

INŽENÝRSKÁ MECHANIKA 2005

NÁRODNÍ KONFERENCE

s mezinárodní účastí

Svratka, Česká republika, 9. - 12. května 2005

IDENTIFICATION OF MICROPLANE MODEL PARAMETERS FROM EXPERIMENTS IN UNIAXIAL COMPRESSION

A. Kučerová**, M. Lepš*, J. Zeman*

Summary: In this paper we present an identification method of microplane model parameters from results of experiments in a uniaxial deformation on concrete. Novelties are usage of the Latin Hypercube Sampling method for generation of training sets, a sensitivity analysis and genetic algorithm based training of a neural network by the evolutionary algorithm SADE. Advantages and disadvantages of this approach together with possible extensions are thoroughly discussed and analyzed.

1. Úvod

Beton jako stavební materiál vykazuje heterogenní chování a proto je jeho numerické modelování značně obtížné. Microplane model [1,4] 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. Hlavní nevýhodou tohoto modelu je však veliké množství fenomenologických materiálových parametrů. Proto je zapotřebí spolehlivá metoda vedoucí k jejich uspokojivému určení. Konkrétně je určitý typ betonu popsán osmi parametry: Youngovým modulem pružnosti E, Poissonovým součinitelem v 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 experimentálních dat.

Běžnou praxí je využití metody pokusu a omylu, kdy se experimentátor snaží manipulací s parametry "trefit" do získaných pracovních diagramů [1,5]. Díky značné nelinearitě odezvy konstrukce je tento postup netriviální, přestože lze v dostupné literatuře nalézt určité meze pro dané parametry. Zvolené meze daných parametrů v této práci jsou uvedeny v Tab. 1.

Vlastní optimalizační problém pak může být formulován například takto: Nalezněte materiálové parametry betonu tak, aby se výsledky numerické analýzy a grafy z experimentu při jednoosé tlakové zkoušce (viz Obr. 1) shodovaly. Objektivní funkcí je potom suma čtverců rozdílů mezi známým pracovním diagramem a výsledky ze simulací s použitím modelu microplane.

Značnou nevýhodou modelu microplane je jeho výpočetní náročnost. Například numerická analýza příkladu jednoosého tlaku (Obr. 1) trvala déle než 23 hodin na počítači osazeném procesorem Pentium II Xeon 400 MHz s 512 MB RAM, viz též [5]. Proto byla v minulých letech využita pouze simulace odezvy jednoho elementu modelu microplane.

^{*} Ing. Matěj Lepš, Ph.D., Ing. Anna Kučerová, Ing. Jan Zeman, Ph.D.: Katedra stavební mechaniky, Fakulta stavební, České Vysoké Učení Technické v Praze; Thákurova 7; 166 29 Praha 6 – Dejvice; tel: +420.224 354 375, fax: +420.224 310 775; e-mail: anicka@cml.fsv.cvut.cz

Tab. 1 Meze parametrů modelu microplane

Ε	\in	(20.0, 50.0) GPa
v	E	(0.1, 0.3)
k_l	E	(0.00008, 0.00025)
k_2	∈	(100.0, 1000.0)
k3	∈	(5.0, 15.0)
k_4	∈	(30.0, 200.0)
C ₃	∈	(3.0, 5.0)
C_{20}	∈	(0.2, 5.0)

V nedávné době bylo vyzkoušeno několik možností jak nastíněný problém řešit. Odhad materiálových parametrů pomocí umělé neuronové sítě trénované na aproximaci pracovních diagramů [6] ukázal, že některé parametry, jako např. modul pružnosti *E* a parametr k_1 , mohou být získány poměrně snadno, ale značné obtíže byly zaznamenány např. u koeficientu c_{20} . V další práci [7] byla využita paralelní verze genetického algoritmu SADE přímo na získání požadovaných materiálových parametrů. Prvním výsledkem bylo zjištění, že se dá značná výpočetní náročnost redukovat paralelní implementací na nezbytné minimum. Druhým důležitým výsledkem byl fakt, že daná optimalizační úloha má mnoho lokálních minim se shodnou objektivní funkcí a je tudíž nesnadné získat optimální hodnoty. S ohledem na nová zjištění v této oblasti je v tomto článku navržena nová metodika získávání parametrů modelu microplane.



