A matematikai modellezés hatása nemlineáris optimalizálási feladatok megoldásának hatékonyságára

Ph.D. értekezés tézisei
Informatika Doktori Iskola

Dobjánné Antal Elvira

Témavezetők:
Dr. Csendes Tibor és Dr. Vinkó Tamás

Számítógépes Optimalizálás Tanszék
Szegedi Tudományegyetem
2017
Szeged

Bevezetés

Az optimalizálási problémák matematikai modellezése során hozott döntések jelentős mértékben befolyásolhatják az alkalmazott megoldó hatékonyságát, a feladat komplexitását. Egy optimalizálási feladat átfogalmazása azonban sokszor az egyéni intuíción túlmutató automatikus módszerekkel is lehetséges. A lineáris programozási feladatok szimplex módszer számára legelőnyösebb formájának felírása például már az 1970-es években is foglalkoztatta a matematikusokat. Ezek a korai eredmények mára az AMPL előfeldolgozóba épülve szinte észrevétlenül hasznosulnak.

Számos kutatás foglalkozik napjainkban a (vegyes-)egészértékű programozási feladatok kedvezőbb, ill. adott körülmények között megoldható alakba történő átfogalmazásának lehetőségeivel, habár ezek az átalakítások általában bizonyos feltételek relaxációjával és a feladat dimenziójának növelésével járnak együtt. Meglepően csekély számú publikáció koncentrál azonban a nemlineáris optimalizálási feladatok ekvivalens átalakításait eredményező, automatizálható szimbolikus eljárásokra. Erre a területre fókuszál dolgozatom első fele.

A fellelhető szakirodalom elméleti eredményeinek ismeretében feltételezhető volt, hogy napjaink nagytudású számítógépes algebra rendszerei alkalmasak egy szimbolikus, tehát a számítási pontosságot garantáló, nemlineáris optimalizálási feladatok ekvivalens átírásait megadó alkalmazás létrehozására, amely lehetőséget ad a korábban publikált algoritmusok integrálására, tesztelésére, továbbá inspirációt nyújthat ezek továbbfejlesztéséhez. Ezzel a céllal elkészült egy, a feltétel nélküli nemlineáris optimalizálási feladatokon nemlineáris koordinátatranszformációkat végrehajtó Maple program. Alapos tesztelés során felderítettem azokat a specifikus területeket, amelyek egy ilyen algoritmus implementálása során kritikusak lehetnek, és megállapítottam, hogy a Maple rendszer néhány dokumentálatlan hibája és általános felhasználásra tervezett, a konkrét célnak nem megfelelő minőségben megvalósított funkciója komoly akadályokat állít a bővítés elé.

Tanulmányoztam a rendelkezésre álló kereskedelmi és szabad szoftveres alternatívákat, és a programozó számára nyújtott rugalmassága, erős fejlesztői háttere és a legújabb matematikai eredményeket felvonultató saját, széles körű funkcionalitása miatt a Mathematica programot találtam a legjobb alapnak a további fejlesztések számára. Ez alapján korábbi programunkat további funkciókkal bővítve átültettem Mathematica alá, és empirikus úton bizonyítottam, hogy előfeldolgozóként alkalmazva az általunk javasolt szimbolikus transzformációk hasznosak egy klasszikus (numerikus) heurisztikus megoldó számára.

Új elméleti eredmények születtek a párhuzamosan végrehajtható nemlineáris koordináta-transzformációkkal és a feladathoz adott feltételekkel kapcsolatban, továbbá készítettem egy online demonstrációs oldalt az eljárás bemutatására. Mindezen eredményeket egy hatástényezős [2] és egy további lektorált folyóiratcikkben [1] publikáltam.

Az értekezés második felében egy konkrét feladathoz, az elosztott tartalom-megosztó rendszerekben felmerülő méltányos (max-min fair) sávszélesség-kiosztás problémájához kapcsolódó modellezési kérdéseket vizsgáltam. A feladatra sikerült egy új, egzakt matematikai programozási modellt adnom, és az ezen alapuló algoritmus helyességét elméleti úton bizonyítanom. Elkészült az algoritmus AMPL nyelvű megvalósítása, aminek a teljesítményét numerikus tesztek révén összehasonlítottam a feladat megoldására korábban létező programmal. Az eredmény több tekintetben pozitív. Ezek az eredmények szintén elfogadásra kerültek egy hatástényezős folyóiratcikkben történő publikálásra [3].

