BME OMIKK
LOGISZTIKA
10. k. 5. sz. 2005. szeptember–október. p. 46–51.
Tanulmánytár * Ellátási/elosztási logisztika

Tabukeresési változatok útvonal-optimalizálási problémára

Csiszár Sándor*
főiskolai adjunktus
Budapesti Műszaki Főiskola, Kandó Kálmán Villamosmérnöki Kar,
Mikroelektronika és Technológia Intézet.

lektorálta:
Bóna Krisztián
egyetemi tanársegéd
Budapesti Műszaki és Gazdaságtudományi Egyetem, Közlekedésmérnöki Kar,
Közlekedésüzemi Tanszék

A tabukeresést eddig sikeresen alkalmazták kombinatorikai optimalizálási problémák, gráfszínezés, ütemezés stb. megoldására. Az útvonal-optimalizálás logisztikai probléma. Az optimalizáló algoritmusok vizsgálatakor egyszerűsítésként a vevők légvonalbeli távolságait vesszük alapul, de felhasználói szoftverek esetén az egyes relációkra (viszonylatokra) vonatkozó tényleges távolságok egy adatbázisba feltölthetők. E probléma kétfázisú hibrid metaheurisztikával történő megoldása kapcsán a második fázis egy tabukeresés, ami Chiang és Russel [1] módszerének néhány új ötlettel kiegészített változata. A tabukeresési változatokat célszerű összehasonlítani adott példákon keresztül, előfordul ugyanis, hogy egy logikus ötlet alkalmazásával járó előny nem számottevő, vagy épp jelentősen megnöveli a program futási idejét, ezért alkalmazása megfontolandó. A cikkben tárgyalt vizsgálatokat M. Solomon [2] teszt segítségével végeztük, n = 100 vevőszám mellett.

Tárgyszavak: útvonal; optimalizálás; járatoptimalizálás; tabukeresés; heurisztika; metaheurisztika.

* E-mail: csiszar.sandor@kvk.bmf.hu

A probléma definíciója

Útvonal-optimalizálási probléma (VRPTW)

Tekintve, hogy a témának bőséges szakirodalma van, másrészt ebben a tanulmányban a tabukeresés vizsgálata a cél, itt csak rövid ismertetőre szorítkozunk. Legyen adva egy központi raktár és *n* vevő, akiket a raktárból kell kiszolgálni. Megfelelően gyors és intelligens szoftver segítségével biztosíthatók a legfontosabb gazdasági, logisztikai célok:

- a lehető legkisebb napi járatszám, ezen belül pedig
- a lehető legrövidebb bejárási útú járatok megtalálása költségoptimumra törekedve.

A probléma megoldásakor az alábbi feltételeket kell figyelembe venni:

- minden jármű a központi raktárból indul, és ide érkezik vissza,
- a vevők csak egyszer látogathatók,
- távolságukon a légvonalbeli távolságokat értjük,
- a vevők elhelyezkedése (térbeli koordinátái), a rendelt áru mennyisége, a járművek szállítókapacitása, a vevők kiszolgálási időintervalluma (időablaka) és kiszolgálási időszükséglete adott,
- a depó nyitvatartási idején belül kell a járműveknek visszatérni, a depónál eltöltött rakodási idő elhanyagolható.

A megoldás nem sértheti a megadott feltételeket. A feladat tehát egy keresési probléma, aminek egzakt megoldására csak $n<50$ esetén érdemes vállalkozni a kombinatorikai robbanás miatt. Általában egy viszonylag jó kezdeti megoldásból indulunk ki, majd ezt próbáljuk tökéletesíteni. A kezdeti megoldás javítását újabban további két fázisra szokás osztani, mivel az elsődleges járatszám minimalizálása és a másodlagos költségcsökkentés sok esetben nem elégíthető ki együtt (a legalacsonyabb költség nem a legkisebb járatszámnál adódik).

Tabukeresés

A tabukeresést Glover [3, 4] alkalmazta először. Ez egy iteratív eljárás, mely eszközt kínál arra, hogy a lokális minimumot elhagyhassuk. A tabukeresés jelentőségének megértéséhez néhány fogalmat kell tisztázni. Egy aktuális megoldás (*s*) egy gráffal megadható. A szomszédsági gráf csomópontjai azon megoldásgráfokat tartalmazzák *(N(s,x)),* melyeket az aktuális megoldásból az adott operátorkészlet (műveletek halmaza: M) mellett elérhetünk. Lokális minimumnak (más problémák esetén maximumot keresünk) nevezzük azt a megoldást (*s*), melyre fennáll: $f(s)<=f(r)$, minden $r \in N(s,x)$-re, ahol *f* a kiértékelő függvény. A lokális keresés során az aktuális megoldást a vizsgált szomszédságban található megengedett legjobb megoldásra cseréljük, ezt addig folytatjuk, amíg található az aktuálisnál jobb megoldás. Itt a lokális keresés véget ér, bár nem állítható, hogy a kapott megoldás globálisan is a legjobb. A tabukeresés során egy rövid távú memóriában (tabulistán) tároljuk a legutóbb bejárt megoldásokat, pontosabban azokat az állapotokat, melyek egy „nemrég” megvizsgált megoldást állítanának vissza. Ez alól általában kivételt teszünk, ha a tabulista megsértésével az eddigi legjobb megoldásnál jobb eredmény adódik. Ha a rövid távú memória a legutolsó (*c*) ciklusra emlékszik, akkor elkerülhető a legfeljebb (*c*) hosszúságú hurok kialakulása. Az intenzifikálás alkalmazása azt jelenti, hogy egy középtávú memóriában a jó megoldások bizonyos tulajdonságaira (jelentős költségcsökkenés, várakozási idő csökkenése, járatszámcsökkenés) emlékezünk és nem fogadunk el egy ideig olyan megoldásokat, melyek ezzel a tulajdonsággal nem rendelkeznek. A diverzifikálás – mely hosszú távú

memóriának is tekinthető – célja, hogy a „távolabbi régiókat” is feltérképezzük, ezért bizonyos jellemzők gyakoriságát regisztráljuk és „büntetjük” azokat a megoldásokat, melyek ezekkel a jellemzőkkel rendelkeznek. A tabukeresés során a tabulista hossza állandó vagy valamilyen listahosszmenedzsment szerint változik. A legsikeresebb lokális keresések Potvin and Rousseau, Russell, valamint néhány kiváló tabukeresés Rochat and Taillard, Chiang and Russell [1], and Cordeau et al. nevéhez fűződik.

Tabukeresési változatok

Alapmegoldás

Alapmegoldásként – egyben összehasonlítási alapként – a már említett Chiang és Russel (Ch-R) reaktív tabukeresés megoldását választjuk három eltéréssel, amiből kettő – a szerző tudomása szerint – eddig még nem publikált megoldás. A reaktív tabukeresés alapgondolata, hogy regisztráljuk az ismétlődő megoldásokat, és ha az ismétlődések száma elér egy megadott értéket, akkor célszerű a lokális keresést máshol folytatni, diverzifikációt – oszcillációt – alkalmazni (lásd részletesen [1]). Egyszerűbb, ha a megoldások helyett azok jellemzőit – a járatszámot, a költséget és a várakozási időt – tartjuk nyilván, és a megoldásokat ismétlődőnek tekintjük, ha minden regisztrált adatuk egyező. Ez igen kis hibával igaz, s mellette jelentős tármegtakarítás érhető el. Ch-R megoldásában a tabulista a következőképpen van megadva: (c, r, t), ahol c a szóban forgó vevő azonosítója, r az út azonosítója és végül t az iterációk száma, amin belül csak akkor térhet vissza a c vevő az r útra, ha minden eddiginél jobb megoldás adódik. Az alapmegoldásban definiált tabulista ettől annyiban tér el, hogy az útvonal azonosítója helyett a c-k gráfél szerepel, ahol k az aktuális megoldásban a c-t követő vevő azonosítója. Ezzel a c vevő valószínűleg nagyobb flexibilitást kap, visszatérhet ugyan az r útra, kivéve azt az esetet, hogy a c-k él visszaálljon. Megjegyzendő azonban, hogy ebben az esetben a c-k él egy r-től különböző úton sem állhat elő, míg ugyanez Ch-R megoldásban megtörténhet. Az említett új megoldások a következők:

1. **A tabulista hosszának menedzselése:** Ch-R megoldásában a tabulista hossza csökken vagy nő, aszerint, hogy a változatlan tabuhossz melletti lépésszám meghalad-e egy átlagos iterációs számot. Ezzel ellentétben az alkalmazott megoldás esetén a tabulista hossza két oszcilláció között lineárisan nő egy minimum és maximum érték között minden ismétlődés esetén. Ezzel „egyre nagyobb körök” megtételére kényszerítjük az algoritmust, ami logikus keresési módszernek tűnhet.
2. **Dinamikus tabulista alkalmazása:** A megoldás alapgondolata az, hogy ha egy nagyobb költségcsökkenést talál az algoritmus, akkor az előző egy relatív rossz megoldás, tehát ide annál kevésbé érdemes visszatérni, minél nagyobb a költség csökkenése. Bevezethetünk egy dinamikus tabulista-hosszt – a szokásos tabulista mellett –, aminek a maximumát a felhasználó beállíthatja. A keresés során ez azt jelenti, hogy a nagyobb költségű pontok ritkábban fognak visszatérni, és várhatóan intenzívebben keressük az alacsonyabb költségű megoldások környékén. A dinamikus lista hossza (Th_d) az algoritmus futása során $TH_d = 0$, ha a költségcsökkenés nem pozitív, egyébként pedig $Th_d = Round\{Th_m\ [dK/(\eta Ks)]\}$, ahol dK az algoritmus által talált költség csökkenés, K_s az aktuális megoldás költsége, η szorzó faktor a dinamikus lista maximális hosszához tartozó költségcsökkenés számításához, Th_m pedig a dinamikus lista maximális hosszúsága.

Alkalmazott szomszédsági operátorok

Korábban érintett kérdés az alkalmazott szomszédsági operátorok köre. Gehring és Homberger [5] ugyanezen problémákra csupán három operátort alkalmazott – nemcsak tabukeresési célra, hanem az evolutionary fázisban is. Ezek a következők:

- egy vevő áthelyezése egy másik pozícióba (insertion move), ami lehet útvonalon belül vagy két útvonal között,
- két vevő cseréje (interchange move),
- a harmadik művelet pedig a 2-Opt, melynek elnevezése onnan származik, hogy két-két gráfél szűnik meg, illetve keletkezik, szemléletesen két útvonal kiválasztott pontjainak átkötésével (feltételezve, hogy ez nem sérti a probléma feltételeit).

A szerző [6] megoldásában a 2-Opt művelet az oszcilláció során kerül alkalmazásra, a másik két említett művelet párhuzamosan mindig kiértékelésre került. Ezek mellett a szerző további három különböző útvonalak közötti műveletet (Interchange-21, Interchange-22, Interchange-32, ez utóbbi például azt jelenti, hogy az egyik útról három, a másikról kettő vevőt helyeznek át) is alkalmazott (feltéve, hogy az első két művelet és sorrendben a megelőző nem talált költségcsökkenést). Ezeken kívül még egy útvonalon belüli 3-Opt művelet is alkalmazásra került az előzőekhez hasonló feltételek mellett.

A tabukeresési változatok összehasonlítása

A cél meghatározása

A tabukeresés során az alkalmazott stratégiától és a probléma hosszától függően vagy a teljes szomszédságot átvizsgálhatjuk (Global Best), vagy a keresést a szomszédsági halmazban addig folytatjuk, amíg az aktuálisnál jobb megoldást találunk (First Best), vagy egy adott lépésszámot el nem érünk. Ez utóbbi esetet általában nagyméretű problémák esetén használjuk, bár felvetődik a kérdés, hogy nagy számú iterációs lépés után válasszuk-e ki a legjobbat (több időt töltve egy iterációval), vagy esetleg a szomszédság részbeli átvizsgálásával válasszuk ki a következő aktuális megoldást. Nyilvánvaló, hogy erre nem lehet általános választ adni, de egyes problématípusokra érdemes vizsgálni a lehetőségeket, hasonlóképpen a szomszédság, illetve az alkalmazott operátorkészlet megválasztását. A további kutatásokhoz – elsősorban nagyméterű problémák esetén – fontos, hogy a felvázolt tabukeresési változatok hatékonyságát, időszükségletét összehasonlítsuk.

