FOGALMAK ÉS MÓDSZEREK

ANONIMUS

Egy új optimálási algoritmus*

Eljárást mutatunk be egy speciális problémafajta megoldására.

1. Bevezetés

Az eddig ismert módszerek, amelyek megoldják azt az optimálási problémát, hogy egy f függvény maximumát egy S tartományban megtaláljuk, a következő hiányosságokat mutatják:

(1) Gyakran különleges követelményeket (pl. differenciálhatóság vagy konkávitás) támasztanak a függvénnyel szemben.

(2) A tartománytól rendszerint megkövetelik, hogy összefüggő, vagy éppenséggel konvex legyen.

(3) Az eljárások gyakran komplikáltak, ezért nehéz megérteni és számítógépre programozni őket.

Ezért én és alább említendő munkatársaim olyan eljárás kifejlesztésére törekedtünk, amelyik nincsen ezeknek a kötöttségeknek alávetve.

2. Matematikai alapok

Legyen S az U szeparábilis metrikus térnek egy részhalmaza. Mint ismeretes, van az U pontjaiból álló olyan $\{P_0, P_1, \ldots\}$ sorozat, amelyik U-ban sűrű.

Ha S rendelkezik a következő tulajdonsággal:

(A) S belsejének zárt burka tartalmazza S-et,

akkor $\{P_0, P_1, \ldots\}$ sűrű S-ben is. Továbbá ha még f folytonos akkor

$$\begin{aligned}\operatorname{Sup}\{f(x) : x \in S\} &= \operatorname{Sup}\{f(x) : x \in S \cap \{P_0, P_1, \ldots\}\} = \\ &= \operatorname{Sup}_k\{f(P_k) : P_k \in S\}.\end{aligned}$$

* A tanulmány eredetileg angol nyelven jelent meg a *Mathematical Programming* 1972. évi 3. kötetében, 124—128. oldalakon, a szerkesztő következő megjegyzésével:
A kéziratot, szakadozottan és szamárfülekkel, Philipp Wolfe küldte el a *Mathematical Programming*-nak, az ő kísérőleveléből idézünk: „Sok olyan cikket lektoráltam már, amelyek optimálási eljárásokat javasoltak, anélkül, hogy hatékonyságukat megvizsgálták volna. Időm kímélésére egy mondatot idézek: »Ez az algoritmus minden olyan problémát megold, amelynek megoldására az ön módszere képes, éspedig, mint a rendelkezésre álló bizonyítékok mutatják, ugyanolyan jól.« Ezért javasolom a cikk megjelentetését . . . és remélem, hogy a szerző felfedi majd kilétét, hogy megkapja . . . , amit bőven megérdemel.”
A SZIGMA szerkesztője köszönetet mond a *Mathematical Programming Society*-nak a közlési engedélyért és *Martos Bélának*, aki a cikkre figyelmét felhívta.

Ezáltal f-nek a megszámlálhatatlan számosságú S tartományban való maximálását egy olyan maximálási feladatra vezettük vissza, amelynél a tartomány sokkal kisebb lett.

3. Az algoritmus

Legyen f, S és a $\{P_k\}$ sorozat a 2. szakaszban mondottak szerint definiálva. Kezdetben legyen $j = 0$, $k = 0$, $x_0 = P_0$.

A k-adik lépés kezdetén adva van $j \leq k$, x_j és P_k.

(i) Ha P_k nem eleme S-nek, tégy k helyébe $k + 1$-et és menj (i)-re. Egyébként menj (ii)-re.

(ii) $[P_k \in S]$. Ha $f(P_k) > f(x_j)$, legyen $x_{j+1} = P_k$, tégy k helyébe $k + 1$-et és menj (i)-re. Egyébként tégy k helyébe $k + 1$-et és ment (i)-re.

(Ebben az alakjában az eljárást az irodalom [1] ,,Wolfe Univerzális Algoritmusa'' néven ismeri.)

4. Az algoritmus konvergenciája

Tétel. A 2. fejezet feltételei mellett az $\{f(x_j)\}$ sorozat az f-nek S-beli supremumához konvergál éspedig monoton növekvően.

Bizonyítás. A tétel egyenesen következik egy klasszikus eredményből [2]. Újabban egy fejlettebb módszereket alkalmazó bizonyítást is találtak [3].