Az implementáció során, részben az elméleti eredményekhez kapcsolódóan, számos modellezési ötlet merült fel, ezért az értekezésben egy külön fejezet foglalkozik ezen technikák hatáselemzésével. A max-min méltányos erőforrás-kiosztást előállító modell tizenkét változatát hasonlítottam össze egy kiterjedt numerikus tesztelés során, két professzionális megoldó és huszonhét nagyméretű tesztfeladat bevonásával. Az eredményeket egy közlésre benyújtott kéziratban foglaltuk össze [7].

Nemlineáris koordináta-traszformációk nemlineáris optimalizálási problémák egyszerűsítésére

Előzmények

Tekintsük a következő, feltétel nélküli nemlineáris optimalizálási problémát:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

ahol $f(x) : \mathbb{R}^n \to \mathbb{R}$ egy képlettel (vagyis egy szimbolikus kifejezéssel) adott sima függvény. A „kifejezés” itt szimbólumok (konstansok, változók, operátorok, függvénynevek és zárójelek) egy véges, szintaktikailag helyes kombinációját jelöli. A népszerű számítógépes algebra rendszerekben, mint például a Maple-ben [8] és a Mathematicában [11] minden kifejezést pointerek egymásba ágyazott listájával tárolnak, ami tulajdonképpen irányított körmentes gráfok (directed acyclic graph, DAG [10]) megvalósításának felel meg.

Az egyszerűsítő módszer célja, hogy (1) olyan ekvivalens átírásait állítsa elő, amelyek kedvezőbbek a következő értelemben: kevesebb aritmetikai művelet szükséges a kiértékeléshez, a probléma dimenziója kisebb, vagy egyéb okból egyszerűbben, gyorsabban megoldható egy bizonyos megoldó számára. Ekvivalensen pedig azt értjük, hogy az eredeti és az átírt feladat optimumai között bijektív leképezés létezik.

Az (1) probléma egyszerűen megoldható, ha f unimodális, vagyis csupán egy vonzáskörzete, egy lokális minimum pontja van (és nincsenek helyi minimum pontok) a megadott $X \subseteq \mathbb{R}^n$ keresési térben. Egydimenziós függvényeknél ez a tulajdonság könnyen felismerhető, de magasabb dimenziókban általában nem triviális annak eldöntése, hogy unimodális-e egy függvény.

Csendes és Rapcsák eredménye értelmében egy függvény implicit unimodalitása egy változó transzformáció alakjában explicitté tehető:

1. TÉTEL [6] A folytonos $f(x)$ függvény akkor, és csakis akkor unimodális az n-dimenziós valós térben, ha létezik egy homeomorf $y = h(x)$ változó transzformáció, amelyre $f(x) = f\left(h^{-1}(y)\right) = y^T y + c$, ahol c egy valós konstans, és az origó $h(x)$ értékkészletében van.

A következő tétel elégséges feltételt mond ki arra vonatkozóan, hogy egy helyettesítés mikor egyszerűsíti a célfüggvényt:

2. TÉTEL [6] Ha $h(\mathbf{x})$ sima és szigorúan monoton az x_i függvényében, akkor a megfelelő transzformáció egyszerűsíti a függvényt abban a tekintetben, hogy a $h(\mathbf{x})$ minden $f(\mathbf{x})$-beli előfordulását egy új változóval helyettesítjük a $g(\mathbf{y})$ átalakítással kapott függvényben, miközben az $f(\mathbf{x})$ minden helyi minimum (maximum) helye a $g(\mathbf{y})$ függvény egy helyi minimum (maximum) helyévé transzformálódik.

Ha pedig az értékkészletre vonatkozó feltétel is teljesül, a helyettesítés bijektív leképezést létesít az eredeti és az átírt célfüggvény optimális megoldásai között:

3. TÉTEL [6] Ha $h(\mathbf{x})$ sima, szigorúan monoton az x_i függvényében, és az értékkészlete $\mathbb{R}$-rel egyenlő, akkor az átalakítással kapott $g(\mathbf{y})$ függvény minden $\mathbf{y}^*$ lokális minimum (maximum) helyére létezik olyan $\mathbf{x}^*$, hogy $\mathbf{y}^*$ az $\mathbf{x}^*$ transzformáltja, és $\mathbf{x}^*$ az $f(\mathbf{x})$ helyi minimum (maximum) helye.

