykovou výztuží. Tímto krokováním je umožněno sledovat vliv jednotlivých stavebních částí nosníku na celkovou cenu.

Kapitola 2

Model železobetonového nosníku

Nejprve bude nastíněna základní optimalizační otázka, tedy cíl optimalizace u spojitého nosníku. Stručně bude zmíněn postup získání vnitřních sil. Dále budou zmíněny některé požadavky kladené na daný nosník, jejich posouzení a začlenění do optimalizované funkce. Proměnné, ovlivňující tuto funkci, jsou diskutovány v poslední části této kapitoly.

2.1 Objektivní funkce

V průběhu optimalizace budeme potřebovat od sebe odlišit jednotlivé návrhy nosníků pomocí jednoho parametru, podle jehož velikosti jsme schopni určit, které řešení je lepší a které horší. Tento parametr je dán objektivní funkcí. Jak jsme se zmínili v úvodu, kritériem našeho rozhodování bude celková cena nosníku. Objektivní funkce tak nabývá tvar

$$f(\mathbf{X}) = \frac{V_c P_c + W_s P_s}{P_{norm}}, \quad (2.1)$$

podmíněný následujícími podmínkami

$$\delta_i \leq \delta_{lim} = \frac{l}{250} \quad (2.2)$$

$$M_{Sd} \leq M_{Rd} \quad (2.3)$$

$$|V_{Sd}| \leq V_{Rd2} \quad |V_{Sd}| \leq V_{Rd3} . \quad (2.4)$$

V rovnici (2.1) je V_c objem betonu a W_s hmotnost oceli; P_c a P_s jsou cena betonu za objemovou jednotku a cena oceli za hmotnostní jednotku. Hodnota P_{norm} je cena porovnávacího návrhu, kterou normujeme hodnotu objektivní funkce. Nerovnice (2.2)-(2.4) představují vybrané omezující podmínky normy Eurokód 2 (dále jen EC2) kladené na železobetonový nosník. Jmenovitě nerovnost (2.2) reprezentuje podmínku mezního stavu přetvoření, kde l je rozpětí a δ_{lim} je maximální povolený průhyb. Nerovnosti (2.3) a (2.4) se týkají mezních stavů únosnosti v ohybu a ve smyku. Jednotlivé podmínky budou vysvětleny později.

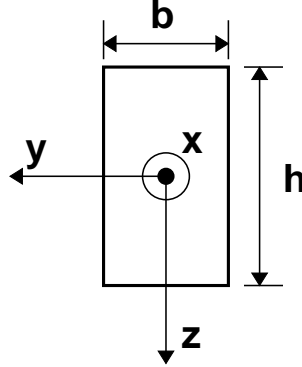
Vnitřní síly označené indexem k jsou odvozené od charakteristických hodnot zatížení, veličiny s indexem d jsou odvozené od návrhových hodnot zatížení. Např. M_{Sd} je moment vnitřních sil (index S značí vnitřní sílu) vyvolaný návrhovým zatížením. Únosnost označujeme indexem R ; je-li odvozena z charakteristických pevností, připojujeme index k , je-li odvozena z výpočtových pevností, pak připojujeme index d . Tak např. M_{Rd} je únosnost průřezu v ohybu. Definice charakteristické/návrhové zatížení a charakteristická/výpočtová pevnost je uvedena v normě Eurokód.

2.2 Výpočetní model

Uvažujme nosník s obdélníkovým průřezem (Obr. 2.1) a danými dimenzemi b a h . Zaměříme svoji pozornost na 2D problém, tj. na ohyb a smyk v rovině xz . Pro návrh a posouzení je nutno znát vnitřní síly v příčném řezu, které se obvykle získají metodou konečných prvků.

V našem případě je použit postup založený na Mindlinově kinematické hypotéze. Pokud vztah pro podélné posunutí u

$$u(x, z) = u_0(x) + \phi_y(x)z. \quad (2.5)$$



Obr. 2.1: Průřez nosníku a systém souřadnic

dosadíme do Hookova zákona, obdržíme výrazy pro vnitřní síly

$$\begin{aligned} N_x &= EA \frac{du_0}{dx}, \\ M_y &= EI_y \frac{d\phi_y}{dx}, \\ Q_z &= kGA \left(\phi_y + \frac{dw}{dx} \right). \end{aligned} \quad (2.6)$$

V rovnicích (2.5)-(2.6) představuje u_0 posun průřezu ve směru osy x a ϕ_y je rotace průřezu kolem osy y ; A a I_y je plocha a moment setrvačnosti; E a G je Youngův a smykový modul železobetonu. Parametr k je smykový opravný součinitel, který pro obdélník nabývá hodnoty $5/6$. Aby nevzniklo smykové ztuhnutí, musí být zkosení průřezu γ

$$\gamma = \left(\phi_y + \frac{dw}{dx} \right),$$

konstantní po celém elementu. Pro tento účel zavedeme bublinovou funkci N_c

$$N_c = 1 - \xi^2.$$

Aproximace svislých posunutí pak nabývá tvaru

$$w = N_1 w_1 + N_2 w_2 + \alpha N_c, \quad (2.7)$$

zatímco pro pootočení průřezu platí standardní isoparametrická interpolace

$$\phi_y = N_1 \phi_1 + N_2 \phi_2, \quad (2.8)$$

kde w_i , ϕ_i jsou uzlové posuny a pootočení; N_1 , N_2 jsou lineární bázové funkce. Aby byl splněn konstantní průběh zkosení, musí být [Bittnar and Šejnoha, 1992]

$$\alpha = \frac{1}{8}(\phi_2 - \phi_1).$$

2.3 Návrh podélné výztuže

Postup uvedený v tomto odstavci je použit pro návrh porovnávacího nosníku, jehož cena je stálou složkou penalizační funkce (2.1). Uvažujme ohýbaný jednostranně vyztužený obdélníkový průřez na Obr. 2.2 . Formulujeme dvě podmínky rovnováhy:

$$F_C - F_S = 0 , \quad (2.9)$$

$$M_{Rd} = F_S z , \quad (2.10)$$

kde

$$F_S = A_S f_{yd} , \quad (2.11)$$

$$F_C = 0.81xb \alpha f_{cd} \quad (2.12)$$

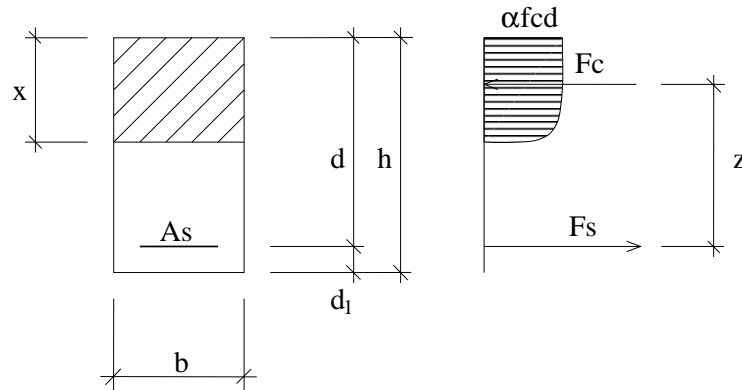
za předpokladu parabolicko-rectangulárního rozdělení napětí v betonu. Z podmínky (2.2) a rovnic (2.9)-(2.12) získáme vztah pro výpočet potřebné plochy výztuže

$$A_S = bd \frac{\alpha f_{cd}}{f_{yd}} \left(1 - \sqrt{1 - \frac{2M_{Sd}}{bd^2 \alpha f_{cd}}} \right). \quad (2.13)$$

Materiálové veličiny, výpočtové napětí v betonu αf_{cd} a výpočtové napětí v oceli f_{yd} , budou zadány a nebudou během optimalizace měněny. Ohybový moment M_{Sd} je závislý na zadaném zatížení a geometrii konstrukce. Ve vzorci (2.13) tak zůstaly skryty tři neznámé veličiny - šířka a výška průřezu b , h a profil zvolené výztuže ϕ , které je nutno zvolit. Účinná výška d , pro pruty umístěné v jedné řadě, je pak určena vztahem

$$d = h - d_1 \quad \text{kde} \quad d_1 = c + \frac{\phi}{2} . \quad (2.14)$$

Velikost krytí výztuže c je tabelována v závislosti na zvoleném profilu ϕ a stupni agresivity prostředí.



Obr. 2.2: Ohýbaný jednostranně vyztužený obdélníkový průřez

2.4 Posouzení podélné výztuže

Musíme zkontrolovat, zda moment únosnosti průřezu M_{Rd} vyhovuje podmínce (2.3). Jeho velikost je dána vztahem

$$M_{Rd} = A_S f_{yd} (d - 0.416x), \quad x = \frac{A_S f_{yd}}{0.81 b \alpha f_{cd}}. \quad (2.15)$$

Dále je nutno ověřit, zda jsou splněny předpoklady minimálního stupně vyztužení

$$A_S \geq \frac{0.6}{f_{yk}} bd, \quad (2.16)$$

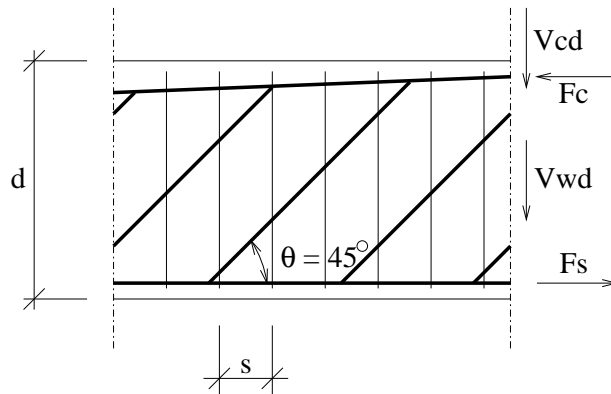
$$A_S \geq 0.0015 bd. \quad (2.17)$$

Nesplnění podmínek (2.16) a (2.17) je řešeno úpravou výšky průřezu h . Z tohoto důvodu není nutné přidávat tuto podmínku do formulace objektivní funkce.

