

Moderné prístupy riešenia optimalizačných problémov pomocou metód evolučných výpočtov

Článok sa venuje problematike evolučných výpočtových techník. Opisuje základné črty genetického algoritmu, evolučných stratégií, diferenciálnej evolúcie, umelého imunitného systému, samoorganizujúceho sa migračného algoritmu a algoritmu Particle Swarm Optimization.

Všetky z uvedených metód pracujú na základe princípov odpozorovaných zo správania živých organizmov v prírode.

Algoritmy patriace do množiny evolučných výpočtov predstavujú účinný nástroj na riešenie zložitých optimalizačných úloh. Medzi ich hlavné prednosti patrí jednoduchosť a ľahká použiteľnosť. Významnou výhodou oproti špecializovaným metódam je univerzálnosť použitia. Jedným algoritmom dokážeme riešiť širokú škálu problémov. Miernou nevýhodou je závislosť času riešenia a kvality nájdeného výsledku od voľby riadiacich parametrov jednotlivých algoritmov. Spomínané parametre nemožno určiť exaktne, ale len odhadnúť na základe predchádzajúcich skúseností.

Napriek uvedeným komplikáciám metódy evolučných výpočtov takmer vždy poskytujú postačujúce výsledky a umožňujú riešiť optimalizačné problémy ťažko riešiteľné inými bežnými metódami.

Úvod

Pod pojmom evolučné výpočty [1] rozumieme množinu metód pracujúcich na princípoch odpozorovaných z prírody. Patria sem genetické algoritmy, evolučné stratégie, diferenciálna evolúcia, umelý imunitný systém, genetické programovanie, samoorganizujúci sa migračný algoritmus, Particle Swarm Optimization a mnohé ďalšie. Výhodne sa využívajú najmä pri riešení zložitých optimalizačných úloh. Vďaka jednoduchej implementácii sú vhodným nástrojom na riešenie rôznych optimalizačných problémov v technickej praxi.

Výhody evolučných výpočtov

Univerzálnosť použitia – jednou metódou z množiny EV dokážeme riešiť široké spektrum úloh, ktoré vieme sformulovať ako optimalizačné úlohy.

Jednoduché nasadenie – oproti bežným optimalizačným metódam nepotrebujeme podrobnú znalosť riešeného problému ani dodatočné informácie (napr. gradient). Jedinou podmienkou je existencia účelovej funkcie, ktorou vieme určovať vhodnosť riešenia. Výhodnotenie účelovej funkcie možno robiť pomocou počítačového modelu optimalizovaného problému alebo reálnou simuláciou.

Vysoká účinnosť – metódy EV dokážu riešiť bežnými metódami ťažko riešiteľné úlohy. Patria sem napríklad hľadanie globálneho extrému mnohorozmerných, nelineárnych, multimodálnych funkcií, ďalej riešenie zložitých kombinatorických a grafovo orientovaných úloh a problémov s rôznymi typmi premenných a s rôznymi druhmi obmedzení.

Základné črty evolučných výpočtov

Evolučné výpočty používajú tri základné princípy odpozorované z prírody. Sú to mutácia, kríženie a princíp prežitia silnejšieho je-

dinca. Mutácia a kríženie zaručujú zmenu vlastností potomkov oproti rodičom. Mutácia je náhodná zmena vlastností jedinca vplyvom prostredia (v prírode je to napríklad žiarením alebo teplotnými a chemickými vplyvmi). Kríženie zabezpečuje zmenu potomkov oproti rodičom pomocou vzájomnej zámeny vlastností rodičov. Princíp prežitia silnejšieho jedinca zaručuje prirodzený výber jedincov s najlepšimi vlastnosťami. V prírode majú jedinci obmedzené prostredie s obmedzeným množstvom potravy, o ktorú musia vzájomne súperiť. Prežijú a reprodukovujú sa len jedince, ktoré sú schopné prispôsobiť sa. Rovnako pri hľadaní najlepšieho riešenia pomocou EV máme záujem len o najlepšie jedince reprezentujúce najlepšie riešenia problému.

Princíp riešenia problémov pomocou EV

Najprv potrebujeme preformulovať riešený problém do podoby riešiteľnej pomocou EV. Ďalej potrebujeme zaviesť transformáciu z oblasti problému do oblasti riešenia (vytvoriť lineárny reťazec premenných reprezentujúci vlastnosti riešenia). Jednotlivé reťazce reprezentujú riešenia a predstavujú jedincov populácie. Na začiatku vytvoríme populáciu s niekoľkými jedincami. V priebehu EV meníme pomocou mutácie a kríženia vlastnosti jedincov populácie a vyberáme jedincov s lepšími vlastnosťami, čím vzniká nová, lepšia populácia, na ktorej opäť vykonávame operácie mutácie a kríženia. Proces vykonávame dovtedy, kým nedosiahneme požadované riešenie. Najlepší jedinec populácie potom predstavuje hľadané riešenie.

Genetické algoritmy

GA [2] [3] sú asi najznámejším a najpoužívanejším prostriedkom z množiny EV. Pri svojej činnosti využíva populáciu jedincov, nad ktorou sa vykonávajú operácie mutácie, kríženia a výberu najlepších jedincov. Fungovanie genetického algoritmu demonštruje schéma na obr. 1.

Úspešnosť a najmä rýchlosť konvergencie algoritmu ku globálnemu optimu dokážeme ovplyvňovať voľbou parametrov algoritmu. Sú to hlavne veľkosť populácie, pravdepodobnosť kríženia a pravdepodobnosť mutácie. Hodnoty uvedených parametrov treba voľiť vzhľadom na riešený problém. Odhad vhodných parametrov je vecou skúsenosti.

Evolučné stratégie (ES)

Prvá verzia ES [4], [5] pracovala s jedným jedincem reprezentovaným dvojicou vektorov:

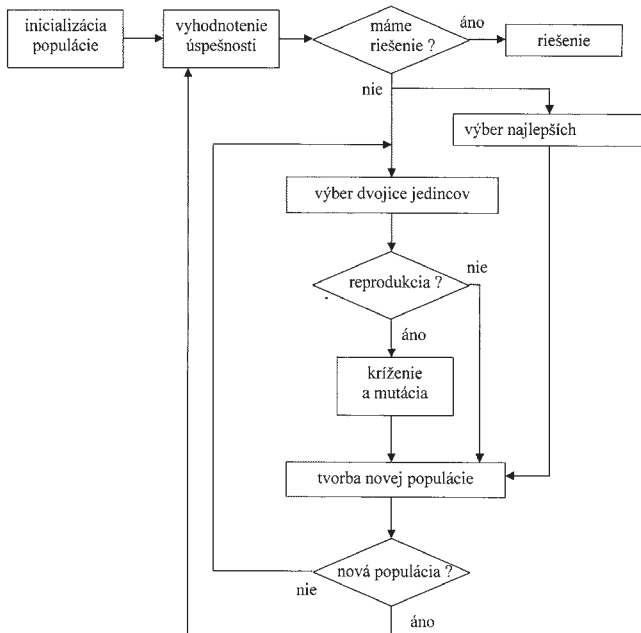
$$v = (x, \delta)$$

kde x predstavuje vektor hľadaných parametrov a δ je vektor smerodajných odchýlok zmien vektora x . Pôvodný algoritmus obsahoval len operáciu mutácie:

$$x^{t+1} = x^t + N(0, \delta)$$

kde N je vektor nezávislých náhodných čísel s bežným rozdelením pravdepodobnosti. Premenná δ ovplyvňuje šírku Gaussovej krivky a nazýva sa veľkosť kroku mutácie. Do ďalšej generácie postupí jedinec s lepšou hodnotou účelovej funkcie.





Obr.1 Schéma fungovania genetického algoritmu

Veľkosť kroku mutácie bola v prvej verzii konštantná. Pozorovaním sa zistilo, že len 1/5 mutácií vedie k lepším jedincom. Preto sa zaviedla takáto korekcia parametra δ :

$$\delta^{t+1} = \delta^t c \quad \text{ak } \phi < 1/5$$

$$\delta^{t+1} = \delta^t / c \quad \text{ak } \phi > 1/5$$

$$\delta^{t+1} = \delta^t c \quad \text{ak } \phi = 1/5$$

Korekcia sa vykonáva po stanovenom počte generácií, počas ktorých sa vykonáva vyhodnotenie miery úspešnosti mutácií ϕ . Konštanta $0,817 < c < 1$ bola zistená empiricky. Korekcia veľkosti kroku zabezpečuje zväčšovanie kroku pri úspešnej mutácii a znižovanie kroku pri neúspešnej mutácii.

Ďalším zlepšením (algoritmus s jedným jedincom mal tendenciu uviaznuť v lokálnom extréme) je zväčšenie počtu jedincov v populácii a zavedenie operácie kríženia.

Diferenciálna evolúcia (DE)

Algoritmus DE [6] je založený na „súperení“ potomkov s rodičmi. Princíp algoritmu:

Ak máme populáciu s N jedincami.

1. Pre jedinca x_i vygenerujeme vektor v :

$$v = x_1 + F(x_2 - x_3), \text{ kde } x_1, x_2 \text{ a } x_3 \text{ sú náhodné jedince z populácie rôzne od jedinca } x_i \text{ a } 0 < F < 2 \text{ je konštanta.}$$

2. Vytvoríme skúšobný reťazec u :

$$u_j = v_j \quad \text{ak } \rho < CR$$

$$u_j = x_{ij} \quad \text{ak } \rho \geq CR$$

kde j je poradové číslo prvku vektorov u a v , CR je parameter pravdepodobnosti kríženia a ρ je náhodné číslo. CR a ρ je z intervalu $(0;1)$.