5. Az algoritmus gépi programja

Hogy az algoritmust az automatikus számolás szempontjából kényelmesebbé tegyük, úgy döntöttünk, hogy az U teret tovább specifikáljuk a valós számok négyzetesen összegezhető sorozatainak Hilbert terévé. Az így előálló ,,Szeparábilis Hilbert Iterációs Módszer'' lehetővé teszi, hogy a $\{P_k\}$ sorozatot belülről generáljuk ahelyett, hogy egy külső forrásból olvasnók be. A tagokat blokkokban generáljuk, a blokkok indexe $N = 1, 2, \ldots$ és az N-edik blokk M-edik eleme az $[a_1, a_2, \ldots, a_N, 0, 0, 0, \ldots]$ vektor, ahol $[a_1, a_2, \ldots, a_N]$ a lexikográfikus sorrend szerinti M-edik csúcspont az origó köré írt N élhosszúságú N-dimenziós kocka $1/N$ élhosszúságú kockákra való felosztásában. (Ily módon P_k csak véges számú nemzéró elemet tartalmaz, ami számítástechnikailag előnyös.) Persze, ha a probléma véges D dimenziójú, akkor P_k-nak minden a D-edik után következő komponense elhagyható. A program ALP/360-ban írva a következő:

$$\nabla\ SZEPAR\acute{A}BHILBERTITER\acute{A}CI\acute{O}$$

[1] $N \leftarrow 0$

[2] $X \leftarrow (N \leftarrow N + 2 + M \leftarrow {}^{-}1)\varrho 0$

[3] $S : \rightarrow (S - M \geq {}^{-}1 + (1 + N * 2) * N),\ X \leftarrow (X \times \sim V) + P \times V \leftarrow$
$\leftarrow (G\,P) \wedge (F\,X) < F\,P \leftarrow$
$\leftarrow ({}^{-}0.5 \times N) + ((N \varrho 1 + N * 2) \top M \leftarrow M + 1) : N$

A felhasználónak csak két függvényt, F-et és G-t, kell specifikálnia. Az F függvénynek meg kell adnia az f célfüggvény értékét, a G függvénynek pedig 1 vagy 0 értéket kell felvennie, aszerint, hogy argumentuma S-hez tartozik-e vagy sem.

Fenti programunk biztosítja ugyan az új algoritmus működését, de hatékonyságát tovább fokozhatjuk, ha kihasználjuk párhuzamos számításokra való alkalmasságát. Ennek megfelelően az eljárást újraprogramoztuk, úgy hogy futtatható legyen T számú párhuzamos aritmetikai regiszterrel rendelkező számítógépen. (Ez a kód sajnos túl hosszú semhogy itt reprodukálni lehessen.) Megmutatható, hogy T megfelelő választásával a futási időt számottevő arányban csökkenteni lehet.

6. A módszer további finomítása és kiterjesztése

Kutatás folyik arra, hogy az algoritmust még hatékonyabbá tegyük optimálási feladatok megoldásában. Bebizonyítottuk, hogy ha S megszámlálható halmaz, akkor elegendő $\{P_k\} \supseteq S$-et választani ahhoz, hogy a 4. fejezetbeli konvergencia tétel érvényes maradjon, és így az algoritmust kiterjeszthetjük olyan esetekre is, amelyben a tétel feltételei nincsenek kielégítve, például arra az esetre, amikor S az E^D tér azon vektoraiból áll, amelynek komponensei egész értékűek. (Ezt ,,integer programozási problémának'' nevezik (lásd [4].) Meg kell jegyezni, hogy az 5. fejezetben programozott módon éppen ezt tettük. Ilyesféle kiterjesztéseket további cikkekben fogunk tárgyalni.

Lekötelezettje vagyok annak a munkatársamnak, aki rámutatott az algoritmus egy lehetséges meggyorsítására. Legtöbb feladatnál az a helyzet, hogy a $\{P_0, P_2, P_4, \ldots\}$ sorozat sűrű a $\{P_0, P_1, P_2, \ldots\}$ sorozatban, amelyből következik, hogy az előbbi sűrű U-ban is. Tehát, ha a 2. fejezetbeli algoritmusban a $\{P_k\}$ sorozatot a $\{P_{2k}\}$ sorozattal helyettesítjük, a konvergenciatétel még mindig áll, de az eljárásban a lépések száma az eredetihez képest 50%-kal csökken. Ez az ötlet további gyorsítási módszereket sugallt és ezek lényeges plusz megtakarítással kecsegtetnek. Kellő időben beszámolunk majd róluk az irodalomban.

7. Az algoritmus hatékonyságának vizsgálata

Mivel az optimálási és egyéb feladatok megoldására sok eljárást ismerünk, gondosan összehasonlítottuk az új algoritmust a korábbiakkal. Mivel T regiszteres parallel számoló berendezés nem állt rendelkezésünkre, felkértük fentebb már említett munkatársunkat, hogy végezze el egy ilyen komputeren való számolás kézi szimulálását. Ő igen megnyugtató előrehaladásról számolt be [4] az optimálási algoritmusok hatékonysági vizsgálatára szolgáló probléma megoldása során.

Egy ettől független hatékonysági vizsgálatként összehasonlítottuk a mi algoritmusunkat 43 más publikált optimálási algoritmussal minden jelentősebb képességük szempontjából: