Algoritmus a járatszerkesztés számításigényének csökkentésére

DR. BENKŐ JÁNOS
GATE, Géptani Intézet

A járatszerkesztés modelljét és a megoldás algoritmusát az 1960-as évek elején az *ATUKI* megbízásából *Krekó Béla* és *Szántó Emil* dolgozta ki [2]. A feladat, mint ismeretes a szállítójárművek, targoncák stb. üresfutásának és az ún. garázsmeneteknek a minimálását teszi lehetővé.

Az *üresfutások minimálására* törekvő feladat gyakorlati megfogalmazása a következő:

Adott n számú állomás ($A_1, A_2, ..., A_n$). Az A_i állomások egyidejűleg egyaránt lehetnek feladók és megrendelők. A j-dik állomás, mint megrendelő y_{ij} számú járat érkezését igényli az i-dik feladótól. Ezenkívül, ismeretes az állomások közötti szállítási távolság vagy a c_{ij} fajlagos szállítási költség. Acél a járművek útvonaltervének kialakítása, amellyel a lehető legkevesebb üresfutással kerül az áru a feladó állomásról a megrendelő állomásra.

Könnyen belátható, hogy a kitűzött célt akkor érjük el, ha az üres meneteket úgy osztjuk el az egyes szállítási viszonylatokra, hogy az x_{ij} üres menetek száma és a c_{ij} szállítási távolságok szorzatainak összege minimális, azaz

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\, x_{ij} \to \min.$$

Látható, hogy ha valamennyi állomáson azonos az elindított és az oda érkező járatok száma, akkor nincs üres menet, és így a feladatnak nincs értelme. A legkedvezőtlenebb esetben viszont minden rakott menetet üres menet követ.

A feladat −ma már klasszikusnak nevezhető− megoldása két lépésből áll. A feladatot először a *szállítási problémának* kezeljük. Az első lépésben programozzuk az üres meneteket. A második lépés a járatkapcsolás, vagyis a rakott és az üres menetek összekötése.

A klasszikus módszer szerint az első lépésben egy olyan szállítási feladatot kell megoldani, amelyben a készletek a feladóhelyekről indítandó, az igények pedig a megrendelőhelyre érkező járatoknak felelnek meg. A távolságmátrix főátlójában csak 0 elemek szerepelnek, így a megoldás után a menetek nagy része a főátlóra kerül. A főátlón kívüli relációkra programozott menetek az üres menetek számát adják, amelyek a mátrix transzponálásával kerülnek a tényleges helyükre. Ezután a járatkapcsoláshoz szükséges ún. *munkamátrixot* a rakott és az üres menetek transzponált mátrixának egymásra illesztésével kapjuk meg.

A javasolt módszer. A továbbiakban ehhez az eljáráshoz hasonló, de ennél sokkal kevesebb számítást igénylő, és talán közérthetőbb algoritmust ismertetünk. Tekintsük először az algoritmust:

(1) $$y_{ij} \ge 0,\ x_{ij} \ge 0,\ i = j = 1, 2, \ldots, n,$$

(2) $$d_k = \sum_{j=1}^{n} y_{kj} - \sum_{i=1}^{n} y_{ik},\ k = 1,2,...,n,$$

(3) $$r_k = \begin{cases} d_k, \text{ ha a } d_k > 0 \\ 0, \text{ ha a } d_k \le 0 \end{cases},$$

(4) $$t_k = \begin{cases} |d_k|, \text{ ha a } d_k < 0 \\ 0, \text{ ha a } d_k \ge 0 \end{cases},$$

(5) $$\sum_{j=1}^{n} x_{kj} = t_k,$$

(6) $$\sum_{i=1}^{n} x_{ik} = r_k,$$

(7) $$c_{ij} = M, \text{ ha } i = j,$$

(8) $$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\, x_{ij} \to \min,$$

ahol: n az állomások száma,
y_{ij} a rakott menetek száma az ij relációban,
x_{ij} az üres menetek száma az ij relációban,
d_k a k-adik állomásról indított és a k-adik állomásra érkező rakott menetek számának a különbsége,
t_k a k-adik állomásról indított üres menetek száma,
r_k a k-adik állomásra érkező üres menetek száma,
c_{ij} a fajlagos szállítási költség.

Az algoritmust összehasonlítva a klasszikus eljárással, az alapvető különbség az *üres menetek számának meghatározásában* van, és mint látni fogjuk, ez lényegesen egyszerűsíti azok *elosztását* is.

Az algoritmus szerint először a (2), (3) és (4) összefüggésekkel kiszámítjuk az üres menetek számát. A rakott menetek **Y** mátrixát soronként és oszloponként összegezzük, majd a k-dik sorösszegből levonjuk a k-dik oszlop összegét. A d_k differenciák előjele a (3) és (4) feltételeknek megfelelően megmutatja, hogy az üres menetet a k-dik állomásról kell-e indítani, vagy a k-dik állomásra kell-e teljesíteni. A $d_k>0$ azt jelenti, hogy a k-dik állomásról több rakott menetet indítanak, mint amennyi oda érkezik. Értelemszerűen ezért a k-dik állomásra $r_k=d_k$ alkalommal kell üresen mennie a járműnek. A $d_k<0$ esetben fordított a helyzet, ilyenkor a k-dik állomásról $t_k = |d_k|$ alkalommal üresen kell indítani a járatot. Ha $d_k=0$, akkor magától értetődően nincs üres járat, vagyis $r_k=t_k=0$.

