Algoritmus a kapacitáskorlátos, egycentrumos járatszerkesztés megoldására

PROF. DR. BENKŐ JÁNOS
SZIE, Gödöllő, Gazdaságtudományi Kar. RGVI

A tanulmány az elosztási hálózatokon gyakran előforduló problémával, a járatszerkesztéssel foglalkozik. A szerző központi telephelyről indított gyűjtő vagy elosztó járatok minimális költségű útvonalainak számítására közöl egy új algoritmust, amelynek lépéseit egy mintapéldán mutatja be, majd az eredményeket általánosítva összefoglalja az eljárást. A kézi számolásra és számítógépes programozásra egyaránt alkalmas módszer, különböző megszorítások, kapacitás, idő stb. korlátok egyidejű figyelembevételét is lehetővé teszi.

1. Bevezetés

Az elosztási vagy felvásárlási tevékenységet folytató vállalatoknál, raktári bázisokon stb. gyakori feladat a járművek menetrendjének és útvonalának tervezése, amit járatszerkesztésnek neveznek. A probléma nagyon sokféle formában jelentkezhet, de az egyik leggyakoribb eset az, amikor egy vállalatnak egy központi telephelyről (egy centrumból) kell ellátnia a fogyasztókat vagy megrendelőket, és az igényeinek kielégítésére a vállalat véges számú és korlátozott kapacitású járműparkkal rendelkezik.

A feladat megoldásának legegyszerűbb módja az, hogy minden fogyasztóhoz egyedi járművel szállítjuk ki a megrendelt mennyiséget. A megrendelt mennyiségek azonban általában nem kötik le a járművek teljes kapacitását, ami lehetővé teszi az utak összevonását, járatokba szervezését. Természetesen az összevonások csak bizonyos megszorítások mellett hajthatók végre, így:

a megrendelők igényét a célállomásokon maradéktalanul ki kell elégíteni,

a szállított mennyiség nem lépheti túl a járathoz rendelt jármű kapacitását,

a jármű által megtett út vagy a szállítási idő nem léphet túl egy előre meghatározott megengedett értéket.

Tekintve, hogy az utak összevonására általában nagyon sokféle lehetőség van, felvetődik a kérdés, található-e az intuitív döntéseknél jobb, az optimálishoz közelálló megoldás. Az irodalomból számos heurisztikus algoritmus ismert a feladat megoldására. Az utóbbi években elsősorban a genetikus algoritmusok alkalmazása vált népszerűvé (*Baker, Barrie M, and Ayechew, M. A., 2003*, *Chakroborty, P. and Dwivedi, T., 2002*, *Christian Prins, 2004*). A továbbiakban a klasszikus megoldások közé sorolható, kézi számolásra és programozásra egyaránt alkalmas új algoritmusra teszünk javaslatot, amelynek lépéseit egy mintapéldán mutatjuk be, majd az eredményeket általánosítva összefoglaljuk az eljárást.

A javasolt megoldás alapja az a triviális tény, hogy az utak összevonása a gyakorlatban általában útmegtakarítást eredményez, ami nagyon egyszerűen belátható. Ha a P_0-P_i-P_0 és a P_0-P_j-P_0 utakat egyesítjük P_0-P_i-P_j-P_0 járattá, akkor az elérhető megtakarítás:

$$s_{ij} = 2c_{0i} + 2c_{0j} - (c_{0i} + c_{ij} + c_{0j}) = c_{0i} + c_{0j} - c_{ij}.$$

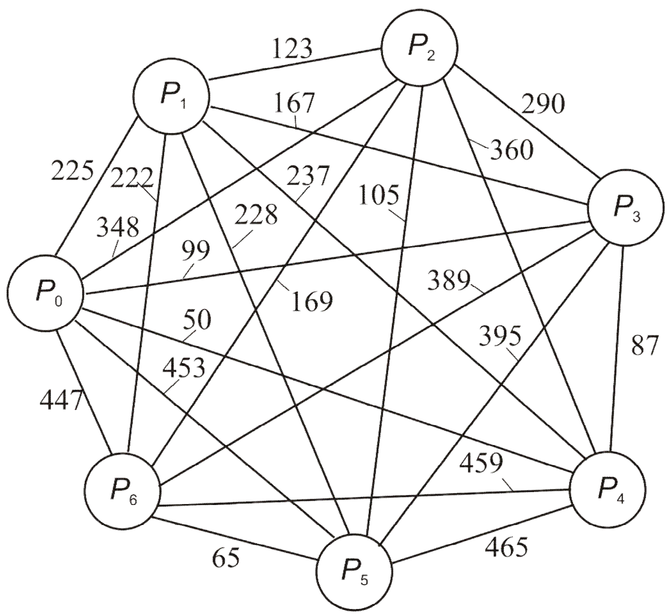
Az is világos, hogy minél nagyobb az adott járathoz rendelt jármű kapacitása, annál több út összevonására van lehetőség. Ezért a járatok szervezésekor a nagyobb teherbírású járműveknek prioritást kell biztosítani.

Természetesen indokoltan vetődhet fel a kérdés, hogy az útmegtakarítás minden esetben költségmegtakarítást jelent-e. A kérdés indoka egyrészt az, hogy az utak összevonása szállításimunka (tkm) növekedéssel járhat, másrészt a nagyobb kapacitású járművek fajlagos üzemeltetési költsége általában nagyobb, mint a kisebbeké. Mégis a kérdésre egyértelmű igennel válaszolhatunk. A szállítási munka növekedése ugyanis nem jelent számottevő költségnövekedést, mivel a terhelt és a terheletlen jármű fogyasztása közötti különbség nagyon kicsi, és az egyéb költségekhez viszonyítva nem számottevő. A nagyobb és a kisebb teherbírású járművek fajlagos üzemeltetési költségében mutatkozó eltérés pedig nem tartozik e feladat lényegéhez, tekintve, hogy e feladatban meglévő jármű állományt kell optimálisan kihasználni. Más kérdés az, ha a járműállomány összetételéről kell dönteni, akkor magától értetődően mérlegelni kell ezeket a fajlagos költségeket is.

A gyakorlatban előfordulhat, hogy egy-egy fogyasztó megrendelése meghaladja a legnagyobb jármű kapacitását. Ebben az esetben a fuvart megosztjuk két vagy több jármű között. Tekintettel arra, hogy az így telítődő járműveknél nincs további lehetőség útösszevonásra, a telített járműveket nem vizsgáljuk, csak a megosztás utáni maradékokat kezeljük megrendelésként.

2. Mintapélda

Legyen adott egy debreceni székhelyű vállalat, P_0, amelynek Debrecenből kell az ország különböző pontjaira (Budapest, Győr stb.) települt P_1, P_2,...,P_n fogyasztókhoz meghatározott r_1, r_2,...,r_n, mennyiségű terméket eljuttatnia. A fogyasztókat, telephelyeiket és az igényeiket az *1. táblázat* tartalmazza.



1. ábra. Az elosztási hálózatot ábrázoló $G(P,E)$ gráf (Forrás: saját szerkesztés)

Ismeretesek továbbá a P_0 centrum és a P_1, P_2,...,P_n célállomások közötti legrövidebb utak c_{0j}, c_{i0}, és c_{ij} (*3. táblázat*). Az állomások és az utak egy $G(P,E)$ gráfot alkotnak (*1. ábra*), ahol a P az állomások és az E a c_{ij} súlyú az élek halmaza. A feladat a megrendelt mennyiségek kiszállítása a centrumból a fogyasztókhoz a rendelkezésre álló járműpark felhasználásával úgy, hogy a járművek által megtett összes út a lehető legrövidebb legyen. A járművek által megtehető utat és a szállítási időt illetően nem élünk megszorításokkal, ami azonban az eljárás lényegét nem érinti. Az összevonások szempontjából csak a járművek kapacitása jelentsen korlátot.

A vállalat 7 db különböző t_k (k=1,2,…,7) teherbírású gépkocsival rendelkezik, a járműpark összetételét a *2. táblázat* foglalja össze.

