

# EVOLUČNÍ ALGORITMY A JEJICH APLIKACE V NELINEÁRNÍ REGRESI

Josef TVRDÍK a Ivan KŘIVÝ  
OU PŘF, KI a KM

**Abstract:** Two stochastic algorithms of optimization are described. Both of them are based on the control random search with a randomized reflection of simplex combined with some ideas of evolution algorithms. The experimental results of application to the estimation of non-linear regression parameters are presented. Both algorithms are found to be fast enough and reliable in finding the global minimum of objective function.

**Резюме:** В работе описаны два стохастические алгоритмы для оптимизации. Эти алгоритмы основаны на управляемом случайном поиске с использованием случайной рефлексии в симплексной связи и некоторых принципов эволюционных алгоритмов. Пресентированы результаты применения этих алгоритмов в оценке параметров моделей нелинейной регрессии. Установлено, что эти алгоритмы достаточно быстрые и надёжные в поиске глобального минимума критериальной функции.

## 1 Úvod

Odhad parametrů nelineárních regresních modelů je většinou úloha hledání minima kritériální funkce v prostoru parametrů. Kritériální funkcí může být např. součet čtverců reziduí (klasická metoda nejmenších čtverců), suma absolutních odchylek, největší absolutní odchylka, medián čtverců odchylek, uřezaná suma čtverců odchylek (*trimmed least squares*) atd. Prostor parametrů v praktických úlohách je zpravidla vymezen intervalem jejich přípustných hodnot, takže odhad parametrů nelineárních regresních modelů je vlastně optimalizační úloha s tzv. *box constraints*. Formálně lze naši úlohu hledání zapsat takto: Najdi  $\mathbf{x}_m \in \Omega$  takové, že  $f(\mathbf{x}_m) \leq f(\mathbf{x})$  pro všechna  $\mathbf{x} \in \Omega$ ;  $\Omega$  je podmnožinou  $d$ -rozměrného euklidovského prostoru,  $\Omega = \prod_{i=1}^d \langle a_i, b_i \rangle$ ,  $a_i < b_i$  pro všechna  $i$ .

Při odhadu parametrů nelineárních regresních modelů nevystačíme pouze s tradičními algoritmy optimalizace využívajícími derivací kritériální funkce. Takové algoritmy často nenajdou globální minimum, v lepším případě skončí v minimu lokálním. Je také známo, že při odhadu parametrů nelineárních regresních modelů běžné softwarové prostředky občas selhávají fatálně [7].

Řadu let jsou známy jiné druhy algoritmů, které hledají minimum prohledáváním prostoru parametrů. Zde připomeneme simplexovou metodu,

kteřou před třiceti lety navrhli Nelder a Mead [8]. Jejich algoritmus lze stručně popsat takto:

V množině  $\Omega$  zvolíme  $d + 1$  lineárně nezávislých bodů (*simplex*), v těchto bodech vyhodnotíme kritériální funkci  $f(\mathbf{x})$ , najdeme bod  $\mathbf{x}_H$ , ve kterém je  $f(\mathbf{x}_H)$  větší než v ostatních bodech, a pak určíme bod  $\mathbf{x}_P$  středově souměrný k  $\mathbf{x}_H$  podle těžiště ostatních bodů simplexu. Této operaci se říká *reflexe* a pro  $d = 2$  je graficky znázorněna na obr. 1. Pokud je  $f(\mathbf{x}_P) \leq f(\mathbf{x}_H)$ , pak nahrazením bodu  $\mathbf{x}_H$  bodem  $\mathbf{x}_P$  získáme nový simplex a opakujeme reflexi, dokud funkční hodnoty ve všech bodech simplexu nejsou dostatečně blízké. Pověšme si, že reflexí se nemusí vždy podařit najít takové  $\mathbf{x}_P$ , aby platilo  $f(\mathbf{x}_P) \leq f(\mathbf{x}_H)$ . Pak je nutno doplnit algoritmus o další operace – viz [8].