Az üres menetek számának meghatározása. után valamely ismert algoritmussal (pl. progresszív módszer, *Vogel-Korda-módszer* stb.) megoldjuk az (5), (6), (7) és (8) feltételeket, illetve a célfüggvényt kielégítő szállítási feladatot. Az új eljárás előnye tulajdonképpen itt jelentkezik, mivel a **C** távolságmátrix ama sorait, illetve oszlopait, ahol $t_k=0$, illetve $r_k=0$ elhagyhatjuk. Így az eredetinél jóval kisebb méretű szállítási feladatot kell megoldani. A legkedvezőtlenebb esetben is n-ről n/2-re csökken a távolságmátrix rendje.

Az előnyök szemléltetésére tekintsünk egy, az irodalomban is megtalá1ható *példát* [1], így az érdeklődő olvasónak alkalma. nyílik a régi és az új eljárás idő- és számításigényének az összehasonlítására.

Példa. Egy üzemben öt, P_1, P_2,...,P_5 munkahely között a szállítást targoncával kívánják megoldani. Az üzemek között teljesítendő rakott menetek számát tartalmazó **Y** mátrixot az 1. *táblázat* mutatja.

1. táblázat

A rakott menetek **Y** *mátrixa*

			H	*o*	*v*	*a*	
H		P_1	P_2	P_3	P_4	P_5	$\Sigma_j y_{kj}$
o	P_1	–	2	3	1	–	6
n	P_2	–	–	2	3	4	9
n	P_3	–	4	–	1	2	7
a	P_4	2	2	1	–	1	6
n	P_5	–	1	2	4	–	7
	$\Sigma_i y_{ik}$	2	9	8	9	7	35

Az üzemek közötti szállítási távolságokat a *2. táblázat* szemlélteti.

2. táblázat

A szállítási távolságok **C** *mátrixa*

		H	*o*	*v*	*a*	
H		P_1	P_2	P_3	P_4	P_5
o	P_1	0	8	10	5	15
n	P_2	8	0	4	3	10
n	P_3	10	4	0	9	8
a	P_4	5	3	9	0	6
n	P_5	15	10	8	6	0

Képezzük a (2) összefüggésnek megfelelő különbségeket, és határozzuk meg (3), (4) feltételek alapján az elosztandó üres meneteket:

$$
\begin{array}{lll}
d_1=4 & r_1=4 & t_1=0 \\
d_2=0 & r_2=0 & t_2=0 \\
d_3=-1 & r_3=0 & t_3=1 \\
d_4=-3 & r_4=0 & t_4=3 \\
d_5=0 & r_5=0 & t_5=0
\end{array}
$$

Ezek után felírhatjuk a szállítási feladat mátrixát (*3. táblázat).* Ha azokat a sorokat, illetve oszlopokat, ahol t_k=0, illetve r_k=0, elhagyjuk, a szállítási feladat egy 2X1-es mátrixra egyszerűsödik, amelynek a megoldása egyértelműen adódik, éspedig:

$$x_{31}=1,\ x_{41}=3.$$

Ezt a *3. táblázatban* a megfelelő elemek jobb felső részében tüntettük fel.

3. táblázat

A szállítási feladat redukált mátrixa

		H	*o*	*v*	*a*		
H		P_1	P_2	P_3	P_4	P_5	t_k
o	P_1						
n	P_2						
n	P_3	10^{1}					1
a	P_4	5^{3}					3
n	P_5						
	r_k	4					

Az üres menetek hossza:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij} = 25.$$

A járatkapcsoláshoz szükséges mátrixot a rakott és az üres menetek mátrixának összegzéséből kapjuk (*4. táblázat),* azaz:

$$\mathbf{Y_m = Y + X.}$$

A rakott és az üres menetek összes hossza:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(y_{ij} + x_{ij}) = 255,$$

vagyis az összes szállítás el végzéséhez 255 távolságegységet kell megtenni, és ebb ől 25 távolságegységet üresen.

4. táblázat

Munkamátrix a járatkapcsoláshoz

		H	*o*	*v*	*a*	
H		P_1	P_2	P_3	P_4	P_5
o	P_1	-	2	3	1	-
n	P_2	-	-	2	3	4
n	P_3	1	4	-	1	2
a	P_4	5	2	1	-	1
n	P_5	-	1	2	4	-

A járatkapcsolástól, mivel az intuitív jelleg ű, és az egyéb korlátozó feltételek is rendkívül változatosak lehetnek, eltekintünk.

IRODALOM

[1] Felföldi L.: Anyag mozgatá si folyamatok tervezése. Műszaki Könyvkiadó, Bp. 1976.
[2] Szántó E.: A körutazási és járatszerkesztési modell. KÖZDOK, 1972.

Publikálva:

A+CS, 1987. XXXII. évfolyam - 5. szám 134-136 p.