FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA XVIII.

Kolozsvár, 2013. március 21–22.

OPTIMÁLÓ ALGORITMUSOK ÉS TESZTFÜGGVÉNYEK FEJLESZTÉSE

BARCSÁK Csaba, MARCSÁK Gábor, JÁRMAI Károly

Abstract

The aim of this research is to develop an advanced web application, which is a standalone collection of modern heuristic optimization algorithms. The algorithms are also a target to improve their efficiency and reliability. There have been test functions developed for comparison of different optimization algorithms.

Key words:

optimization techniques, test functions

Összefoglalás

A kutatás célja hogy kidolgozzon egy fejlett webes alkalmazást, amely egy önálló gyűjteménye a modern heurisztikus optimáló algoritmusoknak. Az algoritmusok fejlesztése is cél, hogy javítsuk a hatékonyságukat és a megbízhatóságukat. Kidolgozásra kerültek teszt függvények, hogy a különböző optimalizálási algoritmusok összehasonlíthatóak legyenek.

Kulcsszavak:

optimáló algoritmusok, tesztfüggvények

1. Bevezetés

Az optimáló algoritmusok a mérnöki munka számos területén alkalmazásra kerülnek. Egyik fontos területe a mérnöki objektum (mérnöki szerkezetek, gépszerkezetek, stb.), másik a mérnöki rendszer (logisztikai, energetikai, stb.) optimalizálása. Az optimalizálás általában azt jelenti, hogy a célfüggvényt (anyag, költség, élettartam, stb.) minimáljuk, vagy maximáljuk a változók (méretek, darabszám, anyagminőség, kapacitás, stb.) értékének módosításával úgy, hogy feleljen meg az objektum, vagy rendszer az előírt feltételeknek (feszültség, alakváltozás, csillapítás, kapacitás, teherbírás, stb.). Nagyon sok optimáló algoritmus létezik, ebből most csak néhányat említünk. Az újabb optimáló módszerek heurisztikusak.

2. Web alkalmazás fejlesztése JavaFX környezetben optimalizálási problémák megoldására

A heurisztikus algoritmusok gyűjteményéhez mindenekelőtt egy keretrendszert kellett kifejleszteni, mely tárolja, rendszerez azokat. A program tervezésekor több fontos szempont lett figyelembe véve.

Az Interneten bárki számára könnyen és gyorsan hozzáférhető web alkalmazás legyen, amit nem kell külön telepíteni, böngészőből használható.

A legtöbb heurisztikus algoritmus implementációja sokkal egyszerűbb modern, objektum orientált paradigmát követő programozási nyelvek segítségével. Annak ellenére, hogy a nem informált kereső eljárásokhoz viszonyítva kevesebb számítás árán jutnak eredményre, bonyolult problémák esetén fontos a választott technológia futási sebessége.

- Alapvető követelmény a könnyen kezelhető, informatív felhasználói felület.
- Fontos volt továbbá, hogy az algoritmusok gyűjteményét könnyen lehessen bővíteni. A keretrendszer ennek megfelelően egyfajta moduláris struktúrát követ.
- A fent vázolt szempontok figyelembevételével a viszonylag újnak számító JavaFX technológiát alkalmaztuk a megvalósításhoz. Egyik legfontosabb előnye a Java technológiából eredő platformfüggetlenség, mely lehetővé teszi használatát Windows, Mac OS X és Linux rendszereken. A közeljövőben várhatóan az Android és iOS mobil operációs rendszerek is támogatottá válnak, ezzel lehetővé téve a technológia használatát okostelefonokon és táblagépeken. A JavaFX programok asztali alkalmazásként és web alkalmazásként egyaránt képesek működni, a két mód között pedig nincs szignifikáns sebességbeli különbség, mivel futtatásukról ugyanaz a Java Virtuális Gép (JVM) gondoskodik.

3. Heurisztikus algoritmusok

Újabban előtérbe kerültek azok a heurisztikus algoritmusok, melyek működési elve valamilyen természeti jelenségen alapul, például Darwin evolúciós elméletén. A különböző élőlények adott helyzetben való viselkedésmintájából merítve jó eredményeket lehet elérni, ennek analógiájára dolgozták ki többek között a madár és halrajok mozgásán (Particle Swarm Optimization **PSO**), a baktérium (Bacterial Foraging **BFO**), a differenciális evolúció (**DE**) és hangyakolóniák (Ant Colony) táplálkozásán, vagy a szentjánosbogarak fényjelenségein (Firefly) alapuló módszereket.

4. Particle Swarm algoritmus

A Particle Swarm Optimization (részcseke-csoport) algoritmust 1995-ben dolgozták ki [1]. Működését a madár és halrajok mozgása inspirálta. A nagyszámú egyed mozgásában egyfajta rendezettség figyelhető meg. Érzékelik egymás helyzetét, bizonyos mértékben pedig emlékeznek a korábbi pozíciókra. A keresés adott számú részecske létrehozásával kezdődik, amik véletlenszerű kiindulási pontokban helyezkednek el. A részecskék tárolják aktuális és addigi legjobb pozíciójukat, valamint a részecske csoport addigi legjobb pozícióját. Hogy az adott pozíció mennyire jó az adott keresés szempontjából, mindig egy fitness függvénnyel határozható meg. Az iterálás addig folytatódik, míg nem teljesül valamilyen leállási feltétel.