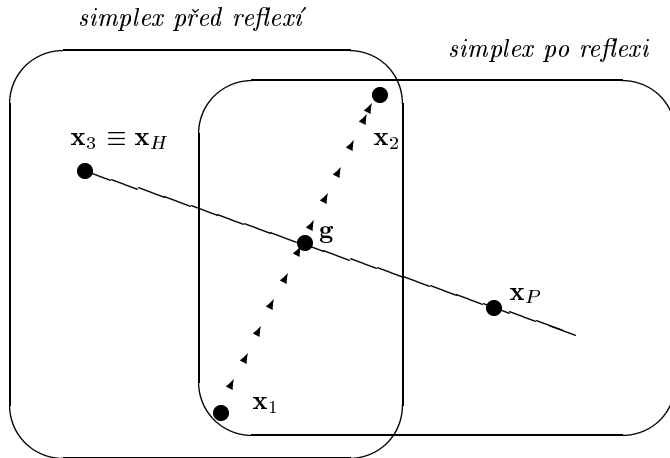


Figure 1: Reflexe pro  $d = 2$ .

Později Price [9] navrhl algoritmus řízeného náhodného prohledávání (*controlled random search*, CRS). Tento algoritmus můžeme zařadit mezi stochastické algoritmy optimalizace, neboť jeho iterační kroky nelze deterministicky predikovat. Na počátku se náhodně vygeneruje  $N$  bodů z  $\Omega$  ( $N \gg d$ ), v každém z těchto bodů se vyhodnotí kritériální funkce  $f(\mathbf{x})$ . Těchto  $N$  bodů z  $\Omega$  spolu s jejich hodnotami kritériální funkce tvoří tzv. *pool*, který je zapamatován a aktualizován během výpočtu. Simplexy  $d + 1$  bodů se vybírají z těchto  $N$  bodů náhodně. Reflexe se provede stejně jako v simplexové metodě, a pokud je  $f(\mathbf{x}_P)$  menší než dosud největší hodnota kritériální funkce  $f(\mathbf{x}_M)$  zapamatované v poolu, nahradí se bod  $\mathbf{x}_M$  v poolu bodem  $\mathbf{x}_P$ . Pool se tak postupně každou úspěšnou reflexí „zlepšuje“. Výpočet končí, když je splněna nějaká vhodná podmínka, např. rozdíl největší a nejmenší hodnoty kritériální funkce zapamatované v poolu je dostatečně malý.

V posledních letech se mezi aplikacemi optimalizačních algoritmů stále více uplatňují algoritmy motivované biologickým vývojem populací, ať již se jedná o genetické algoritmy [2], či evoluční algoritmy v širším smyslu [5, 6]. Základní myšlenky těchto algoritmů lze vyjádřit takto:

- počáteční populace je vybrána náhodně
- jedinci nové populace dědí vlastnosti jedinců předchozí populace
- s malou pravděpodobností mohou vznikat jedinci s novými vlastnostmi (tzv. mutace)

Aplikací těchto myšlenek spolu s využitím řízeného náhodného prohledávání lze navrhnout algoritmy optimalizace, které mohou být úspěšně užity při odhadu parametrů nelineárních regresních modelů. Zde uvedeme dva takové algoritmy - modifikovaný algoritmus řízeného náhodného prohledávání MCRS [3] a na evoluční strategii založený algoritmus ES2 [4, 12] spolu s výsledky experimentálního testování těchto dvou algoritmů na sadě čtrnácti úloh odhadu parametrů nelineárních regresních modelů.

## 2 Algoritmus MCRS

Základní rysy této modifikace Priceova algoritmu byly poprvé uvedeny na Robustu 94 [10]. Modifikace spočívá ve znárodnění reflexe. Definujeme proceduru *Simplex*, která generuje nový bod  $\mathbf{x}$  ze simplexu  $S$  ( $d+1$  lineárně nezávislých bodů z populace bodů  $P$ ) vztahem

$$\mathbf{x} = \mathbf{g} - \Gamma(\mathbf{z} - \mathbf{g}), \quad (1)$$

kde  $\mathbf{z}$  je náhodně zvolený vrchol simplexu  $S$ ,  $\mathbf{g}$  těžiště zbývajících  $d$  vrcholů tohoto simplexu a  $\Gamma$  náhodný multiplikativní koeficient. Dále zapíšeme proceduru *Reflexe* ve tvaru:

```

procedure Reflexe( $P$ , var  $\mathbf{x}$ );
repeat
     $S := (d + 1)$  nahodne vybranych bodu z  $P$ ;
    Simplex( $S$ );
until  $\mathbf{x} \in \Omega$ .

