

# A körutazási feladat megoldása dinamikus programozással

**DR. BENKŐ JÁNOS**

egyetemi adjunktus GATE, Mezőgazdasági Géptani Intézet

*A járművek és targoncák optimális útvonalának meghatározása gyakran vezethető vissza a körutazási problémára. A szerző ennek megoldására közöl új eljárást, amelynek egyik előnye az ismert eljárásokkal szemben, hogy a körutazási feltétel mellett egyéb járulékos megszorítások figyelembevételét is lehetővé teszi.*

A klasszikus értelemben vett tiszta körutazási feladat a szállításszervezési problémákban viszonylag ritkán fordul elő. A tiszta modell formájában általában csak valamely komplex probléma részletkérdései fogalmazhatók meg, vagy pedig további járulékos megszorításokat igényel a feladat megoldása.

A probléma legegyszerűbb megfogalmazása a következő. Adott  $n$  város, ezeket jelöljük:  $P_1, P_2, \dots, P_n$  szimbólumokkal. Az  $i$ -edik és a  $j$ -edik város közötti távolság  $c_{ij}$ . A feladat, valamilyik városból, mondjuk  $P_1$ -ből elindulva ellátogatni minden városba egyszer, de csakis egyszer, majd visszatérni  $P_1$ -be. Kérdés, a városok meglátogatásának mely sorrendje biztosítja a legrövidebb útvonalat.

Az ismertett probléma feltételrendszere a hozzárendelési feltételből:

$$\sum_i x_{ij} = \sum_j x_{ij} = 1, \quad x_{ij} \in \{0,1\} \text{ minden } i, j \text{-re,}$$

és a körutazási feltételből:

$$I = \{i_1 i_2, i_2 i_3, \dots, i_{n-1} i_n, i_n i_1\}$$

áll. Az utóbbi azt írja elő, hogy az 1 értékű változók indexei  $n$  elemű indexláncot alkotnak. E feltételek mellett kell meghatározni a

$$K = \sum_i \sum_j c_{ij} x_{ij} \rightarrow \min$$

célfüggvényt.

Ez a kombinatorikus feladat régóta foglalkoztatja a matematikusokat és a gyakorlati szakembereket. A probléma megfogalmazható integer lineáris programozási feladatként, azonban ezzel szemben komoly kifogásként merülhet fel, hogy nem túl nagy  $n$  esetén is meglehetősen sok változójú *ILP*-t kapunk [4]. Ez indokolja, a feladat szerkezetéhez igazodó, kevesebb számítást igénylő módszerek kifejlesztését. Sok olyan algoritmus ismeretes, amelyek az esetek nagy számában optimumot szolgáltatnak. Ezek többsége a korlátozás és szétválasztás módszerét alkalmazza, amelynek sikere nagymértékben függ az alsó korlát helyes megválasztásától. Ilyen algoritmusok például *Little*, *Murty*, *Sweeney* és *Karel* algoritmusai [8], valamint a *Held* és *Karp* által kidolgozott eljárás [5]. A továbbiakban egy ezektől eltérő, a szekvenciális optimalizációs tételre épülő algoritmust mutatok be [2]. Ennek előnyeként említhető, hogy a körutazási feltétel mellett egyéb járulékos megszorítások is nagyon egyszerűen figyelembe vehetők.

## A szekvenciális optimalizálás

A műszaki-gazdasági szélsőérték feladatok megoldásának egyik leghatékonyabb eszköze a szekvenciális optimalizálás, amit gyakran dinamikus programozásnak neveznek. Az optimalitási tételre alapuló módszert *R. Bellman* [1] amerikai matematikus dolgozta ki. A tétel diszkrét, determinisztikus rendszerekre a következők szerint értelmezhető [6].

Legyen egy rendszerünk, amelynek állapota az egyes  $k$  fázisokban ( $k=0, 1, \dots, n$ ) döntés következtében megváltozhat. A rendszer a  $k$  fázisokban megszámlálható számú állapotban lehet. Ha a döntések egy bizonyos sorozatát  $k=0$ -tól  $k=n$ -ig politikának, a kapcsolódó döntések sorozatát pedig, amelyek valamely politika részei alpolitikának nevezzük, akkor igaz a következő tétel: Egy optimális politika csak optimális alpolitikákból állhat. Ez könnyen belátható a következő gondolatmenet alapján:

Tekintsünk egy alpolitikát, amely valamely optimális politika része. Ha ez az alpolitika nem lenne optimális, akkor lenne egy jobb alpolitika, amely lehetővé tenné a politika javítását, ez pedig ellentmondana a hipotézisünknek.

