

INVERSE ANALYSIS USING SOFT-COMPUTING METHODS: A REVIEW

A. Kučerová*, M. Lepš*, J. Zeman*

Summary: *The problem of an inverse analysis occurs in many engineering tasks. The problem itself has several different forms and can be solved by many very distinct methods. In this paper, we present an overview of two basic philosophies of an inverse analysis with an emphasis put on the area of soft-computing methods. Practical aspects will be shown on several identification tasks, where parameters of highly non-linear material models are searched.*

1. Úvod

Inverzní analýza se při řešení inženýrských úloh vyskytuje v různých podobách a bývá taktéž nazývána různými termíny. Obecně lze konstatovat, že cílem úlohy je na základě známých výstupů určit zpětně (inverzně) příslušné vstupy. V inženýrských aplikacích pak na základě výsledků experimentů určit původní podmínky a veličiny, v moderní době pak parametry modelu, kterým se snažíme provedený experiment simulovat. V této práci tedy předpokládáme existenci modelu nějakého reálného experimentu a budeme se snažit nalézt vstupní parametry tohoto modelu tak, aby výstupy z modelu souhlasily s provedeným experimentem. Pro takto definovaný problém existují dva základní způsoby řešení. **Dopředný postup** (*forward mode*) je založen na definici chybové funkce rozdílu odezvy modelu a výstupu z experimentu. Úlohou je pak minimalizace této funkce. V tomto bodě se nejčastěji uplatňují metody evolučních optimalizací. Jelikož je většina současných modelů založena na náročných výpočtech pomocí metody konečných prvků, najdou zde uplatnění i algoritmy vhodné na aproximaci těchto modelů. Druhý postup, **zpětný** (*inverse mode*), předpokládá, že existuje inverzní vztah mezi výstupy a vstupy. Tento vztah je nejčastěji modelován umělými neuronovými sítěmi.

V dalších částech příspěvku bude ukázáno, že dopředný postup identifikace vede na multimodální optimalizační problém. Tento problém je většinou řešen vhodným rozšířením nějakého evolučního algoritmu. Pro zpětný postup se nejčastěji používá umělá neuronová síť trénovaná evolučním algoritmem. V příspěvku jsou rozebrány detaily jednotlivých postupů spolu s použitými metodami z oblasti umělých inteligencí. Použití v inženýrské praxi je pak ukázáno na příkladu identifikace nelineárních materiálových modelů jako jsou horniny, zeminy a beton.

* Ing. Anna Kučerová, Ing. Matěj Lepš, Ph.D., Ing. Jan Zeman, Ph.D.: Katedra 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: leps@cml.fsv.cvut.cz

2. Dopředný postup inverzní analýzy

Jak již bylo naznačeno v úvodu této práce, problém inverzní analýzy lze formulovat na základě existence experimentu E , který, ať už fyzicky či virtuálně, poskytuje ze známých parametrů X^E požadované výstupy (měření) Y^E . Formálně lze tento vztah zapsat jako

$$Y^E = E(X^E). \quad (1)$$

Inverzní analýza je pak definována jako hledání neznámých X^E ze známých výstupů Y^E , tedy zpětně (inverzně) ve vztahu k experimentu E . V inženýrských úlohách je pak většinou experiment E nahrazen nějakým virtuálním modelem M , nejčastěji programem obsahujícím některou z numerických metod, obvykle metodu konečných prvků. V tomto okamžiku tato práce předpokládá, že je k dispozici dostatečně přesný model M experimentu E , neboli, že můžeme klást

$$E \equiv M, \quad (2)$$

z čehož automaticky vyplývá

$$Y^E = M(X^E). \quad (3)$$

Tento krok je důležitý zejména z hlediska ceny, kdy se předpokládá, že finanční náklady na vyhodnocení modelu M jsou řádově menší či přímo zanedbatelné k finanční náročnosti fyzického experimentu E .

Za výše uvedených podmínek je **dopředný postup** (*forward, classical mode/direction*) inverzní analýzy definován na základě definice chybové funkce rozdílu výstupů z modelu a experimentu, nebo-li

$$\min f(X) = |Y^E - M(X)|. \quad (4)$$

Za předpokladu, že je tato funkce vzhledem k metrice nezáporná, je řešením problému jedno či více řešení X^* , pro které platí jak $f(X) = 0$, tak i $X^* \equiv X^E$, což bylo cílem úlohy.

Tradičně se k řešení problému (4) používaly gradientní optimalizační metody. V současné době je ale čím dál častěji model M uzavřen v konečně-prvkovém programu omezeném licenčními podmínkami, uzavřeným kódem apod. a tudíž znalost prvních a vyšších derivací funkce $f(X)$ je nedostupná nebo samotná funkce $f(X)$ není spojitě diferencovatelná. Z tohoto důvodu se v této oblasti začínají používat optimalizační metody z oblasti umělých inteligencí, ať už jsou to obecně stochastické metody využívající pouze jedno řešení k nalezení optima jako je metoda *simulovaného žíhání* [1,2] či metody pracující s více řešeními nezávisle jako jsou *evoluční algoritmy* [3,4].

Výhodou takto definované úlohy oproti **zpětnému postupu**, popsanému dále, je, že dopředný postup je obecný ve všech rovinách a vždy je schopný poskytnout přesné řešení, pokud takové existuje. Speciálními příklady dokládajícími obecnost dopředného postupu je například stav kdy:

- Pro více X existuje jen jedno Y . Tento stav při dopředném postupu vede na multimodální optimalizaci [5]. Problém je pak snadno řešitelný zavedením optimalizační metody schopné vyhledat více optim.
- Pro jedno X existuje více Y , nejčastěji jde o případy stochastických výpočtů či experimentů zatížených šumem či jinými chybami. Tento problém lze řešit buď úpravou chybové funkce (4) pro více hodnot Y nebo zavedením stochastických parametrů jak na vstupu, tak i na výstupu.

- c) Existuje více než jeden experiment E . Např. v případě dvou experimentů $Y^{E1} = E1(X^E)$ a $Y^{E2} = E2(X^E)$, lze chybovou funkci (4) prostě sčítat přes Y^{E1} a Y^{E2} , nebo lze celý problém řešit jako vícekritériální optimalizační úlohu [6,7].

2.1 Řešení nevýhod dopředného postupu

Největší nevýhodou dopředného postupu je nutnost provést obvykle několik tisíců až miliónů výpočtů funkce (4), která, jak již bylo řečeno, většinou zahrnuje náročný výpočet nějakým programem. Tento problém je řešitelný dvěma postupy: první využívá vzájemné nezávislosti jakýchkoliv dvou vektorů z X . Tento jev je často nazýván *implicitní paralelizmus*. Díky tomu je velice jednoduché naprogramovat tuto úlohu jako globální paralelní model [8], kde hlavní procesor (tzv. *master*, nebo *root*) řídí optimalizaci, ale vlastní vyhodnocení modelu M je počítáno na jiných (tzv. *slave*) procesorech nebo počítačích. Díky vzájemné nezávislosti je pak dosahováno lineárního zrychlení výpočtu (tzv. *linear speed-up*) až do velkého počtu použitých procesorů, tedy s minimálními komunikačními ztrátami.

Druhý postup řešení je založen na podobné myšlence, jakou jsme již uplatnili v rovnici (2). Předpokládá existenci nějakého jiného modelu $\underline{M}$, který v tomto případě není totožný s modelem M (pouze se mu co nejvíce přibližuje), ale hlavní podmínkou je, aby vyhodnocení náhradního modelu $\underline{M}$ trvalo co nejkratší dobu. Zde tedy neplatí identita (2), neboli platí

$$\underline{M} \neq M, \quad (5)$$

a tudíž se optimální řešení X^* získané optimalizací modelu $\underline{M}$ musí vždy zkontrolovat použitím rovnice (3), zda odpovídá původnímu požadavku (1). Nastává otázka, jak sestavit vhodný model $\underline{M}$. Z mnoha možných řešení se zaměříme na dva, v poslední době nejčastější, přístupy. První přístup, nazvěme ho **interpolační**, se snaží interpolovat model M , bez jakékoliv znalosti jeho vnitřní struktury. K řešení se používají tradiční algoritmy matematické interpolace, nejčastěji pak metody plochy odezvy (*response surface methods*, dále jen RSM) [9], z oblasti umělých inteligencí pak interpolační umělé neuronové sítě, jako jsou sítě s radiální bází (*radial basis function network*, dále jen RBFN) [10,11]. Požadavek na interpolaci je v tomto případě zásadní, neboť takto vytvořený model $\underline{M}$, kromě již spočtených bodů $\underline{Y} \equiv Y$, neodpovídá modelu M a tudíž je mimo známé body pokládán za nespolehlivý. Získání dobré interpolace modelu M je pak většinou založeno na iteračním postupu, kdy je sestaven model $\underline{M}$ na základě již známých bodů/řešení (dvojic X_k, Y_k), multi-modální optimalizací jsou nalezena minima $\underline{Y}^* = \underline{M}(X^*)$ modelu $\underline{M}$ (tento proces by tedy měl být s ohledem na časovou nenáročnost modelu $\underline{M}$ časově nenáročný), jsou spočteny odpovídající přesné hodnoty modelu M , neboli $Y_{k+1} = M(X^*)$, a ty jsou přidány k původní množině již známých bodů. Iterace pokračuje tak dlouho, dokud není dosaženo minima funkce (4). Dolní index k je zde použit jako čítač iterací. Cílem celého postupu je pak minimalizovat počet vyhodnocení funkce (4). Jednotlivé postupy různých autorů, viz též [11,12] nebo [13], se pak liší právě v tom, jak tohoto cíle dosáhnout.