```

Pak můžeme algoritmus MCRS napsat takto:

```

procedure MCRS;
begin  $P :=$  populace  $N$  nahodne generovanych bodu v  $\Omega$ ,
    repeat
        Reflexe( $P$ ,  $\mathbf{x}$ );
        if  $f(\mathbf{x}) < f(\mathbf{x}_{max})$  then  $\mathbf{x}_{max} := \mathbf{x}$ ;

```

**until** splněna podmínka ukončení;  
**end** {MCRS};

$\mathbf{x}_{max}$  je ten bod ze zapamatované populace  $N$  bodů, ve kterém je hodnota kritériální funkce nejvyšší. Lze očekávat, že s rostoucím rozptylem náhodné veličiny  $\Gamma$  bude prohledáván prostor parametrů důkladněji, a tedy tím větší bude naděje, že algoritmus nalezne globální minimum. Uspokojivé výsledky byly dosaženy, když  $\Gamma$  bylo rovnoměrně rozděleno na intervalu  $(0, \alpha)$  a hodnota vstupního parametru  $\alpha$  byla mezi 4 až 8 [3].

Při aplikacích iteračních algoritmů v nelineární regresi lze jako podmínku ukončení užívat vztahu

$$\frac{f(\mathbf{x}_{max}) - f(\mathbf{x}_{min})}{f_0} \leq \varepsilon_0, \quad (2)$$

kde  $\mathbf{x}_{min}$  je bod s nejmenší hodnotou kritériální funkce,  $\varepsilon_0$  vstupní parametr algoritmu ( $\varepsilon_0 > 0$ ) a  $f_0$  vhodná charakteristika variability závisle proměnné. Pokud např. minimalizujeme součet čtverců reziduí (RSS),  $f_0$  je celková suma čtverců odchylek pozorovaných hodnot závisle proměnné od jejich průměru [3]. Tato podmínka ukončení (2) je univerzálněji použitelná než kritérium  $f(\mathbf{x}_{max}) - f(\mathbf{x}_{min})$ , navrhané Conlonem [1]. Vstupními parametry algoritmu jsou  $N$  (počet zapamatovávaných bodů tvořících populaci),  $\alpha$  (určující maximální vzdálenost nově vytvořeného bodu od těžiště v rovnici (1)),  $\varepsilon_0$  v podmínce ukončení (2) a specifikace  $\Omega$ .

### 3 Evoluční algoritmus ES2

Evoluční algoritmy (viz např. [5, 6]) simulují mechanismus přirozeného výběru, který obecně směřuje k dosažení optima. Algoritmus ES2, podobně jako dříve popsany algoritmus ES1 [11], vychází z následujících principů:

- Počáteční populace bodů se generuje náhodně ze zadaného parametrického prostoru  $\Omega$ .
- Nová populace vznikající v každém iteračním kroku dědí vlastnosti staré populace, a to dvojím způsobem:
  - body s nejnižší hodnotou  $f$  se přímo kopírují ze staré populace do nové,
  - zbývající body nové populace se vytvářejí pomocí reflexe z bodů staré populace.
- V každém iteračním kroku může vzniknout s malou pravděpodobností  $p_0$  bod se zcela novými vlastnostmi (dokonce i s vyšší hodnotou funkce  $f$ ), což je obdoba mutace v genetických algoritmech.