A módszer alkalmazásának feltétele, hogy a  $\mathbf{p}$  politikát fel tudjuk bontani

$$(p_0, p_1), (p_1, p_2), \dots, (p_{n-1}, p_n)$$

alpolitikákra úgy, hogy a  $K$  költség csupán az egyes állapotok kezdő- és végpontját jellemző  $(p_{i-1}, p_i)$  paraméterektől függő

$$u_i(p_{i-1}, p_i)$$

rész költségek összegére bomlik, azaz

$$K(\mathbf{p}) = u_1(p_0, p_1) + u_2(p_1, p_2) + \dots + u_n(p_{n-1}, p_n).$$

Tekintsük a következő  $n \neq 1$  változós függvényt:

$$F(\mathbf{x}) = u_1(x_0, x_1) + u_2(x_1, x_2) + \dots + u_i(x_{i-1}, x_i) + u_n(x_{n-1}, x_n)$$

Ez a függvény felbontható  $n$  számú elemi függvény összegére. Keressük az  $F$  függvény minimumát, tudva, hogy minden  $x_i$  olyan tartományban változhat, amely csak  $x_0$ -tól és  $x_{i+1}$ -től függ, akármilyen értéket vesz fel  $i$  az 1 és  $n$  között.

Nézzük az 1 és 2 fázisokat együttesen, nevezzük  $f_{02}(x_0, x_2)$ -nek az  $u_1(x_0, x_1) + u_2(x_1, x_2)$  összeg optimális értékét, ekkor  $x_1$ -et változtatjuk a tartományában, amely csak  $x_0$ -tól és  $x_2$ -től függ. Tehát minimum keresés esetén a következő képletet kapjuk:

$$f_{02}(x_0, x_2) = \min\{u_1(x_0, x_1) + u_2(x_1, x_2)\},$$

ahol  $x_1 \in X_1(x_0, x_2)$ , ami azt jelenti, hogy  $x_1$  az  $X_1$  értékalmazhoz tartozik, ez pedig csak  $x_0$ -tól és  $x_2$ -től függ. Az  $x_1$ -nek az az értéke, amely optimálissá teszi az  $u_1(x_0, x_1) + u_2(x_1, x_2)$  összeget, egyben az optimális alpolitikát is meghatározza az 1 és 2 fázisokban a vizsgált  $x_0$  és  $x_2$  pár esetében.

Vegyük most az 1, 2 és 3 fázisokat együtt, és nevezzük  $f_{03}(x_0, x_3)$ -nak az  $u_1(x_0, x_1) + u_2(x_1, x_2) + u_3(x_2, x_3)$  összeg optimális értékét, amikor  $x_1$ -et és  $x_2$ -t változtatjuk a tartományaikban. Az optimalitási tétel alapján felírhatjuk:

$$f_{03}(x_0, x_3) = \min\{f_{02}(x_0, x_2) + u_3(x_2, x_3)\},$$

ahol  $x_2 \in X_2(x_0, x_3)$ .

Általánosan:

$$f_{0i}(x_0, x_i) = \min\{f_{0i-1}(x_0, x_{i-1}) + u_i(x_{i-1}, x_i)\},$$

ahol  $x_{i-1} \in X_{i-1}(x_0, x_i)$  és  $f_{01}(x_0, x_1) = u_1(x_0, x_1)$ .

A fenti formula rekurzív alkalmazása lehetővé teszi, hogy kiszámítsuk az egymás után következő optimális alpolitikákat az 1 és 2 fázisokat együttvéve, az 1, 2 és 3 fázisokat együttvéve, ..., az 1, 2, ...,  $i$ , ...,  $n$  fázisokat együttvéve. Az optimális politikát a

$$F(x_0, x_n) = \min \{f_{0n-1}(x_0, x_{n-1}) + u_n(x_{n-1}, x_n)\}$$

képlet adja.

### Az új eljárás algoritmus

A bevezetőben használt jelöléseket megtartva, jelentse  $c_{ij}$  a  $P_i$ -ből a  $P_j$ -be mutató él költségét, ahol  $i=1, 2, \dots, n$ ,  $j=1, 2, \dots, n$ . Legyen  $a_j^{(k)}$  a valamely  $P_i^{(1)}$  csúcsból a  $P_j^{(k)}$  csúcsba mutató,  $k$  élt tartalmazó legrövidebb út hossza, amely út ugyanazt a csúcsot csak egyszer érinti. A fázisok száma:  $k=1, 2, \dots, n$ .

Ha nincs ilyen út, akkor az

$$a_j^{(k)} = M,$$

ha  $k=1$ , akkor az

$$a_j^{(1)} = \min \{c_{ij}\},$$

különben az

$$a_j^{(k+1)} = \min \{a_i^{(k)} + c_{ij}\}.$$

A körutazási feltételt a  $k+1$ -edik fázisban úgy biztosítjuk, hogy az  $a_i^{(k)}$  helyére  $M$ -t írunk, amennyiben az  $a_i^{(k)}$  hosszúságú út a  $j$  csúcsot tartalmazza. Így rekurzíven az  $a_i^{(k)}$  értékeket meghatározva, legfeljebb  $k=n-1$  lépés után megkapjuk a  $P_i^{(1)}$ -ből a  $P_j^{(n-1)}$ -be vezető,  $n-1$  élt tartalmazó legrövidebb utakat, amelyekben valamennyi csúcs különböző, feltéve, hogy ilyen utak léteznek.