1. táblázat

Megrendelések

Fogyasztó	P_1	P_2	P_3	P_4	P_5	P_6
Telephely	Budapest	Győr	Miskolc	Nyíregyháza	Szombathely	Zalaegerszeg
r_i[t]	4	3	4	2	3	2

2. táblázat

Járműpark

Teherbírás [t]	**Darabszám**
10	2
6	5

3. táblázat

Távolsági mátrix (km)

	Debrecen P_0	Budapest P_1	Győr P_2	Miskolc P_3	Nyíregyháza P_4	Szombathely P_5	Zalaegerszeg P_6
Debrecen P_0	0	225	348	99	50	453	447
Budapest P_1	225	0	123	167	237	228	222
Győr P_2	348	123	0	290	360	105	169
Miskolc P_3	99	167	290	0	87	395	389
Nyíregyháza P_4	50	237	360	87	0	465	459
Szombathely P5	453	228	105	395	465	0	65
Zalaegerszeg P_6	447	222	169	389	459	65	0

A megoldás **első lépése** az ún. **megtakarítási mátrix** felállítása, amelynek elemei az előzőek alapján azt mutatják, hogy mennyi útmegtakarítás érhető el azzal, ha a P_0-P_i-P_0 és a P_0-P_j-P_0 utakat egyesítjük P_0-P_i-P_j-P_0 járattá. A c_{ij} távolságmátrixból (*3. táblázat*) számított megtakarítási mátrixot (s_{ij}) a *4. táblázat* tartalmazza.

4. táblázat

Megtakarítási mátrix (km)

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0	450	157	38	450	450
P_2	450	0	157	38	696	626
P_3	157	157	0	62	157	157
P_4	38	38	62	0	38	38
P_5	450	696	157	38	0	835
P_6	450	626	157	38	835	0

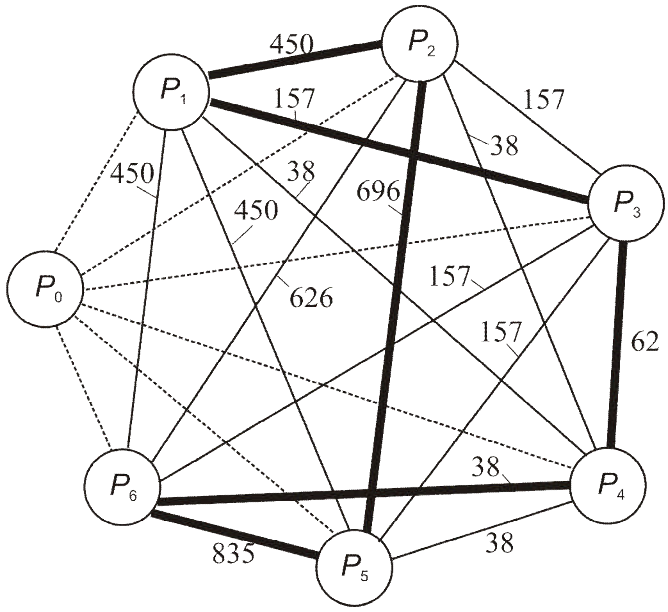
Például, a P_0-P_1-P_0 és a P_0-P_2-P_0 utak összevonásakor a P_2 sor és a P_1 oszlop metszéspontjában, a megtakarítás:

$$s_{21} = c_{01} + c_{02} - c_{21} = 225 + 348 - 123 = 450\,.$$

A megtakarítási mátrixot ábrázolja az $M=(P^*,S)$ gráf (*2. ábra*), ahol az n elemű P^* a P részhalmaza, amely a P_0 pontot nem tartalmazza. Az S pedig az s_{ij} súlyú élek halmaza.

A megoldás **második lépésében** kijelöljük a megtakarítási mátrixot (s_{ij}) ábrázoló $M=(P^*,S)$ gráfhoz tartozó maximális súlyú körutat. A körutazási feladat megoldáshoz számos heurisztikus eljárás áll rendelkezésre, és ezek az irodalomban elérhetők (*Benkő J., 2000*). A körutat szimbolizáló gráf $K=(P^*,S^*)$, ahol az S^* a körutat alkotó élek halmaza. A maximális súlyú körút éleinek súlyát az *5. táblázat* tartalmazza, a körút vastag vonallal kiemelt élei pedig a *2. ábrán* láthatók. (A szaggatott vonallal rajzolt élek nem részei az S halmaznak.) A körút elemei:

$$P_1–P_3–P_4–P_6–P_5–P_2–P_1.$$



2. ábra. Az $M=(P^*,S)$ gráfon kijelölt $K=(P^*,S^*)$ maximális súlyú körút (Forrás: saját szerkesztés)