Druhý přístup, zde nazvaný **aproximační**, předpokládá, že k fyzikálnímu modelu M existuje nějaký nepřesný fyzikální model $\underline{M}$, který je časově méně náročný než model M . Tato situace je přítomná v případech, kdy existují pro jeden fyzikální jev dvě různé vysvětlující teorie, např. vlnová vs. částicová. Daleko častějším jevem pak bude situace, kdy lze studovaný jev popsat geometrií o rozdílném počtu konečných elementů, rozdílným stupněm aproximace hledaných veličin, jednoduchým či složitým materiálovým modelem, dvourozměrným vs. prostorovým modelem apod. Problémem, před kterým stojí tento přístup

je fakt, že nejspíše pro většinu řešení bude platit nerovnost (5). Přičemž není zcela jednoznačné, jakým způsobem tento rozdíl zanést zpět do zjednodušeného modelu (narozdíl od interpolačního přístupu). Aproximační přístup je „horkou novinkou“ v oblasti globální optimalizace a tudíž není známo mnoho poznatků ohledně nastíněného problému (viz několik příspěvků [14,15]). Jako slibný se jeví vícekritériální přístup [16], kdy je zvlášť optimalizována funkce $f1(X) = |Y^E - M(X)|$ a funkce $f2(X) = |Y^E - \underline{M}(X)|$. Cílem je pak pouze hledání minima na $f1$. Za účelem minimalizace času je snaha využívat více vyhodnocení funkce $f2$, ale je otázka, odkud brát odpovídající nespočtené hodnoty $f1$. V příspěvku [16] je celý problém řešen vícekritériálním evolučním algoritmem, kdy jsou neznámé hodnoty $f1$ brány od předchozích, tzv. rodičovských, řešení.

Druhou nevýhodou dopředného postupu, která vyplývá přímo z jeho definice, je skutečnost, že pro opakování inverzní analýzy (např. pro novou sadu experimentálních dat) je nutné provést náročný optimalizační cyklus pokaždé znova (neplatí pouze, pokud bychom měli dostatečně „rychlou“ aproximaci modelu – viz. odstavec výše). Tímto je tento postup handicapován pro časté automatické použití. Opak je pak pravdou pro druhý postup inverzní analýzy.

2.2 Příklady použití dopředného postupu

Aplikace použití dopředného postupu byla vyzkoušena na pracovišti řešitelů již v minulých letech na jednoduchých, avšak značně nelineárních modelech materiálů a zemin. Použití dopředného postupu je zde velice vhodné, neboť vyhodnocení byť i tisíců vyhodnocení chybové funkce je stále maximálně v řádu minut a výhody přesného řešení dopředného postupu tento postup upřednostňují. Jako optimalizační algoritmus byl použit evoluční algoritmus SADE, vyvíjený na pracovišti autorů, viz např. [17] nebo [18].

Prvními dvěma příklady byly Lemaitrovy materiálové modely pro dotvarování hornin [19]. Konkrétně se jednalo o modely pro dotvarování hornin bez poškození a s poškozením horniny, které byly charakterizovány třemi, respektive šesti, nezávislými parametry. Výsledkem byly optimální parametry těchto materiálových modelů, které v tomto případě výrazně předčily výsledky z lineární regrese, viz např. [20].