Vizsgált változatok

Az egyes megoldások hatásainak elemzésére az alább bemutatott öt változat vizsgálatára került sor. A megoldásokat a kötőjel utáni számokkal különböztetjük meg:

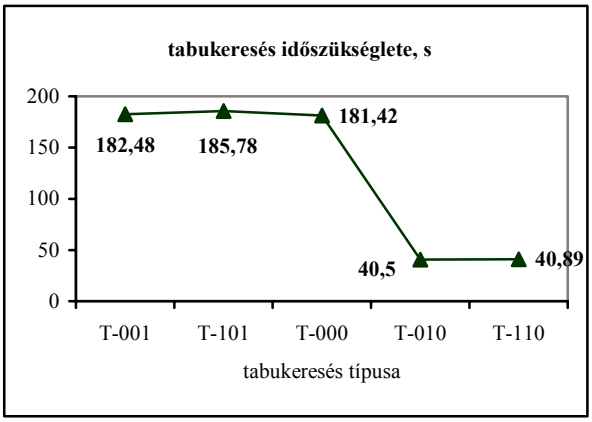
- az első szám 1 vagy 0 aszerint, hogy a maximálisan 10 hosszúságú dinamikus tabulista alkalmazva volt-e vagy sem,
- a második szám 1 vagy 0 aszerint, hogy a S6 vagy a GH3 operátoros verzió volt-e alkalmazva,
- a harmadik szintén 1 vagy 0 aszerint, hogy a reaktív tabukeresést alkalmaztuk-e vagy sem.
- Így a vizsgált öt megoldás: T-001, T-101, T-000, T-010, T-110. Az ezekre a változatokra épülő szoftver által adott eredményeket a Solomon teszt problémákon való futtatással (R100, R105, R201, R205, C101, C105, C201, C205, RC101, RC105, RC201, RC205 így minden típus szerepel a vizsgálatban) értékel-

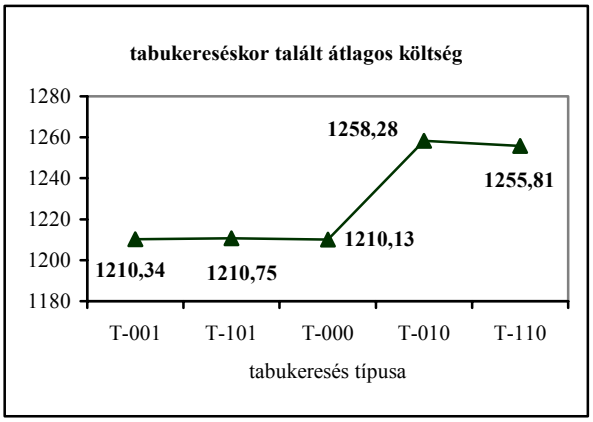
jük. Az értékelés a GH3 operátoros verziók esetén problémánként 10 egyébként 5 futtatás (összesen 120 ill. 60 adat) adatait tartalmazza. Az összehasonlításhoz a futtatási eredmények átlagát használjuk fel. Az eredmények az *1. ábrán* (időszükséglet és költség) láthatók. A számítógépes program Delphi 5 fejlesztő rendszerrel készült, a futtatás 1.7 GHz-es processzorral és 120 MB RAM-al rendelkező számítógépen történt.

Az eredmények értékelése

A futtatási adatok és az *1. ábra* alapján a következő megállapításokat tehetjük:

1. Amint ez várható volt az előzőekben leírtak alapján a Gehring és Homberger által használt háromoperátoros véletlen kiválasztású változatok (T-010, T-110) időszükséglete jóval alacsonyabb (20–25%) az (S6) változatnál, ugyanakkor a költség 4–5%-kal magasabb, mint az S6 változat költsége.
2. A dinamikus tabulista hatása a nagyon intenzív keresés mellett (S6 verziók: T-001, T-101 és T-000) nem mutatható ki, azonban a T-010, T-110 összevetésében kis mértékű költségcsökkenés tapasztalható gyakorlatilag változatlan időigény mellett, tehát alkalmazása indokolt lehet.
3. Megállapítható azt is, hogy a reaktív tabukeresés (vagyis a bejárt megoldások regisztrálása) bár kétségtelenül logikus megoldás, de nincs kimutatható befolyása a költségekre egy bizonyos iterációszám (3000) fölött, igaz időszükséglete sem túl nagy. Ez különösen igaz lehet nagyméretű problémákra. A T-000 változat esetén a reaktív keresés ki volt kapcsolva, de nem okozott észrevehető változást. Ennek magyarázatához meg kell gondolni, hogy a tabukeresés – eltekintve a végrehajtott oszcillációktól – determinisztikus. Ha egy korábbi megoldás visszatér, a keresés nem szükségképpen ugyanúgy folytatódik, mint a korábbi esetben, mert ehhez az is kellene, hogy a tabulisták állapota is megegyezzék, aminek esetenként igen kicsi lehet az esélye. Az oszcilláció akkor lenne igazán hasznos, ha a megoldástérnek egy olyan régiójába tudnánk eljutni, amit az adott pillanatig nem jártunk be, vagy az eddiginél jobb megoldást tartalmaz. Erre pillanatnyilag nincs eszközünk.





1. ábra Tabukeresési változatok időszükséglete és a talált költség

4. A tabukereséssel elért költséget leginkább az alkalmazott operátorkészlet befolyásolja, igaz ezzel az átvizsgálandó szomszédsági tér is megnő és végső soron keresési idővel kell érte fizetni. Ebből az következik, hogy az adott keresési célnak megfelelően kell az operátorkészletet megválasztani.

Ha eltekintünk az oszcillációtól, és a tabukeresést determinisztikus jellegűnek gondoljuk, akkor a kapott eredmény lényegében a kezdeti megoldás, az iterációs lépésszám és a szomszédsági operátorok függvénye. A szomszédsági operátorok köre a gyakorlatban két útvonal közötti mozgásokra korlátozódik, lényegében egyszerűségi és számítás igényességük miatt. Az iterációs lépésszámot pedig a feladat és a felhasználó keresési igénye dönti el. Ezzel elégedettek is lehetnénk, ha tudnánk, hogy a keresés során „egyenletesen" bejártuk a teljes keresési teret, és a megoldásunk a keresési idő megválasztásának megfelelően közelít a lehetséges legjobb megoldáshoz. Sajnos erre pillanatnyilag semmilyen garancia nincs. A keresési tér valamilyen szintű bejárása is rendkívül nehéz feladat, helyette célszerűnek találnám különböző megoldásokból kiindulva elegendően finomítani a keresés metodikáját. Erre szeretnék megoldási javaslatot kidolgozni a közeljövőben.

Támogatók, köszönetnyilvánítás

A szerző köszönetét fejezi ki *Prof. Dr. Monostori Lászlónak* (egyetemi tanár BMKE, igazgatóhelyettes MTA SZTAKI), *Dr. Kis Tamásnak* (MTA SZTAKI) hasznos tanácsaikért, valamint, *Dr. Turmezei Péternek* (igazgató, Mikroelektronikai és Technológia Intézet, BMF Kandó Kálmán Villamosmérnöki Kar) támogatásáért.

Irodalom

[1] Chiang, W. C.; Russell R. A.: A reactive tabu search metaheuristic for the vehicle routing problem with time windows. = INFORMS Journal on Computing. 1997. 9. sz. p. 417–30.

[2] Solomon, M. M.: Algorithm for the vehicle routing and scheduling problem with time window constraints = Operation Research, 1987. 35. sz. p. 254–265.

[3] Glover, F. Tabu search, part I. = ORSA J. Computing, 1989. 1. sz. p. 190–206.

[4] Glover, F. Tabu search, part II. = ORSA J. Computing, 1990. 2. sz. p. 4–32.

[5] Gehring, H.; Homberger, J.: A two-phase hybrid metaheuristic for the vehicle routing problem with time windows = European Journal of Operation Research, 162. sz. 2005. p. 220–238.

[6] Csiszár, S.: Route elimination heuristic for vehicle routing problem with time windows. = Acta Polytechnica Hungarica, 2005. (nyomdai előkészítés alatt).

[7] Bräysy, O.; Gendreau, M.: Tabu Search heuristics for the vehicle routing problem with time windows. Sociedad de Estadística e Investigación Operativa, Madrid, 2002. dec.