Az utolsó tétel értelmében egy $g(\mathbf{y})$ célfüggvény ekvivalens az (1)-ben szereplő $f(\mathbf{x})$-szel, ha a következő átalakítással állítjuk elő:

- alkalmazunk egy helyettesítést $f(\mathbf{x})$-re:

  $$y_i := h(\mathbf{x})\,, \quad 1 \leqslant i \leqslant n,$$

  ahol $h(\mathbf{x})$ egy folytonos függvény $\mathbb{R}$ értékkészlettel, és szigorúan monoton legalább egy x_i változóban,

- átnevezzük a megmaradó változókat:

  $$y_j := x_j\,, \quad j = 1, \ldots, i-1, i+1, \ldots, n, \quad \text{és}$$

- elhagyjuk azokat az y_i változókat, amelyek az előálló célfüggvényben nem szerepelnek.

Azt mondjuk, hogy $h(\mathbf{x})$ *lefedi* x_i *változót* $f(\mathbf{x})$-ben, ha $h(\mathbf{x})$ az x_i minden $f(\mathbf{x})$-beli előfordulását jellemzi, vagyis x_i teljesen eltűnik $f(\mathbf{x})$-ből, ha $h(\mathbf{x})$-et y_i-vel helyettesítjük.

*Alkalmas helyettesítés*nek hívjuk az $y_i = h(\mathbf{x})$ helyettesítést, ha

- $h(\mathbf{x})$ sima, monoton legalább egy x_i változóban, és értékkészlete $\mathbb{R}$,
- $h(\mathbf{x})$ lefed legalább egy x_i változót, továbbá
- $y_i = h(\mathbf{x})$ nem egyszerű átnevezés, vagyis, $h(\mathbf{x}) \neq x_i$, $i = 1, \dots, n$.

Egy alkalmas $y_i = h(\mathbf{x})$ helyettesítés végrehajtása után $\mathbf{y}$ dimenziója nem nagyobb, mint $\mathbf{x}$-é. Kedvező esetben, ha $h(\mathbf{x})$ kettő, vagy több változót fed, az átalakítás után kapott feladat kevesebb változót tartalmaz, mint az eredeti. Másképp fogalmazva, az egyszerűsítő eljárás magában rejti a lehetőségét, hogy megmutassa, ha a modell az eredetinél kevesebb változóval is formalizálható. A redundáns változók felismerése igen hasznos lehet, és általában egyáltalán nem triviális. Ily módon az átírás eredménye hasznosítható lehet akkor is, ha a probléma nem hozható unimodális alakra.

Tekintsük például azt a feladatot, ahol az $f(x_1, x_2) = (x_1 + x_2)^2$ célfüggvény minimumát keressük. Ekvivalens probléma a $g(y_1) = y_1^2$ minimalizálása. Az eredeti x_1 és x_2 változók optimális értéke előállítható az $y_1 = x_1 + x_2$ összefüggés alapján, ami egy alkalmas helyettesítés. Ezen a módon ráadásul megoldható a végtelen számú optimumhely kezelése, ami a numerikus megoldók számára lehetetlen lenne.

Az egyszerűsítő eljárás egyik fő célja, hogy olyan alkalmas helyettesítéseket állítson elő, amelyek redundáns változók eliminációjával járnak. Csendes és Rapcsák vonatkozó eredménye a következő két állítás:

1. ÁLLÍTÁS [6] Ha x_i változó egy sima $f(\mathbf{x})$ függvényben mindenhol a $h(\mathbf{x})$ kifejezés részeként fordul elő, akkor a $\partial f(\mathbf{x})/\partial x_i$ parciális derivált felírható a $(\partial h(\mathbf{x})/\partial x_i)\, p(\mathbf{x})$ formában, ahol $p(\mathbf{x})$ folytonosan differenciálható.

2. ÁLLÍTÁS [6] Ha x_i és x_j változók egy sima $f(\mathbf{x})$ függvényben mindenhol a $h(\mathbf{x})$ kifejezés részeként fordulnak elő, akkor a $\partial f(\mathbf{x})/\partial x_i$ és $\partial f(\mathbf{x})/\partial x_j$ parciális deriváltak egyenként $(\partial h(\mathbf{x})/\partial x_i)\, p(\mathbf{x})$ és $\big(\partial h(\mathbf{x})/\partial x_j\big)\, q(\mathbf{x})$ alakú szorzattá alakíthatók, és teljesül a $p(\mathbf{x}) = q(\mathbf{x})$ egyenlőség.