5. táblázat

Maximális súlyú körút (km)

	P_1	P_2	P_3	P_4	P_5	P_6
P_1			157			
P_2	450					
P_3				62		
P_4						38
P_5		696				
P_6					835	

Ha rendelkezünk elegendően nagy kapacitású járművel, akkor megtehetjük azt is, hogy a körútból a legkisebb s_{ij} elemhez tartozó élet eltávolítjuk ($P_4–P_6$), és az így kapott ún. hamiltoni út végpontjaihoz (P_4 és P_6 pontok) kapcsoljuk a P_0 centrumot. Az eredmény egy új, maximális súlyú körút, amely a centrumból indul, valamennyi fogyasztó telephelyét egyszer érinti, majd visszatér a centrumba. Ebben az esetben egyetlen jármű (járat) látogatja meg a fogyasztóhelyeket.

A járművekre előírt kapacitáskorlát (t_k) miatt azonban a $K=(P^*,S^*)$ körutat olyan, egy vagy több élt tartalmazó, maximális útmegtakarítást eredményező, utakra kell felbontani, amelyek eleget tesznek a kapacitáskorlátokra vonatkozó előírásoknak, azaz a

$$m_k = \sum_{u=1}^{\rho} r_u \le t_k$$

feltételnek, ahol:

ρ a k-adik járattal felkeresett fogyasztók száma,

r_u a u-adik fogyasztó igénye,

m_k a k-adik járattal kiszállított mennyiség,

t_k a k-adik jármű kapacitása.

Végül a kijelölt utaknak a végpontjait összekötjük a centrummal, így maximális útmegtakarítást szolgáltató részkörutakat kapunk.

A **harmadik lépésben** a következő eljárással megszerkesztjük a járatokat, vagyis a fogyasztóhelyeket, illetve kiszállítandó mennyiségeket a járművekhez, illetve a járatokhoz rendeljük. Gyakorlatilag a maximális súlyú körutat a kapacitáskorlátoknak megfelelő utakra bontjuk.

Az *5. táblázatot* egészítsük ki a megrendelőpontokba szállítandó r_i mennyiségekkel, és a t_1, t_2,...t_l terhelhetőség szerint csökkenő sorba rendezett a J_1, J_2,...,J_l járműveket és az m_k terhelésüket tartalmazó *7. táblázattal*.

6. táblázat

	P_1	P_2	P_3	P_4	P_5	P_6	r_i
P_1			157				4
P_2	450^3						3
P_3				62			4
P_4						38	2
P_5		696^3					3
P_6					835^2		2

7. táblázat

Jármű	J_1	J_2	J_3	J_4	J_5	J_6	J_7
t_k [t]	10	10	6	6	6	6	6
m_k [t]	2+3+3	0	0	0	0	0	0

Indítsuk el az első 10 t teherbírású járművet (J_1), és a körúton irányítsuk a legnagyobb megtakarítást eredményező élre, azaz keressük meg a *6. táblázatban* a körút legnagyobb költségelemét. Ez a P_6–P_5 élhez tartozó s_{65}=835. Vizsgájuk meg, hogy a J_1 járművel az 6. és 5. pontok összevont kiszolgálása megvalósítható-e. Ennek feltétele:

$$r_6 + r_5 \leq t_1 .$$

Ha a kapacitáskorlátra vonatkozó feltétel nem teljesül, akkor nincs lehetőség a P_0–P_6–P_0 és a P_0–P_5–P_0 utak összevonására. Mivel ebben az esetben a P_6 és a P_5 fogyasztók kiszolgálását nem lehet összevont járatba szervezni, a járatösszevonást elvetjük. Ekkor a megtakarítási mátrixban az s_{65} és a s_{56} értékét 0-ra állítjuk, és új maximális súlyú körutat keresünk.