2.5 Návrh a posudek smykové výztuže

K návrhu a posouzení smykové výztuže je užitá standardní metoda výpočtu smykové únosnosti [Procházka, 1995]. Vychází z analogie příhradové konstrukce s tlakovými diagonálami skloněnými pod úhlem $\theta = 45^\circ$ (Obr. 2.3). Posouvající síla V_{cd} přenášená podle EC2 tlačným betonem je dána výrazem

$$V_{cd} = bd \tau_{Rd} k (1.2 + 40 \rho_1), \quad (2.18)$$



Obr. 2.3: Model smykového prvku podle standardní metody EC2

kde

$k = 1$ pro prvky s 50% spodní výztuže dovedené do podpory, jinak

$$k = 1.6 - d [m] \geq 1.0 ,$$

$$\rho_1 = \frac{A_s}{bd} \leq 0.02 ,$$

$$\tau_{Rd} = \frac{1}{4} \frac{f_{ctk0.05}}{\gamma_C} \quad \text{je základní hodnota smykové pevnosti betonu.}$$

Požadovaný stupeň smykového vyztužení musí dosáhnout hodnoty

$$\rho_{wd} = \frac{|V_{Sd}| - V_{cd}}{f_{ywd} b \cdot 0.9d} . \quad (2.19)$$

Smyková výztuž je tvořena pouze třímínky. Při jejich návrhu jsou neznámé tři veličiny: vzdálenost třímínek s , průměr třímínků ϕ_{sw} a počet větví třímínek n_s . Dvě z těchto hodnot je nutné volit a třetí určíme výpočtem:

$$s_d = \frac{\pi \phi_{sw}^2 n_s}{4 \rho_{wd} b} \leq s_{max} , \quad (2.20)$$

$$\phi_{swd} = \sqrt{\frac{4}{\pi} \frac{\rho_{wd} b s}{n_s}} \leq \phi_{sw,max} , \quad (2.21)$$

$$n_{sd} = \frac{4}{\pi} \frac{\rho_{wd} b s}{\phi_{sw}} \geq n_{s,min} \quad (s_t \leq s_{t,max}). \quad (2.22)$$

Hodnoty maxim ve vzorcích (2.20)-(2.22) jsou uvedeny např. v [Procházka et al., 1997].

Při posouzení smyku v tomto mezním stavu nesmí dojít k vyčerpání únosnosti tlakových diagonál (2.4a), kde

$$\begin{aligned} V_{Rd2} &= 0.5 \nu f_{cd} b 0.9d , \\ \nu &= 0.7 - f_{ck}/200 \geq 0.5 . \end{aligned} \quad (2.23)$$

Smyková síla na mezi únosnosti třmínkové výztuže

$$V_{wd} = \frac{\pi}{4} \frac{\phi_{sw}^2 n_s}{s} f_{ywd} 0.9d , \quad (2.24)$$

po dosazení do vztahu pro celkovou smykovou únosnost průřezu

$$V_{Rd3} = V_{cd} + V_{wd} , \quad (2.25)$$

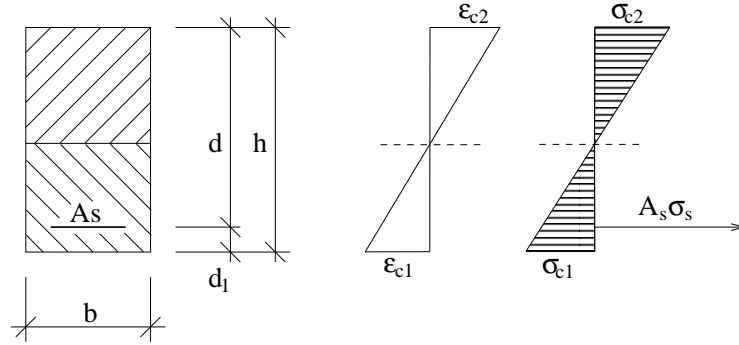
musí vyhovět podmínce (2.4b).

2.6 Posouzení mezního stavu přetvoření

Požadavky druhého mezního stavu ověříme porovnáním vypočtených přetvoření s limitním průhybem podle nerovnice (2.2). Postup vychází z modelu prvku složeného ze dvou částí - části bez trhlin a části s vyloučením působení betonu v tahu, což odpovídá vzniku trhliny.

2.6.1 Průřez bez trhliny

V průřezu bez trhliny se předpokládá plné působení betonového průřezu a pružné chování betonu i výztuže v tahu i tlaku (Obr. 2.4).



Obr. 2.4: Průřez bez trhliny

Nejprve je nutno určit veličiny ideálního průřezu. Uvažuje se poměr $\alpha_e = E_s/E_{cm}$, kde $E_s = 200$ MPa je modul pružnosti betonářské výztuže a E_{cm} je sečnová hodnota modulu pružnosti betonu. Plocha ideálního průřezu pak je

$$A_i = A_c + \alpha_e A_s \quad (2.26)$$

a vzdálenost těžiště ideálního průřezu od horního okraje je dána vztahem

$$a_{gi} = \frac{A_c a_c + \alpha_e A_s d}{A_i} . \quad (2.27)$$

Odpovídající moment setrvačnosti získáme ze vztahu

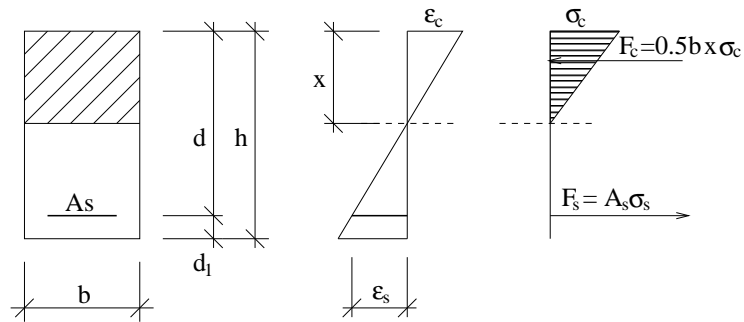
$$I_i = I_c + A_c (a_{gi} - a_c)^2 + \alpha_e A_s (a_{gi} - d)^2 . \quad (2.28)$$

Ve vzorcích (2.26)-(2.28) je A_c plocha betonové části průřezu, A_s plocha betonářské výztuže, I_c moment setrvačnosti betonové části průřezu k vlastní těžišťové ose a a_c vzdálenost těžiště betonu od horních vláken.

Pokud napětí betonu v dolních vláknech průřezu namáhaného ohybovým momentem M_{Sk}

$$\sigma_{c1} = \frac{M_{Sk}(h - a_{gi})}{I_i} \quad (2.29)$$

nepřesáhne střední hodnotu pevnosti v tahu, tedy $\sigma_{c1} \leq f_{ctm}$, nevzniknou trhliny kolmé ke střednici prvku a do výpočtu mezního stavu se použijí veličiny spočteného ideálního průřezu. V opačném případě je nutno uvažovat vznik trhliny.



Obr. 2.5: Průřez s trhlinou za ohybu

2.6.2 Průřez s trhlinou

Výpočet průřezu s trhlinou vychází z předpokladu rovinnosti deformovaného průřezu a z předpokladu nepůsobení betonu v tažené oblasti (Obr. 2.5). Pro výpočet přetvoření na úrovni prvku zavádí Eurokód tzv. průměrnou křivost průřezu $\frac{1}{r_m}$ a koeficient ζ udávající spolupůsobení betonu mezi trhlinami

$$\frac{1}{r_m} = (1 - \zeta) \frac{1}{r_1} + \zeta \frac{1}{r_2}, \quad (2.30)$$

$$\zeta = 1 - \beta_1 \beta_2 \left(\frac{M_{cr}}{M_{Sk}} \right)^2, \quad \text{kde je} \quad (2.31)$$

$$\frac{1}{r_1} = \frac{f_{ctm}}{E_c(h - a_{gi})} \quad \text{křivost průřezu na mezi vzniku trhlin (při } M_{Sk} = M_{cr}),$$

$$\frac{1}{r_2} = \frac{\varepsilon_s + \varepsilon_c}{d} \quad \text{křivost průřezu po vzniku trhliny (} M_{Sk} > M_{cr}),$$

β_1 součinitel vyjadřující vliv soudržnosti výztuže,

β_2 součinitel vlivu doby trvání nebo opakování zatížení,

$$M_{cr} = \frac{I_i f_{ctm}}{h - a_{gi}} \quad \text{moment při vzniku trhlin.}$$

Ze vzorce (2.30) lze vypočítat sníženou tuhost EI (2.32) porušeného průřezu všech prvků konstrukce a s novými tuhostmi znovu přepočítat deformace celé konstrukce

$$\frac{1}{r_m} = \frac{M_{Sk}}{EI} . \quad (2.32)$$

Dvě neznámé veličiny ε_s a ε_c lze určit ze silové podmínky

$$\begin{aligned} F_s &= F_c , \text{ kde} \\ F_s &= A_s \varepsilon_s E_s , \\ F_c &= 0.5bx\varepsilon_c E_{cm} , \end{aligned} \quad (2.33)$$

z momentové podmínky k působišti síly F_c

$$M_{Sk} = F_s \left(d - \frac{x}{3} \right) \quad (2.34)$$

a z jednoho geometrického vztahu:

$$\varepsilon_s = \frac{M_{Sk}}{A_s E_s \left(d - \frac{x}{3} \right)} , \quad (2.35)$$

$$\varepsilon_c = \frac{\varepsilon_s x}{d - x} , \text{ kde} \quad (2.36)$$

$$x = \frac{\sqrt{(A_s E_s)^2 + 2bE_{cm}dA_s E_s} - A_s E_s}{bE_{cm}} . \quad (2.37)$$

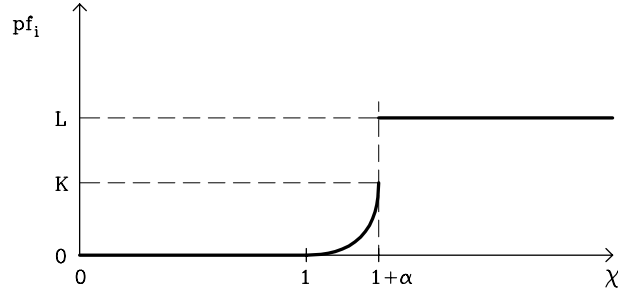
2.7 Penalizační funkce

Pro zahrnutí vlivu podmínek (2.2)-(2.4) na cenu nosníku je použit přístup pomocí penalizačních funkcí [Michalewicz, 1995]. Původní objektivní funkce (2.1) je rozšířena o penalty, vyjadřující splnění či nesplnění daných omezení

$$f(\mathbf{X}) = V_c P_c + W_s P_s + \sum_{i=1}^4 p f_i . \quad (2.38)$$

Penalizační funkce $p f_i$ zavádíme ve tvaru, který je vyobrazen na Obr. 2.6. Uvažujme proměnnou Φ_i , která nesmí překročit mezní hodnotu $\Phi_{i,max}$

$$\Phi_i \leq \Phi_{i,max} . \quad (2.39)$$



Obr. 2.6: Penalizační funkce

K popisu penalizační funkce pf_i zavádíme parametr χ tak, že

$$\chi = \frac{\Phi_i}{\Phi_{i,max}} \quad . \quad (2.40)$$

Jak je vidět na Obr. 2.6, funkce pf_i je na uzavřeném intervalu $\langle 0, 1 \rangle$ rovna nule. Pokud je χ větší než hodnota $1 + \alpha$, nabývá funkce hodnoty L a na otevřeném intervalu $(1, 1 + \alpha)$ má pf_i tvar

$$pf_i = K_i \left(\frac{\chi - 1}{\alpha_i} \right)^{\beta_i} \quad , \quad (2.41)$$