Ha $\partial f(\mathbf{x})/\partial x_i$ nem alakítható szorzattá, akkor bármely, az x_i-ben monoton alkalmas helyettesítés lineárisan függ az x_i-től.

Ez az elmélet képezi az alapját annak a konstruktív szimbolikus algoritmusnak, amely implementációját Maple és Mathematica nyelven elkészítettük.

Az állítások azt sugallják, hogy számítsuk ki az $\partial f(\mathbf{x})/\partial x_i$ parciális deriváltakat minden x_i változóra, alakítsuk ezeket szorzattá, majd keressünk a szorzótényezők között alkalmas helyettesítéseket.

Természetesen a megfelelő $(\partial h(\mathbf{x})/\partial x_i)\,p(\mathbf{x})$ faktorizálás nem feltétlenül könnyű, és sok egyéb szorzat alak létezhet az 1. állításban említetteken felül. Mégis, a függvények egy bő osztályára létezik kanonikus alak. Például egy $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ alakú, n gyökű polinom, ahol a gyököket $x_1, \ldots, x_n$ jelöli, standard $a_n(x - x_1)(x - x_2)\cdots(x - x_n)$ szorzat alakja mindig létezik.

További kritika lehet, hogy ha a kívánt szorzat alak meghatározható is, nem triviális egy jó $h(\mathbf{x})$ helyettesítés megtalálása. Az említett feltétel tehát elégséges, de nem szükséges előfeltétele az egyszerűsítő átalakításoknak.

Saját eredmények

I. 1. *Az automatikus szimbolikus egyszerűsítő Maple és Mathematica implementációja.*

Két programozási környezetben, a Maple és Mathematica rendszerekben készült el a hivatkozott elméleti eredmények alapján kidolgozott eljárás megvalósítása, ami a feltétel nélküli nemlineáris optimalizálási feladat célfüggvénye képletének ismeretében automatikusan képes előállítani a matematikai modell hasznosnak tűnő átírásait [2, 1]. A javasolt átírások révén felismerhetővé válhat a modell rejtett redundanciája, lehetőség nyílhat a feladat dimenziójának csökkentésére.

Az értekezés 3.4 szakaszában megvizsgáltam a Maple implementáció kapcsán felmerülő főbb problémákat. Összehasonlítottam a Maple és Mathematica változatokat az előállított helyettesítések szempontjából. Itt különösen a saját, erre a célra készült tesztfeladataim vizsgálatára támaszkodtam. Kiderült, hogy a Mathematica rugalmas helyettesítő rutinja erre a célra megfelelőbb, és ez a megbízható intervallum aritmetikával kiegészítve a Maple-nél jobb alapot biztosít egy ilyen jellegű alkalmazás számára.

I. 2. *Számítógépes tesztelés standard globális optimalizálási tesztfeladatokon, és egy numerikus megoldó előfeldolgozójaként.*

Az értekezés 3.4 szakaszában összesen 45 jól ismert globális optimalizálási tesztfeladatot vizsgáltam, és ezek közül 8 esetben talált ekvivalens átírást a Mathematica program [1], ami a teszthalmaz 18%-a. Ha figyelembe vesszük, hogy nem ismert más módszer, amivel hasonló jellegű helyettesítések automatikusan előállíthatók lennének, ez az arány jelentősnek tekinthető.

Az automatikus egyszerűsítő Mathematica változatával a standard és saját tesztfeladatokra előállított helyettesítések mindegyike helyes. Annak eldöntésére, hogy hasznosak-e, vagy jelentéktelenek a produkált helyettesítések, az értekezés 4. fejezetében az automatikus egyszerűsítő által produkált átírások hatását egy numerikus globális optimalizáló teljesítményén vizsgáltam. Konkrétan a GLOBAL nevű heurisztikus multi-start megoldó futási idejét és függvény-kiértékeléseinek a számát hasonlítottam össze az eredeti, és a bemutatott automatikus egyszerűsítő által átírt probléma-alakra vonatkozóan [1]. A vizsgált standard globális optimalizálási tesztfeladatok és egyéb mesterségesen konstruált példák vizsgálatából származó eredményeim azt mutatják, hogy a GLOBAL a legegyszerűbbnek tűnő automatikus átírásból is profitálni tud. Mind a futási időre, mind a függvény-kiértékelések számára kedvezően hatott az átalakítás, az összes tesztfeladat átlagában az átírásnak köszönhető relatív javulás mindkét tekintetben 32% volt.