Algoritmus ES2 lze formálně popsat takto:

```
procedure ES2;
begin     $P :=$  populace  $N$  bodu generovaných náhodně v  $\Omega$ 
         $Sort(P)$  vzestupně podle hodnot funkce  $f$ ;
    repeat
         $m :=$  náhodně celé číslo z intervalu  $\langle 1, M \rangle$ 
        for  $j := 1$  to  $m$  do  $Q[j] := P[j]$ ;
        for  $j := m + 1$  to  $N$  do  $Q[j].f := P[N].f$ ;
         $i := m$ ;
        repeat
            repeat  $Reflexe(P, \mathbf{x})$ 
            until  $f(\mathbf{x}) < f(P[N].f)$ 
             $i := i + 1$ ;
             $BinaryInsert \langle \mathbf{x}, f(\mathbf{x}) \rangle$  into  $Q$ ;
        until  $i = N$ ;
        if  $random < p_0$  then
            begin  $i :=$  náhodně celé číslo z intervalu  $\langle 1, N \rangle$ ;
                nahrad  $Q[i]$  bodem generovaným pomocí  $Reflexe(P, \mathbf{x})$ 
                 $Sort(Q)$  vzestupně podle hodnot funkce  $f$ ;
            end {if};
         $P := Q$ ;
    until splněna podmínka pro ukončení optimalizace
end {ES2}
```

V algoritmu ES2, stejně jako v algoritmu MCRS, může být použito kterékoli z kritérií optimalizace zmíněných v úvodu. K zastavení optimalizačního procesu lze také využít vztahu (2).

Ve srovnání s algoritmem MCRS vyžaduje evoluční algoritmus ES2 navíc dva vstupní parametry, a to:

- pravděpodobnost mutace  $p_0$ ,
- nezáporné celé číslo  $M$  takové, že  $M < N$ .  $M$  je nejvyšší počet bodů, které může zdědit nová populace ze staré.

Na tomto místě poznamenejme, že algoritmus ES2 je v podstatě ekvivalentní algoritmu MCRS, pokud zvolíme  $p_0 = 0$  a  $M = 0$ .

Je zřejmé, že oba naše algoritmy (ES2 i MCRS) jsou velmi blízké genetickým algoritmům (GA), přitom reflexe v našich algoritmech přebírá úlohu křížení v GA. Oba typy algoritmů pracují s populacemi bodů (nikoli s jediným bodem), využívají pouze hodnot optimalizované funkce  $f$  (nikoli jejich derivací) a respektují stochastické principy reprodukce (selekce a křížení).

Existují ovšem jisté rozdíly mezi našimi algoritmy a GA.

1. Populace v našich algoritmech jsou tvořeny body (vektory s reálnými složkami), kdežto GA pracují s populacemi binárních řetězců. Tento rozdíl se však postupně vytrácí, protože novější verze GA (viz např. [6]) operují na chromozomech tvořených reálnými vektory.
2. Naše algoritmy generují nové body (potomky) reflexí z množiny  $d + 1$  náhodně vybraných simplexových bodů (rodičů), zatímco pro GA je typická párová reprodukce. V našem případě jde tedy o jakousi zobecněnou vícenásobnou reprodukci.

## 4 Experimentální výsledky

Popsané algoritmy byly ověřovány na 14 příkladech, modelových datech (pro nelineární regresi) sestavených tak, že většina „klasických“ optimalizačních postupů založených na derivaci funkce  $f$  při jejich zpracování selhává. Tato data jsou spolu s odpovídajícími regresními modely uvedena v naší dřívější práci [11].

Oba algoritmy byly implementovány v jazyku *TurboPascal*, verze 6.0, a příslušné výpočty provedeny na PC 486DX, 66MHz. Ve všech případech jsme jako optimalizačního kritéria používali reziduální součet čtverců.

Vstupní parametry společné oběma algoritmům byly nastaveny shodně, a to:

- $N = 5d$ ,
- $\alpha = 4$ ,
- $1E-16$ .

Zbývající vstupy pro ES2 algoritmus:  $p_0 = 0,01$  a  $M = \text{Int}(N/2)$ .

Experimentální výsledky jsou uvedeny v tab. 1. V této tabulce  $\bar{t}$  označuje průměrnou dobu výpočtu určenou z minimálně 400 (zpravidla více) nezávislých běhů programu,  $sd$  směrodatnou odchylku doby výpočtu a *fail* procento selhání, tj. procento případů, kdy výpočet skončil v lokálním minimu.