Dalším studovaným modelem byl problém retenční čáry zemin, který vyjadřuje funkční závislosti sacího tlaku zeminy na stupni jejího nasycení. Konkrétně se jednalo o model podle van Genuchtena, viz např. [21], který byl charakterizován třemi materiálovými parametry. I zde byl dopředný postup schopen najít velice rychle optimální parametry, viz např. [22].

2.2.1 Identifikace parametrů mikroploškového modelu

Dalším studovaným modelem byl mikroploškový (*microplane*) model betonu. Mikroploškový model [23,24] 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 velké množství fenomenologických materiálových parametrů. Konkrétně je určitý typ betonu popsán osmi parametry: Youngovým modulem pružnosti E , Poissonovým součinitelem ν 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. Největší nevýhodou tohoto modelu je jeho

výpočetní náročnost. Například numerická analýza příkladu jednoosého tlaku trvala déle než 23 hodin na počítači osazeném procesorem Pentium II Xeon 400 MHz s 512 MB RAM, viz též [25].

Proto byla v práci [26] k dopřednému postupu využita paralelní verze evolučního algoritmu SADE 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. Tento problém je v současné době plánován k řešení interpolačním přístupem.

3. Zpětný postup inverzní analýzy

Tento postup vychází ze stejných předpokladů, které byly uvedeny na začátku druhé kapitoly, neboli uvažuje platnost rovnic (1)-(3). **Zpětný postup** (*inverse, backward mode*) inverzní analýzy je pak definován za předpokladu, že existuje inverzní postup/vztah k použitému modelu M , neboli, že existuje matematický algoritmus M^{INV} , který je schopen splnit následující rovnici:

$$X = M^{INV}(Y), \quad (6)$$

pro libovolné Y a jemu příslušné X . Obecně tento „inverzní“ model nemusí existovat, v této práci budeme předpokládat, že lze nalézt M^{INV} tak přesné, že bude vyhovovat pro zrovna studovanou oblast. Jinak řečeno, rovnice (6) bude splněna pouze na určité (třeba i nespojitě či diskrétní) oblasti. Dalším ústupkem bude, že nebudeme striktně dbát na vyjádření zpětného vztahu, ale připustíme myšlenku, že je předpis M^{INV} natolik složitý, že se spokojíme s pouhou aproximací tohoto modelu. Kvalitu této aproximace je pak velice snadné zkontrolovat, neboť dvojice X, Y získaná z rovnice (6) musí opět splňovat původní rovnici (3). Konečné použití tohoto postupu je již triviální, neboť požadovanou hodnotu X^E získáme pouhým dosazením Y^E do rovnice (6).

Z předchozího odstavce jsou jasně patrné největší výhody zpětného postupu. Za předpokladu, že jsme našli dostatečně přesný model M^{INV} , je pak získání požadovaných vstupů X^E otázkou sekund (závisí na složitosti popisu M^{INV}) a to i opakovaně. Tudíž se tento postup hodí pro časté použití s jedním modelem M a při nasazení do automatické inverzní analýzy pro jeden konkrétní model.

Naopak, největší nevýhodou, kromě dosti silného předpokladu existence takového inverzního vztahu, je značně náročný proces získávání tohoto inverzního vztahu. V dnešní době se k tomuto popisu nejčastěji používají vrstevnaté umělé neuronové sítě (*layered artificial neural net*, dále jen neuronová síť, NN) [27,28,29,30], které jsou dostatečně robustní, aby byly schopné popsat jakýkoliv nelineární vztah.

