



# **TVORBA NEZÁVISLÝCH VEKTORŮ S UŽITÍM METODY SIMULOVANÉHO ŽÍHÁNÍ**

Priezvisko, meno študenta : Zuzana Vitingerová  
Ročník a odbor štúdia : 5. ročník, Inžénýrství ŽP  
Vedúci práce : Ing. Matěj Lepš  
Katedra : K. stavební mechaniky

## Obsah

Abstrakt .....	2
1. Úvod .....	2
1.1 Metoda LHS .....	4
1.2 Simulované žíhání.....	4
2. Generování vstupních dat pro neuronové sítě .....	6
2.1 Trénování umělých neuronových sítí .....	6
2.2 Metoda .....	7
2.3 Výsledky.....	8
2.4 Sensitivní analýza .....	10
3. Generování vstupních dat pro optimalizaci.....	13
3.1 Náhodné vektory pro stochastické optimalizace .....	13
3.2 Popis metody a výsledky.....	14
4. Závěr .....	17
Literatura .....	17

## Abstrakt

Tato práce je úzce svázána s některými metodami umělých inteligencí a to konkrétně se stochastickými optimalizačními metodami a umělými neuronovými sítěmi. Jednotícím prvkem je v tomto případě tvorba nezávislých a/nebo náhodných vektorů reálných čísel, které v prvním případě složí k nalezení nových, neprozkoumaných oblastí účelové funkce, v druhém případě pak k tvorbě reprezentativních vstupních dat. Cílem tohoto výzkumu je pak v první řadě minimalizace potřebných iterací dané metody či zlepšení konvergence příslušného iteračního procesu. Tento příspěvek se detailněji zabývá dosud nepublikovanými aspekty tvorby nezávislých vektorů zejména s ohledem na nejnovější trendy z oblasti stochastických optimalizačních metod s důrazem na aplikace při multikriteriálních, multimodálních a aproximativních optimalizacích.

## Abstract

This work is closely connected to selected methods of applied artificial intelligence – namely stochastic optimization algorithms and artificial neural networks. The unifying topic of the presented study is a generation of independent and/or random vectors of real numbers which are used for (i) exploration of new, yet unsearched, parts of definition domain of an objective function or (ii) generation of representative input data. The particular goal of the presented research is a decrease of a number of function calls and/or improvement of the convergence rate of the pertinent iteration process. The contribution studies in more detail novel aspects and approaches in the generation of independent vectors, which are currently not available in the literature. The attention is paid to recent trends in stochastic optimization methods with emphasis on applications to multicriterial, multimodal and surrogate function-based optimization methods.

## 1. Úvod

Hned na začátku tohoto článku je nutné se zmínit o významu slova *náhodný*. Většina vědecké literatury<sup>1</sup> používá toto slovo pouze pro některé fyzikální děje a procesy, na jejichž základě je případně možné získat náhodné realizace v digitální podobě. Posloupnosti (sekvence) náhodných čísel daných matematickými algoritmy, které se nejčastěji používají v programech pak postrádají reálnou náhodnost (po určité době se produkovaná čísla začínají opakovat) a proto se v tomto kontextu nazývají *pseudo-náhodná*. Typickým příkladem je standardizovaná funkce `rand()` dostupná téměř ve všech programovacích jazycích. Jelikož se tato práce zabývá tvorbou náhodných vektorů na běžných osobních počítačích, budeme pod pojmem *náhodná čísla a náhodné vektory* nadále uvažovat pouze jejich pseudo-náhodné varianty.

Dalším důležitým pojmem jsou takzvané *kvazi-náhodné* posloupnosti. Ty si zachovávají svoji jistou vnitřní "náhodnost", ale zároveň se snaží rovnoměrně pokrýt daný definiční obor. Tyto protichůdné požadavky, tedy náhodnost proti rovnoměrnosti, do jisté míry splňují některé matematické posloupnosti, např. známé

---

<sup>1</sup> viz např. [3] nebo [15]

Sobolovovy sekvence [15], nebo metody založené na metodě Monte Carlo a jejich *kvazi-* verzích. Metoda Latin Hypercube Sampling (LHS) použitá v této práci patří právě do této poslední skupiny.

V současné době se v oblasti aplikované vědy často setkáváme s problémy, které nejsme schopni v reálném čase vyřešit s pomocí známých, deterministických metod. Díky tomu se otvírají dveře metodám přibližným (aproximativním), nepřesným a stochastickým a zejména metody z oblasti takzvaných umělých inteligencí získávají v posledních letech na oblibě. Jako příklad přibližných metod zde dále budeme odkazovat na metody umělých neuronových sítí, viz např. [8] nebo [18], k nepřesným metodám bychom zařadili veškeré "fuzzy" přístupy, viz např. [21], a ke stochastickým metodám zařazujeme veškeré postupy využívající bezprostředně ke své funkci náhodná čísla nebo náhodné vektory. Tato prezentovaná práce se pak zabývá tvorbou náhodných vektorů pro první a třetí výše citovanou oblast.

## 1.1 Metoda LHS

Metoda Latin Hypercube Sampling (LHS) je jednou ze simulačních metod typu Monte Carlo (MC) [3]. Byla vyvinuta k odhadu statistických parametrů odezvy (konstrukce) a oproti jednoduché metodě MC výrazně snižuje počet potřebných simulací při zachování požadované přesnosti odhadů statistik odezvy [17].

Definiční obor funkce hustoty  $f(X_i)$  každé základní náhodné veličiny  $X_i$  je rozdělen na  $N_{sim}$  ( $N_{sim}$  je počet simulací) disjunktních intervalů. Intervaly jsou voleny tak, aby měly stejnou pravděpodobnost  $1/N_{sim}$ . Tím dosáhneme rovnoměrného pokrytí celého rozsahu každé základní náhodné veličiny vzhledem k distribuční funkci a zajistíme, že žádná hodnota není předem vyloučena a každá reálná hodnota je použita právě jednou.

Metodu LHS bychom mohli rozdělit do dvou fází. Tou první je výběr vzorků představujících intervaly rozdělení pravděpodobnosti náhodné veličiny. Obvykle se vzorky vybírají inverzní transformací pravděpodobnostní funkce v polovině vrstvy rovnoměrně rozděleného oboru hodnot kumulativní distribuční funkce. To může vést ke zkreslení požadovaného rozptylu, především kvůli krajním vrstvám. Proto je lépe provádět výběr vzorků jako střední hodnotu každého intervalu, tedy s pomocí přídavné integrace.

Ve druhé fázi LHS zavádí statistickou závislost mezi vzorky. Požadujeme, aby jednotlivé vektory matice  $\mathbf{X}$  byly nezávislé, tomu odpovídá i nezávislost statistických souborů tvořících jednotlivé realizace veličiny. O statistické nezávislosti matice  $\mathbf{X}$  se můžeme přesvědčit výpočtem pomocí např. Pearsonova či jiného statistického korelačního koeficientu. Korelační koeficient poté měníme směrem k požadovanému už jen změnami pořadí vzorků u jednotlivých veličin, ne změnou jejich hodnoty. Tím stále udržíme požadované charakteristiky veličiny jako rozptyl nebo střední hodnotu.

## 2.2 Simulované žíhání

Metoda simulovaného žíhání (Simulated Annealing - SA) [5,6] je stochastická optimalizační metoda, která má svůj základ spíše ve fyzice než v biologii. Jejím vzorem je proces žíhání v pevných látkách. Ve fyzice se pojmem žíhání označuje proces, kdy je pevné těleso v peci vyhřáto na vysokou teplotu, teplota se potom pomalu snižuje tak, aby jednotlivé částičky materiálu měly čas zaujmout na dané teplotní hladině stav s minimální energií. Jak klesá teplota materiálu, klesá i celková energie až dosáhne své minimální hodnoty. Proces se využívá například k odstraňování vnitřních defektů tělesa.