Obr. 1 Výpočetní model tlakové zkoušky a) na začátku a b) na konci zatěžování

Vlastní článek je organizován následovně: další kapitola je věnovaná popisu navrhovaného postupu získávání materiálových parametrů betonu z výsledků zatěžovací zkoušky v jednoosém tlaku. Konkrétně je popsána aplikace umělé neuronové sítě trénované algoritmem SADE. Taktéž je zmíněno použití simulační metody Latin Hypercube Sampling a stochastické senzitivní analýzy na získávání nezávislých trénovacích množin. V závěrečné kapitole jsou shrnuty získané výsledky a poznatky.

2. Navrhovaná metodologie

a)

Neuronová síť

Navrhovaný postup využívá aplikace umělých neuronových sítí [8]. Konkrétně je představena trojvrstvá, plně propojená neuronová síť (viz Obr. 2).



Obr. 2 Architektura umělé neuronové sítě

Obecně je tato síť určena k projekci vstupního vektoru $I = (I_0, I_1,...I_m)$ na výstupní vektor $T = (T_0, T_1,...T_n)$. Síť je popsána L vrstvami označenými $l_0, l_1...l_{L-1}$, kde l_0 je vstupní a l_{L-1} je výstupní vrstva. V jedné vrstvě l_i je N_i neuronů označených $n_{i, 1}, n_{i, 2},...n_{i, Ni}$. Všechny vrstvy kromě výstupní mají tzv. "bias neuron" $n_{i,0}$. Spojení jednotlivých neuronů je charakterizováno váhami $w_{1,i,j}$, kde l je příslušná hladina, $i = 0, 1...N_{I-1}$ je index neuronu v předchozí hladině (i = 0 pro bias) a $j = 1, 2...N_I$ je index neuronu v hladině l. Výstup jednoho neuronu $n_{l, j}$ je pak definován jako

$$O_{l,j} = f_{act} \left(\sum_{i=0}^{N_{l-1}} O_{l-1,i} \cdot w_{l,i,j} \right), l = 1, 2...L - 1, j = 1, 2...N_l,$$

$$O_{0,j} = I_j, j = 1, 2...N_0,$$

$$O_{l,0} = 1, l = 1, 2...L - 1,$$
(1)

kde f_{act} je aktivační funkce, která má v naší implementaci podobu

$$f_{act}(\Sigma) = \frac{1}{1 + e^{-\alpha/\Sigma}}.$$
(2)

Parametr α ve vzorci (2) odpovídá strmosti aktivační funkce v okolí počátku a je nastavený v našich výpočtech na hodnotu 0.8. Nakonec je výstup z každé hladiny označen jako $O_i = (O_{i,1}, O_{i,2,...}, O_{i,Ni})$. Jeden průchod neuronovou sítí lze tedy zapsat následovně:

Nechť l = 1.

Vypočti $O_{l, i}$ pro $i = 1, 2...N_{l}$. l = l + 1.

Pokud $l \le L$ pak jdi do bodu 2., jinak O_{L-1} je aproximace vstupu T.

Chyba měřená na výstupu, která slouží jako měřítko úspěšnosti odhadu, může být definována například takto:

$$\varepsilon = \frac{1}{2} \sum_{i=1}^{N_{L-1}} (T_i - O_{L-1,i})^2.$$
 (3)

Výstup neuronové sítě je určen předchozím trénováním, které spočívá v nalezení příslušných vah $w_{l,i,j}$. Z našeho pohledu se jedná o optimalizační problém, kdy se snažíme

změnou těchto vah minimalizovat chybu na výstupu. V takovémto případě tedy váhy slouží jako proměnné optimalizační úlohy. Jak bylo ukázáno např. v [6], genetické algoritmy, v tomto konkrétním případě algoritmus SADE, mohou být efektivně využity na úkor tradičních metod.