I. 3. *Az elmélet általánosítása együttesen végrehajtható párhuzamos helyettesítésekre és feltételes NLP feladatokra.*

A Csendes és Rapcsák által leírt nemlineáris koordináta-transzformációkat két irányban általánosítottam [1], az eredményeket az értekezés 3.5. szakasza tartalmazza. Egyrészt az alkalmas párhuzamos helyettesítések leírásával a korábbinál bonyolultabb ekvivalens átírásokra adtam elégséges feltételeket, másrészt kiterjesztettem az alkalmazás lehetőségét a feltételes nemlineáris optimalizálási feladatokra.

Az eredeti, átírás előtti nemlineáris optimalizálási feladat most feltételeket is tartalmazhat:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ c_i(\mathbf{x}) \quad & = 0 \\ c_j(\mathbf{x}) \quad & \leqslant 0, \end{aligned} \tag{2}$$

ahol $f(\mathbf{x}), c_i(\mathbf{x}), c_j(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ sima, valós, képlettel adott függvények, továbbá $i = 1, \dots, p_1$ és $j = p_1 + 1, \dots, p_2$ egészértékű indexek.

Az átalakított feltételes optimalizálási feladat a

$$\begin{aligned} \min_{y \in \mathbb{R}^m} \quad & g(\mathbf{y}) \\ d_i(\mathbf{y}) \quad & = 0 \\ d_j(\mathbf{y}) \quad & \leqslant 0. \end{aligned} \tag{3}$$

Itt $g(\mathbf{y}) : \mathbb{R}^m \to \mathbb{R}$ az $f(\mathbf{x})$ átírt formája, $d_i(\mathbf{y}), d_j(\mathbf{y}) : \mathbb{R}^m \to \mathbb{R}$ ismét sima, valós függvények, méghozzá a $c_i(\mathbf{x}), c_j(\mathbf{x})$ feltételek átírásai, $i = 1, \dots, p_1$ és $j = p_1 + 1, \dots, p_2$.

Jelölje X az $f(\mathbf{x})$ célfüggvényben és $c_k(\mathbf{x})$, $k = 1, \dots, p_2$ feltételekben előforduló változókat jelölő szimbólumok halmazát. Y az átírás után kapott $g(\mathbf{y})$ és $d_k(\mathbf{y})$, $k = 1, \dots, p_2$ függvényekben szereplő változók szimbólumainak a halmaza. Mivel a dimenziószám növelését most nem engedjük meg a transzformációs lépésben, $m \leqslant n$ és $|Y| \leqslant |X|$. Az egyszerűsítő eljárás futásának kezdetén $Y := X$.

Legyen C az eredeti korlátok baloldalain álló kifejezések halmaza:

$$C := \{c_k(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}\,,\ k = 1, \dots, p_2\}.$$

Az F jelölje azt a halmazt, ami $f(\mathbf{x})$ és $c_k(\mathbf{x}) \in C$ függvények összes részkifejezéseit (vagyis az eredeti kifejezések részeként előforduló minden, szintaktikailag helyes kifejezést) tartalmazza.

Az algoritmusunk döntő része a *transzformációs, avagy átírási lépés*. Ha egy $H \subset F$ kifejezés-halmaz fedi a $V \subseteq X$ változó-halmazt (tehát a $v \in V$ változók egyike sem fordul elő H kifejezésein kívül az $f(\mathbf{x})$, vagy $c_k(\mathbf{x}) \in C$ képletében), továbbá $|H| \leqslant |V|$, akkor alkalmazzunk minden H-hoz kapcsolódó helyettesítést $f(\mathbf{x})$-re és C-re a következőképpen. Helyettesítsünk egy új y_i változót minden $h_i(\mathbf{x}) \in H$ helyébe az $f(\mathbf{x})$ képletében, és hasonlóképp minden $c_k(\mathbf{x}) \in C$ képletében. Továbbá frissítsük a változók halmazát az átírt problémában: $Y := (Y \cup y_i) \setminus V$.