Algoritmus SA funguje podobně - je vytvořeno počáteční řešení, parametr nazývaný *teplota* je nastaven na startovací teplotu a je vytvořeno náhodně nové řešení v okolí již známého řešení. Je-li nové řešení podle požadovaných kritérií „lepší“ je automaticky přijato do populace. Je-li horší, je přesto s určitou pravděpodobností přijato do populace.

Tato pravděpodobnost se pak řídí Boltzmannovým rozdělením a je závislá na stavu populace a aktuální teplotě. Vztah

$$Pr(E) = \frac{1}{1 + \exp\left(\frac{-\Delta E}{\kappa_B T}\right)},$$

kde  $\kappa_B$  je Boltzmannova konstanta, popisuje rozdělení energie systému, který je v tepelné rovnováze s teplotou  $T$ , mezi všemi rozdílnými energetickými stavy.

Proces je opakován několikrát při konstantní teplotě, která je postupně snižována do předepsaného minima.

Simulované žihání je možné rozšířit o některé prvky známé z genetických algoritmů, konkrétně o populace chromozómů a genetické operátory. Obecně můžeme tento rozšířený algoritmus stručně zapsat takto:

1.  $T = T_{max}$ ,  $t = 0$

Počáteční teplotu  $T_{max}$  je třeba zvolit tak, aby poměr přijatých jedinců ke všem vytvořeným byl cca 50%.

2. vytvoř  $P_0$ , ohodnot  $P_0$

3. **while** (**not** zastavovací podmínka) {

Jako zastavovací podmínka je nastaveno překročení maximálního počtu iterací.

4. `count = succ = 0`

5. **while** (`count < countmax  $\wedge$  succ < succmax`) {

Proměnná  $count_{max}$  označuje počet všech iterací a  $succ_{max}$  počet úspěšných iterací na dané teplotní hladině. Doporučený poměr těchto hodnot je  $count_{max} = 10 succ_{max}$ .

6. `count = count + 1, t = t + 1`

7. vyber jedince  $I_t$  z  $P_{t-1}$

8. vyber operátor  $O$

Je vybrána mutace nebo křížení.

9. změň  $I_t$  operátorem  $O$ , výsledek je  $I_t'$

10. `p = 1/(1+exp ((F( $I_t'$ ) - F( $I_t$ ))/T))`

Zde je počítána zmíněná pravděpodobnost nahrazení rodiče horším potomkem. Díky snižování teploty se tato pravděpodobnost směrem ke globálnímu optimu snižuje.

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  $T$

Tento krok je nazýván ochlazování. Je použit jednoduchý vztah pro snižování teploty ve tvaru  $T_{i+1} = T_{mult} T_i$ . Součinitel  $T_{mult}$  (redukční součinitel), je možné nastavit na určitou hodnotu (obecně je doporučována hodnota 0,95) nebo jej vypočítat podle následujícího vztahu:

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

aby při zvoleném počtu iterací  $iter_{max}$  byla dosažena minimální teplota  $T_{min}$ .

18. }

## 2. Generování vstupních dat pro neuronové sítě

### 2.1 Trénování umělých neuronových sítí

První část této práce se tedy bude zabývat tvorbou nezávislých vektorů pro umělé neuronové sítě (dále jen neuronové sítě). Ty byly původně vytvořeny k simulaci procesů v lidském mozku, nicméně se posléze zjistilo, že jednoduché spojení několika umělých neuronů (perceptronů) může sloužit k řešení mnoha problémů jako je rozpoznávání obrazců (*pattern recognition*), různé druhy aproximací a interpolací nebo i různých systémů řízení. V této práci se budeme zabývat pouze dílčí oblastí neuronových sítí, konkrétně vrstevnatými sítěmi s dopředným šířením (*feed-forward layered neural networks*), které se používají zejména pro aproximaci, interpolaci a predikci dat.

Výstup tohoto typu neuronových sítí je dán jednak její topologií (tou se ale nebudeme v této práci zabývat) a pak hlavně trénovacím procesem (učení), při kterém jsou nastaveny váhy jednotlivých synaptických vah. Proces učení lze taktéž chápat jako optimalizační úlohu, při které je minimalizována chyba mezi známými výstupy, které chceme získat, a těmi, které získáme pomocí predikce neuronové sítě. Jinými slovy, na počátku trénovacího procesu je nutné poskytnout množinu vzorů a obrazů, které chceme, aby neuronová síť věrně reprodukovala, a při trénovacím procesu minimalizujeme rozdíl mezi původními a obdržnými obrazy.

V současných inženýrských aplikacích neuronových sítí nejsou problémem ani tak vlastnosti té které neuronové sítě, ale množství trénovacích dat. Současné experimenty jsou značně nákladné a proto je obecnou snahou reálné experimenty přenést do počítačů. A tady vyvstává problém, jak vytvořit množinu reprezentativních dat. Jsou zde dva protichůdné požadavky - jednak snaha o maximální počet vzorků tak, aby byly postiženy všechny možné stavy materiálu a/nebo konstrukce, a zároveň, aby jejich počet byl minimální, aby se zjednodušil proces trénování neuronové sítě. Zde je jasná podobnost s protichůdnými požadavky *kvazi-náhodných* čísel a posloupností, které se v současné době při učení neuronových sítí začínají nejčastěji používat. Porovnání několika generátorů *kvazi-náhodných* sekvencí bylo nedávno publikováno v [18], kde F. Tong ukázal, že jako nejvhodnější se jeví použití matematických generátorů typu Sobolevovských sekvencí. V České Republice se pak touto problematikou zabývá D. Lehký na VUT v Brně, viz např. [10] nebo [11]. Ten konkrétně používá metodu Latin Hypercube Sampling (LHS) kombinovanou s metodou simulovaného žihání k získání nezávislých náhodných vektorů, viz též [19]

a [20]. V této práci jsme aplikovali stejnou metodologii a zaměřili jsme se na grafické výstupy z tohoto postupu.

## 2.2 Metoda

Pro generování vstupních dat pro neuronové sítě bylo použito spojení metody LHS se simulovaným žíháním. Nejprve byla s pomocí LHS vytvořena počáteční populace, resp. matice vzorků  $\mathbf{X}$ , počet sloupců je dán počtem veličin  $N_v$ , počet řádků je roven počtu vzorků  $N_{sim}$ . Jelikož v práci bylo uvažováno pouze rovnoměrné rozdělení, při kterém nedochází k žádnému zkreslení na krajích definičního oboru, bylo možné vzorky volit ve středu intervalů.

Dále bylo požadováno, aby vzorky byly vzájemně nezávislé. Proto byla zavedena matice  $\mathbf{S}$  obsahující korelační koeficienty mezi jednotlivými veličinami v aktuálním vzorku a matice  $\mathbf{K}$  obsahující požadované korelační koeficienty, v případě nezávislosti to je jednotková matice. K vyhodnocení kvality celkových statistických závislostí v matici vzorků  $\mathbf{X}$  byla zavedena norma  $E$ , která zohledňuje deviace všech koeficientů:

$$E = \sqrt{\sum_{i=1}^{N_v-1} \sum_{j=i+1}^{N_v} (S_{i,j} - K_{i,j})^2},$$

v případě požadované nezávislosti veličin je možné matici  $\mathbf{K}$  z výpočtu vypustit (nad diagonálou jsou pouze nuly).