Algoritmus SADE

Genetické algoritmy patří v současné době mezi jedny z nejoblíbenějších optimalizačních metod. Jsou založeny na principu analogie s procesy, které formovaly život na této planetě v průběhu miliónů let. Největší rozdíl od tradičních optimalizačních metod lze spatřovat v tzv. populaci jedinců, což není nic jiného než množina možných řešení daného problému. Na tato řešení jsou pak aplikovány operátory křížení, mutace a výběru ve snaze docílit postupného zlepšení všech prvků v populaci. Takto formulovaný algoritmus poprvé navrhl J. Holland [9] a od té doby se genetické algoritmy rozšířily do mnoha oblastí (doporučujeme knihy D. Goldberga [10] a Z. Michalewicze [11] pro další reference a podrobné studium).

Genetické algoritmy ve své původní formulaci pracovaly s populací kódovanou do binárního řetězce (analogie s dědičnou informací uloženou v DNA). Tato formulace ale není příliš vhodná pro inženýrské úlohy, kde se velice často setkáváme s reálnými čísly. Proto byly objeveny reálně kódované genetické algoritmy, které pro reálná čísla musí mít jiné operátory křížení a mutace. Základy reálných genetických algoritmů jsou popsány např. v [11]; v této práci navazujeme na reálně kódovaný algoritmus "Diferenciální Evoluce", který byl představen R. Stornem v [12].

V této práci představujeme modifikovaný genetický algoritmus SADE (Simplified Atavistic Differential Evolution). V našich předchozích pracích [3,13] bylo ukázáno, že je tento algoritmus schopný vyřešit i rozsáhlé úlohy se stovkami neznámých a je schopen si poradit i s několika lokálními extrémy. Nyní ve stručnosti představíme tento algoritmus, zvídavého čtenáře odkazujeme na práci [3], kde je algoritmus rozepsán do podrobností.

V tradičních genetických algoritmech se nejčastěji v prvním kroku náhodně vytvoří počáteční populace. Algoritmus pak opakuje procesy křížení, mutace a výběru, dokud není splněno některé ze zastavovacích kritérií, kterými nejčastěji bývá maximální počet vyhodnocení optimalizované funkce. V naše případě pracujeme s populací 10*n* chromozómů, kde *n* značí celkový počet neznámých objektivní funkce. Proces optimalizace je pak řízen těmito operacemi:

Mutace: Nechť je $\mathbf{x}_i(g)$ *i*-tý chromozóm v generaci *g*. Pokud je tento chromozóm vybrán k operaci mutace, je nejprve náhodně vytvořen jiný chromozóm *RP* a nové řešení je pak dáno vztahem:

$$\mathbf{x}_{k}(\mathbf{g}+1) = \mathbf{x}_{i}(\mathbf{g}) + MR(\boldsymbol{RP} \cdot \mathbf{x}_{i}(\mathbf{g})) .$$
(4)

Parametr *MR* je konstantní a v našich příkladech má hodnotu 0.5. Počet nových řešení vytvořených tímto operátorem je definován pomocí parametru "radioaktivita", který je opět konstantní a v našich příkladech je nastaven na hodnotu 0.1, tj. 10% z nové populace je vytvořeno právě tímto operátorem.

Lokální mutace: Záměrem tohoto operátoru je lokální prohledávání okolí již nalezených optim. Jeho podoba spočívá v drobných změnách všech proměnných jednoho zvoleného řešení.

Křížení: je velice důležitý operátor genetických algoritmů. Jeho cílem je kombinovat "dobré" vlastnosti existujících řešení k dosažení lepších hodnot. Tento operátor vytváří nový chromozóm $\mathbf{x}_i(g + 1)$ tak, že vybere náhodně dvě existující řešení $\mathbf{x}_q(g)$ a $\mathbf{x}_r(g)$, vypočte jejich rozdíl, vynásobí ho koeficientem *CR* a výsledek přičte k třetímu, taktéž náhodně vybranému řešení $\mathbf{x}_p(g)$, neboli

$$\mathbf{x}_{i}(g+1) = \mathbf{x}_{p}(g) + CR(\mathbf{x}_{q}(g)-\mathbf{x}_{r}(g)).$$
(5)

Každá z proměnných, která tímto překročí vymezené hranice je nastavena na hraniční hodnotu. Parametr *CR* má pravděpodobně největší dopad na chování celého algoritmu: je odpozorováno, že pro vyšší rychlost konvergence jsou vhodnější nižší hodnoty (mezi 0.05 a 0.1), zatímco vyšší hodnoty (v našem případě 0.3) zvyšují pravděpodobnost nalezení globálního optima.