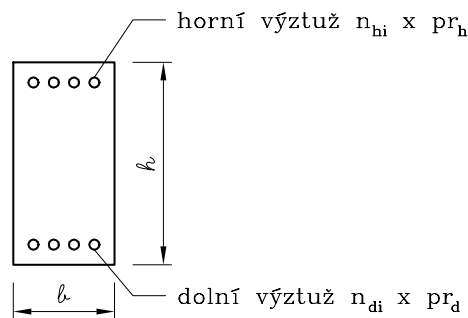
kde α , β , K a L jsou uživatelem definované parametry. Číslo α udává procentuální hranici mezi přípustnou a nepřípustnou hodnotou Φ_i . Parametr β určuje křivost křivky a obvykle je $\beta \geq 1$. Parametr K je menší nebo roven L . Pokud nebudou splněny podmínky (2.2)-(2.4) a proměnné z těchto podmínek se budou nalézat v hranicích daných parametrem α , můžeme objektivní funkci (2.38) zapsat ve tvaru

$$\begin{aligned} f(\mathbf{X}) = V_c P_c + W_s P_s \quad + \quad & K_1 \left(\frac{\frac{\delta_i}{\delta_{lim}} - 1}{\alpha_1} \right)^{\beta_1} + K_2 \left(\frac{\frac{M_{Sd}}{M_{Rd}} - 1}{\alpha_2} \right)^{\beta_2} \\ & + K_3 \left(\frac{\frac{V_{Sd}}{V_{Sd2}} - 1}{\alpha_3} \right)^{\beta_3} + K_4 \left(\frac{\frac{V_{Sd}}{V_{Rd3}} - 1}{\alpha_4} \right)^{\beta_4} \quad . \quad (2.42) \end{aligned}$$

Volené parametry, nastavené pro všechny podmínky shodně, jsou vypsány v tabulce 2.1.

K	L	α	β
1.0	20.0	0.15 – 0.01	2.0

Tab. 2.1: Parametry penalizačních funkcí



Obr. 2.7: Průřez nosníku

2.8 Optimalizované parametry nosníku

Začneme-li studovat podrobněji rovnici (2.38), první neznámou je objem betonu. Rozpětí nosníku zůstává v průběhu optimalizace konstantní, takže sledovanými proměnnými jsou dimenze průřezu (Obr. 2.7), kde h a b je výška a šířka průřezu. V tabulce 2.2 jsou vypsány horní a dolní meze rozměrů spolu s požadovanou přesností. Jsou zvoleny dva přístupy, z nichž první, “inženýrský”, užívá diskrétní hodnoty skokově se měnící po 25 mm, zatímco druhé řešení, “matematické”, využívá spojitých hodnot.

Další neznámé v rovnici (2.38) se týkají ocelových výztužných prvků. V tabulce 2.3 je uvedeno zbývajících 19 parametrů, které jsou potřebné k popisu ohybové a smykové výztuže. Je potřeba zdůraznit, že všechny tyto proměnné jsou celá čísla.

Proměnné pr_h a pr_d v tabulce 2.3 reprezentují průřezy ohybové výztuže, shodné po celé délce prutu, umístěné u horního a dolního povrchu (Obr. 2.8). Velikosti profilů pr_h a pr_d jsou vybírány z hodnot uvedených v tabulce 2.4. Parametry $n_{dI} - n_{dIII}$ odpovídají

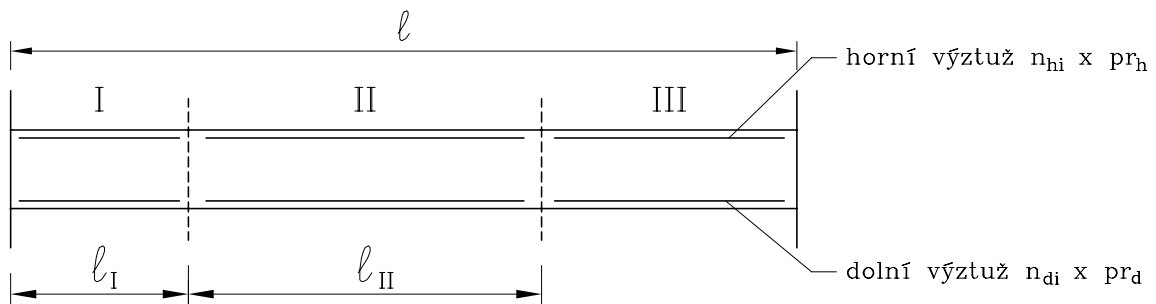
Proměnná	Jednotky	Minimum	Maximum	Přesnost	Poznámka
h	[m]	0.15	0.85	0.025	Diskrétní hodnoty
		0.15	0.85	0.001	Spojitě hodnoty
b	[m]	0.15	0.45	0.025	Diskrétní hodnoty
		0.15	0.45	0.001	Spojitě hodnoty

Tab. 2.2: Rozměry příčného řezu

Proměnná	Minimum	Maximum	Přesnost	Poznámka
OHYBOVÁ VÝZTUŽ				
pr_h, pr_d	1	16	1	Indexy vektoru reálných čísel
$n_{dI} - n_{dIII}$	0	31	1	
$n_{hI} - n_{hIII}$	0	31	1	
l_I, l_{II}	0	12	1	Příslušný počet elementů
SMYKOVÁ VÝZTUŽ				
pr_w	1	4	1	Indexy vektoru reálných čísel
l_{w1}, l_{w2}	0	12	1	Příslušný počet elementů
$s_{w1} - s_{w3}$	0.05	0.40	0.025	Vzdálenost mezi třmínky
$p_{w1} - p_{w3}$	2	4	2	Počet stříhů

Tab. 2.3: Parametry ohybové a smykové výztuže

počtům prutů umístěných u dolního povrchu v úsecích $I - III$ a $n_{hI} - n_{hIII}$ jsou počty profilů u horního povrchu (Obr. 2.8). Proměnné l_I a l_{II} jsou spojeny s počtem elementů přidružených k jednotlivým úsekům I a II . Velikost úseku III se dopočte z jednoduché aritmetiky. Tři intervaly jsou zvoleny z důvodů snadné aplikace užitých postupů na optimalizaci rámu. Podobný význam mají parametry pro smykovou výztuž; pr_w představuje průměr třmínků shodný po celé délce prutu, vybíraný z hodnot v tabulce 2.4. Počet stříhů $p_{w1} - p_{w3}$ a vzdálenosti mezi třmínky $s_{w1} - s_{w3}$ odpovídají jednotlivým intervalům

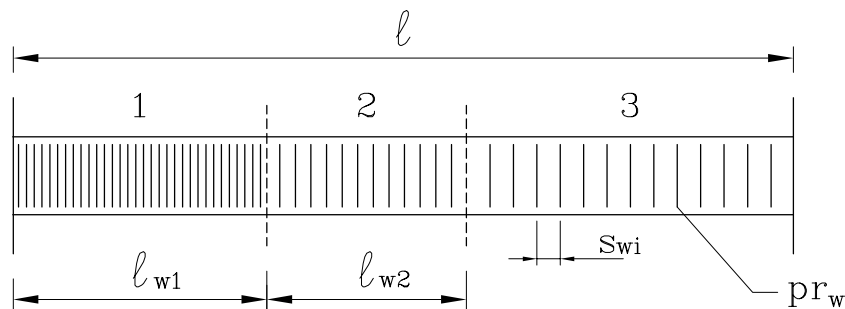


Obr. 2.8: Dělení oblastí ohybové výztuže

Typ výztuže	Průměry [mm]								
Ohybová	5.5	6.0	6.5	7.0	8.0	10.0	12.0	14.0	
	16.0	18.0	20.0	22.0	25.0	28.0	32.0	36.0	
Smyková	4.0	6.0	8.0	10.0					

Tab. 2.4: Zvolené řady profilů

$l_{w1} - l_{w3}$ (Obr. 2.9). Všechny výše uvedené parametry mohou být uloženy ve vektoru \mathbf{X} , rovnice 2.38, který se v tomto případě skládá z 21 proměnných.



Obr. 2.9: Dělení oblastí smykové výztuže

V článku [Laníková, 1999] je řešena podobná úloha, jednodušší z pohledu počtu proměnných. Autorka neuvažuje vliv průměrů výztužných vložek a soustřeďuje se na minimalizaci plochy výztuže. Tento přístup značně urychluje optimalizační fázi zvláště u výpočetně náročných problémů. Z praktického hlediska je ale nutné uvažovat vliv velikosti profilu na krytí vložek a následně na rameno vnitřních sil. Ve velkém množství případů též nelze nalezenou plochu výztuže pokrýt vybranými profily, ať z důvodů konstrukčních nebo z důvodů cenových, a proto byla v této práci velikost profilů implementována.

Kapitola 3

Optimalizační postupy

Než přistoupíme k popisu jednotlivých algoritmů, bude zmíněn způsob interpretace proměnných při optimalizaci. Následně budou stručně zmíněny některé genetické algoritmy, přičemž bude největší pozornost věnována metodě rozšířeného simulovaného žíhání.

3.1 Kódování dat

Ve většině případů optimalizační algoritmus nepracuje přímo s hodnotami proměnných objektivní funkce, ale s jejich obrazem transformovaným do určitého vhodného číselného prostoru. Převod hodnot mezi optimalizovanou funkcí a optimalizačním algoritmem nazveme kódování. V současné době se užívají dva typy kódování. První je převod do oblasti reálných čísel [Michalewicz, 1992], druhý využívá binární reprezentace dat [Goldberg, 1989]. V této práci byl z důvodů velkého množství celočíselných hodnot zvolen druhý způsob.

Uvažujme funkci $f(\mathbf{X}) = f(x_1, x_2, \dots, x_n)$, kde \mathbf{X} je vektor n proměnných, celých nebo reálných čísel x_i , definovaných na uzavřeném intervalu

$$\min_i \leq x_i \leq \max_i, \quad (3.1)$$

kde min_i a max_i jsou hranice příslušné k dané proměnné x_i z množiny reálných nebo celých čísel D_i . Dále předpokládejme, že každé číslo x_i je reprezentováno s určitou přesností p_i , která je definovaná jako nejmenší jednotka, které může x_i dosáhnout. Požadujeme-li přesnost např. na pět desetinných míst, je $p_i = 0.00001$. Pokud $D_i \subseteq N_0$ (přirozená čísla včetně nuly), potom $p_i = 1$. V případě, že $p_i = 2$, číslo x_i bude nabývat lichých nebo sudých hodnot podle zvoleného minima (3.3). Každá takováto proměnná x_i může být převedena na nezáporné celé číslo $y_i \in N_0$ (3.2).

$$y_i = \left\lceil \frac{x_i - min_i}{p_i} \right\rceil. \quad (3.2)$$

Zpětná transformace má tvar

$$x_i = y_i p_i + min_i. \quad (3.3)$$

Číslo y_i lze přirozeně v počítači implementovat jako řetězec binárních čísel délky k tak, že

$$\frac{max_i - min_i}{p_i} \leq 2^k. \quad (3.4)$$

Hodnota čísla k je dána vztahem

$$k = \left\lceil \frac{\ln \left(\frac{max_i - min_i}{p_i} \right)}{\ln 2} \right\rceil, \quad (3.5)$$

kde operátor $[z]$ značí zaokrouhlení čísla z na nejbližší vyšší celé číslo. Je zřejmé, že délka binárního řetězce je velice závislá na požadované přesnosti. Vysoká přesnost u reálných čísel tak může vést k neúspěchu celé optimalizace. V našem případě se ale tato slabost binárního kódování neprojeví.