Mivel azonban a kapacitáskorlátra vonatkozó feltétel teljesül, ezért a P_6–P_5 élre a 6. fogyasztóhoz szállítandó mennyiséget, r_6=2 tonnát, a P_5 ponthoz közvetlen kapcsolódó P_5–P_2 élre pedig az 5. fogyasztóhoz szállítandó r_5=3 t mennyiséget programozzuk. A programozott mennyiségeket a szállítási feladatok megoldásánál alkalmazott szintaxist követve a felső indexbe írjuk (*6. táblázat*).

A programozással a 6. és az 5. pontok felkeresését összevontuk, és a P_6–P_5 élet hozzárendeltük az első járathoz. Az összevonás eredményeként kialakult P_0–P_6–P_5–P_0 járattal elérhető útmegtakarítás 835 km, és a J_1 jármű terhelése m_1=2+3=5 t, amit a *7. táblázat* második oszlopában tüntettünk fel. Ezt követően az 5. oszlopot és 6. sort lefedjük (inaktívvá tesszük), mivel az P_5 pontba a P_6-ból vezet él, a P_6 pontból pedig már nem indulhat más él.

Az összevonást követően a J_1 jármű még nem telített, ezért próbáljuk meg a járatot bővíteni. A járat bővítést a körúton a P_6–P_5 élet megelőző P_4–P_6, vagy követő P_5–P_2 élek bevonásával folytathatjuk. Azt az élet választjuk, amelyhez nagyobb s_{ij} érték tartozik, vagyis amelytől nagyobb megtakarítást remélhetünk, és amely eleget tesz a járműterhelési feltételnek is.

Mivel az

$$s_{52}=696>s_{46}=38,$$

továbbá az

$$r_6+r_5+r_2=2+3+3=8\le 10,$$

a P_5–P_2 élet is a járathoz adjuk, és a P_2 ponthoz kapcsolódó P_2–P_1 élre r_2=3 t mennyiséget programozunk. Fedjük le az 5. sort és a 2. oszlopot, mivel a P_2 pontba a P_6–P_5–P_2 él sorozat vezet (*6. táblázat*). A kialakult járat élei: P_0–P_6–P_5–P_2–P_0, az útmegtakarítás 835+696= 1531 km, és a gépkocsi terhelése m_1=2+3+3=8 t (*7. táblázat*).

Ezt követően kíséreljük meg a járatot a kapcsolódó, P_4–P_6 megelőző, vagy a P_2–P_1 követő élekkel bővíteni. Ismét a nagyobb útmegtakarítást eredményező élet választjuk,

$$s_{21}=450>s_{46}=38,$$

és megvizsgáljuk, hogy a P_2–P_1 él hozzáadható-e a járathoz. Ennek feltétele:

$$r_6+r_5+r_2+r_1<=10.$$

Tekintettel arra, hogy $r_6+r_5+r_2+r_1$=2+3+3+4=12, a kapacitáskorlátra vonatkozó feltétel nem teljesül, ezért a P_2–P_1 él járatba szerkesztését elvetjük. Az első járatot ezért a P_2 pontnál lezárjuk. A lezárást követően a P_6 pontba már csak P_0-ból és a P_2 pontból pedig már csak a P_0-ba vezethet él, ezért lefedjük a 6. oszlopot és a 2. sort (*6. táblázat*). Ezzel az első járat szerkesztése befejeződött, az érintett pontok: P_0–P_6–P_5–P_2–P_0, a jármű terhelése m_1=8 t, és az elért megtakarítás:

$$s_{65}+s_{52}=835+696=1531 \text{ km}.$$

8. táblázat

	P_1	P_3	P_4
P_1	0	157	38
P_3	157	0	62
P_4	38	62	0

Az új, a második járat indítása előtt az *5. táblázatból* hagyjuk el a *6. táblázat* lefedett sorait és oszlopait, majd az így kapott maradék megtakarítási mátrixon (*8. táblázat*) kijelöljük a maximális súlyú körutat. Ennek eredményét mutatja a *9. táblázat*, amelynek utolsó oszlopába a megrendelőpontokba szállítandó r_i mennyiségeket is beírtuk.