Z údajů v tab. 1 vyplývá, že algoritmus MCRS je ve všech testovaných případech rychlejší než algoritmus ES2, u některých modelů však selhává (např. pro data č. 11 ve více než jedné čtvrtině nezávislých běhů). Algoritmus ES2 je sice pomalejší (v některých případech až čtyřikrát), ale procento selhání je prakticky zanedbatelné.

## 5 Závěr

Oba prezentované algoritmy jsou nedeterministické. V obou jsou náhodně generovány počáteční populace, vrcholy simplexu jsou vybírány náhodně a rovněž procedura reflexe je nedeterministická. U ES2 navíc přistupují

Tab. 1: Přehled experimentálních výsledků

Ident. no.	MCRS			ES2		
	$t$ [s]	$sd$ [s]	$fail$ [%]	$t$ [s]	$sd$ [s]	$fail$ [%]
1	2.4	0.1	0	5.8	0.8	0
2	6.4	0.9	2	27.1	8.2	0.3
3	3.6	0.3	0	12.7	1.4	0
4	2.4	0.3	0	6.6	0.8	0
5	32.1	1.3	0	51.3	12.6	0
6	1.1	0.1	0	3.2	0.4	0
7	44.8	2.7	0	72.8	24.0	0
8	106.4	12.7	0	169.7	22.3	0
9	1.8	0.2	0	5.4	1.2	0
10	91.4	38.8	0	106.4	33.6	0
11	5.7	1.7	28	18.6	3.4	0.3
12	5.6	0.5	0	23.0	2.3	0
13	179.2	59.0	0	268.9	31.5	0
14	22.6	1.5	0	32.1	3.5	0

další dva stochastické kroky - velikost dědené části staré populace je určována náhodně a mutace nastává jen s jistou malou pravděpodobností a s nepředvídatelným výsledkem. Přesto oba algoritmy až překvapivě rychle a algoritmus ES2 s vysokou spolehlivostí našly globální minimum ve všech 14 obtížných testovacích příkladech. Lze je tedy doporučit přinejmenším jako alternativní postup odhadu regresních parametrů nelineárních modelů v takových situacích, kdy běžné softwarové prostředky selhávají. Oba algoritmy jsou snadno implementovatelné a nevyžadují na vstupu zadání počátečních hodnot odhadovaných parametrů.

## References

- [1] Conlon, M.: *The Controlled Random Search Procedure for Function Optimization*. Commun. Statist.-Simula. Comput., **21**(3) (1992) 919-923.
- [2] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Addison Wesley 1989.
- [3] Krivý, I., Tvrđík, J.: *The Controlled Random Search Algorithm in Optimizing Regression Models*. Comput. Statist. and Data Anal. **20**(2) (1995) 229-234.

- [4] Křivý, I., Tvrđík, J.: *Stochastic Algorithms in Estimating Regression Models*. In: A. Prat (Ed.), COMPSTAT 1996. Proceedings in Computational Statistics (Physica-Verlag, Vienna, 1996) 325-330.
- [5] Kvasnička, V.: *A Hybrid of Simplex Method and Simulated Annealing*. Chemometrics, to appear.
- [6] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Springer Verlag 1992.
- [7] Militký, J.: *Nonlinear Regression on Personal Computers*. In: R. Dutter and W. Grossmann (Eds.), COMPSTAT 1994. Proceedings in Computational Statistics (Physica-Verlag, Vienna, 1994) 395-400.
- [8] Nelder, J.A., Mead, R. : *A simplex method for function minimization*. Computer J., **7**(1) (1964) 308-313.
- [9] Price, W.L.: *A controlled random search procedure for global optimization*. Computer J., **20**(4) (1976) 367-370.
- [10] Tvrđík, J.: *Odhad regresních parametrů metodou řízeného náhodného výběru*. In: Sborník ROBUST 94. Praha, JČMF 1994, 153-159,
- [11] Tvrđík, J., Křivý, I.: *Stochastic Algorithms in Estimating Regression Parameters*. In: Proceedings of MME '95. Ostrava, VŠB-TU 1995, 217-228.
- [12] Tvrđík, J., Křivý, I.: *A Genetic Type Algorithm for Optimization*. In: Proceedings of MENDEL'96, 2nd International Conference on Genetic Algorithms. Brno, VUT 1996, 176-180.