Výběr: představuje jádro genetického algoritmu. Cílem je zredukovat velikost populace se zachováním "lepších" řešení. To lze zařídit výběrem nejlepších jedinců, což ale ve většině případů vede na předčasnou konvergenci (z angl. *premature convergence*) neboli dojde ke ztrátě diverzity populace. V našem případě je použit modifikovaný turnajový výběr, kdy jsou z populace vybráni dva jedinci (turnaj) a horší z nich je z populace vyřazen. Tento přístup má dvě velké výhody – jednak zajišťuje přežití nejlepšího jedince do další generace a jednak umožňuje "přežití" i nejhoršího řešení, pokud nebude vybráno do souboje.



Obr. 3 Pracovní diagramy vytvořené ze 30 simulací metodou LHS

Metoda Latin Hypercube Sampling (LHS) a stochastická senzitivní analýza

Ve snaze zlepšit předchozí výsledky získané pomocí neuronových sítí [6] byla implementována stratifikační metoda Latin Hypercube Sampling pro vytváření vstupních trénovacích dat. Metoda LHS je totiž schopna již při malém počtu simulací zajistit reprezentativnost těchto vzorků a díky tomu umožnit neuronové síti lépe aproximovat daný problém. Tato metoda je ještě vylepšena metodou simulovaného žíhání pro zaručení nezávislosti jednotlivých vstupů. Konkrétně je použit algoritmus programu FREET [14].



Obr. 4 Porovnání deseti pracovních diagramů získaných ze simulací na jednom elementu modelu a ze simulací na celém betonovém válci

Využití metody LHS, a tím snížení počtu nutných simulací pro trénování neuronové sítě, umožnilo přechod od simulace jednoosého tlaku pouze na jenom elementu modelu microplane k simulaci této zkoušky pro celý betonový válec. Výstupem z této metody pak jsou pracovní diagramy pro takto vytvořená data, jak je vidět na Obr. 3. Jistý rozdíl mezi pracovními diagramy získanými ze simulace na jednom elementu a na celém válci, který může být významný při idenfikaci některých parametrů modelu, je možné vidět při porovnání grafů na Obr. 4.



Obr. 5 Senzitivní analýza vlivu materiálových parametrů na průběh pracovního diagramu pro osovou deformaci

Na 30ti simulacích získaných metodou LHS byla provedena sensitivní analýza pro určení vlivu jednotlivých materiálových parametrů na nelineární odezvu. Konkrétně byl použit Pearsonův korelační koeficient daný rovnicí

$$cor = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}},$$
(6)

jehož hodnota byla vyčíslena ve dvaceti po sobě jdoucích bodech na pracovních diagramech pro osovou deformaci, viz Obr. 5, a ve stu po sobě jdoucích bodech na pracovních diagramech pro boční deformaci, viz Obr. 6. Z výsledků je vidět vliv jednotlivých materiálových parametrů na nelineární odezvu materiálu, kde např. vliv modulu pružnosti *E* zejména v elastické větvi je jednoznačný. Také významný vliv parametru k_1 je patrný z obou uvedených grafů, zatímco Poissonovo číslo ovlivňuje významně pouze průběh boční deformace a to v počátku zatěžování. Jistý vliv má pak ještě parametr c_{20} a to na sestupnou větev pracovního diagramu pro osovou deformaci.