Nyní chceme matici  $\mathbf{X}$  změnit tak, aby korelační koeficienty mezi jednotlivými veličinami a tedy i norma  $E$  byly co nejnižší. Iterační proces bychom mohli rozdělit do dvou kroků: mutace a selekce. Během mutace dochází k náhodné změně vzorků. Tuto změnu uskutečňujeme změnou pořadí jednotlivých hodnot vzorků ne změnou těchto hodnot. Pokud bychom chtěli vyzkoušet všechny možnosti, bylo by to  $(N_{sim})^{N_v-1}$  iterací, což je při větším množství vzorků a veličin nerealizovatelné.

Proto se při změně vzorků používá algoritmus simulovaného žíhání, který při relativně malém počtu iterací dosáhne dostatečně nekorelovaných vzorků. Optimalizovanou (minimalizovanou) funkcí je norma  $E$ . V každém iteračním kroku se provede pouze jedna náhodná záměna dvojice vzorků u jedné veličiny. Protože známe umístění změněných vzorků, je možné nově spočítat jen dotčené korelační koeficienty, což výrazně uspoří výpočtový čas.

Po změně vzorků a výpočtu nové normy dochází k selekci. Pokud je nová norma menší než původní, je nová matice vzorků automaticky přijata. Pokud je nová norma větší, aplikuje se simulované žíhání. S pravděpodobností

$$Pr(E) = \frac{1}{1 + \exp\left(\frac{-\Delta E}{\kappa_B T}\right)},$$

kde  $\Delta E$  je rozdíl norem  $E$  po záměně a před záměnou, je přijata i norma horší. Hodnotu  $\kappa_B$  (Boltzmannovu konstantu) můžeme z výpočtu vypustit (tedy položit rovnou jedné), protože normu  $E$  a teplotu  $T$  uvažujeme ve stejném rozměru.

Maximální teplotu nastavujeme tak, aby v počátečních iteracích byl poměr všech iterací a úspěšných iterací 50%. Nastavení  $T_{max}$  nemusí v tomto případě



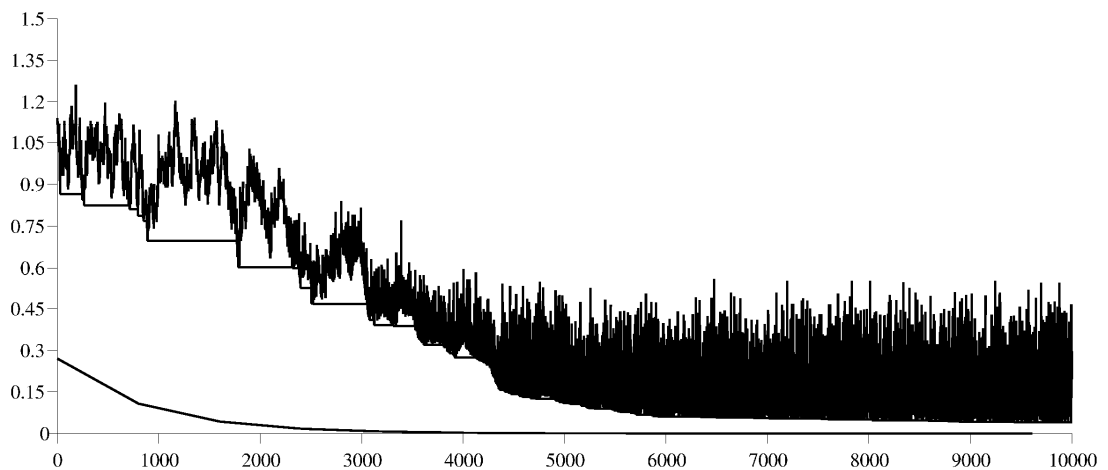
probíhat metodou „pokus - omyl“, protože známe požadovanou výslednou matici  $\mathbf{K}$  a víme, že členy optimalizované matice  $\mathbf{S}$  leží v intervalu  $\langle -1; 1 \rangle$ . Takže můžeme maximum normy spočítat pomocí počtu vzorků  $N_v$  a teoretického největšího rozdílu mezi požadovaným a aktuálním korelačním koeficientem:

$$T_{max} = 0.03(N_v(N_v - 1)/2).$$

Minimální teplota by měla být velmi nízká, v předkládané práci je používána hodnota  $T_{min} = 0,000001T_{max}$ , aby byl směrem k minimu vliv simulovaného žíhání potlačen a nedošlo tak k zavržení již nalezeného minima.

Optimalizační proces byl zastaven při dosažení maximálního počtu iterací  $iter_{max}$ .

Typický průběh simulovaného žíhání, snižování teploty i aktuální normy  $E$  je na **Obr. 1**



**Obr. 1** Typický průběh simulovaného žíhání: (odspodu) průběh teploty, nejlepší a aktuální hodnota účelové funkce.

## 2.3 Výsledky

Díky popsanému optimalizačnímu procesu skutečně dochází k významnému snížení korelačních koeficientů a normy  $E$ . Pro ilustraci jsou dále uvedeny průměrné hodnoty normy při generování 30 vzorků o 8 proměnných.

Na začátku optimalizace, tedy po vytvoření vzorků pomocí LHS, byla průměrná norma  $E = 1,049219$ . Pro ilustraci je uvedena první matice  $\mathbf{S}$ :

$$\mathbf{S}_1 = \begin{pmatrix} 1 & 0.009411 & 0.17909 & -0.33482 & 0.08565 & -0.28365 & -0.13637 & -0.13771 \\ 0.009411 & 1 & -0.25161 & -0.08743 & 0.02870 & -0.48654 & -0.14216 & -0.00868 \\ 0.17909 & -0.25161 & 1 & 0.03760 & -0.16574 & 0.04961 & -0.05228 & -0.15284 \\ -0.33482 & -0.08743 & 0.03760 & 1 & -0.07408 & -0.01980 & -0.19555 & 0.09366 \\ 0.08565 & 0.02870 & -0.16574 & -0.07408 & 1 & 0.06474 & 0.05718 & -0.44116 \\ -0.28365 & -0.48654 & 0.04961 & -0.01980 & 0.06474 & 1 & 0.04650 & -0.14661 \\ -0.13637 & -0.14216 & -0.05228 & -0.19555 & 0.05718 & 0.04650 & 1 & 0.04071 \\ -0.13771 & -0.00868 & -0.15284 & 0.09366 & -0.44116 & -0.14661 & 0.04071 & 1 \end{pmatrix}$$

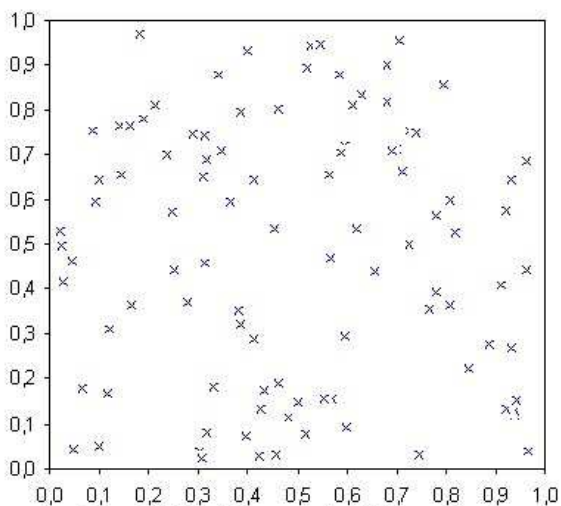
Po proběhnutí maximálního počtu iterací ( $iter_{max}$ ) byly průměrné normy  $E$  následující:

$iter_{max}$	$E$
10 000	0,016222
100 000	0,013303
1 000 000	0,011734

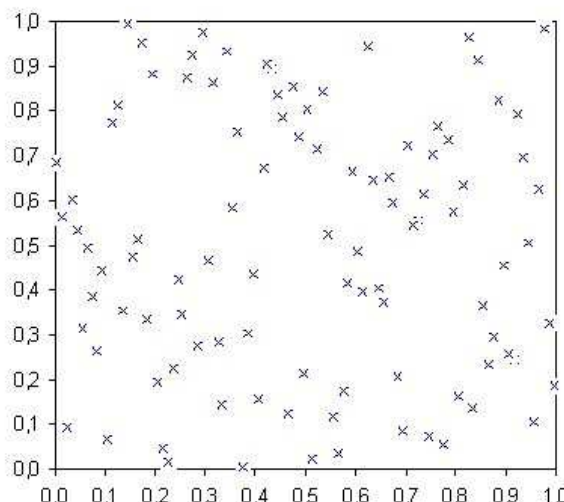
Pro představu, jak velké je snížení jednotlivých korelačních koeficientů, je uvedena konečná matice  $S$ :

$$S_{1000000} = \begin{pmatrix} 1 & -0.00200 & 0.00067 & 0.00334 & 0.00067 & 0.00022 & 0.00200 & -0.00156 \\ -0.00200 & 1 & 0.00512 & -0.00156 & -0.00200 & 0.00334 & -0.00067 & 0.00022 \\ 0.00067 & 0.00512 & 1 & -0.00067 & 0.00200 & -0.00111 & -0.00200 & 0.00156 \\ 0.00334 & -0.00156 & -0.00067 & 1 & 0.00156 & 0.00334 & 0.00289 & 0.00156 \\ 0.00067 & -0.00200 & 0.00200 & 0.00156 & 1 & -0.00200 & 0.00111 & -0.00156 \\ 0.00022 & 0.00334 & -0.00111 & 0.00334 & -0.00200 & 1 & 0.00378 & 0.00022 \\ 0.00200 & -0.00067 & -0.00200 & 0.00289 & 0.00111 & 0.00378 & 1 & 0.00156 \\ -0.00156 & 0.00022 & 0.00156 & 0.00156 & -0.00156 & 0.00022 & 0.00156 & 1 \end{pmatrix}$$

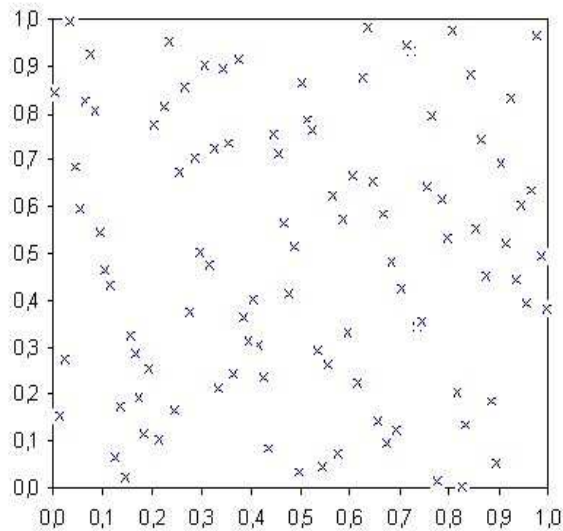
Přestože snížení korelačních koeficientů je skutečně výrazné, při grafickém znázornění vzorků se zdánlivě neliší rozdělení vzorků před optimalizačním procesem, po proběhnutí simulovaného žihání a zcela náhodné rozdělení vzorků. Přitom při optimalizaci souboru 100 vzorků o dvou proměnných byla norma  $E$  snížena na nulu. Na následujících obrázcích jsou znázorněny výsledky těchto tří způsobů rozmístění: zcela náhodné rozdělení 100 vzorků (viz **Obr. 2**), na **Obr. 3** je vidět rozmístění vzorků po aplikaci metody LHS a na **Obr. 4** je výsledné rozmístění po proběhnutí 10 000 iteračních cyklů. U všech obrázků je zapsána jejich norma  $E$ .



**Obr. 2** Náhodné rozmístění 100 vzorků  
 $E = 0,045768$



**Obr. 3** Rozmístění 100 vzorků pomocí LHS  
 $E = 0,052037$

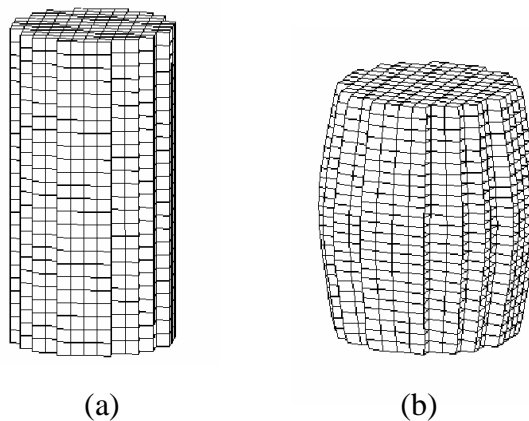


Obr. 4 Rozmístění 100 vzorků po SA  
 $E = 0,000000000000$

## 2.4 Sensitivní analýza

Přestože předchozí grafické výsledky ukazují zanedbatelný vliv zahrnutí statistické nezávislosti, její výsledný účinek bude patrný až v nějakém konkrétním případě s danou neuronovou sítí. Testování na několika případech s různými typy neuronových sítí by přesáhlo rozměr této práce a proto jsme se zaměřili pouze na jeden problém – v tomto případě identifikaci materiálových parametrů mikroploškového modelu betonu.

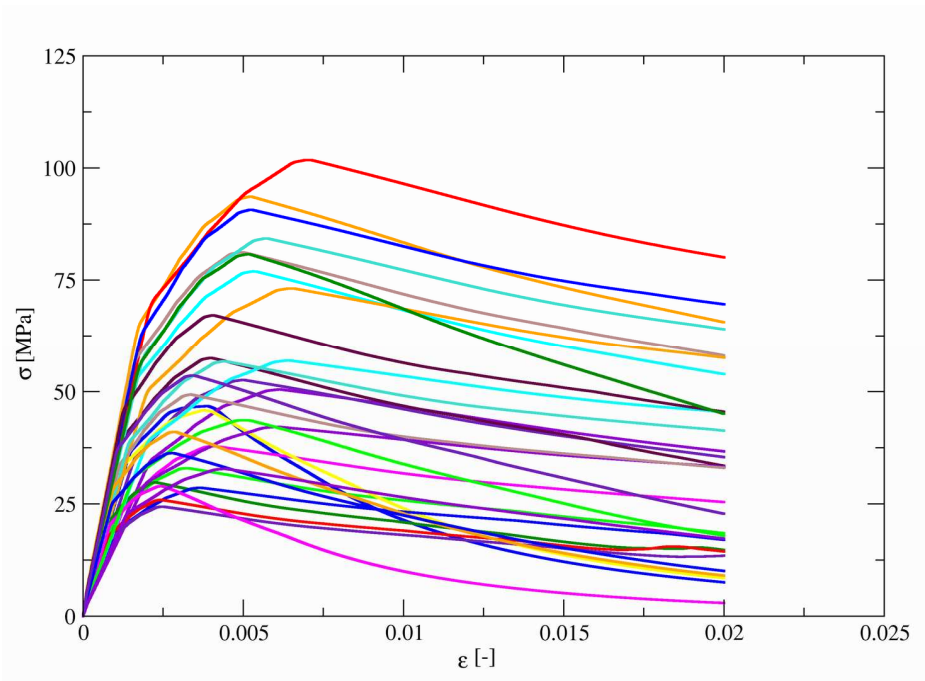
Mikroploškový model [2,7] je trojrozměrný materiálový model, který je schopen popsat různorodou odezvu betonu, změkčení v tahu i v tlaku, porušení materiálu, odtěžování i cyklické zatěžování včetně materiálové anisotropie. Určitý typ betonu je pak popsán osmi parametry: Youngovým modulem pružnosti  $E$ , Poissonovým součinitelem  $\nu$  a dalšími šesti parametry ( $k_1, k_2, k_3, k_4, c_3, c_{20}$ ), které nemají jednoznačnou fyzikální interpretaci a tudíž jsou velice těžko získatelné z experimentů.