A fenti mozzanatra (a H-hoz kapcsolódó) transzformációs lépésként fogunk hivatkozni. Abban a speciális esetben, ha $|H| = |V| = 1$ és $p_1 = p_2 = 0$, visszakapjuk Csendes és Rapcsák [6] algoritmusát a feltétel nélküli esetre.

Az $\mathbf{y} := H(\mathbf{x})$ jelölést az $y_i := h_i(\mathbf{x})$, $i = 1, \ldots, |H|$ rövidítéseként használjuk.

4. TÉTEL Amennyiben teljesül, hogy H helyettesítéseknek egy helyes halmaza, és minden $h_i(\mathbf{x}) \in H$ olyan sima valós függvény, amely szigorúan monoton minden $v \in V \subseteq X$ változó függvényében, továbbá H elemszáma nem több V elemszámánál, és $h_i(\mathbf{x})$ értelmezési tartománya minden $h_i(\mathbf{x}) \in H$-ra egyenlő $\mathbb{R}$-rel, akkor a H-hoz kapcsolódó transzformációs lépés oly módon egyszerűsíti az eredeti problémát, hogy az eredeti feladat minden $\mathbf{x}^*$ helyi minimum (maximum) helye az átalakított probléma egy $\mathbf{y}^*$ helyi minimum (maximum) helyévé transzformálódik.

5. TÉTEL Amennyiben teljesül, hogy H helyettesítéseknek egy helyes halmaza, és minden $h_i(\mathbf{x}) \in H$ olyan sima valós függvény, amely szigorúan monoton minden $v \in V \subseteq X$ változó függvényében, továbbá H elemszáma nem több V elemszámánál, és $h_i(\mathbf{x})$ értelmezési tartománya és értékkészlete minden $h_i(\mathbf{x}) \in H$-ra egyenlő $\mathbb{R}$-rel, akkor a H-hoz kapcsolódó transzformációs lépés oly módon egyszerűsíti az eredeti problémát, hogy az átalakított probléma minden $\mathbf{y}^*$ helyi minimum (maximum) helye előáll az eredeti probléma egy $\mathbf{x}^*$ helyi minimum (maximum) helyének átírásával.

Max-min méltányos sávszélesség-kiosztás BitTorrent hálózatokban

Előzmények

A BitTorrent közösségekben felmerülő max-min méltányos sávszélesség-kiosztás problémáját rajközi (inter-swarm) szinten vizsgáltam. Ez a felhasználók viselkedésének egyik legkomplexebb vizsgálható szintje, ahol a modell megengedi, hogy a felhasználók több tartalom le- és feltöltésében vegyenek részt egyszerre.

Capotă és szerzőtársai [5] nyomán egy BitTorrent közösség aktuális állapotát – vagyis, hogy egy adott időpillanatban ki kinek tölthet fel, milyen adatátviteli korlátok érvényesek, stb. – egy speciális *páros gráf reprezentáció*val írhatjuk le. Ezt az irányított, súlyozott páros gráfot $G = \left(\{U, L, D\}, E, f, c\right)$ jelöli. A közösség alapvető elemei a *felhasználók halmaza* (I) és a *torrentek halmaza* (T). Minden $i \in I$ felhasználó rendelkezik μ_i *feltöltési kapacitás*sal és δ_i *letöltési kapacitás*sal, továbbá

$U = \{ u_i \mid i \in I \}$: a *feltöltő csúcsok* halmaza, ahol u_i az i felhasználó feltöltési (seeding vagy leeching) potenciálját reprezentálja;

$D = \{ d_i \mid i \in I \}$: a *letöltő csúcsok* halmaza, ahol d_i az i felhasználó letöltési (leeching) potenciálját reprezentálja;

$L = \left\{ l_i^t \mid i \in I,\ t \in T \right\}$: a *leeching csúcsok* halmaza, ahol az l_i^t (úgynevezett *leeching session*) létezése azt jelöli, hogy az i felhasználó éppen leech-eli, letölti a t torrentet;

E: az élek halmaza; $E = E_U \cup E_D$, ahol $E_U = \bigcup_{i,j,t}(u_i, l_j^t)$ a *feltöltő élek* halmaza, és $E_D = \bigcup_{j,t}(l_j^t, d_j)$ a *letöltő élek* halmaza;