Obr. 6 Senzitivní analýza vlivu materiálových parametrů na průběh pracovního diagramu pro boční deformaci

Výsledky identifikace

V tomto bodě byla již dříve představená neuronová síť natrénována na množině třiceti nezávislých vstupů získaných pomocí metody LHS. Kvalita natrénování je pak zkoušena na deseti jiných náhodně zvolených vzorcích. Jako trénovací algoritmus byl použit algoritmus SADE. Pro každý identifikovaný parametr byla trénována jiná síť s jediným neuronem ve výstupní vrstvě. To umožnilo výbrat vhodná vstupní data pro neuronovou síť s ohledem na výsledky sensitivní analýzy pro jednotlivé parametry.

parameter	absolut	ní chyba	relativní chyba [%]	
	průměrná	maximáln í	průměrná	maximáln í
E [GPa]	312,4	627,7	1,00	1,93
k_{I}	2,588e-06	6,654e-06	1,75	4,49
C20	0,7989	1,8909	18,65	50,99

Tab. 2: Chyby v odhadech Youngova modulu pružnosti a parametru k_i .

V první fázi výpočtů je využito jako vstupů pouze bodů z pracovních diagramů pro osovou deformaci. Pro identifikaci parametru k_l se také jako vstupní hodnoty uplatnily hodnoty napětí a deformace, které v těchto diagramech odpovídají "peakům", tzn. bodům s největší hodnotou napětí. Hodnota korelačního koeficientu mezi parametrem k_l a hodnotu napětí je 0,9751 a pro hodnotu deformace je 0,6918. Výstupem byly natrénované neuronové sítě, přičemž jedna

z nich je schopna spolehlivě identifikovat hodnotu Youngova modulu pružnosti a druhá hodnotu parametru k_1 . Jen dosti přibližné odhady byly získány také pro parametr c_{20} . Chyby v odhadech těchto tří parametrů jsou uvedeny v Tab. 2. Ostatní sítě se uspokojivě natrénovat nepodařilo.

Cílem dalšího postupu bylo zpřesnění odhadů hodnot parametru c_{20} . Jelikož není jisté, že jednotlivé parametry modelu microplane se vzájemně ve svých účincích neovlivňují, je možné použít známé hodnoty některých parametrů pro přesnější identifikaci parametrů dalších. Proto bylo v této fázi jako vstupů pro neuronové sítě použito i odhadnutých hodnot Youngova modulu pružnosti a parametru k_1 z předchozích výpočtů. Tento postup vedl k jistému zpřesnění odhadů parametru c_{20} , identifikaci ostatních parametrů však neovlivnil.



Obr. 7 Naměřený pracovní diagram a odpovídající simulace modelem microplane pro dva různé odhady parametru c_{20}

Takto natrénované sítě, které identifikují hodnoty parametrů pouze z pracovního diagramu pro osobou deformaci, byly použity pro odhad parametrů modelu ze skutečných naměřených dat. Pracovní diagram z naměřených dat a odpovídající simulace pro dva různé odhady parametrů c_{20} je možné porovnat na Obr. 7.

parameter	absolut	ní chyba	relativní chyba [%]				
	průměrná	maximáln í	průměrná	maximáln í			
v	0,004976	0,01125	2,22	4,32			
C_{20}	0,4119	0,7706	9,87	25,03			

Tab. 3: Chyby v odhadech Poissonova čísla a parametru c_{20} .

V poslední fázi výpočtů bylo využito pracovních diagramů pro boční deformaci jako vstupů pro neuronovou síť a také již získané odhady Youngova modulu pružnosti a parametru k_1 . Výstupem byly natrénovaná síť schopna uspokojivě identifikovat hodnotu Poissonova čísla a tento dobrý odhad byl poté opět znovu použit pro další zpřesnění identifikace parametru c_{20} . Výsledné chyby v odhadech těchto dvou parametrů jsou uvedeny v Tab. 3. Ostatní sítě se stále uspokojivě natrénovat nepodařilo.

Vzory pracovních diagramů pro deset testovacích příkladů a jejich obrazy získané z parametrů odhadnutých natrénovanými sítěmi jsou zobrazeny na Obr. 8 a 9.