Poznamenejme, že binární zápis vektoru \mathbf{X} je často nazýván *chromozom*, zatímco pro jednotlivé proměnné je užíván název *geny*. Pokud je vektor \mathbf{X} složen ze dvou proměnných x, y , z nichž je každé reprezentováno 8 bity, potom je *chromozom* složen ze dvou *genů* a obsahuje 16 binárních číslic.

3.2 Stochastické optimalizační metody

Při optimalizaci některých funkcí často selhávají tradiční gradientní metody, které vycházejí ze znalosti dané funkce. V mnoha případech takovéto postupy vyžadují splnění určitých předpokladů, které nejsme schopni zajistit. U takovýchto funkcí je pak zapotřebí prohledat prostor možných řešení. Na toto pole nastupují stochastické algoritmy, u nichž je zaručeno 100% nalezení minima za předpokladu nekonečného počtu iterací. Úloha se tak v podstatě redukuje z hledání minima objektivní funkce na minimalizaci počtu iterací zvolené stochastické metody. V následujících částech jsou některé tyto metody a postupy představeny.

3.2.1 Genetické algoritmy

Genetické algoritmy jsou formulovány na základě analogie s biologickým vývojem v přírodě a ve své podstatě vychází z principů Darwinovy evoluční teorie. Základní charakteristikou tohoto algoritmu je využití populace jedinců, jejichž postupným vývojem dochází k prohledávání dostupného prostoru paralelně, což zajišťuje tomuto algoritmu požadovanou robustnost. Stejně jako v přírodě, i zde spolu jednotlivá idividua soupeří v zájmu vlastního přežití tak, že silnější jedinci přežívají do dalších generací, zatímco slabší obvykle zahynou. Tento proces lze stručně popsat algoritmem 3.1.

Algoritmus 3.1 zahrnuje základní kroky cyklu jednoduchého genetického algoritmu: reprodukci (#5), rekombinační fázi (#6) a výběr nové populace (#7). V následující části budou popsány operátory, zajišťující změny na řádku 6, vysvětlení řádků 5 a 7 bude následovat.

3.2.2 Genetické operátory

Vývin nového jedince (rekombinační fáze #6), který vzniká kombinací charakteristik rodičů k dosažení lepších vlastností, je v genetickém algoritmu (dále jen GA) vyjádřen v podobě křížení (kombinační fáze) a mutace (vliv náhodnosti). Jednotlivé implemen-

```

1   $t = 0$ 
2  vytvoř populaci  $P_0$ , ohodnoť  $P_0$ 
3  while (not zastavovací - podmínka ) {
4     $t = t + 1$ 
5    vyber  $M_t$  z  $P_{t-1}$  (aplikace výběru)
6    změň  $M_t$  (aplikace genetických operátorů)
7    vytvoř  $P_t$  z  $M_t$  a ohodnoť  $P_t$  (vlození nových jedinců do  $P_t$ )
8  }
```

Algoritmus 3.1: Princip genetického algoritmu

tace těchto operátorů jsou závislé na volbě kódování dat (kap. 3.1). Detailní popis operátorů pro reálné kódování lze nalézt např. v [Houck et al., 1995], pro binární kódování v [Beasley et al., 1993a]. Zaměříme zde pozornost na základní binární operátory *křížení* a *mutace*.

šablona	0 1 0 0 1 1 1 0 0 1
rodič 1	1 1 1 0 1 0 0 1 0 0
	\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow
potomek 1,2	1 1 1 0 1 0 0 1 1 0
	\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow
rodič 2	1 0 1 0 0 1 0 1 1 1
	\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow

Obr. 3.1: Uniformní křížení

Začneme s *uniformním křížením*¹, které ze dvou rodičů vytvoří dva nové potomky, jak je vidět na Obr. 3.1. Nejprve je náhodně vygenerována šablona stejné délky jako rodiče. Na vytvoření *potomka 1* postupujeme následovně. Pokud je bit v šabloně roven 1, potom je odpovídající bit z *rodiče 1* nakopírován do *potomka 1*, pokud je bit v šabloně roven 0, použije se odpovídající bit z *rodiče 2*. K vytvoření *potomka 2* stačí vyměnit

¹z ang. *uniform crossover*

pořadí rodičů. Základní a nejužívanější formy tohoto operátoru, *jednobodové* a *dvoubodové* křížení [Beasley et al., 1993b], jsou dva speciální případy, skládající se z řad nul a jedniček (Obr. 3.2). Jednoduše řečeno, u jednobodového křížení náhodně vybereme bod křížení a prohodíme u rodičů konce řetězců.



Obr. 3.2: Šablona pro a) jednobodové a b) dvoubodové křížení



Obr. 3.3: Mutace

Mutace je aplikovaná pouze na jeden chromozom. V binární formě pouze náhodně, s malou pravděpodobností, změní jedničku na nulu a obráceně (Obr. 3.3). Tento operátor vnáší do populace novou informaci, pozitivní nebo negativní. Po určitém počtu iterací se GA může octnout v oblasti lokálního minima a pak jedině operátor mutace může vnést novou informaci vedoucí ke globálnímu optimu.

3.2.3 Výběr jedinců

V řádku 5 algoritmu 3.1 jsou jedinci vybírání do rekombinační fáze na základě jejich relativní úspěšnosti, často nazývané jako jejich “síla” nebo též “váha”. Při optimalizaci funkcí je síla většinou úměrná hodnotě objektivní funkce. V případě minimalizace je předpokládán počet kopií e_i daného jedince i dán vztahem

$$e_i = \frac{s_i}{\sum_1^N s_i} K, \quad s_i = \frac{1}{\delta + f_i}, \quad f_i \geq 0,$$

kde N je počet jedinců ve staré populaci P_{t-1} , K je počet vybíraných jedinců do M_t a f_i je funkční hodnota příslušná jedinci i ; při maximalizaci funkce je $s_i = f_i$. Parametr δ je malé pozitivní číslo bránící dělení nulou.

Obvykle není účelné vybírat jedince přímo v závislosti na velikosti jejich síly. V některých případech může z výrazně nejlepšího jedince vzniknout více kopií v jedné iteraci, což po několika cyklech GA způsobí podobnost všech členů populace a pád do lokálního minima. K potlačení tohoto jevu se používá několik metod. Jednou z nich je *lineární úprava velikosti sil*², která převádí původní váhy s_i na upravené s'_i lineární transformací $s'_i = as_i + b$. Standardní postup určení konstant a a b vychází z požadavku rovnosti průměrných sil před a po transformaci, tj. $\bar{s}' = \bar{s}$, a z požadavku na maximální sílu $s'_{max} = C_{mult} \bar{s}'$, kde je obvykle použito $C_{mult} \in \langle 1.2; 2.0 \rangle$. Příslušné vzorce pro a a b jsou uvedeny v [Goldberg, 1989]. Na základě původních (nebo upravených) sil dochází k výběru jedinců algoritmy, z nichž několik reprezentativních je uvedeno níže:

Metoda rulety: V prvním kroku je vybráno náhodné číslo r z uzavřeného intervalu $\langle 0; 1 \rangle$. Je vybrán jedinec i , jehož kumulativní pravděpodobnost výběru

$$p_i = \frac{\sum_1^i s'_i}{\sum_1^N s'_i}$$

je nejbližší menší než r . Postup je opakován do zaplnění výběrové množiny. Tato metoda je názorně popsána např. v [Topping and Khan, 1994]. Je nutno poznamenat, že způsob výběru v této metodě je velice často popisován analogicky s jednou otočkou kola v ruletě, kdy velikosti jednotlivých výsečí tohoto kruhu odpovídají silám jedinců a místo dopadu náhodného “hodu” vybere odpovídající výseč. Průběh výběru si můžeme přiblížit na příkladě uvedeném v tabulkách 3.1 a 3.2.

Chromozom	s'_i	p_i
1	0.663	0.256
2	0.258	0.356
3	0.944	0.721
4	0.722	1.000

Tab. 3.1: Příklad populace s odpovídajícími kumulativními pravděpodobnostmi

²z ang. *linear scaling*

Náhodné číslo r	0.606	0.941	0.665	0.312	0.178	0.115	0.665	0.306
Vybraný chromozom	3	4	3	2	1	1	3	2

Tab. 3.2: Příklad výběru chromozomů

Stochastic universal sampling (SUS): Tato metoda byla navržena v [Baker, 1987] a v podstatě zobecňuje metodu rulety. Její největší výhoda spočívá v tom, že jako jedinná z prezentovaných algoritmů každému jedinci zaručuje (přesně) zahrnutí očekávané hodnoty e_i . Pomyslný kruh je rovnoměrně rozdělen na úseky podle požadovaného počtu vybíraných jedinců a místo náhodného dopadu v prvním úseku je okopírováno do všech úseků. Na rozdíl od předchozí metody tak vybere z první náhodné hodnoty celý výběr najednou. Celý postup je v programovacím pseudo kódu zapsán v algoritmu 3.2. Implementace vychází z poznatku, že pro velikost celkového součtu S očekávaných hodnot e_i jedinců i platí $S = \sum_1^N e_i = K$, kde N je velikost populace a K je počet vybíraných chromozomů.

```

1 bod = náhodné číslo  $u[0, 1]$ 
2 suma = 0
3 for (  $i = 0 ; i < N ; i = i + 1$  ) {
4     for (  $suma = suma + e_i ; suma > bod ; bod = bod + 1$  ) {
5         Vyber jedince  $i$ 
6     }
7 }
```

Algoritmus 3.2: Výběrová metoda SUS

Remainder Stochastic Independent Sampling (RSIS): [Baker, 1987] Jedinci jsou v této metodě vybírání nejprve na základě celočíselné části e_i , které se počet kopií jedince v M_t rovná. Pokud je např. $e_i = 2.6$, budou po této fázi ve výběrové množině

2 kopie chromozomu i . Zbylé desetinné části (v našem příkladě hodnota 0.6) jsou zahrnuty tak, že je vygenerováno náhodné číslo r z intervalu $\langle 0; 1 \rangle$ a pokud je zbylá část menší, zahrne se do výběrové množiny a hodnota zbytku se nastaví na nulu. Postupuje se k porovnání dalšího jedince. Celý cyklus se opakuje do zaplnění výběru. Metodu lze symbolicky rozepsat do algoritmu 3.3.

```
1  for (  $i = 0 ; i < N ; i = i + 1$  ) {
2      for ( ;  $e_i \geq 1.0 ; e_i = e_i - 1$  ) {
3          Vyber do  $M_t$  jedince  $i$ 
4      }
5  }
6   $i = 0$ 
7  while ( není množina  $M_t$  plná ) {
8      if ( náhodné číslo  $u[0, 1] \leq e_i$  ) {
9          Vyber jedince  $i$ 
10          $e_i = 0.0$ 
11     }
12      $i = i + 1$ 
13     if (  $i > N$  ) {
14          $i = 0$ 
15     }
16 }
```

Algoritmus 3.3: Výběrová metoda RSIS

3.2.4 Genetický algoritmus I (GA I)

Touto částí začíná popis jednotlivých optimalizačních algoritmů. První algoritmus (GA I) je označován jako jednoduchý GA. Je navrhnut podle definice v [Goldberg, 1989], doznal jen několika změn, které se týkají zejména výběru. Princip algoritmu je založený pouze na výběru za předpokladu nenulové pravděpodobnosti výběru každého jedince. V kontextu s algoritmem 3.1 jsou nejdůležitější řádky rozepsány níže:

Krok 5 K výběru je použita metoda RSIS spolu s lineárně upravenými silami jedinců.