$c : U \cup L \cup D \to \mathbb{N}$: a *kapacitás függvény*, amely a résztvevők sávszélesség-korlátait reprezentálja :

$$c(u_i) = \mu_i, \quad c(d_i) = \delta_i, \quad c(l_i^t) = \infty;$$

$f : E \to \mathbb{R}^+$: a *folyam függvény*, a kiosztott sávszélességet reprezentálja azokon az éleken, amelyek kielégítik a *folyam-megmaradási tulajdonság*ot:

$$\sum_{u_i \in U} f(u_i, l_j^t) = f(l_j^t, d_j) \qquad \forall l_j^t \in L,$$

valamint a *kapacitás korlátok*at:

$$\sum_{t,j} f(u_i, l_j^t) \leqslant \mu_i \qquad \forall (u_i, l_j^t) \in E_U,$$

$$\sum_{t} f(l_j^t, d_j) \leqslant \delta_j \qquad \forall (l_j^t, d_j) \in E_D.$$

Ebben a hálózati folyam modellben a sávszélesség-kiosztás akkor *max-min méltányos*, ha bármely (l_j^t, d_j) letöltő élre igaz, hogy az $f(l_j^t, d_j)$ folyam csak egy másik $(l_{j'}^{t'}, d_{j'})$ letöltő él $f(l_{j'}^{t'}, d_{j'})$ folyam értéke csökkentésével növelhető, amelyre $f(l_{j'}^{t'}, d_{j'}) < f(l_j^t, d_j)$.

Mivel a feladat folytonos és konvex halmazon értelmezett [5], a max-min méltányos kiosztás egyértelműen létezik [4, 9].

Capotă és szerzőtársai [5] megmutatták, hogy a BitTorrent protokoll standard sávszélesség-kiosztását alkalmazó BitTorrent közösségek átlagos teljesítménye (nem túl meglepő módon) szuboptimális a max-min méltányosság tekintetében. Ez a méltányossági mérték megfelel például a napjainkban peer-to-peer rendszerekkel is kapcsolatba hozott videóközvetítő (streaming) szolgáltatások céljainak.

Az általunk kidolgozott algoritmus [3] Capotă és szerzőtársai munkájának továbbfejlesztése. A Radunović és Le Boudec által is összefoglalt [9] általános max-min programozási algoritmus sémáját követve a letöltő élek max-min méltányos súlyait iteratív módon számítjuk, minden körben a legkisebb meghatározatlan súllyal rendelkező változókat rögzítve.

Saját eredmények

II. 1. *Egzakt matematikai program, helyesség-igazolással.*

Fő eredményem, hogy Capotă és szerzőtársai komplikált szitamódszerét egy egzakt matematikai programmal helyettesítettem, majd az erre épülő *mMaxMin* algoritmus helyességét elméleti úton bizonyítottam. Az algoritmus részletes leírását a bizonyítással együtt egy hatástényezős folyóiratcikkben [3], valamint az értekezés 5.4. szakaszában közöltem.

Különösebb előkészítés nélkül is érthető a következő jelentős segédtétel és tétel.

2. LEMMA Az *mMaxMin* bármely k. iterációjában fennáll a

$$\sigma_k = \sigma$$

összefüggés, ahol σ egy konstans, a hálózat maximális átvitelének mértéke abban az esetben, amikor

$$\forall(l_j^t, d_j) \in E_D \ : \ f(l_j^t, d_j) \geqslant \phi.$$

6. TÉTEL Az *mMaxMin* véges számú lépés után véget ér, és a max-min méltányos tulajdonságot garantálja minden letöltő élre.

II. 2. *AMPL implementáció és numerikus tesztek.*

A bizonyítottan helyes algoritmust AMPL nyelven, elméleti eredményeimet és néhány további átírási lehetőséget is alkalmazva valósítottuk meg. Az értekezés 5.5. szakaszában összehasonlítottuk a korábbi *MM* [5] és a saját *MaxMin-r* [3] algoritmust a BitSoup.org nevű BitTorrent közösség pillanatfelvételeiből származó nagyméretű feladatokon. Az eredmények alapján a *MaxMin-r* a nagyobb problémákon gyorsabban teljesít, mint az *MM*, ráadásul már néhány kezdeti iteráció után nagyon jó közelítését adja a max-min méltányos allokációnak. Ez a közelítés, ami a feladat fízibilis megoldása, nagyon gyorsan előállítható. Például a *MaxMin-r* első iterációja 46 másodpercet igényelt a 23 670 csúcsot és 7 326 letöltő élet tartalmazó példán, míg az *MM* algoritmus hasonlóan jó eredményt biztosító első 910 iterációja több, mint nyolc órán keresztül futott.

