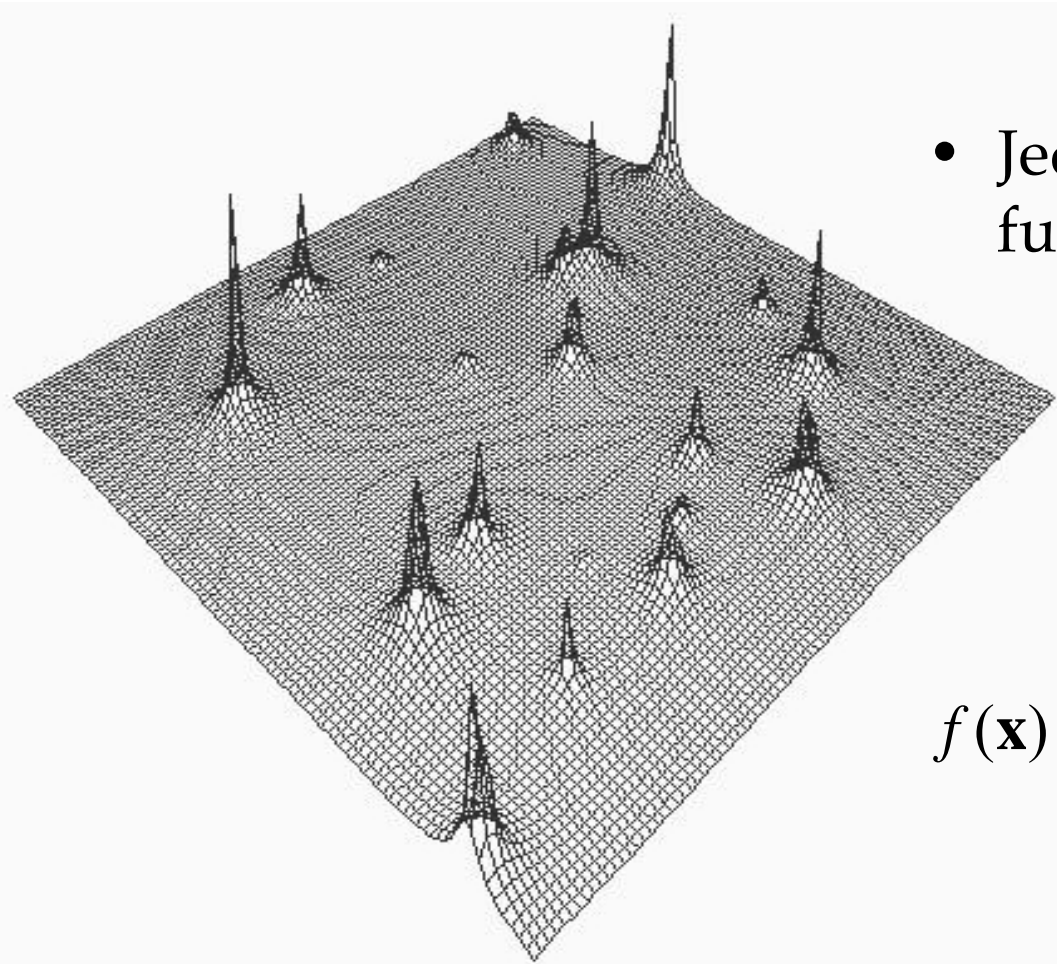


Multimodální optimalizace

- **Optimalizace funkcí, které mají více než jedno lokální minimum**
- **Indikátor - několik spuštění optimalizačního algoritmu dává rozdílné výsledky**
- **Úzce souvisí s předčasnou konvergencí**

Multimodální problémy ①

- Příklad: Funkce “Type 0”



- Jednoduchá multimodální funkce:

$$f(\mathbf{x}) = \sum_i^n y_i \left(\frac{\pi}{2} - \arctan \frac{\|\mathbf{x} - \mathbf{x}_i\|}{r_i} \right)$$

Multimodální problémy ②

- Příklad: Optimalizace ŽB nosníku

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

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

[Lepš, Diplomová práce, 2000]

Multimodální problémy ②

- Příklad: Optimalizace ŽB nosníku pokr.

9 007 199 254 740 992	Možných řešení
100 000 000	MC bodů, z toho:
24 698 222	Platných řešení
722	S cenou pod 1000 Kč
55	S cenou pod 900 Kč
0	S cenou pod 800 Kč

Minimum 573,50 Kč

Metody řešení

- **Re-start**
 - Opakované spuštění optimalizace z jiného startovacího bodu
 - Nejčastější použití u metod matematického programování
- **Zapamatování si již navštívených bodů**
 - Viz např. TABU Search metoda, nebo-li použití „paměti“

Metody řešení

- **Re-annealing**
 - Skoková změna teploty u Simulovaného žíhání
- **Metody na zachování diverzity u algoritmů s populací**
 - Např. rozdělením populace na oddělené části, viz paralelizace evolučních algoritmů
 - nebo potlačením elitismu při výběru nové populace

Metody řešení

- **Fitness Sharing**

- Změna cílové funkce podle počtu řešení sdílejících oblast lokálního minima

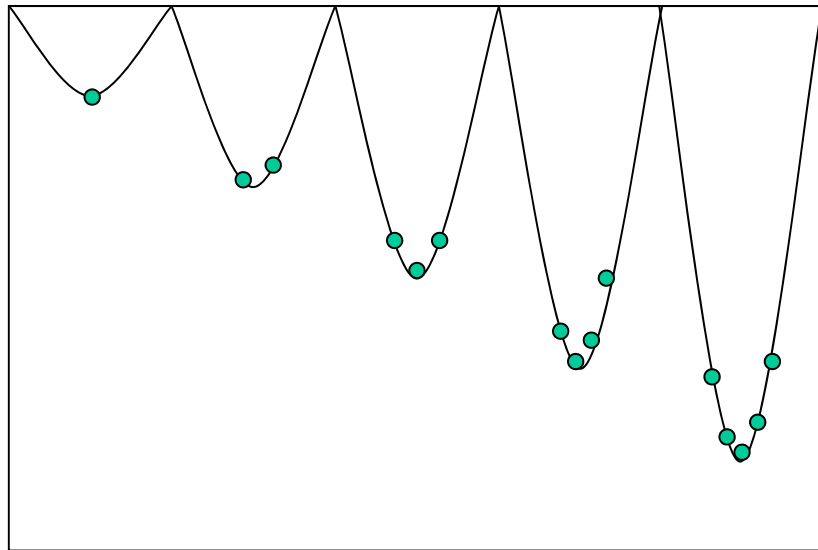
$$F'(i) = \frac{F(i)}{\sum_j sh(d(i, j))},$$
$$sh(d) = \begin{cases} 1 - (d / \sigma_{share})^\alpha & \text{pokud } d \leq \sigma_{share}, \\ 0 & \end{cases}$$

- Dva parametry: α - tvar

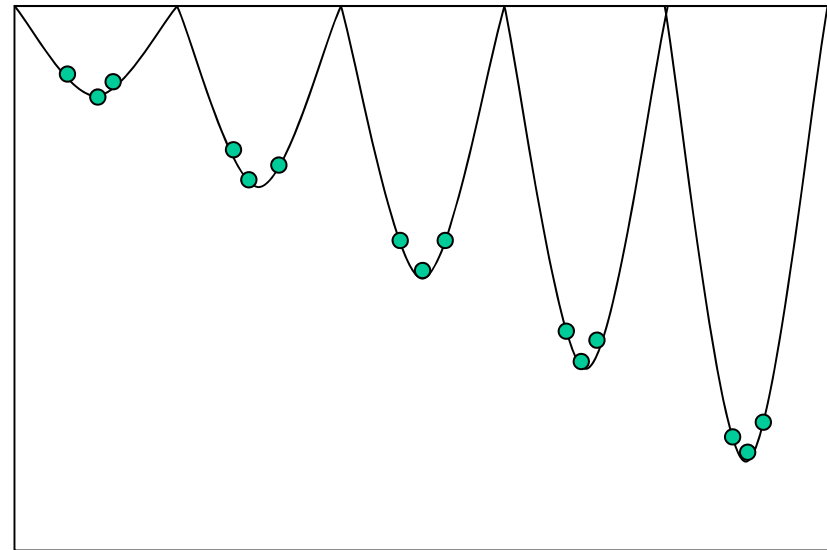
σ_{share} - kolik řešení tvoří okolí

Metody řešení

- **Crowding**
 - Steady-state selekce
 - Potomci nahrazují pouze „blízké“ rodiče