Nejprve je proveden výběr M jedinců určených pro reprodukci a poté je aplikován výběr ještě jednou a vybere se $N - M$ jedinců, kde N je počet jedinců v populaci. Druhý výběr bude pouze nakopírován do nové populace.

Krok 6 Jsou použity genetické operátory na prvních M jedinců.

Krok 7 Všichni jedinci vzniklí v kroku 6 a všichni jedinci z druhého výběru jsou nakopírováni do nové populace.

Za povšimnutí stojí, že nejsou vybíráni jedinci, kteří “zahynou”. Budou to všichni jedinci, kteří se neobjeví ani v jednom výběru.

3.2.5 Genetický algoritmus II (GA II)

Mnozí příznivci GA tento algoritmus nazývají *Steady state* nebo *On thy fly*. Je založený na změně a následném zatřídění jen malého počtu jedinců. Jednoduchá verze tohoto přístupu je znovu nastíněna na jednotlivých krocích algoritmu 3.1 :

Krok 5 Metodou rulety jsou vybrány dva chromozomy.

Krok 6 Na tyto jedince je aplikováno křížení a na dva nově vzniklé je aplikován operátor mutace. Operátory jsou aplikovány na základě předem předepsaných pravděpodobností.

Krok 7 Metodou inverzní rulety jsou vybráni dva jedinci. V případě, že jsou horší než jedinci vzniklí v kroku 6, jsou těmito nahrazeni. Metoda rulety zde pracuje s převrácenými hodnotami sil (“inverzně”) a tak s větší pravděpodobností vybere z pohledu optimalizace horší chromozom.

3.3 Rozšířené simulované žíhání

S ohledem na výsledky, obdržené při aplikaci výše uvedených algoritmů (kap. 4), se ukázaly nedostatky těchto přístupů. Při porovnání jednotlivých optimalizačních algoritmů v [Matouš et al., 2000] se ukázala jako jedna z nejprogresivnějších *metoda rozšířeného simulovaného žíhání* (AUSA)³ [Kvasnička, 1994]. Efektivita této metody spočívá ve využití principů GA (populace jedinců oproti jednomu vyhledávacímu bodu) spojených s teorií minimálních energetických stavů využívaných v simulovaném žíhání (SA)⁴. Jednotlivé stavební prvky algoritmu AUSA jsou popsány v následujících podkapitolách.

3.3.1 Simulované žíhání (SA)

Simulované žíhání je stochastická optimalizační metoda, která používá náhodných operátorů. Optimum hledá náhodně v okolí již známého řešení. V této metodě je nový vektor přijat i přesto, že nedojde ke snížení hodnoty optimalizované funkce, a to na základě pravděpodobnosti, která se řídí Boltzmannovým rozdělením [Press et al., 1992]

$$Pr(E) \sim e^{\left(\frac{-E}{k_B T}\right)} .$$

Uvedený vztah popisuje rozdělení energie systému, který je v tepelné rovnováze s teplotou T , mezi všemi rozdílnými energetickými stavy, kde k_B je Boltzmannova konstanta.

³z ang. *augmented simulated annealing method*

⁴z ang. *simulated annealing*

3.3.2 Algoritmus AUSA

Pokud chceme popsat princip algoritmu AUSA, můžeme vybírat ze dvou pohledů. První popisuje tuto metodu jako výše uvedené simulované žhání rozšířené o více jedinců a genetické operátory [Mahfoud and Goldberg, 1992], druhý popis přikládá větší důležitost genetickému algoritmu, do kterého je ve fázi rozhodování začleněna taktika SA [Kvasnička, 1993]. Nezávisle na sporu lze algoritmus stručně zapsat do algoritmu 3.4 :

```

1   $T = T_{max}, t = 0$ 
2  vytvoř  $P_0$ , ohodnot  $P_0$ 
3  while (not zastavovací - podmínka) {
4       $count = succ = 0$ 
5      while(  $count < count_{max} \wedge succ < succ_{max}$ ) {
6           $count = count + 1, t = t + 1$ 
7          vyber jedince  $I_t$  z  $P_{t-1}$ 
8          vyber operátor  $O$ 
9          změň  $I_t$  operátorem  $O$ ,  $I'_t$  je výsledek
10          $p = \exp((F(I'_t) - F(I_t))/T)$ 
11         if ( náhodné číslo  $u[0, 1] \leq p$ ) {
12              $succ = succ + 1$ 
13             vlož  $I'_t$  do  $P_t$ 
14             ohodnot  $P_t$ 
15         }
16     }
17     sniž  $T$ 
18 }
```

Algoritmus 3.4: Rozšířené simulované žhání (AUSA)

Krok 1 Počáteční teplotu T_{max} je potřeba zvolit tak, aby poměr přijatých jedinců ke všem vytvořeným byl $\approx 50\%$. Nastavení této hodnoty vyžaduje experimentování metodou “pokusu a omylu”.

Krok 3 Jako zastavovací podmínka bylo zvoleno překročení maximálního počtu iterací.

Krok 5 Parametr $count_{max}$ udává maximální počet všech iterací a $succ_{max}$ počet úspěšných iterací na dané teplotní hladině. Doporučený poměr těchto hodnot je $count_{max} = 10succ_{max}$.

Krok 7 V této implementaci AUSA vychází z již dříve popsaneho *Steady state* genetického algoritmu (GA II). Proto jsou v tomto kroku metodou SUS vybráni pouze dva jedinci.

Krok 8 Je vybrána mutace nebo křížení. Je doporučena větší pravděpodobnost výběru pro operátor mutace, v [Kvasnička, 1994] je doporučen poměr mezi křížením a mutací ≈ 0.1 .

Krok 9 & 10 Tento krok řídí zatřídění jedinců do nové populace. Je použito Metropolisovo kritérium, uvedené např. v [Vidal, 1993], které umožňuje výměnu rodiče horším potomkem pouze s určitou pravděpodobností. Tato pravděpodobnost se s konvergencí do globálního optima snižuje. V případě $F(I'_t) > F(I_t)$, je pravděpodobnost příjmu větší než jedna. To odpovídá stavu, kdy je potomek lepší než rodič, který je v tomto případě vždy nahrazen.

Krok 17 Tento krok je nazýván *ochlazování*⁵. Je použit jednoduchý vztah pro snižování teploty ve tvaru $T_{i+1} = T_{mult}T_i$. Součinitel T_{mult}

$$T_{mult} = \left(\frac{T_{min}}{T_{max}} \right)^{\frac{succ_{max}}{iter_{max}}}$$

je nastaven tak, aby při zvoleném maximu počtu iterací $iter_{max}$ dosáhl minimální teploty T_{min} , jejíž doporučená hodnota je $T_{min} = 0.01T_{max}$.

⁵z ang. *cooling schedule*

3.3.3 Návaznost penalizačních funkcí

Připomeňme si parametr α z rovnice (2.41), který ovlivňoval přísnost penalizace nesplnění zadaných podmínek (Obr. 2.6). Jeho velikost má přímý dopad na počet nepřipustných řešení, které se objeví v další populaci.

V průběhu optimalizace je tento parametr zmenšován tak, aby na začátku optimalizace umožnil přijmout i nepřipustné jedince a aby na konci optimalizace už takováto řešení nepřijal. Tento postup dovoluje použít nepřístupné jedince jako pomocná řešení k dosažení přípustných (nepenalizovaných) jedinců zvláště v optimalizačních případech, kdy je ve vyhledávacím prostoru velké množství penalizovaných řešení.

Jako nejjednodušší řešení snižování parametru α se jeví návaznost na proces snižování teploty (krok 17 algoritmu 3.4). Pokud dojde ke snížení teploty, sníží se i α a to na velikost

$$\alpha = \alpha_{max} \left(\frac{\alpha_{min}}{\alpha_{max}} \right)^{\frac{iter_{curr}}{iter_{max}}}, \quad (3.6)$$

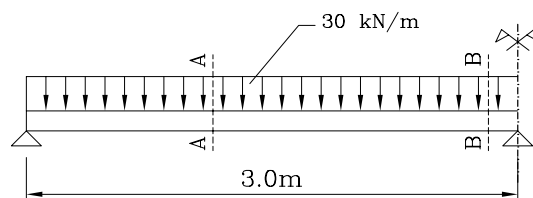
a zůstane konstantní do další změny teploty. V rovnici (3.6) je α_{max} počáteční nastavení, α_{min} nejnižší hodnota parametru α , $iter_{max}$ je maximální počet iterací a $iter_{curr}$ je počet již proběhlých iterací v daný okamžik.

Kapitola 4

Výsledky

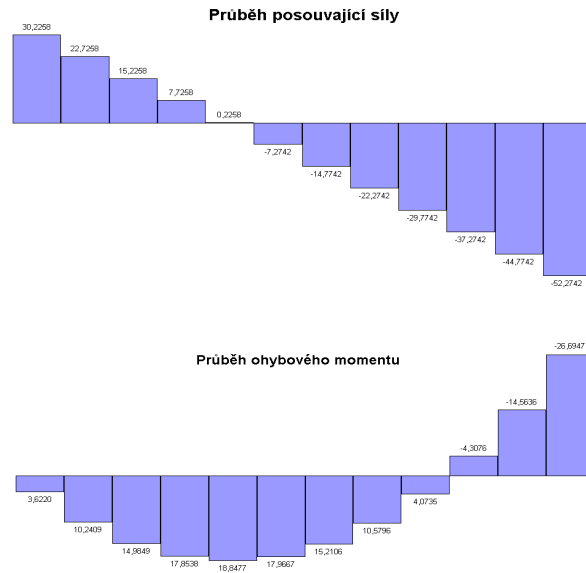
4.1 Příklad nosníku

Za testovací příklad byl vybrán spojitý nosník zatížený konstantním spojitým zatížením (Obr. 4.1). S ohledem na symetrii byla řešena pouze polovina nosníku. Průběhy vnitřních sil, ohybového momentu a posouvající síly (Obr. 4.2), byly zjištěny deformační variantou metody konečných prvků (kap. 2.2). Velikosti parametrů vztahujících se k materiálu jsou vzaty z tabelovaných hodnot pro beton C 20/25 a ocel E 10 216, přičemž jsou zvoleny ceny $P_c = 1350.0 \text{ Kč/m}^3$ pro beton a $P_s = 50.0 \text{ Kč/kg}$ pro ocel. Parametry



Obr. 4.1: Zatížený spojitý nosník

srovnávacího nosníku (Tab. 4.1) byly navrženy podle zásad a doporučení uvedených v [Procházka et al., 1997]. Pokud je řešena optimalizace jen některých částí nosníku, zbylé hodnoty jsou vzaty z tohoto návrhu.



Obr. 4.2: Průběhy vnitřních sil

4.2 Výsledky optimalizace tvaru průřezu

Tento příklad je spojen s redukcí ceny pouze v závislosti na změně tvaru nosníku, tj. na změně dimenzí b a h . Výztuž není v průběhu optimalizace měněna.

Z pohledu optimalizačního algoritmu se jedná o dvourozměrný problém. Pokud uvažujeme diskrétní hodnoty, má binární chromozom délku 9 bitů, v případě spojitých hodnot je délka binárního řetězce 19 bitů.

Byly aplikovány genetické algoritmy **GA I** a **GA II** (kap. 3.2.4 a 3.2.5), se shodnými počátečními parametry. Startovací populace P_0 o 200 jedincích byla vytvořena zcela náhodně. Pravděpodobnost křížení $p_c = 1$ a mutace $p_m = 0.03$ zůstaly konstantní

Řez		Rozměry		Parametry smykové výztuže							Cena
$A - A$	$B - B$	b [mm]	h [mm]	pr_w	l_{w1}	s_{w1}	l_{w2}	s_{w2}	l_{w3}	s_{w3}	[Kč]
3 \emptyset 10.0	6 \emptyset 10.0	200	300	8.0	8	200	4	125	-	-	1002.65

Tab. 4.1: Parametry porovnávacího nosníku

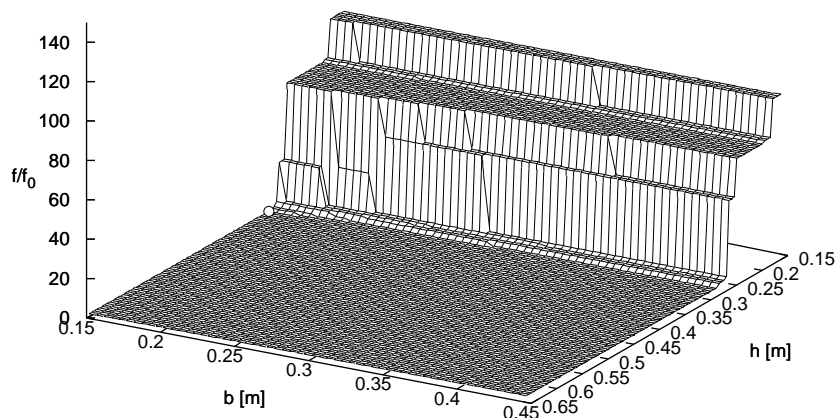
Hodnoty	b [mm]	h [mm]	Cena [Kč]
Spojité	150	294	898.48
Diskrétní	150	300	906.39

Tab. 4.2: Výsledky optimalizace tvaru průřezu

po celý průběh optimalizace. Zastavovací kritérium spočívalo v překročení 100 000 iterací. Výsledky jsou vypsány v tabulce 4.2, kde jsou uvedeny optimální dimenze průřezu s odpovídající cenou pro spojité a diskrétní hodnoty sledovaných veličin. Oba algoritmy vždy našly přesné minimum zobrazené na Obr. 4.3 (prázdný kroužek).

4.3 Optimalizace tvaru průřezu a ohybové výztuže

V tomto příkladě je redukována cena změnou dimenzí průřezu a ohybové výztuže. Objektivní funkce je závislá na 12 proměnných, přičemž délka binárního chromozomu je 55 bitů s uvažováním diskrétních dimenzí a 65 bitů pro případ se spojitými rozměry. Parametry optimalizačních algoritmů **GA I** a **GA II** byly nastaveny stejně jako v předchozím příkladu. V tabulce 4.3 jsou uvedeny minimální, maximální a průměrné ceny odpovídající nejlepšímu jedinci v populaci z 20 nezávislých spuštění optimalizace. Směrodatná odchylka je uvedena k dokreslení hodnotící statistiky.



Obr. 4.3: Průběh objektivní funkce

Algoritmus	Hodnoty	Cena [Kč]			
		minimum	průměr	maximum	směr. odchylka
GA I	Spojité	851.27	866.24	880.83	9.01
	Diskrétní	959.07	977.09	1024.12	17.10
GA II	Spojité	837.22	844.39	859.15	7.89
	Diskrétní	824.43	829.48	834.22	3.23

Tab. 4.3: Výsledky optimalizace tvaru průřezu a ohybové výztuže

Z výsledků v tabulce 4.3 je zřejmé, že zahrnutím ohybové výztuže do optimalizačního procesu lze ušetřit další investice. Při porovnání nejlepšího jedince s diskretními hodnotami dimenzí s porovnávacím nosníkem činí rozdíl cen 18 %. Další snížení ceny lze docílit optimalizací smykové výztuže, což je popsáno v následující kapitole 4.4.

Přestože byly oba algoritmy testovány pouze na uvedených příkladech, z tabulky 4.3 je patrná převaha *steady state* genetického algoritmu **GA II** nad “tradičním” genetickým algoritmem **GA I**, zejména při uvažování diskretních hodnot průřezových veličin. Špatná konvergence byla v těchto případech způsobena problémem pádu algoritmu do lokálního minima v okolí nejlepšího jedince v počáteční populaci.

V tabulce 4.4 jsou rozepsány výsledky **GA II** z 5 nezávislých spuštění. Je zde patrná variabilita obou dimenzí a hlavně výztužných profilů jak u dolního (řez $A - A$), tak u horního (řez $B - B$) povrchu nosníku (Obr. 4.1).

4.4 Maximální optimalizační problém

V tomto příkladě je optimalizován nosník se všemi 21 proměnnými, které jsou definovány v kapitole 2.8. Binární řetězec má délku 80 a 90 bitů pro problém s diskretními respektive spojitými hodnotami rozměrů průřezu.

Po přidání smykových veličin do optimalizované funkce nebyly genetické algoritmy **GA I** a **GA II** schopny dokonkovat do přijatelných hodnot. Z důvodu porovnání algoritmů v [Lepš et al., 1999] a [Matouš et al., 2000] byla aplikována metoda *rozšířeného*

Hodnoty	Řez		Rozměry		Cena [Kč]
	$A - A$	$B - B$	b [mm]	h [mm]	
Spojité	6 \emptyset 6.0	10 \emptyset 6.5	151	385	837.22
	5 \emptyset 6.5	10 \emptyset 6.5	150	393	842.21
	7 \emptyset 5.5	12 \emptyset 6.0	150	388	838.16
	6 \emptyset 6.0	11 \emptyset 6.0	150	408	859.15
	6 \emptyset 6.0	10 \emptyset 6.5	150	388	838.91
Diskrétní	5 \emptyset 7.0	10 \emptyset 7.0	150	350	824.43
	6 \emptyset 6.5	10 \emptyset 7.0	150	350	830.30
	7 \emptyset 6.0	12 \emptyset 6.5	150	350	833.18
	6 \emptyset 6.5	12 \emptyset 6.5	150	350	834.22
	7 \emptyset 6.0	10 \emptyset 7.0	150	350	829.27

Tab. 4.4: Výsledky **GA II** z 5 nezávislých spuštění

simulovaného žíhání (AUSA) (kap. 3.3). Uspokojivých výsledků bylo dosaženo až s úpravami algoritmu AUSA, které jsou popsány v následujících odstavcích.

Navržený postup kombinuje klasickou metodu AUSA se speciálním typem přezíhání¹. První krok spočívá ve vygenerování dostatečně velké populace sub-optimálních řešení použitím standardního algoritmu AUSA (algoritmus 3.4). Je předepsán počet spuštění optimalizačního algoritmu. Po každé je počáteční populace vytvořena zcela náhodně. Díky tomu obsahuje velké procento jedinců, kteří nesplňují předepsaná omezení (2.2)-(2.4). Na konci tohoto kroku jsou nejlepší jedinci obvykle v některém z mnoha lokálních minim, ale ve všech případech splňují omezující podmínky. Tato řešení jsou uschována pro proces přezíhání.

¹z ang. *reannealing*

č./ p_m	SUS – Náhodně			SUS – Inverzní SUS			SUS – Rodič		
	0.6	0.75	0.9	0.6	0.75	0.9	0.6	0.75	0.9
1	813.99	781.72	683.17	821.16	811.69	606.41	797.46	742.59	825.74
2	778.87	775.05	632.47	691.35	686.05	806.43	815.06	807.24	861.28
3	707.26	831.11	687.14	660.19	774.12	778.64	801.01	837.88	823.18
4	792.11	717.82	809.02	715.43	635.62	771.81	832.51	796.87	859.58
5	742.93	753.71	774.16	669.97	793.35	786.57	824.13	807.97	772.46
6	758.21	776.03	583.02	810.81	776.16	700.69	807.33	818.35	805.15
7	748.54	674.65	652.65	701.98	725.33	729.52	777.27	811.15	827.11
8	787.30	725.65	695.72	773.35	778.36	761.83	753.55	871.24	833.38
9	667.26	744.52	780.23	848.93	805.60	719.23	790.89	830.88	821.63
10	661.10	737.33	809.91	766.86	697.31	599.47	774.06	804.57	895.99
Min.	661.10	674.65	583.02	660.19	635.62	599.47	753.55	742.59	772.46
Pr.	738.06	744.75	699.14	738.20	738.11	714.56	793.35	806.48	827.09
Max.	813.99	831.11	809.91	848.93	811.69	806.43	832.51	871.24	895.99
Sm.Od.	49.61	40.46	74.63	63.74	56.02	68.76	22.99	31.16	31.98

Tab. 4.5: Ceny z počátečních řešení

4.4.1 Optimalizace s diskrétními hodnotami dimenzí

Výsledky z 10 nezávislých spuštění s diskrétními hodnotami dimenzí z prvního kroku jsou uvedeny v tabulce 4.5. Byly studovány tři různé způsoby jak vybrat a zatřídit nové jedince vzniklé v průběhu optimalizačního cyklu (řádky 10 – 15 v algoritmu 3.4). Výběr byl ve všech případech proveden v závislosti na síle jedinců metodou SUS (kap. 3.2.3).