Obr. 5 Výpočetní model tlakové zkoušky a) na začátku a b) na konci zatěžování

Učení neuronové sítě by zde bylo opět příliš náročné, ale z některých prací (např. [9]) vyplývá, že schopnost neuronové sítě naučit se některý z těchto parametrů je přímo úměrná jeho statistickému vlivu na odezvu materiálu. Proto jsme výše popsanými metodami vytvořili sadu materiálových parametrů betonu pro mikroploškový model z mezích daných **Tab. 1**. Ze 30 simulací zkoušky v jednoosém tlaku (viz. **Obr. 5** a **Obr. 6**) jsme vypočetli Pearsonův korelační koeficient vlivu jednotlivých parametrů

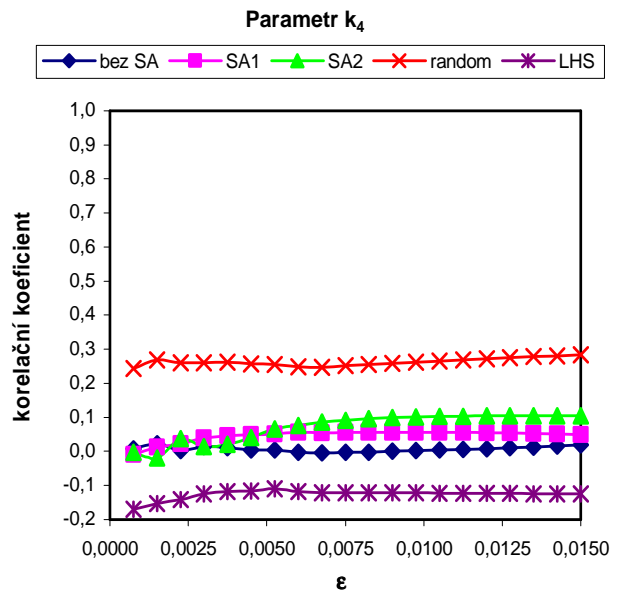
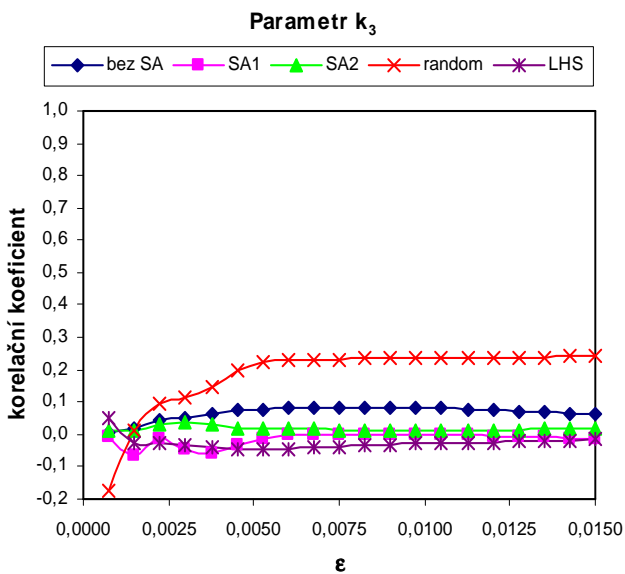
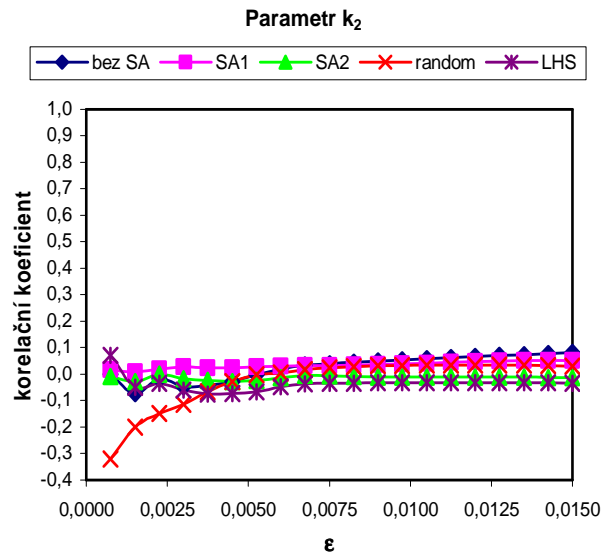
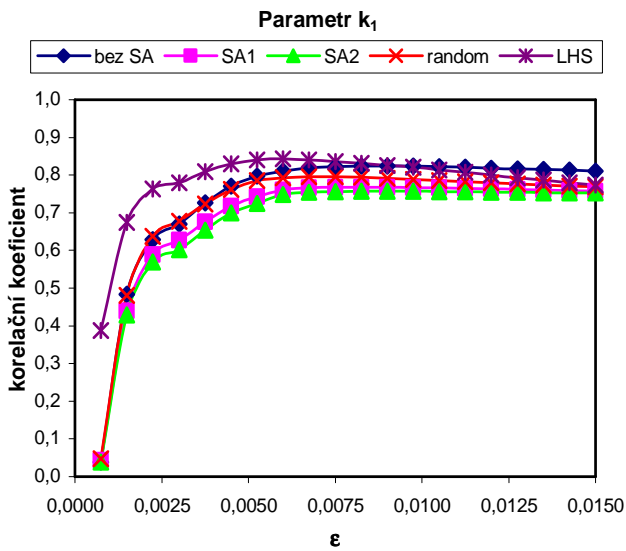
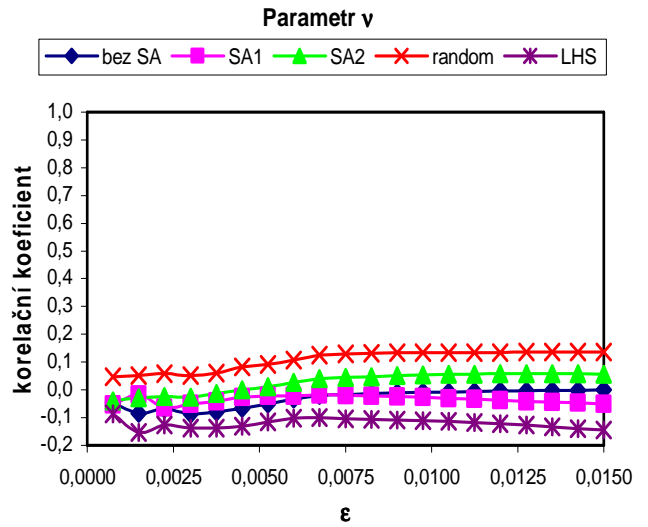
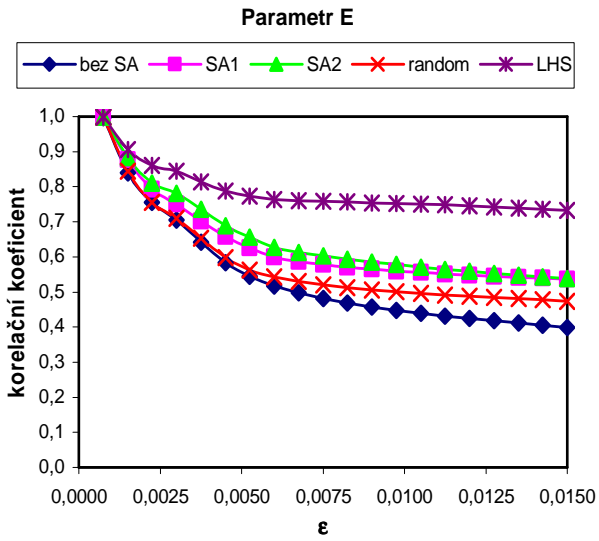
$$cor = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