Fitness Sharing



Crowding

Metoda CERAF

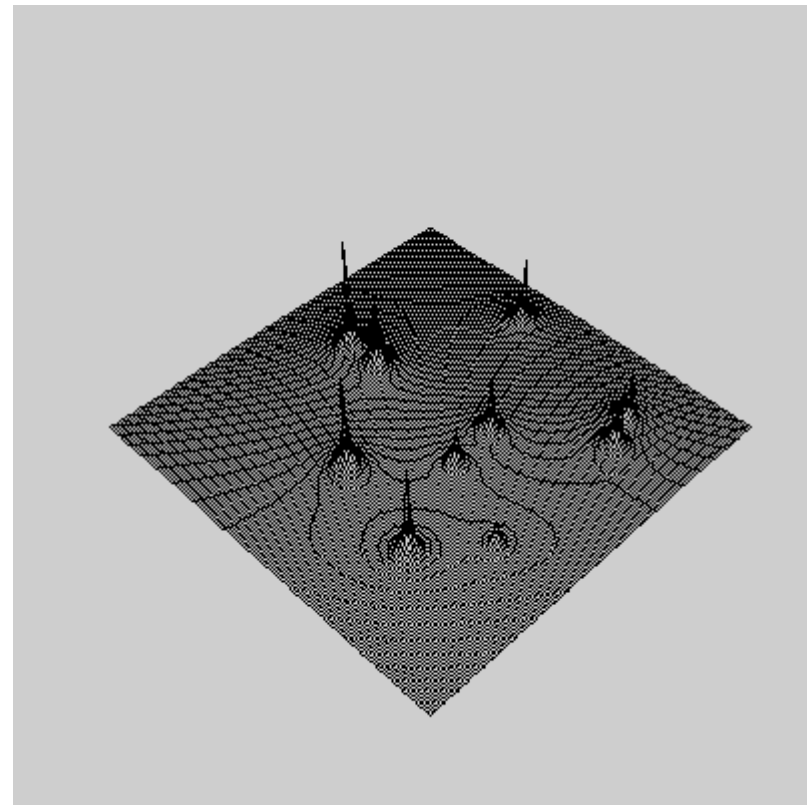
- **CERAF** (La méthode de Centre Radio-Actif)
- Navržený Hrstkou a Kučerovou, 2003
- Nezávislá nadstavba pro optimalizační algoritmus s populací
- Re-startovací algoritmus s „pamětí“
- Ukládá v paměti nalezené lokální extrémy
- Brání optimalizačnímu algoritmu konvergovat znovu do stejného lokálního optima
- Zvyšuje robustnost optimalizačního algoritmu
- Uživatel si může zvolit jedno z nalezených optim

Metoda CERAF

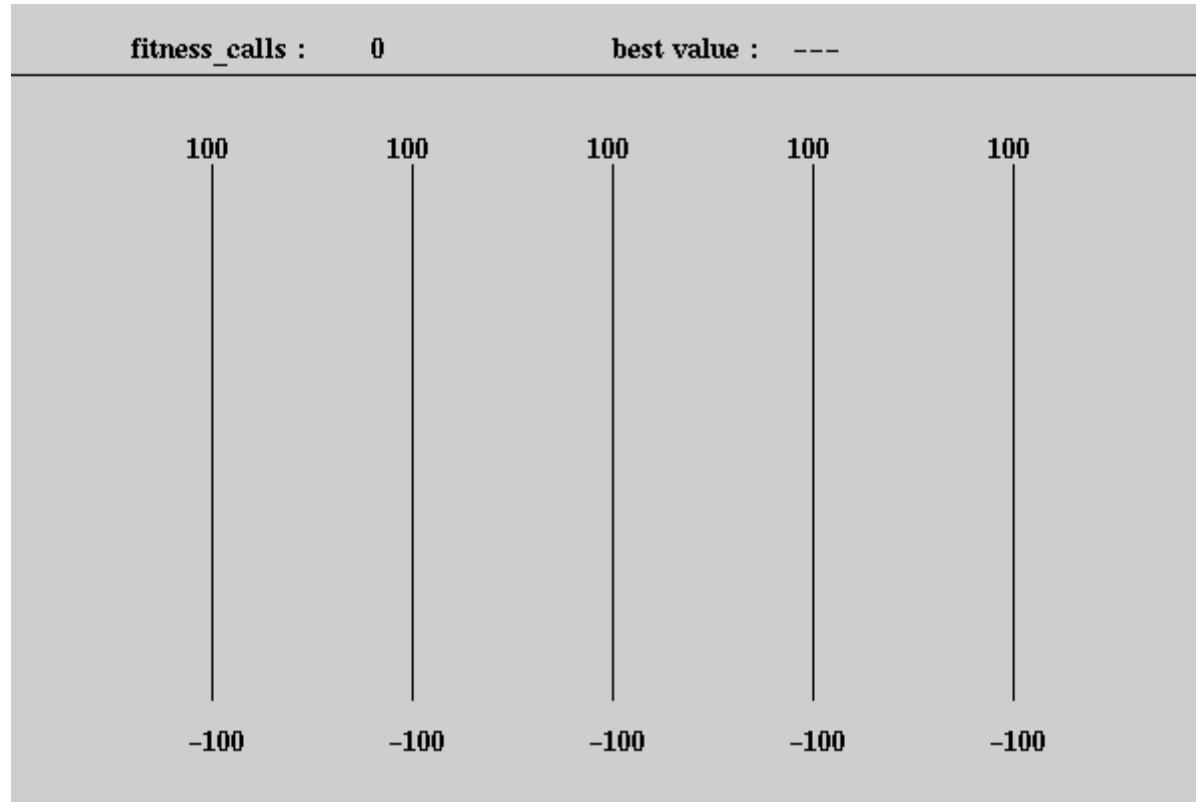
- Pokud se nejlepší nalezená hodnota nezlepšuje po daný počet generací algoritmu o více než zvolenou hodnotu, je daný bod označen jako lokální optimum.
- Optimalizační algoritmus je restartován a optimalizace začíná znovu s náhodnou populací.
- Kolem lokálního optima je vytvořena eliptická „radioaktivní“ zóna.
- Každý jedinec vytvořený algoritmem uvnitř zóny je v průběhu selekce „zničen“ a nahrazen novým, mimo zónu.
- Pokud byl jedinec vytvořen operátorem křížení, je zároveň zóna nepatrně zmenšena, což umožní nalezení jiného optima uvnitř zóny, pokud má algoritmus tendenci směrem k ní konvergovat.

Použití metody CERAF ①

- Optimalizace funkce
Typu 0 s využitím
algoritmu **SADE**
rozšířeného o metodu
CERAF