První tři sloupečky odpovídají případu, kdy noví jedinci soupeří s náhodně vybranými členy staré populace namísto s rodiči. Tato modifikace umožňuje vnést do populace nové informace stejně jako umožní nahrazení horších jedinců. Následující tři sloupce jsou analogické k předchozímu postupu, pouze je jako metoda výběru jedinců ze staré populace aplikována inverzní metoda SUS, tedy metoda SUS používající inverzní síly chromozomů.

Poslední sloupce byly vygenerovány standardní metodou AUSA, která je uvedená v algoritmu 3.4. Ve všech případech nedosáhl poslední algoritmus optimálních hodnot a jak je patrné z nízkých směrodatných odchylek, populace uvázly v oblasti lokálních minim. Neúspěch lze připsat na úkor velikému počtu nepřípustných jedinců v počáteční populaci, kteří při takto nastaveném schématu výběr – zatřídění nemají možnost v průběhu optimalizace populaci opustit. Z těchto důvodů se jako progresivní jeví první dva přístupy, mezi nimiž nejsou velké rozdíly ve výsledcích.

Též byla provedena analýza vlivu pravděpodobnosti aplikace jednotlivých binárních operátorů v kroku 8 algoritmu 3.4. Výsledky v tabulce 4.5 ukazují, že vyšší hodnoty pravděpodobnosti mutace p_m dosahují lepších výsledků; pravděpodobnost křížení p_c je dána vztahem $p_c = 1 - p_m$. Z tohoto důvodu byla již nadále v prvním kroku optimalizačního procesu nastavena pravděpodobnost mutace na hodnotu 0.9.

Druhý krok v modifikované metodě AUSA spočívá v přežívání. Vhodná počáteční populace je vytvořena následovně: První krok byl spuštěn 50 krát. Pokaždé byla náhodně vygenerována populace o 100 jedincích. Po dosažení maximálního počtu iterací bylo uschováno 20 nejlepších jedinců pro další použití. Tak vznikla populace čítající 1000 sub-optimálních řešení. Z této množiny je jako počáteční populace do procesu přežívání náhodně vybráno 400 jedinců. Ještě jednou je nastavena maximální teplota T_{max} (krok 1 algoritmu 3.4). Výsledky z 10 nezávislých spuštění jsou uvedeny v tabulce 4.6. Pro výběr byla použita metoda SUS, k zatřídění náhodný výběr. Z tabulky 4.6 též vyplynulo, že lineární úprava sil chromozomů (scaling) není v tomto případě důležitá.

V tabulce 4.7 jsou vypsána výsledná řešení z 10 nezávislých spuštění. Nebyla použita metoda scaling a byla nastavena $p_m = 0.9$ (poslední sloupec v tabulce 4.6). Je zde patrná variabilita obou dimenzí a výztužných profilů u dolního (řez $A-A$) a u horního (řez $B-B$) povrchu nosníku (Obr. 4.1) spolu s profilem a vzdálenostmi tímínkové výztuže. Jednotlivé výrazy jsou popsány v tabulce 2.3.

Pokud porovnáme minimální dosaženou hodnotu u maximálního optimalizačního problému s výsledkem obdržným z předchozí optimalizace, tedy redukce ceny změnou di-

č./ p_m	užita metoda scaling			bez metody scaling		
	0.6	0.75	0.9	0.6	0.75	0.9
1	658.60	649.05	637.49	702.87	582.44	593.00
2	612.89	636.93	586.85	613.82	673.42	660.44
3	736.72	640.54	598.14	626.07	639.04	618.92
4	625.71	618.68	658.39	675.54	583.83	616.77
5	597.93	665.09	628.20	615.48	611.70	583.83
6	660.44	610.78	619.05	701.48	580.25	590.34
7	610.78	588.34	656.28	580.25	609.80	618.13
8	678.07	607.06	599.03	578.81	586.57	618.66
9	665.09	651.14	612.60	597.52	583.83	582.58
10	608.88	654.48	594.99	629.74	657.87	578.81
Min.	597.93	588.34	586.85	578.81	580.25	578.81
Pr.	641.19	628.22	616.17	627.31	610.66	603.6
Max.	736.72	665.09	658.39	702.87	673.42	660.4
Sm. Od.	40.53	23.47	24.19	43.69	33.03	23.97

Tab. 4.6: Cena z 10 nezávislých spuštění po přežhání

menzí a ohybové výztuže, dostaneme velikost úspory cca 30 % . Při detailním pohledu na jednotlivá řešení je patrné, že úspora není jen v objemu smykové výztuže. Na druhou stranu se ukazuje, že návrh ohybové a smykové výztuže je spolu daleko více svázán, než by se z uvedených vzorců (kap. 2.5) mohlo zdát. Při porovnání s návrhem podle EC 2 (Tab. 4.1) dosahuje úspora hodnoty nad 42 % , což zřejmě odpovídá vyváženému vztahu obou výztuží k daným dimenzím průřezu.

4.4.2 Zamyšlení nad volbou parametrů

V navrženém algoritmu modifikované metody rozšířeného simulovaného žhání je veliké množství parametrů, pomocí kterých lze upravovat a regulovat konvergenci algoritmu.

Řez		Rozměry		Parametry smykové výztuže							Cena
$A - A$	$B - B$	b [mm]	h [mm]	pr_w	l_{w1}	s_{w1}	l_{w2}	s_{w2}	l_{w3}	s_{w3}	[Kč]
4 \emptyset 6.0	9 \emptyset 5.5	150	550	4.0	9	400	3	75	-	-	593.00
4 \emptyset 6.5	9 \emptyset 5.5	175	550	4.0	10	425	2	100	-	-	660.44
5 \emptyset 5.5	9 \emptyset 5.5	150	550	4.0	9	400	3	75	-	-	618.92
6 \emptyset 5.5	8 \emptyset 6.5	150	450	6.0	9	400	3	125	-	-	616.77
5 \emptyset 5.5	9 \emptyset 5.5	150	525	4.0	9	425	3	75	-	-	583.83
4 \emptyset 6.0	8 \emptyset 6.0	150	550	4.0	9	425	3	75	-	-	590.34
4 \emptyset 6.0	10 \emptyset 5.0	150	550	4.0	9	425	3	75	-	-	618.13
6 \emptyset 6.0	8 \emptyset 7.0	150	400	6.0	9	425	3	100	-	-	618.66
4 \emptyset 6.0	8 \emptyset 6.0	150	525	4.0	9	425	3	75	-	-	582.58
4 \emptyset 6.0	9 \emptyset 5.5	150	525	4.0	9	425	3	75	-	-	578.81

Tab. 4.7: Výsledná řešení bez metody scaling a s $p_m = 0.9$

Jejich nastavení je závislé na optimalizovaném problému, a proto nelze formulovat obecné vztahy na určení jejich hodnot. Z těchto důvodů se v odborné literatuře objevují pouze doporučené poměry některých parametrů, které vycházejí z optimalizace více funkcí a jejichž vliv se dá zobecnit. Zbývá metoda “pokusu a omylu”, ale před jejím použitím je vhodné prozkoumat vliv jednotlivých parametrů na konvergenční charakteristiky.

Představme si objektivní funkci jako dno vysychajícího moře a naším úkolem je zůstat ve vodě až do konce. Úroveň hladiny odpovídá parametru teploty, který určuje na jaké oblasti hledáme požadované řešení. Můžeme se pohybovat pouze pod hladinou; je-li dané řešení penalizované, jsme v oblasti mělčiny.

První parametr, maximální teplota T_{max} , odpovídá počátečnímu nastavení hladiny. Doporučená hodnota musí navodit stav, kdy je 50 % všech řešení ve vodě. V případě vysoké hodnoty bude trvat neúměrně dlouho, než se dostaneme do oblasti minim, za nízkého stavu hladiny nebudeme mít možnost “proplouvat” do okolních oblastí. Minimální teplota by měla být nastavena tak, aby na konci iterace byla voda jen v oblasti

globálních minim. Pokud bude T_{min} velké, budeme se neustále pohybovat v okolí velkého množství lokálních minim, pokud nastavíme T_{min} příliš malé, hladina klesne pod nejnižší bod moře a my se ocitneme na suchu i mimo lokální minimum.

Dalšími parametry jsou veličiny ovlivňující penalizační funkce a tím i velikost “mělčín”. Zásady jsou podobné jako pro teplotu - počáteční hodnota udržuje v rovnováze maximální počet nutných iterací a velikost vyhledávacího prostoru, minimální vymezuje prostor kolem oblasti minim.

Komplikace nastávají v případě přežhání. Všichni jedinci v počáteční populaci se nacházejí v některém z minim, které odpovídá naposledy nastavené teplotě a parametrům penalizačních funkcí. Pokud zvolíme postup jako u nové populace, novým nastavením parametrů změní tvar objektivní funkce a algoritmus tak musí začít znovu hledat oblasti minim. Logicky se tak nabízí pro počáteční parametry použít koncové hodnoty teploty a parametru α z prvního kroku modifikované metody AUSA. K nastavení minimálních hodnot po procesu přežhání je nutno opět použít metodu “pokusu a omylu”.

Výše nastíněný postup byl vyzkoušen při optimalizaci maximálního problému se spojitými hodnotami dimenzí (kap. 4.4.3) a výsledky potvrdily uvedené hypotézy. Např. teplota T_{max} by musela být nastavena na 100 krát vyšší hodnotu než v prvním kroku, aby se splnil předpoklad přijetí 50 % nových jedinců. Je zajímavé, že v předchozím případě s diskrétními hodnotami dimenzí pokusně získané optimální parametry odpovídaly stavu po prvním kroku.

4.4.3 Optimalizace se spojitými hodnotami dimenzí

Do objektivní funkce vstupují hodnoty dimenzí se spojitými hodnotami (Tab. 2.2). Je použita modifikovaná metoda AUSA, jak je nastíněna v kapitole 4.4.1. Nastavení první fáze je shodné s řešením použitým při maximálním problému s diskrétními hodnotami; byla vytvořena populace sub-optimálních řešení čítající 1000 jedinců.