5. A PSO algoritmus bővítése gradiens becsléssel

A standard algoritmus a célfüggvény kiszámított értékein kívül más információkkal nem rendelkezik a függvényről, pedig több esetben hasznos lenne a függvény egyes lokális tulajdonságainak ismerete, mivel ezen információk nem kötődnek a véletlenhez és általuk hatékonyabbá tehetnénk az eljárást. Az egyik ilyen lokális tulajdonság a gradiens, melyet, mivel csak diszkrét pontokban rendelkezünk mintákkal, becsülnünk kell. A véges differencia alapú megoldások gyors és hatékony megoldást nyújtanak a gradiens becslésére diszkrét adatok esetén. Mindegyik véges differencia alapú séma a differenciálendő függvény Taylor-sorából indul ki, amely egy egydimenziós függvény $f(x)$ esetében a következőképp írható fel:

$$f(x_0 + h) \approx f(x_0) + \frac{f'(x_0)}{1!}h + \frac{f''(x_0)}{2!}h^2 + \dots = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}h^n \quad (1)$$

Abban az esetben, ha a Taylor-sort abbahagyjuk a második tagnál és kifejezzük a deriváltat, a következő formula adódik:

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h} \quad (2)$$

Ezt a képletet az irodalom backward difference (hátravett differenciák) becslés néven említi. Ez a megoldás egyszerű, gyorsan kiszámolható, hátránya viszont az, hogy kevésbé pontos. Léteznek összetettebb gradienst becselő megoldások az irodalomban, de azok számításigényesebbek az előbbieknél leírt eljárásoknál, és kettőnél több mintavételi pontot igényelnek. Egy adott részecske mozgása során adott pillanatig érintett pontokhoz tartozó függvény értékeket felhasználhatjuk az adott pontokban vett gradiensek becslésére. Az általunk implementált algoritmusban a backward difference módszer került beépítésre. A gradienseket az elkészült algoritmus a részecskék sebességének beállítására használja, ezáltal a részecskék a függvény értelmezési tartományának egyes intervallumaiban gyorsabban, míg más intervallumokban lassabban mozognak. Minden egyes részecske a pozíció és sebesség adatok mellett tárolja, hogy a már érintett pontokban hány egymás utáni esetben talált pozitív előjelű gradienst. Az algoritmus a szimulációk során összességében jobban teljesített, mint a standard változat.

6. Az algoritmusok tesztelése

Az algoritmusok teszteléséhez a Rastrigin's tesztfüggvény kétváltozós változatát használtuk, mely az 1. ábrán látható [2]:

A tesztfüggvénynek az $x=0.0$, $y=0.0$ helyen van globális minimuma, melynek értéke 0.0. A teszt célja a következő kérdések megválaszolása volt:

- Milyen hatékonysággal találják meg az algoritmusok az optimális megoldást?

- Ha mindhárom algoritmus esetén azonos számú populációt adok meg, valamint leállási feltételként is

1. ábra. A kétváltozós Rastrigin's függvény ugyanazon iterációs korlátot kapják, akkor megállapítható, hogy melyik algoritmus működik a leggyorsabban. A leggyorsabb működés természetesen nem feltétlenül jelent egyúttal leghatékonyabbat is.

- Fejlettebb leállási feltétel definiálásával milyen mértékben növelhető a hatékonyság?

1. táblázat. Teszteredmények

	Idő [ms]	x	y	Fitnesz
PSO	26	$-7,92 \cdot 10^{-10}$	$2,27 \cdot 10^{-11}$	$1,40 \cdot 10^{-9}$
PSO*	18	$3,82 \cdot 10^{-10}$	$-6,94 \cdot 10^{-10}$	$8,25 \cdot 10^{-9}$
DE	30	$5,17 \cdot 10^{-6}$	$-1,25 \cdot 10^{-7}$	$4,43 \cdot 10^{-6}$
BFO	43	$3,41 \cdot 10^{-6}$	$2,18 \cdot 10^{-5}$	$9,63 \cdot 10^{-5}$

7. Összefoglalás

Kidolgozásra került egy JavaFX bázisú keretrendszer, mely tárolja, rendszerezi a heurisztikus algoritmusokat. A beépítésre került, illetve kerülő módszerek: részecskecsoporthoz módszer (Particle Swarm Optimization PSO), a differenciális evolúció (DE), a baktérium (Bacterial Foraging BFO) és a hangyakolóniák (Ant Colony) táplálkozásán, illetve a szentjánosbogarak fényjelenségein (Firefly) alapuló módszerek. Kifejlesztésre került egy új megközelítés a PSO algoritmus gyorsítására, mely a mintavételi pontokban véges differencia alapú becsléssel határozza meg a gradienst és ezt használja fel a részecskék sebességének megváltoztatására, a módszer nem igényel több mintavételezést, mint az eredeti eljárás és nem függ a rendszer kezdőállapotától. A módszer hatékonysága függ attól, hogy mekkora egymást követő pozitív gradiens szám esetén változtatjuk a részecske sebességét, valamint a sebességet milyen mértékben változtatjuk. Minden célfüggvény esetén más és más beállítások lehetnek hatékonyak. Tesztfeladatokon elvégeztünk összehasonlításokat a módszerek között.

Köszönetnyilvánítás

A bemutatott kutató munka a TÁMOP-4.2.1.B-10/2/KONV-2010-0001 jelű projekt részeként az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával, továbbá az OTKA 75678 számú projekt támogatásával valósult meg.

Irodalom

- [1] Kennedy, J., Eberhart, R.: *Particle swarm optimization*, IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948, 1995.
- [2] Molga, M., Smutnicki, C.: *Test functions for optimization needs*, 2005, <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>

Barcsák Csaba, BSc. hallgató, **Marcsák Gábor**, MSc. hallgató, **Dr. Jármái Károly**, DSc, egy. tanár Miskolci Egyetem, Gépészmérnöki és Informatikai Kar, Anyagmozgatási és Logisztikai Tanszék, H-3515 Miskolc-Egyetemváros, Magyarország, Telefon: +36 46 565 111/20-28, altjar@uni-miskolc.hu