3.1 Použití umělé neuronové sítě

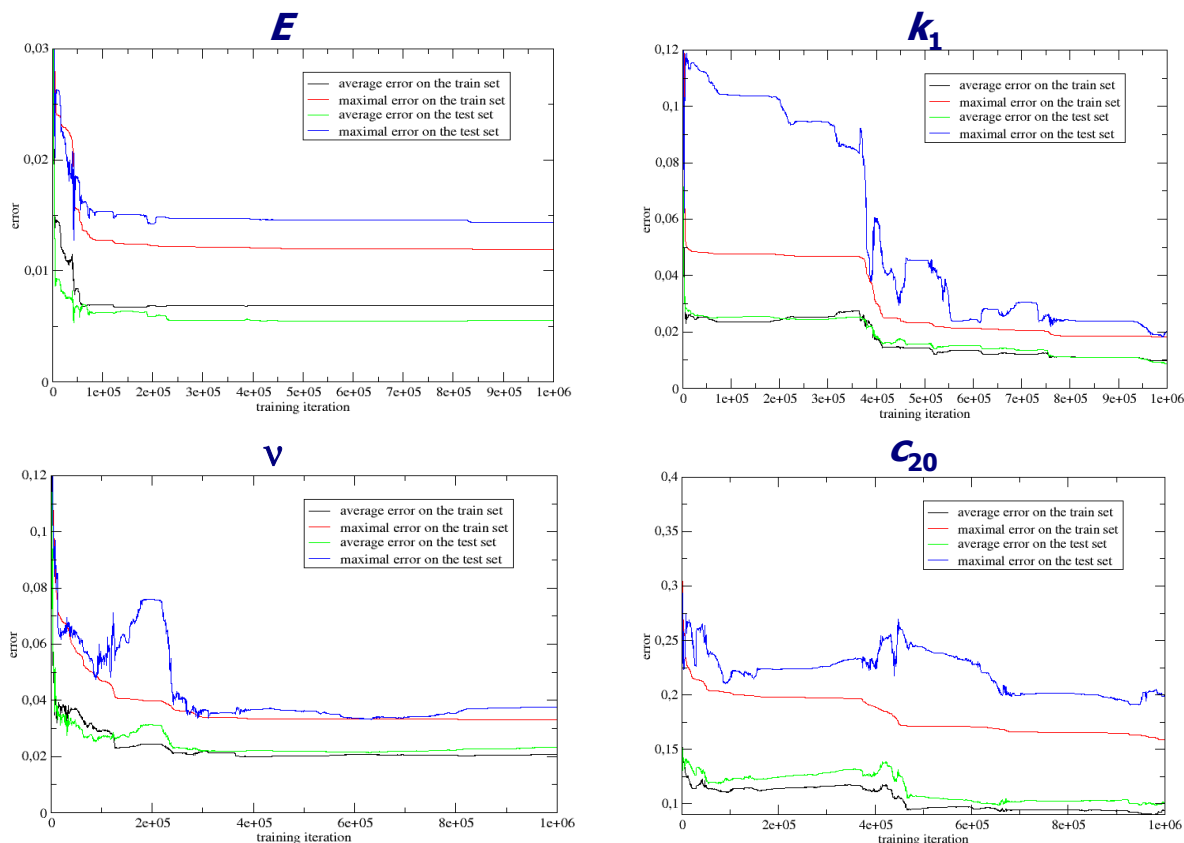
Nebudeme se zde zabývat podrobným popisem fungování vrstevnaté umělé neuronové sítě (NN), ten lze nalézt např. v [31] nebo v [32]. Nicméně je nutno poznamenat, že NN je nelineární algoritmus, který na zadané vstupy I dává výstupy O , v našem případě by tedy bylo požadováno $I \equiv Y^E$ a $O \equiv X^E$. Proces tvorby nelineárního modelu NN spočívá nejprve v nastavení počtu vnitřních nelineárních vazeb, zde dáno počtem tzv. vrstev a počtem tzv.

neuronů v každé vrstvě. Pro toto nastavení neexistuje jasný předpis, nicméně jsou dány odhady, v jakých mezích by se konkrétní čísla měla pohybovat, viz např. [28]. Druhým krokem je pak nalezení tzv. synoptických vah. Tento krok se v terminologii neuronových sítí nazývá „učení“, konkrétně pak s „učitelem“. Předpokládá, že máme k dispozici množinu vzorů a obrazů, zde tzv. **trénovací** množinu, neboli sadu I^{TR} a jim odpovídající výstupy O^{TR} , které chceme, aby neuronová síť uměla aproximovat. Proces učení pak spočívá v minimalizaci chyby mezi získaným výstupem O a požadovaným výstupem O^{TR} . Takto formulovaná úloha je v podstatě optimalizační problém (zkušenosti říkají, že značně multimodální), který je nejčastěji řešen metodou zpětné propagace chyby (*back-propagation method*, dále jen BP), která patří do oblasti metod největšího spádu. V současné době se místo metody BP používají například některé evoluční algoritmy schopné řešit nastíněnou multimodalitu, viz např. [31].

S tématem učení se svazují dva základní problémy – „podučení“ (*under-fitting* nebo *under-learning*) a „přeučení“ (*over-learning* nebo *over-fitting*). Oba dva pojmy se vztahují k tomu, jak je schopna neuronová síť aproximovat danou sadu I^{TR} , O^{TR} . Tento stav se kontroluje za pomoci tzv. **testovací sady**, o které se předpokládá, že je ze stejné oblasti jako trénovací sada, ale je maximálně rozdílná od trénovací sady. Stav „podučení“ pak nastává za situace, že má daná neuronová síť řádově méně vnitřních stupňů volnosti (v tomto případě počet neuronů a tudíž i vah) než je potřeba na dostatečnou aproximaci. Tento jev je pak snadno dokumentován velikou chybou v průběhu učení jak na trénovací, tak i na testovací množině. Naopak „přeučení“ je způsobeno větším počtem stupňů volnosti sítě, než kolik je jich potřeba na dobrou aproximaci. V tomto případě je minimální chyba v průběhu učení na trénovací množině, ale neklesá nebo naopak roste chyba na testovací množině. Obecně je doporučeno, aby byla neuronová síť „lehce“ podučena, aby si zachovávala své dobré aproximační vlastnosti. Typické „optimální“ průběhy chyb v závislosti na průběhu učení lze vidět na Obr. 1, kde postupně klesá chyba jak na trénovací (známé) tak i na testovací (pro neuronovou síť neznámé) množině.

3.2 Příklad použití při identifikace parametrů mikroploškového modelu

V tomto konkrétním případě byla použita umělá neuronová síť k odhadu materiálových parametrů mikroploškového modelu pro betonu, který byl již představen v kapitole 2.2.1. Jelikož se lze dívat na učení neuronové sítě jako na optimalizační problém, byla pro tento problém aplikována již výše zmíněná metoda SADE. Množství potřebných trénovacích a testovacích bodů bylo zredukováno pomocí metody LHS rozšířené o simulované žíhání, viz např. [33,34]. Stochastická senzitivní analýza byla použita nejen k získání vlivu jednotlivých materiálových parametrů na nelineární odezvu, ale též posloužila jako odhad vyprodukovaných chyb. Typický průběh chyb na trénovací a testovací množině je pak zaznamenán na Obr. 1. V současné době jsou uspokojivě vyřešeny identifikace většiny parametrů mikroploškového modelu, kromě parametrů k_2 , k_3 , a c_{20} , viz např. [35,36] nebo [37].



Obrázek 1: Průběh chyby během trénovacího procesu.

3. Závěr

V článku byly představeny dva základní postupy řešení problémů inverzní analýzy. Konkrétně pak dopředný postup, který převádí problém inverzní analýzy na optimalizační problém. Je výhodný pro časově nenáročné funkce a jen občasné použití. V případě výpočetně náročných problémů je možné použít paralelní implementace nebo aproximace daného modelu. Jednotlivé aspekty těchto postupů byly důkladně prostudovány s několika příklady z inženýrské praxe.

Druhý postup, zpětný, řeší problém inverzní analýzy hledáním aproximace zpětného vztahu mezi výstupy a vstupy. V článku je ukázána, v poslední době nejčastější, aplikace s využitím umělých neuronových sítí. Jsou probrány některé aspekty tohoto přístupu a jejich důležitost je komentována na příkladu identifikace materiálových parametrů mikroploškového modelu betonu.

4. Poděkování

Autoři by rádi poděkovali za finanční podporu grantem Ministerstva školství a tělovýchovy České Republiky, číslo MSM 6840770003.

5. Literatura

- [1] L. Ingber: Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, (1993), 18(11):29-57.
- [2] R.V.V. Vidal, editor: *Applied Simulated Annealing*, volume 396 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, (1993).
- [3] D.E. Goldberg: Genetic algorithms in search, optimization and machine learning, Addison-Wesley, (1989).
- [4] Z. Michalewicz: Genetic Algorithms+Data Structures=Evolution Programs, Springer, (1992).
- [5] S.W. Mahfoud: *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, (1995).
- [6] C.A.C. Coello: List of references on evolutionary multi objective optimization, <http://www.lania.mx/~ccoello/EMOO/EMOObib.html>.
- [7] K. Miettinen: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Dordrecht, (1999).
- [8] E. Cantú-Paz: *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, (2001).
- [9] J. Lee, P. Hajela: Application of classifier systems in improving response surface based approximations for design optimization. *Computers & Structures*, (2001), 79:333-344.
- [10] M.K. Karakasis, K.C. Giannakoglou: On the use of surrogate evaluation models in multi-objective evolutionary algorithms, *ECCOMAS*, (2004).
- [11] H. Nakayama, K. Inoue, Y. Yoshimori: Approximate optimization using computational intelligence and its application to reinforcement of cable-stayed bridges, *ECCOMAS*, (2004).
- [12] 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.
- [13] A. Kučerová, M. Lepš, J. Skoček: Large Black-box Functions Optimization Using Radial Basis Function Networks, *Proceedings of Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, (2005).
- [14] L.F. Gonzalez, E.J. Whitney, J. Périaux, M. Sefrioui, K. Srinivas: Multidisciplinary aircraft conceptual design and optimisation using a robust evolutionary technique. *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Jyväskylä, (2004).
- [15] J. Wang, J. Periaux, M. Sefrioui: Parallel evolutionary algorithms for optimization problems in aerospace engineering. *Journal of Computational and Applied Mathematics*, (2002), 149:155-169.
- [16] D. Quagliarella: Airfoil design using Navier-Stokes equations and an asymmetric multi-objective genetic algorithm, *Evolutionary Methods for Design, Optimization and Control*:

Applications to Industrial and Societal Problems, Eurogen 2003, International Center for Numerical Methods in Engineering (CIMNE), (2003).

- [17] O. Hrstka, A. Kučerová, M. Lepš, J. Zeman: A competitive comparison of different types of evolutionary algorithms, *Computers and Structures*, (2003), 81/18-19:1979-1990.
- [18] 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.
- [19] J. Lemaitre, J.-L. Chaboche: *Mechanics of solid materials*, Cambridge University Press, USA, (2000).
- [20] A. Kučerová, M. Mühlbauer, Z. Bittnar: Parameter Identification for Rock Long Term Behaviour Simulation, *NAFEMS world congress 2003*, Orlando, Florida, U.S.A., (2003).
- [21] M. Kutílek, V. Kuráž, M. Císlarová: Retenční čáry půdní vlhkosti, *Hydropedologie*, 10, (2000), 102-107.
- [22] V. Kuráž, A. Kučerová, M. Kuráž: Využití genetických algoritmů pro aproximaci retenčních čar, *Pedologické dny 2003*, (2003), 107-111.
- [23] 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.
- [24] M.E. Jirásek, Z.P. Bažant: *Inelastic analysis of structures*, John Wiley & Sons, (2001).
- [25] J. Němeček: Modelling of compressive softening of concrete, *CTU Reports*, 4(6), (2000).
- [26] 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).
- [27] G. Yagawa and H. Okuda: *Neural networks in computational mechanics*, CIMNE, (1996).
- [28] S. Haykin: *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 2nd edition, (1998).
- [29] Ch. Bishop: *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, (1995).
- [30] J. Hertz, A. Krogh, R.G. Palmer: *An Introduction to the Theory of Neural Computation*, Addison Wesley, (1991).
- [31] 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).
- [32] M. Lepš: *Single and Multi-Objective Optimization in Civil Engineering with Applications*, PhD thesis, CTU in Prague, (2005).
- [33] D. Lehký, D. Novák: Identification of material model parameters using stochastic training of neural network. In *Proceedings of 5th International PhD Symposium in Civil Engineering*, Delft, Netherlands, (2004).

- [34] D. Novák, M. Vořechovský, R. Rusina: Small-sample probabilistic assessment - software FREET. In *Proceedings of 9th International Conference on Applications of Statistic and Probability in Civil Engineering - ICASP 9*, pages 91-96, San Francisco, USA. Millpress, Rotterdam, (2003).
- [35] A. Kučerová, M. Lepš, J. Zeman: Microplane model parameters identification using soft computing methods. In S. M. Sivakumar, A. Meher Prasad, B. Daltaguru, S. Narayanan, A.M. Rajendran, and S.N. Atluri, editors, *Proceedings of International Conference on Computational & Experimental Engineering and Sciences*, pages 980-985, Forsyth, USA, (2005). Tech Science Press.
- [36] A. Kučerová, M. Lepš, J. Zeman: Estimating parameters of microplane material model using soft computing methods,. In J. Herskovic, Mazorche S., and Canelasm A., editors, *WCSMO6 - Sixth World Congresses of Structural and Multidisciplinary Optimization*, COPPE Publications, (2005).
- [37] A. Kučerová, M. Lepš, J. Zeman: Back analysis of microplane model parameters using soft computing methods. In Z. Waszczyszyn and M. Słowski, editors, *NNSC-2005: International Symposium on Neural Network and Soft Computing in Structural Engineering*, Cracow, Poland, (2005).