II. 3. *Modellezési technikák hatáselemzése.*

Az értekezés 6. fejezetében egy közlésre benyújtott kéziratunk [7] alapján azt vizsgáltam, hogy a max-min méltányos sávszélesség-kiosztást előállító, nagyméretű lineáris és nemlineáris vegyes-egészértékű programokat is magában foglaló optimalizálási feladat megoldásának sebességét mennyiben befolyásolja különféle modellezési technikák alkalmazása. Elemeztem a matematikai modellezés során felmerülő átírási lehetőségek hatásait. Kiterjedt numerikus tesztelést végeztem tizenkét modell-változat, huszonhét teszteset és kettő professzionális megoldó minden lehetséges kombinációjával.

Az elvégzett kísérletek számos érdekes eredménnyel szolgáltak. Egyrészt, nem találtam olyan modell-változatot, ami minden esetben a leggyorsabban oldaná meg a feladatot. Kiderült továbbá, hogy a tesztelt korszerű megoldók számára nem jelent problémát a bilineáris felírás hatékony megoldása. A Gurobi és a MOSEK is könnyebben boldogult a kisebb méretű bilineáris feladattal, mint a McCormick-átírás révén előálló ekvivalens, nagyobb méretű lineáris változattal. Kiemelkedett egy, az AMPL-ben is megvalósított LP előmegoldó technika hatása, mely az összes teszt átlagában a futási idő mintegy 10%-os javulását eredményezte.

Hivatkozások

[1] Antal, E. – Csendes, T.: Nonlinear symbolic transformations for simplifying optimization problems. In *Acta Cybernetica*, 22. évf. (2016) 4. sz., 715–733. p.

[2] Antal, E. – Csendes, T. – Virágh, J.: Nonlinear transformations for the simplification of unconstrained nonlinear optimization problems. In *Central European Journal of Operations Research (CEJOR)*, 21. évf. (2013) 4. sz., 665–684. p.

[3] Antal, E. – Vinkó, T.: Modeling max–min fair bandwidth allocation in BitTorrent communities. In *Computational Optimization and Applications*, 2016. 18 p., http://dx.doi.org/10.1007/s10589-016-9866-5. Közlésre elfogadva.

[4] Bertsekas, D. P. – Gallager, R. G.: *Data Networks*. 2nd. kiad. Englewood Cliffs, NJ, USA, Prentice Hall, 1992. ISBN 0132009161.

[5] Capotă, M. – Andrade, N. – Vinkó, T. – Santos, F. – Pouwelse, J. – Epema, D.: Inter-swarm resource allocation in BitTorrent communities. In *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P 2011)* (konferenciaanyag). 2011, 300–309. p.

[6] Csendes, T. – Rapcsák, T.: Nonlinear coordinate transformations for unconstrained optimization I. Basic transformations. In *Journal of Global Optimization*, 3. évf. (1993) 2. sz., 213–221. p.

[7] Dobjánné Antal E. – Vinkó T.: Egy nemlineáris vegyes-egészértékű optimalizálási feladat különféle modelljeinek komparatív elemzése. 2016. http://www.inf.u-szeged.hu/~antale/en/research/DAntal_MIPmodels2016.pdf. Közlésre benyújtott kézirat.

[8] Heck, A.: Internal data representation and substitution. In *Introduction to Maple*. New York, NY, Springer New York, 2003, 153–174. p. ISBN 978-1-4613-0023-6.

[9] Radunović, B. – Le Boudec, J.-Y.: A unified framework for max-min and min-max fairness with applications. In *IEEE/ACM Transactions on Networking*, 15. évf. (2007) 5. sz., 1073–1083. p.

[10] SCHICHL, H. – NEUMAIER, A.: Interval analysis on directed acyclic graphs for global optimization. In *Journal of Global Optimization*, 33. évf. (2005) 4. sz., 541–562. p.

[11] Wolfram Language & System Documentation center. Wolfram Language tutorial: Basic internal architecture. `http://reference.wolfram.com/language/tutorial/BasicInternalArchitecture.html`. [2017.01.12.].