3. Vygenerujeme i -tého jedinca novej populácie (do novej populácie postúpi úspešnejší z dvojice jedincov u a x_i):

$$x_i^{t+1} = u \quad \text{ak } f(u) < f(x_i^t)$$

$$x_i^{t+1} = x_i^t \quad \text{ak } f(u) \geq f(x_i^t)$$

kde t je číslo aktuálnej populácie. Uvedený vzťah platí pre prípad hľadania minima.

4. Pre všetky jedince populácie opakujeme body 1 až 3.

Umelý imunitný systém (UIS)

Algoritmus napodobňuje činnosť imunitného systému živých organizmov. Princíp fungovania je takýto [7]:

1. náhodné vytvorenie populácie jedincov a vyhodnotenie ich účelovej funkcie (úspešnosti riešenia),
2. test ukončovacích podmienok,
3. vyhodnotenie vektora účelovej funkcie celej populácie,
4. vygenerovanie klonov každého jedinca populácie (vytvorenie rozšírenej populácie),
5. mutácia každého klonu (nepriamo úmerne úspešnosti jedinca, menej úspešní jedinci sú zmenení viac ako úspešnejší) podľa vzťahu:

$$x^{t+1} = x^t + \frac{N}{\beta} e^{-f_n(x)}$$

kde x^{t+1} je mutovaný jedinec rodiča x^t , $0 < N < 1$ je náhodné číslo s bežným rozdelením pravdepodobnosti, β je konštanta prispôbujúca veľkosť mutácie danej populácii a $f_n(x)$ je normalizovaná fitness funkcia:

$$f_n(x) = \frac{f(x) - \min f(x)}{\max f(x) - \min f(x)}$$

kde $f(x)$ je fitness v prípade maximalizácie,

6. výber najúspešnejšieho zmutovaného klonu, ktorým sa nahradí rodičovský jedinec,
7. vyhodnotenie vektora účelovej funkcie celej populácie a porovnanie s hodnotou v bode 3, pri zlepšení pokračujeme bodom 4,
8. určenie vzájomnej afinity (euklidovskej vzdialenosti) jedincov rozšírenej populácie a odstránenie jedincov s malou (vhodne stanovenou) afinitou voči najlepším jedincom,
9. vytvorenie novej populácie z náhodných jedincov pôvodnej populácie a jedincov rozšírenej populácie, pokračovanie bodom 2.

Samoorganizujúci sa

migračný algoritmus (SOMA)

Základná myšlienka pochádza z algoritmu DE. SOMA [8] používa upravené operácie mutácie a kríženia. Mutácia je premenovaná na tzv. pertubáciu. Pri pohybe jedincov v priestore možných riešení je ich pohyb rušený (tzn. pertubovaný) a nie mutovaný. Kríženie je nahradené putovaním jedincov po danej n -rozmernej ploche. Pri putovaní jedinca priestorom možných riešení dochádza k mapovaniu priestoru po diskretných skokoch, čím sa generuje sekvencia potomkov, z ktorej prežije len ten najlepší. Pri pohybe je pre každého jedinca uchovaná súradnica pozície, na ktorej našiel najlepšie riešenie, ktoré postupuje do ďalších migračných kôl. Existuje niekoľko verzií (stratégií) algoritmu, napríklad All-to-one, kde ostatní jedinci putujú k najlepšímu jedincovi populácie. Pri verzii All-to-all putuje každý jedinec ku každému. Ďalšou zaujímavou verziou je All-to-onerand, kde ostatní jedinci populácie putujú k jednému náhodne vybranému (nie najlepšiemu) jedincovi. Použitím vhodnej stratégie a voľbou vhodných parametrov algoritmu dokážeme výrazne urýchliť konvergenciu k hľadanému riešeniu.

Particle Swarm Optimization (PSO)

Ide o veľmi jednoduchý a efektívny algoritmus. Inšpirácia pre vznik tohto algoritmu pochádza zo správania sa jedincov umiestnených v skupine [9]. Podobné správanie sa pozorovalo pri spoložití skupiny vtákov v prírode alebo pri skupine ľudí riešiacich sociálne problémy v spoločnosti.

Každé riešenie (častica) je reprezentované reťazcom parametrov. Začiatková populácia obsahuje častice náhodne umiestnené v celom prehľadávanom priestore riešení. Každá častica si pamätá svoje najlepšie riešenie p_{naj} . Ďalej si pamätá najlepšie riešenie populácie g_{naj} . V každej generácii sa každá častica premiestni na náhodne určené miesto:

$$p_i = p_i + 2 \cdot rand \cdot p_{inaj} - p_i + 2 \cdot rand \cdot g_{inaj} - p_i$$

kde $rand$ je náhodné číslo z intervalu $(0,1)$ a p_i je i -ty parameter jedinca p .



Genetické programovanie

Všetky dosiaľ uvedené metódy EV sa používajú na hľadanie optimálneho riešenia pri problémoch s pevne danou štruktúrou (modelov alebo objektov). Genetické programovanie [10] sa používa na hľadanie optimálnej štruktúry systémov (prvkov a väzieb medzi nimi). Na reprezentáciu optimalizovaných štruktúr sa používajú stromové štruktúry, ktoré sa prepisujú do reťazcov pomocou Readovho kódu. Takéto reťazce potom predstavujú jednotlivých jedincov populácie. Samotný beh algoritmu je podobný genetickému algoritmu. Používajú sa upravené operácie mutácie a kríženia.

Genetické programovanie sa používa napríklad pri symbolovej regresii, automatizovanom generovaní programov, návrhu logických automatov, grafov, schém a konštrukcií. Zaujímavým príkladom použitia je návrh energeticky úsporného domu alebo programu na riešenie šachu.

Záver

GA je jednoducho implementovateľný a rozšírený nástroj dosahujúci výborné výsledky pri riešení širokého spektra optimalizačných úloh.

ES sú vhodné na optimalizáciu bežných (číselných) funkcií. Algoritmus ES dokáže samostatne meniť riadiace parametre, čím sa sám prispôbuje (adaptuje) riešenému problému.

DE je jednoduchý a výkonný algoritmus použiteľný (vzhľadom na rýchlu konvergenciu) na optimalizáciu bežných funkcií. Nevýhodou je možnosť uviaznutia v lokálnu extrémnu pri multimodálnych funkciách (funkciách s viacerými lokálnymi extrémami).

UIS je zaujímavým, ale výpočtovo náročným algoritmom (väčší počet vyhodnotení fitness funkcie oproti iným metódam EV a náročný výpočet afinít jedincov populácie).

SOMA algoritmus predstavuje výkonný a rýchly prostriedok na riešenie optimalizačných problémov a pri mnohých úlohách vykazuje najlepšie vlastnosti (doba riešenia, schopnosť nájsť globálny extrém).

PSO je ukázkou jednoduchého, elegantného a silného algoritmu využiteľného na numerickú optimalizáciu.

Uvedené metódy sú silnými nástrojmi na riešenie optimalizačných problémov. Nevýhodou je nemožnosť exaktne dokázať konvergenciu riešenia, avšak praktické skúsenosti ukazujú, že metódy EV takmer vždy dokážu nájsť globálne optimum. Silnou stránkou EV je ich prirodzený paralelizmus (samostatne pôsobiace jedince populácie) umožňujúci distribuovane riešiť veľmi zložitú a rozsiahle úlohy.

S narastajúcim výpočtovým výkonom dnešných počítačov sa ďalej rozrástá aj množina úloh riešiteľných metódami EV.

Literatúra

- [1] SEKAJ, I.: Evolučné výpočty a ich použitie pri riešení praktických problémov. STU 2005.
- [2] GOLDBERG, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Inc., 1989.
- [3] KVASNIČKA, V., POSPÍCHAL, J., TIŇO, P.: Evolučné algoritmy. STU 2000.
- [4] RECHENBERG, I.: Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog Stuttgart, 1973.
- [5] SCHWEFEL, H. P.: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Birkhauser, Basel, 1977.

[6] LAMPINEN, J.: Solving Problems Subject to Multiple Non-linear Constraints by the Differential Evolution. Zborník konferencie Mendel 2001, 6. – 8. 6. 2001, Brno, Česká republika.

[7] DE CASTRO, L. N., TIMMIS, J.: An Artificial Immune Network for Multimodal Function Optimization. Zborník konferencie Congress on Evolutionary Computation CEC 2002, IEEE Press, 2002, p. 699 – 704.

[8] ZELINKA, I., LAMPINEN: SOMA – Self-Organizing Migrating Algorithm. Mendel 2000, 6th International Conference on Soft Computing, Brno, Česká republika.

[9] ZHANG, L., ZHOU, C., LIU, X., MA, Z., MA, M., LIANG, Y.: Solving multi objective optimization problems using particle swarm optimization. Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), Canbella, Australia, 2003, p. 2 400 – 2 405.

[10] KOZA, J. R.: Genetic Programming. MIT Press, Cambridge, MA, 1992.

Ing. Jozef Štefanka
doc. Ing. Peter Hubinský, PhD.

43

Katedra automatizácie a regulácie
Ilkovičova 3, 812 19 Bratislava
e-mail: jozef.stefanka@stuba.sk