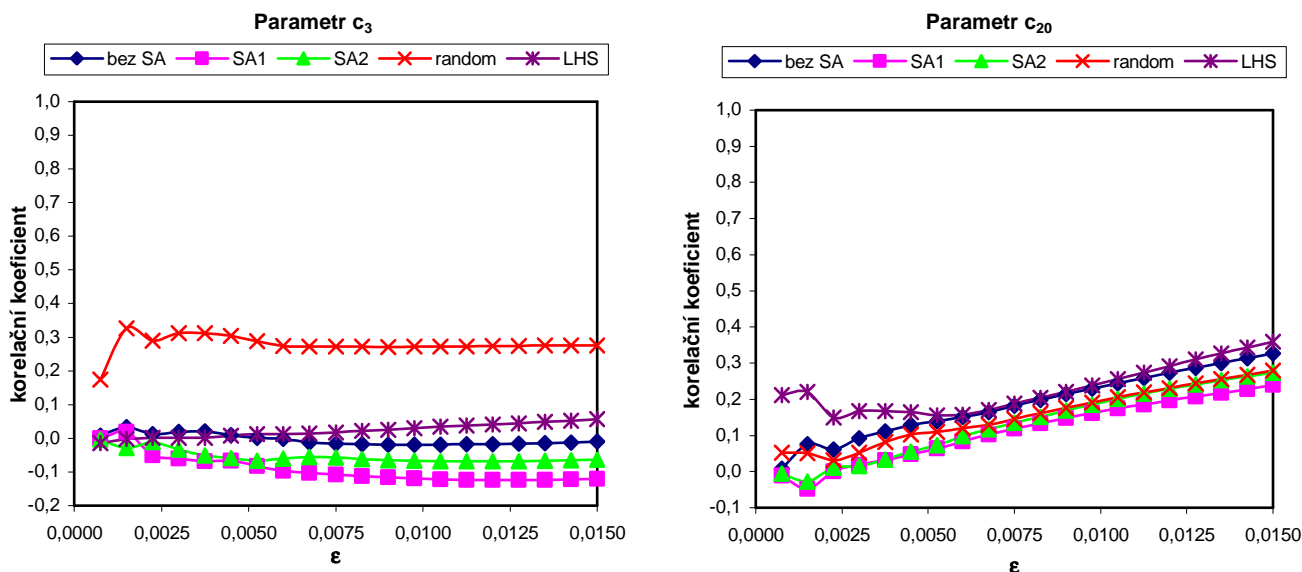


**Obr. 6** Pracovní diagramy získané simulací zkoušky v jednoosém tlaku

$E$	∈	(20.0, 50.0) GPa
$\nu$	∈	(0.1, 0.3)
$k_1$	∈	(0.00008,
$k_2$	∈	(100.0, 1000.0)
$k_3$	∈	(5.0, 15.0)
$k_4$	∈	(30.0, 200.0)
$c_3$	∈	(3.0, 5.0)
$c_{20}$	∈	(0.2, 5.0)

**Tabulka 1:** Meze mikroploškových parametrů





**Obr. 7** Srovnání korelačních koeficientů jednotlivých metod pro vyjmenované parametry mikroploškového modelu („bez SA“ značí výběr pouze lepšího jedince v metodě SA, „SA1“ značí námi popisovanou metodu, „SA2“ je získáno pomocí komerčního softwaru FREET [14], „random“ je vytvořeno pomocí standardního generátoru náhodných čísel a „LHS“ značí výsledky obdržené pouze s pomocí LHS.

Z obrázků je patrné, že u této konkrétní aplikace jsou rozdíly mezi jednotlivými metodami tvorby nezávislých vektorů minimální. Lze pouze konstatovat, že obecný generátor náhodných čísel produkuje vyšší korelační koeficienty u parametrů s minimálním vlivem. Opak je pak pravdivý pro osamocenou metodu LHS, která dosáhla maximálních korelačních koeficientů u parametrů  $E$ ,  $k_1$  a  $c_{20}$ , které mají největší vliv na tlakovou odezvu betonu.

### 3. Generování vstupních dat pro optimalizaci

#### 3.1 Náhodné vektory pro stochastické optimalizace

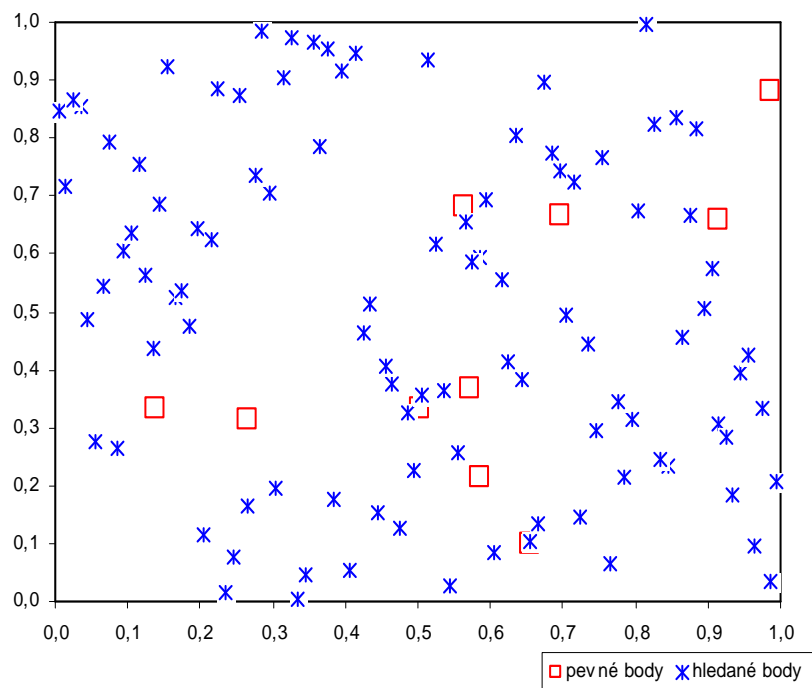
Přestože druhá část této práce míří do oblasti globálních stochastických optimalizačních metod, jako je výše představené simulované žíhání, nebo oblast evolučních algoritmů jako jsou genetické algoritmy [4], evoluční strategie [1] či diferenciální evoluce [16], výsledky tohoto výzkumu mohou být aplikovány i v oblasti tradičních optimalizačních metod, kam budeme řadit všechny metody (gradientní) využívající znalosti prvních a vyšších derivací účelové funkce. Většina výše zmiňovaných metod je přímo založena na náhodných procesech, nebo alespoň jednou využívá generátor náhodných čísel, např. pro tvorbu startovacího bodu. Cílem je tedy vytvořit takovou posloupnost náhodných čísel nebo vektorů, která bude minimalizovat počet potřebných iterací optimalizační metody nebo bude zvyšovat robustnost dané metody. Touto problematikou se ve svých pracích v současné době zabývá H. Maaranen, viz. [12] a [13]. Konkrétně studuje vliv náhodně vytvořené počáteční populace na konvergenční charakteristiky genetického algoritmu.

Tato práce se proto zaměřila na dosud neprobádanou oblast multimodálních a multikriteriálních optimalizačních problémů. Ty jsou charakterizovány velkým množstvím lokálních minim na jedné straně a množinou nedominantních řešení (tzv.