Při následném přežhání bylo přihlédnuto ke skutečnosti, že u spojitého problému je $2^{10} \approx 1000$ krát více řešení než u problému diskrétního. Počáteční populace byla

č. spuštění	minimální teplota		
	$T_{min} = 0.001$	$T_{min} = 0.0005$	$T_{min} = 0.0001$
1	609.27	678.09	646.64
2	580.22	575.78	726.99
3	621.38	569.87	674.60
4	650.37	640.54	699.12
5	659.61	634.51	629.42
6	650.57	667.25	727.94
7	621.77	717.82	726.99
8	641.41	591.57	609.27
9	659.61	631.83	647.22
10	659.61	598.93	643.83
Minimum	580.22	569.87	609.27
Průměr	630.37	625.10	667.39
Maximum	659.61	717.82	727.94
Směrodatná Od.	25.22	45.32	42.08

Tab. 4.8: Cena po přežívání se spojitými hodnotami dimenzí

vytvořena výběrem 400 jedinců z řešení získaných v prvním kroku. Další 400 jedinců bylo vytvořeno zcela náhodně, což vneslo do populace opět novou informaci, ale hlavně tento krok umožnil algoritmu nastartovat konvergenci s velkým počtem nepenalizovaných jedinců. Podle předpokladů popsaných v předchozí kapitole byly počáteční parametry algoritmu nastaveny na koncové hodnoty z první fáze. Dále došlo k “zprísnění” požadavků na řešení - penalizační parametr α je redukován z hodnoty 0.01 na hodnotu 0.005 . Algoritmus je zastaven po dosažení 400000 iterací, tedy dvakrát více než v předchozím případě.

Výsledky z 10 nezávislých spuštění jsou vypsány v tabulce 4.8. Je zde patrný vliv hodnoty minimální teploty. V prvním případě byla T_{min} příliš vysoko a tři řešení (číslo 5, 9, 10)

Řez		Rozměry		Parametry smykové výztuže							Cena
$A - A$	$B - B$	b [mm]	h [mm]	pr_w	l_{w1}	s_{w1}	l_{w2}	s_{w2}	l_{w3}	s_{w3}	[Kč]
8 \emptyset 5.5	8 \emptyset 7.0	150	394	6.0	7	375	5	100	-	-	678.09
4 \emptyset 6.0	9 \emptyset 5.5	150	521	4.0	9	425	3	75	-	-	575.78
4 \emptyset 6.0	8 \emptyset 6.0	150	523	4.0	9	425	3	75	-	-	569.87
6 \emptyset 5.5	8 \emptyset 6.5	152	437	4.0	9	275	3	50	-	-	640.54
4 \emptyset 6.5	6 \emptyset 6.5	182	532	4.0	10	425	2	100	-	-	634.51
4 \emptyset 6.0	9 \emptyset 5.5	151	527	8.0	11	425	1	300	-	-	667.25
4 \emptyset 7.0	9 \emptyset 5.5	215	507	4.0	10	425	2	150	-	-	717.82
5 \emptyset 5.5	10 \emptyset 5.0	158	500	4.0	9	425	3	75	-	-	591.57
4 \emptyset 6.5	8 \emptyset 5.5	166	569	4.0	10	375	2	100	-	-	631.83
4 \emptyset 6.5	11 \emptyset 5.0	159	447	4.0	9	400	3	50	-	-	598.93

Tab. 4.9: Výsledná řešení se spojitými hodnotami dimenzí

skončily v počátečním lokálním minimu. V posledním případě byla konečná teplota naopak nízko a opět dvě hodnoty (číslo 2, 7) nebyly schopny opustit startovací řešení.

V tabulce 4.9 jsou uvedena výsledná řešení s hodnotou $T_{min} = 0.0005$ (průměrní sloupec v tabulce 4.8). Opět je patrná variabilita mezi jednotlivými řešeními. Zobrazené výsledky potvrzují výskyt velkého počtu lokálních minim v oblasti s nízkými cenami, zjištěné při optimalizaci problému s diskrétními hodnotami.

Kapitola 5

Závěr a budoucí směry výzkumu

V této práci jsou shrnuty poznatky o optimalizaci železobetonového spojitého nosníku s využitím některých stochastických algoritmů. Byl představen model nosníku s návazností na zvolené podmínky normy Eurokód 2 a výsledná optimální řešení jeho návrhu.

V kapitole 2 je nastíněn cenový pohled na uvažovaný problém a jednotlivé parametry, které ovlivňují výslednou cenu. Představený diskretizovaný model železobetonového spojitého nosníku je dostatečně výstižný, aby byl schopen pokrýt jak požadavky jednoduchosti a efektivnosti požadované praxí, tak žádosti fyzikálního modelu z důvodu pokrytí normových požadavků. Omezení daná normou jsou detailně zpracovány pro kombinace namáhání ohybem a smykem. Pro úplnou implementaci odezvy materiálu na zatížení zbývá doplnit vliv normálové síly. Než se přistoupí k řešení rámu je též potřeba smysluplně popsat konstrukci na úrovni stavebních prvků - trámů, průvlaků a sloupů, spolu s jejich vzájemnou provázaností např. v dimenzích nebo ve výztuži. Hlavní výhodou dobře navrženého modelu bude možnost optimalizovat konstrukci po částech, neboť ani v blízké budoucnosti výpočetní výkon počítačů nebude schopen optimalizovat stavební konstrukci globálně se všemi proměnnými. Možnost rozdělení konstrukce na jednotlivé nezávislé části spolu se souběžným výpočtem řešení v průběhu optimalizační metody povede na paralelní výpočet návrhu stavební konstrukce.

Jsou představeny dvě modifikace standardního genetického algoritmu a relativně nová

metoda upraveného simulovaného žhání. U posledně jmenované metody je v závislosti na obdržných výsledcích navržena úprava spočívající v opakovaném spuštění optimalizovaného algoritmu. Dále je v kapitole 3 podrobně popsáno binární kódování dat v optimalizačním algoritmu. Je prokázána efektivita tohoto přístupu při optimalizaci funkcí s velkým počtem diskrétních (celočíslných) hodnot. Mnohé současné vědecké práce však ukazují, že by i reálné kódování dat s vhodnými operátory mohlo na tomto poli uspět.

Obdržné výsledku (kap. 4) dokládají různorodost reálného světa. Ačkoliv by nejoptimálnější řešení nebylo z pohledu praxe nevhodnější z důvodů např. veliké stavební výšky, ukazuje se, že je možné navrženým algoritmem řešit i značně složitý problém. Výsledky jsou už pak pouze závislé na popsáných omezujících podmínkách.

Literatura

- [Baker, 1987] Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In Grefenstette, J., editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 101–111. Lawrence Erlbaum Associates.
- [Beasley et al., 1993a] Beasley, D., Bull, D., and Martin, R. (1993a). An overview of Genetic Algorithms: Part 1, Foundations. *University Computing*, 15(2):58–69. Též dostupný na ftp://ralph.cs.cf.ac.uk/pub/papers/GAs/ga_overview1.ps.
- [Beasley et al., 1993b] Beasley, D., Bull, D., and Martin, R. (1993b). An overview of Genetic Algorithms: Part 2, Research topics. *University Computing*, 15(4):17–69. Též dostupný na ftp://ralph.cs.cf.ac.uk/pub/papers/GAs/ga_overview2.ps.
- [Bittnar and Šejnoha, 1992] Bittnar, Z. and Šejnoha, J. (1992). *Numerické metody mechaniky*. ČVUT.
- [Cerny, 1985] Cerny, J. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Opt. Theory Appl.*, 45:41–51.
- [Chudoba, 1995] Chudoba, R. (1995). *Sensitivity analysis of RC framed structures: Object-oriented approach*. PhD thesis, ČVUT, Praha.
- [Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- [Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. MIT Press.

- [Houck et al., 1995] Houck, C. R., Joines, J. A., and Kay, M. G. (1995). A Genetic Algorithm for function Optimization: A Matlab implementation. Technical Report NCSU-IE TR 95-09, North Carolina State University. Též dostupný na http://www.fmmcenter.ncsu.edu/fac_staff/joines/papers/gaot.ps.
- [Ingber, 1993] Ingber, L. (1993). Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, 18(11):29–57.
- [Ingber, 1995] Ingber, L. (1995). Adaptive simulated annealing (ASA): Lessons learned. *Control and Cybernetics*, 25(1):33–54.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, Jr., C., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220:671–680.
- [Kvasnička, 1993] Kvasnička, V. (1993). Stochastické metody optimalizace funkcí N proměnných. In *Analýza dat*, Pardubice.
- [Kvasnička, 1994] Kvasnička, V. (1994). Augmented simulated annealing adaption of feed-forward neural networks. *Neural Network World*, 3.
- [Lampinen and Zelinka, 1999] Lampinen, J. and Zelinka, I. (1999). Mixed variable nonlinear optimization by differential evolution. In *Nostradamus '99*, VUT, Zlín.
- [Laníková, 1999] Laníková, I. (1999). Optimalizovaný návrh výztuže rámových konstrukcí. *Betonářské dny '99*. Česká společnost pro beton a zdivo.
- [Lepš et al., 1999] Lepš, M., Matouš, K., and Bittnar, Z. (1999). Genetic algorithms in optimization of reinforced concrete beam. *Acta Polytechnica*, 39(2):145–154.
- [Mahfoud and Goldberg, 1992] Mahfoud, S. and Goldberg, D. E. (1992). A genetic algorithm for parallel simulated annealing. In Manner, R. and Manderick, B., editors, *Parallel Problem from Nature*, volume 2. North-Holland, Amsterdam. Též dostupný na <http://gal4.ge.uiuc.edu/illegal.home.html>.

- [Mare and Holy, 2000] Mare, T. and Holy, S. (2000). Linear programming in structural optimization. In *Workshop 2000*. ČVUT.
- [Matouš et al., 2000] Matouš, K., Lepš, M., Zeman, J., and Šejnoha, M. (2000). Applying genetic algorithms to selected topics commonly encountered in engineering practice. *Computer Methods in Applied Mechanics and Engineering*. In print.
- [Michalewicz, 1992] Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. AI Series. Springer-Verlag, New York.
- [Michalewicz, 1995] Michalewicz, Z. (1995). Genetic algorithms, numerical optimization and constraints. *Proceedings of the 6th International Conference on Genetic Algorithms*. Též dostupný na <http://www.coe.uncc.edu/~zbyszek/papers.html>.
- [Press et al., 1992] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in C*. Cambridge University Press. Též dostupný na <http://www.nr.com>.
- [Procházka, 1995] Procházka, J. (1995). *EUROCODE 2, výtah z ČSN P ENV 1992-1-1 s komentářem*. PROCON.
- [Procházka et al., 1997] Procházka, J., Bradáč, J., Krátký, J., Filipová, J., and Hanzlová, H. (1997). *Betonové konstrukce, příklady navrhování podle EUROCODE 2*. PROCON.
- [Storn, 1996] Storn, R. (1996). On the usage of differential evolution for function optimization. In *NAPHIS 1996*, pages 519–523. Berkeley.
Též dostupný na <http://www.icsi.berkeley.edu/~storn/bisc1.ps.gz>.
- [Topping and Khan, 1994] Topping, B. and Khan, A. (1994). *Parallel element computations*. Saxe-Coburg Publications.
- [Vidal, 1993] Vidal, R. V. V. (1993). Introduction. In *Applied Simulated Annealing*, volume 396 of *Notes in Economics and Mathematical Systems*. Springer-Verlag.