Vegyük a következő 10 tonnás járművet (J_2), és az előzőleg követett eljárást ismételjük meg az új körút még nem programozott fedetlen élein. Keressük meg a körút legnagyobb súlyú, még nem programozott, fedetlen élét, ez a P_1–P_3, amelynek a súlya s_{13}=157 (*9. táblázat*).

9. táblázat

	P_1	P_3	P_4	r_i
P_1		157^4		4
P_3			62^4	4
P_4	38^2			2

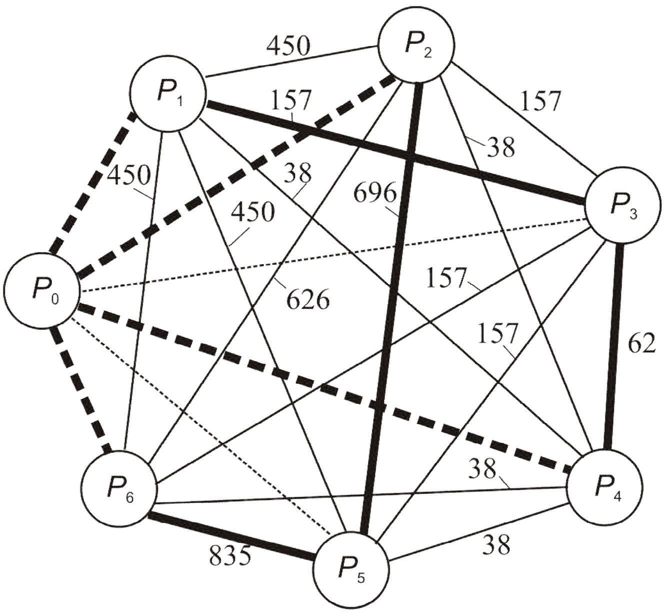
Mivel az

$$r_1 + r_3 = 4 + 4 \leq t_2 = 10\,,$$

a P_1–P_3 élre r_1=4 t és a P_3 ponthoz közvetlen kapcsolódó P_3–P_4 élre a 3. fogyasztóhoz szállítandó r_3=4 t mennyiséget programozunk, majd lefedjük le a P_1 sort és a P_3 oszlopot (*9. táblázat*). Ezzel a 1. és az 3. pontok felkeresését összevontuk, és a P_1–P_3 élet hozzárendeltük a második járathoz. Az így kialakult P_0–P_1–P_3–P_0 járattal elérhető útmegtakarítás 157 km, és a J_2 jármű terhelése m_2=4+4=8 t lesz (*10. táblázat*).

10. táblázat

Jármű	J_1	J_2	J_3	J_4	J_5	J_6	J_7
t_k [t]	10	10	6	6	6	6	6
m_k [t]	2+3+3	4+4+2	0	0	0	0	0



3. ábra. A mintapélda megoldása a szállítási hálózaton (Forrás: saját szerkesztés)

A második járat bővítését a körút a P_1–P_3, élet megelőző vagy követő, járatba még nem kapcsolt fedetlen éleivel folytathatjuk. A megelőző P_2–P_1 él már lefedett (a P_2 pontot az első járat már érintette), ezért csak az s_{34}=62 súlyú P_3–P_4 élet választhatjuk. Az él kapcsolásának feltétele:

$$r_1 + r_3 + r_4 = 4 + 4 + 2 \leq t_2 = 10$$

teljesül, ezért a P_3–P_4 élet hozzáadjuk a második járathoz. A P_4 ponthoz közvetlen kapcsolódó, s_{41}=38 súlyú P_4–P_1, élre r_4=2 t mennyiséget programozunk (*9. táblázat*). A második járat P_1–P_3–P_4 éleinek megfelelően a 3. sort és az 4. oszlopot lefedjük. A második jármű terhelése így r_1+r_3+r_4=4+4+2=10 t, vagyis telítődött, ezért a P_4 pontnál a második járatot is lezárjuk, és egyidejűleg lefedjük a 4. sort. Értelemszerűen a P_4–P_1 él már nem része a második járatnak. A második járat által érintett pontok: $P_0–P_1–P_3–P_4–P_0$, a J_2 jármű terhelése m_2=4+4+2=10 t, és az elért megtakarítás:

$$s_{13}+s_{34}=157+62=219 \text{ km.}$$

A kiszállítandó mennyiségeket a két járatra teljesen elosztottuk, így a feladat megoldása befejeződött. A szerkesztett járatok:

$$P_0–P_6–P_5–P_2–P_0 \text{ és } P_0–P_1–P_3–P_4–P_0.$$

Az összes útmegtakarítás 1531+219 = 1750 km. A megoldás grafikusan a *3. ábrán* látható, ahol a vastagvonallal rajzolt élek szemléltetik a járatokat.

3. Az algoritmus leírása

Legyen $G=(P, E)$ egy összefüggő, irányítatlan gráf, ahol a P a fogyasztókat jelképező csomópontok $n+1$ elemű halmaza, az E pedig a csomópontokat összekötő élek halmaza. A P halmaz elemeit jelölje p_i (i=1,2,...,n), az E halmaz elemeit pedig e_{ij} (i=j=1,2,...,n). Ha a p_i össze van kötve p_j-vel, akkor e_{ij}=1, különben e_{ij}=0. A csomópontokhoz r_i mennyiségek (a fogyasztók igényei), E minden eleméhez pedig c_{ij} költségek tartoznak (ezek a távolságmátrix elemei).

Legyen J a rendelkezésre álló járművek halmaza, amelynek minden j_k (k=1,2,... ,l) eleméhez hozzárendeljük a járműveket jellemző t_k teherbírás- és az m_k terhelésvektorokat.

1. A távolságmátrixból az

$$s_{ij} = c_{0i} + c_{0j} - c_{ij}, \text{ ha } e_{ij} = 1,$$

illetve az

$$s_{ij} = 0, \text{ ha az } e_{ij} = 0$$

képletekkel kiszámítjuk az s_{ij} megtakarítási mátrix elemeit, és azokat súlyként rendeljük az $M=(P^*,S)$ gráf S éleihez, amelyben a P^* a csomópontok (fogyasztók) n elemű halmaza.

2. Rendezzük a J halmazt a t_k teherbírás szerint csökkenő sorrendbe.

3. Az $M=(P^*,S)$ gráf aktív élein jelöljük ki a maximális súlyú körutat, ez legyen a $K=(P^*,S^*)$, ahol S^* a körutat alkotó k_{ij} élek n elemű halmaza, és a k_{ij} élek mindegyikéhez s_{ij} súlyok rendeltek.

4. A k-adik iterációban vegyük a k-adik üres járművet, amelynek teherbírása t_k.

5. Keressük meg a K körút aktív k_{ij} éleihez tartozó s_{ij} elemek közül a legnagyobbat

$$s_{\beta\gamma} = \max\{s_{ij}\}.$$

Ha nincs aktív k_{ij} él, akkor az eljárás végére értünk, különben a 6. lépéssel folytatjuk. Az él akkor aktív, ha az i-edik sor és a j-edik oszlop fedetlen.

6. Vizsgáljuk a kapacitáskorlátra vonatkozó $r_\beta+r_\gamma<t_k$ feltétel teljesülését.

Ha a feltétel teljesül, akkor a $k_{\beta\gamma}$ élre r_β, a $k_{\gamma\delta}$ élre r_γ mennyiséget programozunk, (ahol $k_{\gamma\delta}$ a γ ponthoz kapcsolódó él az K körúton), a $k_{\beta\gamma}$ élet a járathoz adjuk. A k-adik jármű terhelését $m_k:=r_\beta+r_\gamma$-ra állítjuk, kiszámítjuk az útmegtakarítást: $s_k=s_{\beta\gamma}$, lefedjük a β sort és a γ oszlopot, majd az 7. lépéssel folytatjuk.

Különben a járatösszevonást elvetjük (ez azt jelenti, hogy a β és a γ fogyasztók kiszolgálását nem lehet összevont járatba szervezni). Legyen az

$$s_{\beta\gamma} := 0 \text{ és az } s_{\gamma\beta} := 0,$$

majd visszatérve a 3. lépéshez, új körutat keresünk.