*Pareto množiny*) na straně druhé. V průběhu řešení těchto algoritmů nastává situace, že je již část definičního prostoru prozkoumána a je zapotřebí prozkoumat jiné oblasti. Úloha může být tedy definována takto: Vytvořte sekvenci náhodných vektorů, které mají maximálně rovnoměrné rozdělení, ale jsou také maximálně rozdílné od již známých (objevených, ohodnocených) vektorů. Pro řešení jsme využili nejprve metodiku shodnou z předchozího případu neuronových sítí - tedy vytvořit nové vektory jako statisticky nezávislé na již známých vektorech pomocí metody LHS a simulovaného žíhání. Jako druhou možnost jsme zvolili taktéž optimalizační přístup, kdy je na výsledky z metody LHS použito simulovaného žíhání, ale účelovou funkcí je tentokrát součet vzdáleností nových vektorů od již známých, pevně daných vektorů. Podrobný popis jednotlivých částí této metody následuje.

### 3.2 Popis metody a výsledky

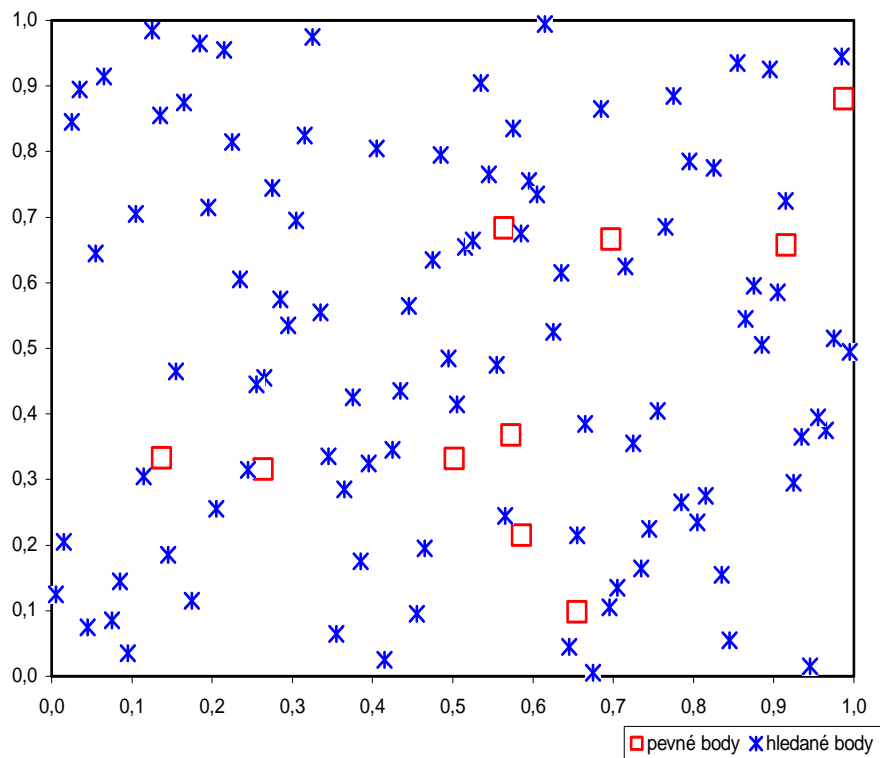
Nejprve bylo zcela náhodně zvoleno 10 bodů. Tyto body představují již nalezená řešení a během iteračních cyklů se s nimi již nehýbe. Jejich počet označme  $M$ . Poté k nim bylo pomocí metody LHS vygenerováno dalších 100 bodů (počet označme  $N_{sim}$ ), které by měly být na původních 10 pevných bodech co nejvíce nezávislé, tedy co nejvíce vzdálené. Situace po vygenerování nových bodů je znázorněna na **Obr. 8**. K dosažení nezávislosti byly zvoleny dva odlišné přístupy, jeden statistický a druhý geometrický.



**Obr. 8** Počáteční rozmístění vzorků pomocí LHS

Statistický přístup je v podstatě totožný s postupem popsáným v předchozí kapitole. Matice vzorků  $\mathbf{X}$  má  $N_{sim}+M$  řádků (100 nových optimalizovaných bodů + 10 pevných bodů) a  $N_v$  sloupců (počet proměnných). Během mutace může být měněno pouze  $N_{sim}$  bodů a norma  $E$  je počítána podle vzorce uvedeného v 2.2. Výsledná situace po proběhnutí 100000 iteračních cyklů je zachycena na **Obr. 9**. Přestože došlo ke snížení celkové normy  $E$ , graficky se opět konečná situace příliš neliší od

výchozího stavu. To by bylo možné vyřešit zavedením vah některých korelačních koeficientů.



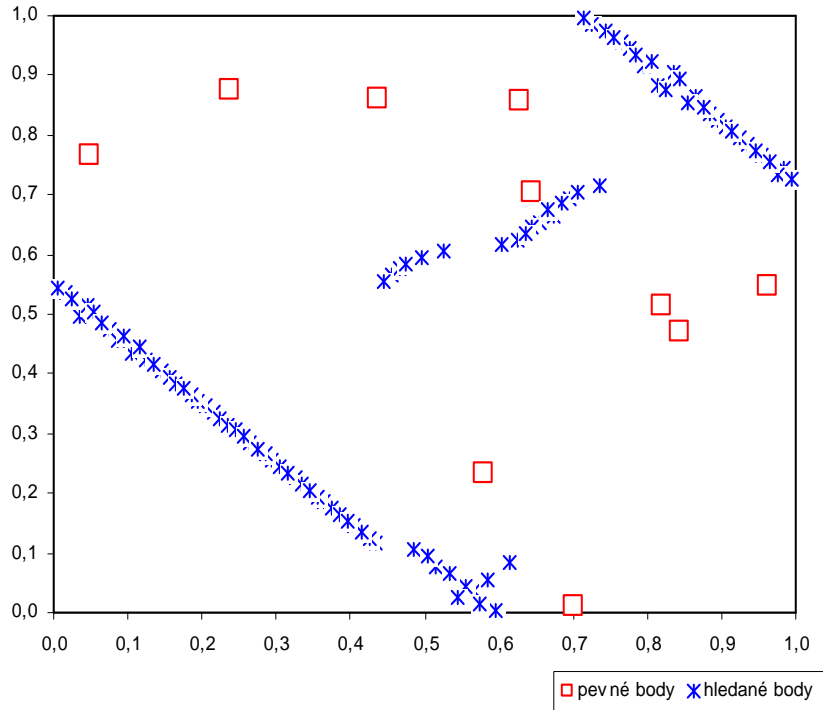
**Obr. 9** Výběr vzorků pomocí statistické korelace

Geometrický přístup je možné dále rozdělit na dvě alternativy. V první je optimalizovanou funkcí součet všech vzdáleností optimalizovaných vzorků od pevných vzorků. Minimalizovaná funkce  $E$  tedy vypadá takto:

$$E = \sqrt{\sum_{i=0}^{Nsim} \sum_{j=0}^M \sum_{k=0}^{Nv} (x_{i,k} - x_{j,k})^2}$$

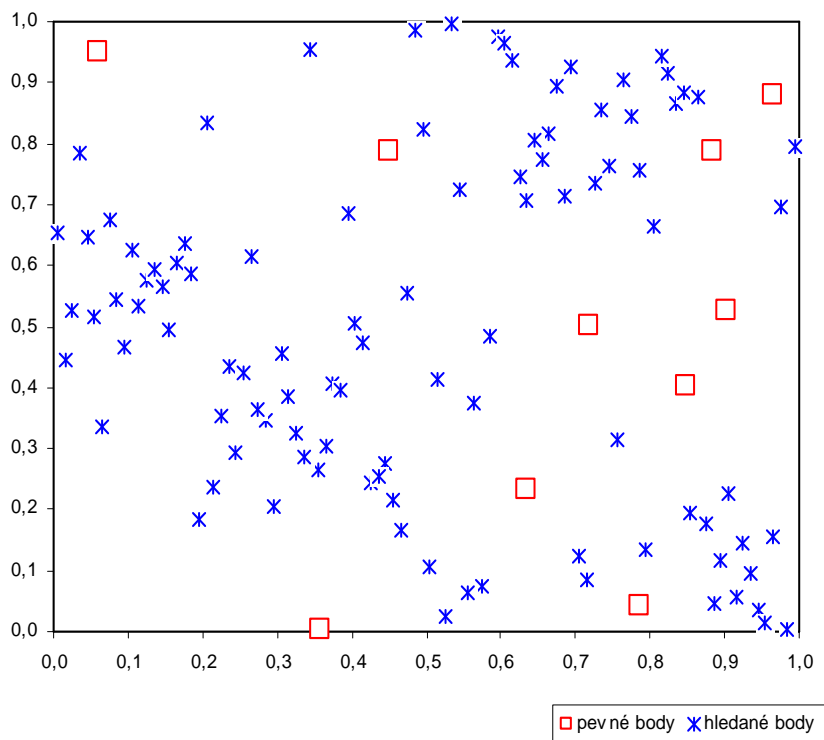
Jak je vidět z **Obr. 10**, tento způsob vede k zajímavým, leč nepoužitelným výsledkům.