Obr.8 Porovnání prvních pěti pracovních diagramů pro původní a odhadnuté parametry



Obr. 9 Porovnání druhých pěti pracovních diagramů pro původní a odhadnuté parametry

3. Závěr

V tomto příspěvku byla jedním z algoritmů umělých inteligencí řešena jedna z inženýrských úloh, která je velmi těžko řešitelná pomocí tradičních metod. Konkrétně byla použita umělá neuronová síť k odhadu materiálových parametrů modelu microplane pro beton z pracovních diagramů získaných při zatěžovací zkoušce betonových válců v jednoosém tlaku. Jelikož se lze dívat na učení neuronové sítě jako na optimalizační problém, byla pro tento problém aplikována metoda SADE z oblasti genetických algoritmů. Množství potřebných simulací bylo zredukováno pomocí metody LHS rozšířené o simulované žíhání, což umožnilo použití simulací zatěžovací zkoušky na celých betonových válcích. Neuronová síť natrénovaná

na datech z těchto simulací pak byla použita k identifikaci parametrů modelu z reálných naměřených dat. Stochastická senzitivní analýza byla použita nejen k získání vlivu jednotlivých materiálových parametrů na nelineární odezvu, ale též posloužila jako účinný nástroj při volbě vhodných vstupních hodnot pro neuronovou síť. Protože získané odhady nejsou identické, stále je zde možnost je vylepšit delším trénovacím procesem, změnou topologie neuronové sítě nebo zvětšením trénovací množiny. Zejména je ale třeba v dalších krocích přistoupit k simulaci jiných typů zatěžovacích zkoušek, ze kterých bude možné identifikovat parametry k_2 , k_3 , k_4 a c_3 . Největší výhodou nastíněné metodologie je především fakt, že jednou naučená neuronová síť může být kdykoliv použita znovu bez nutnosti náročných numerických výpočtů.

4. Poděkování

Autoři by rádi poděkovali za finanční podporu grantem Ministerstva školství a tělovýchovy České Republiky, číslo MSM 210000003.

5. Literatura

[1] Z.P. BAŽANT, F.C. CANER, I. CAROL, M.D. ADLEY, S.A. AKERS: Microplane model M4 for concrete. Part I: Formulation with work-conjugate deviatoric stress, Part II: Algorithm and calibration, Journal of Engineering Mechanics-ASCE, 126, (2000), 944-953, 954-961.

[2] B. PICHLER, R. LACKNER, H.A. MANG: Back analysis of model parameters in geotechnical engineering by means of soft computing, Int. J. for Num. Methods in Eng., 57, (2003), 1943-1978.

[3] O. HRSTKA, A. KUČEROVÁ: Improvements of the different types of binary and real coded genetic algorithms preventing the premature convergence, Advances in Engineering Software, (2004), 35(3-4):237-246.

[4] M.E. JIRÁSEK, Z.P. BAŽANT: Inelastic analysis of structures, John Wiley & Sons, 2001.

[5] J. NĚMEČEK: Modelling of compressive softening of concrete, CTU Reports, 4(6), 2000.

[6] J. DRCHAL, A. KUČEROVÁ, J. NĚMEČEK: Optimizing synaptic weights of neural networks, in Proceedings of the Third International Conference on Engineering Computational Technology, Civil-Comp Press, 2002.

[7] A. KUČEROVÁ, M. LEPŠ, J. NĚMEČEK: Estimation of microplane model parameters using parallel genetic algorithm, in Proceedings of The Seventh International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering, Civil-Comp Press, 2003.

[8] G. YAGAWA AND H. OKUDA: Neural networks in computational mechanics, CIMNE, 1996.

[9] J. H. HOLLAND: Adaptation in natural and artificial systems, MI, Internal report, 1975.

[10] D.E. GOLDBERG: Genetic algorithms in search, optimization and machine learning, Addison-Wesley, 1989.

[11] Z. MICHALEWICZ: Genetic Algorithms+Data Structures=Evolution Programs, Springer, 1992.

[12] R. STORN: On the usage of differential evolution for function optimization, in Biennial Conference of the North American Fuzzy Information Processing Society, pages 519-523, 1996.

[13] O. HRSTKA, A. KUČEROVÁ, M. LEPŠ, J. ZEMAN: A competitive comparison of different types of evolutionary algorithms, Computers and Structures, 81/18-19:1979-1990, 2003.

[14] DRAHOMÍR NOVÁK ET AL: FREET Feasible Reliability Engineering Efficient Tool, Program documentation, Brno University of Technology, FCE / Červenka Consulting, Praha, 2002.