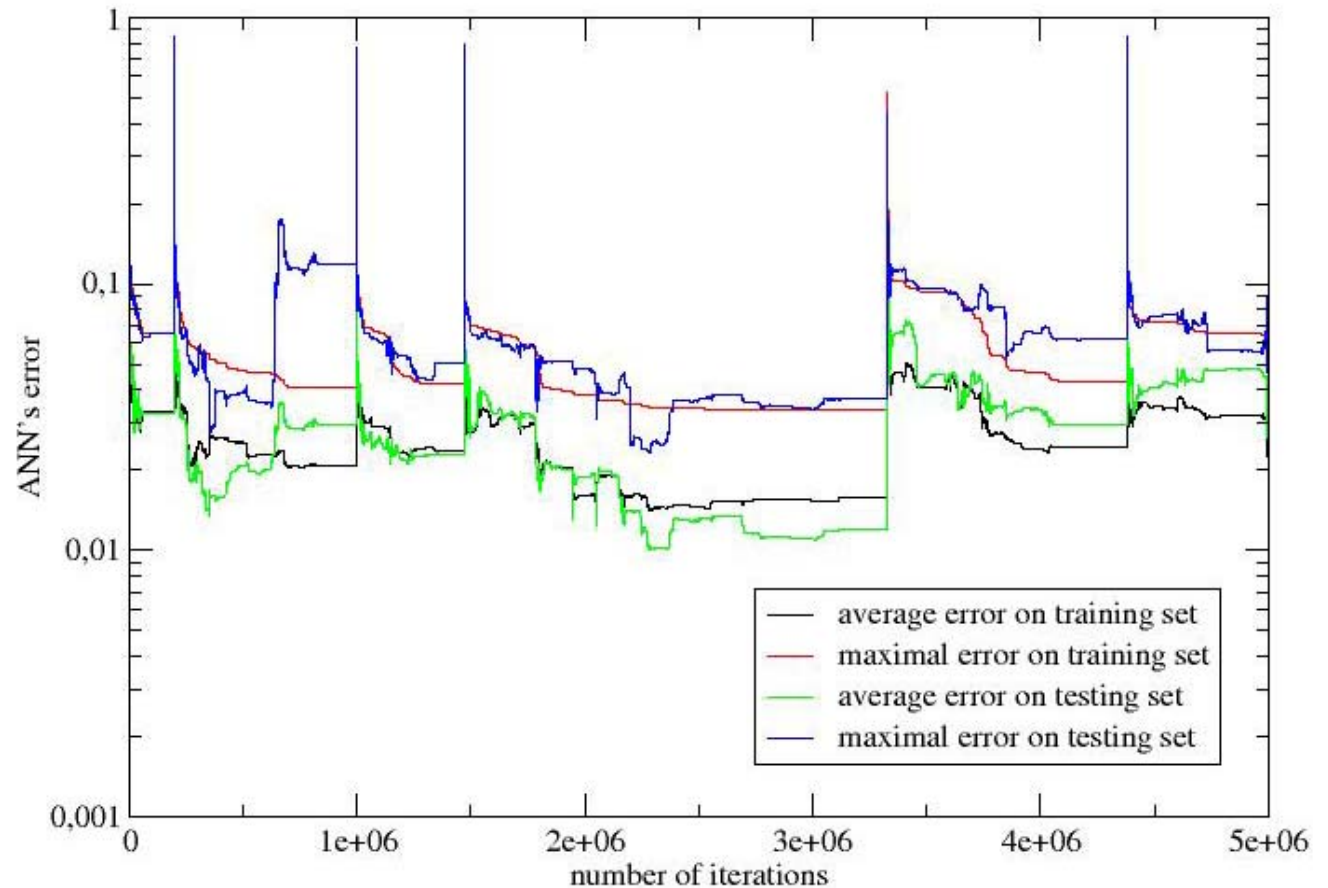


Použití metody CERAF ②



Použití metody CERAF ③

- Optimalizace vah neuronové sítě



Porovnání několika algoritmů

Table 3
Comparison of reliability

Function	SGA	EGA	DE	SADE	CERAF
F1	×	×	×	×	×
F3	×	×	×	×	×
Branin		×	×	×	×
Camelback	×	×	×	×	×
Goldprice		×	×	×	×
PShubert 1		×		×	×
PShubert 2		×		×	×
Quartic		×		×	×
Shubert		×		×	×
Hartman 1		×	×	×	×
Shekel 1		×		×	×
Shekel 2		×		×	×
Shekel 3		×		×	×
Hartman 2					×
Hosc 45			×	×	×
Brown 1			×		×
Brown 3			×	×	×
F5n		×	×	×	×
F10n					×
F15n		×	×	×	×

[Hrstka & Kučerová, 2004]

Reference

- [1] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

- [2] Mahfoud, S. W. (1995). Niching methods for genetic algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

- [3] Hrstka, O. and Kučerová, A. (2004). Improvements of real coded genetic algorithms based on differential operators preventing the premature convergence. *Advances in Engineering Software*, 35(3-4):237-246.

- [4] A. E. Eiben, J. E. Smith, Agoston E. Eiben, J. D. Smith (2003). *Introduction to Evolutionary Computing*. Springer.

Prosba. V případě, že v textu objevíte nějakou chybu nebo budete mít námět na jeho vylepšení, ozvěte se prosím na **matej.leps@fsv.cvut.cz**.

Datum poslední revize: 16.11.2007

Verze: 001