**Obr. 10** Výběr vzorků pomocí součtu vzdáleností

Druhou možností je maximalizovat vzdálenost každého optimalizovaného bodu od nejbližšího pevného bodu. Minimalizovanou funkcí  $E$  je tedy záporná hodnota této minimální vzdálenosti. V každém iteračním cyklu probíhá mutace stejným způsobem jako v Kapitole 2. Potom je nalezena nová nejmenší vzdálenost  $E$  a probíhá selekce se simulovaným žiháním. Na **Obr. 11** je znázorněno výsledné rozmístění bodů. Je patrné, že tento přístup je jediný, který vede k očekávanému rozmístění, tedy že kolem pevných bodů je určitý prostor volný.



**Obr. 11** Výběr vzorků pomocí nejmenší vzdálenosti

## 4. Závěr

Tato práce se zabývala tvorbou náhodných a nezávislých vektorů nejprve pro trénovací množiny pro umělé neuronové sítě. Druhá část práce je pak věnována náhodným číslům a vektorům pro oblast stochastických optimalizačních algoritmů. Pro obě oblasti byla aplikována metoda LHS spolu s algoritmem simulovaného žíhání. Z výsledků vyplývá, že tento postup dává v případě neuronových sítí uspokojivé výsledky, nicméně srovnatelné výsledky byly získány s mnohem jednoduššími metodami. Je možné, že v kombinaci s konkrétní neuronovou sítí na konkrétním příkladu metoda LHS spolu s SA povede k lepším výsledkům, ale z důvodu výpočetní náročnosti nelze doporučit tuto metodu k obecnému použití.

Také v druhém případě se nepodařilo s kombinací LHS, SA a statistické normy dosáhnout uspokojivých výsledků. Těch bylo dosaženo až s normou založenou na vzdálenosti nových vektorů k nejbližšímu pevnému vektoru. Zde se již naplno projevila výhoda metody LHS, která zaručuje rovnoměrné rozdělení náhodných vektorů po celém definičním oboru. Grafické výstupy z této poslední metody jsou velice slibné, nicméně jejich praktický dopad se bude muset vyzkoušet až na konkrétních aplikacích.

## Poděkování

Autorka by ráda poděkovala za finanční podporu grantem GAČR 106/03/0180.

## Literatura

- [1] T. Bäck and H.-P. Schwefel, Evolution Strategies I: Variants and their computational implementation, *Genetic Algorithms in Engineering and Computer Science* (Periaux and Winter, eds.), John Wiley & Sons Ltd., 1995, pp. 111–126.
- [2] Z. P. Bažant, F. C. Caner, I. Carol, M. D. Adley, and S. A. Akers, Microplane model M4 for concrete. Part I: Formulation with work-conjugate deviatoric stress, *Journal of Engineering Mechanics* **126** (9), 944–953, September 2000.
- [3] Russel E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, *Acta Numerica* (1998), 1–49.
- [4] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
- [5] L. Ingber, Simulated Annealing: Practice versus theory, *Mathematical and Computer Modelling*, vol. **18** (11), 29-57, 1993.
- [6] L. Ingber, Adaptive Simulated Annealing (ASA): Lessons learned, *Control and Cybernetics*, vol. **25** (1), 33-54, 1995.
- [7] M. E. Jirásek and Z. P. Bažant., Inelastic analysis of structures, John Wiley & Sons, 2001.

- [8] Marios K. Karakasis and Kyriakos C. Giannakoglou, On the use of surrogate evaluation models in multi-objective evolutionary algorithms, In P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, P. Périaux, and D. Knörzer (eds.), European congress on computational methods in applied sciences and engineering (Eccomas 2004), Jyväskylä, 24–28 July 2004.
- [9] A. Kučerová, M. Lepš, J. Zeman, Soft Computing Methods for Estimation of Microplane Model Parameters, Computational Mechanics Abstracts, Volume II: Abstracts of the Papers Presented at the Regular Sessions of the Sixth World Congress on Computational Mechanics in conjunction with the Second Asian-Pacific Congress on Computational Mechanics. Beijing: Thinghua University Press & Springer Verlag, 2004, s. 69.
- [10] D. Lehký, Material parameters identification for concrete failure modeling, 2<sup>nd</sup> PhD Workshop Brno-Wiener-Prague-Wien<sub>BOKU</sub>, Book of abstracts (J. Vrba, Z. Keršner, and P. Frantík, eds.), Brno University of Technology, Czech Republic, 2003, p. 20.
- [11] D. Lehký and D. Novák, Identification of material model parameters using stochastic training of neural network, Proceedings of 5th International PhD Symposium in Civil Engineering (Delft, Netherlands), 2004.
- [12] K. Miettinen, H. Maaranen and M. M. Mäkelä, Quasi-random initial population for genetic algorithms, *Computers and Mathematics with Applications* (2004), In print.
- [13] K. Miettinen, H. Maaranen and A. Penttinen, How to generate an initial population for genetic algorithms, In P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, P. Périaux, and D. Knörzer (eds.), *European congress on computational methods in applied sciences and engineering (Eccomas 2004)*, Jyväskylä, 24–28 July 2004.
- [14] Drahomír Novák et al: FREET Feasible Reliability Engineering Efficient Tool, Program documentation, Brno University of Technology, FCE / Červenka Consulting, Praha, 2002.
- [15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical recipes in C, 2nd ed., Cambridge University Press, 1992, Also available as [www.nr.com](http://www.nr.com).
- [16] R. Storn and K. Price, Differential Evolution : A simple and efficient adaptive scheme for global optimization over continuous spaces, Tech. Report TR-95-012, University of Berkeley, 1995.
- [17] B. Teplý and D. Novák, Spolehlivost stavebních konstrukcí, *Lecture notes*, Brno University of Technology, CERM, s.r.o., Brno, 1999.
- [18] F. Tong and X.L. Liu, On training sample selection for artificial neural networks using number-theoretic methods, *Proceedings of the Fourth International Conference on Engineering Computational Technology* (B.H.V. Topping and C.A. Mota Soares, eds.), Civil-Comp Press, Stirling, Scotland, 2004.
- [19] M. Vořechovský, Stochastic fracture mechanics and size effect, Ph.D. thesis, Brno University of Technology, 2004.
- [20] M. Vořechovský, D. Novák and R. Rusina, A new efficient technique for samples correlation in Latin Hypercube Sampling, *Proceedings of the VII Scientific*

*International Conference, Applied Mathematics* (Fakulta stavební, Košice, Slovensko), 2002.

[21] L.A. Zadeh, Fuzzy sets, *Information Control*, vol. **8** (3), 338-352, 1965.