7. Megvizsgáljuk a k-adik jármű telítettségét. Ha az $m_k<t_k$, akkor a járatot az K körút $k_{\beta\gamma}$ élét megelőző ($k_{\alpha\beta}$) vagy követő ($k_{\gamma\delta}$) éllel bővíthetjük, feltéve, hogy az él aktív, és a kapacitáskorlátra vonatkozó feltétel is teljesül.

Ha az

$$s_{\gamma\delta}>s_{\alpha\beta}, \text{ és az } r_\beta+r_\gamma+r_\delta<t_k,$$

akkor a $k_{\gamma\delta}$ élet hozzávesszük a járathoz, és a δ ponthoz kapcsolódó követő élre r_δ mennyiséget programozunk. A jármű terhelését r_δ mennyiséggel növeljük:

$$m_k:=r_\beta+r_\gamma+r_\delta,$$

és kiszámítjuk az útmegtakarítást:

$$s_k^\xi:=s_k^{\xi-1}+s_{\gamma\delta},$$

ahol ξ a 7. lépés ciklusváltozója. A $k_{\gamma\delta}$ élet inaktívvá tesszük, lefedjük a γ sort és a δ oszlopot.

Ha az

$$s_{\gamma\delta}<s_{\alpha\beta}, \text{ és az } r_\beta+r_\gamma+r_\alpha<t_k,$$

akkor a $k_{\alpha\beta}$ élet hozzávesszük a járathoz, és a β ponthoz kapcsolódó megelőző élre r_α mennyiséget programozunk. A jármű terhelését r_α mennyiséggel növeljük:

$$m_k:=r_\beta+r_\gamma+r_\alpha,$$

és kiszámítjuk az útmegtakarítást:

$$s_k^\xi:=s_k^{\xi-1}+s_{\alpha\beta},$$

lefedjük a α sort és a β oszlopot.

A 7. lépést addig ismételjük, amíg a kapacitáskorlátra vonatkozó feltétel teljesül. Az ismétlés előtt a ciklusváltozót növeljük: $\xi:=\xi+1$. Ha a jármű telítődött, akkor $s_k:=s_k^\xi$, és a 8. lépéssel folytatjuk.

8. A k-adik járatot lezárjuk, amelyben a felkeresett csomópontok (u=1,2,..ρ), a szállított mennyiség

$$m_k:=\sum_{u=1}^{\rho} r_u,$$

az elért útmegtakarítás s_k. Végül lefedjük a járat első csomópontjával azonos oszlopot és az utolsó csomópontjával azonos sort. A lefedett sorokban és oszlopokban az s_{ij} értékét 0-ra változtatjuk.

9. A ciklusváltozót növeljük $k:=k+1$, és visszatérünk a 3. lépéshez. (Új körutat keresünk!)

4. Következtetések

A mintapéldában a járatok szerkesztésekor csak a járművek kapacitáskorlátait vettük figyelembe, de mint utaltunk rá, annak sincs akadálya, hogy egyéb megszorításokat tegyünk, például, hogy a jármű által megtehető utat vagy időt a kapacitáskorláthoz hasonlóan az útösszevonás feltételeként vizsgáljuk. Az sem szorul különösebb magyarázatra, hogy az algoritmus nemcsak elosztási, hanem gyűjtőjáratok, továbbá személyszállítást végző autóbuszok járatainak szerkesztésére is alkalmas.

IRODALOM

Baker, Barrie M, and Ayechew, M. A. (2003): A Genetic Algorithm for the Vehicle Routing Problem. Computers & Operations Research, 30: 787-800.

Benkő, J. (2000): Logisztikai tervezés. Dinasztia Kiadó, Budapest, ISBN 963 657 271 2, 200 p.

Chakroborty, P. and Dwivedi, T. (2002): Optimal route network design for transit systems using genetic algorithms. Engineering Optimization, (1), 83-100.

Christian Prins (2004): A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem. Computers & Operations Research, 31: 1985-2002.

Laporte, G. (1992): The Vehicle Routing Problem: an Overview of Exact and Approximate Algorithms. European Journal of Operational Research, 59: 345-358.

Publikálva:

Logisztikai Évkönyv 2014, Magyar Logisztikai Egyesület, 2014. 22-32 p.