Az  $n$ -edik lépésben a legrövidebb utakhoz hozzáadjuk a  $P_j^{(n-1)}$ -ből a  $P_i^{(1)}$ -be mutató élt:

$$a_i^{(n)} = a_j^{(n-1)} + c_{ji}.$$

Így a  $P_i^{(1)}$ -ből induló és a  $P_i^{(1)}$ -be vezető körutakat kapunk, amelyek közül a legrövidebb a körutazási feladat megoldása.

### Az eljárás alkalmazása

Az ismertett módszer gyakorlati alkalmazását egy olyan mintapéldán mutatom be, amelynél a *Little* és társai által javasolt algoritmus elakad. Legyen a költségmátrix:

$$C = \begin{vmatrix} M & 4 & 5 & M & M & M \\ 4 & M & 6 & 6 & M & M \\ 5 & 6 & M & M & 7 & M \\ M & 6 & M & M & 10 & 11 \\ M & M & 7 & 10 & M & 12 \\ 7 & M & M & 11 & 12 & M \end{vmatrix}$$

Először az

$$a_j^{(1)} = \min \{c_{ij}\}$$

képletet alkalmazva meghatározzuk, hogy az 1. fázisban a  $P_j$  állomások mely állomásokról érhetőek el a legrövidebb úton:

$$\begin{aligned} a_1^{(1)} &= \min\{M, 4, 5, M, M, 7\} = c_{21} = 4 \\ a_2^{(1)} &= \min\{4, M, 6, 6, M, M\} = c_{12} = 4 \\ a_3^{(1)} &= \min\{5, 6, M, M, 7, M\} = c_{13} = 5 \\ a_4^{(1)} &= \min\{M, 6, M, M, 10, 11\} = c_{24} = 6 \\ a_5^{(1)} &= \min\{M, M, 7, 10, M, 12\} = c_{35} = 7 \\ a_6^{(1)} &= \min\{M, M, M, 11, 12, M\} = c_{46} = 7 \end{aligned}$$

$P_j$ -be vezető utak és a hozzájuk tartozó indexláncok:

$$\begin{aligned} a_1^{(1)} &\rightarrow \{21\} \\ a_2^{(1)} &\rightarrow \{12\} \\ a_3^{(1)} &\rightarrow \{13\} \\ a_4^{(1)} &\rightarrow \{24\} \\ a_5^{(1)} &\rightarrow \{35\} \\ a_6^{(1)} &\rightarrow \{46\} \end{aligned}$$

A második fázistól az

$$a_j^{(k+1)} = \min\{a_i^{(k)} + c_{ij}\}.$$

formulát használjuk, ahol  $a_i^{(k)} = a_j^{(k)}$ .

$$\begin{aligned} a_1^{(2)} &= \min\{M, M+4, M+5, M, M, 11+7\} = a_6^{(1)} + c_{61} = 18 \\ a_2^{(2)} &= \min\{M+4, M, 5+6, M+6, M, M\} = a_3^{(1)} + c_{32} = 11 \\ a_3^{(2)} &= \min\{4+5, 4+6, M, M, M+7, M\} = a_1^{(1)} + c_{13} = 9 \\ a_4^{(2)} &= \min\{M, 4+6, M, M, 7+10, M+11\} = a_2^{(1)} + c_{24} = 10 \\ a_5^{(2)} &= \min\{M, M, 5+7, 6+10, M, 11+12\} = a_3^{(1)} + c_{35} = 12 \\ a_6^{(2)} &= \min\{M, M, M, 6+11, 7+12, M\} = a_4^{(1)} + c_{46} = 17 \end{aligned}$$

A körutazási feltételt az  $a_i^{(1)} = M$ -mel biztosítottuk, ahol az  $a_i^{(1)}$ -hez tartozó indexláncban szerepelt a  $j$ . Például az  $a_3^{(1)}$ -hez tartozó indexlánc 1-3, ezért a 1(2) sorában az  $a_3^{(1)} = M$ .

Az utakhoz tartozó indexláncok:

$$\begin{aligned} a_1^{(2)} &\rightarrow \{46, 61\} \\ a_2^{(2)} &\rightarrow \{13, 32\} \\ a_3^{(2)} &\rightarrow \{21, 13\} \\ a_4^{(2)} &\rightarrow \{12, 24\} \\ a_5^{(2)} &\rightarrow \{13, 35\} \\ a_6^{(2)} &\rightarrow \{24, 46\} \end{aligned}$$

A továbbiakban a formulát mechanikusan alkalmazzuk az  $n-1$ -edik fázisig.

$$a_1^{(3)} = \min\{ M, M+4, M+5, M, M, 17+7\} = a_6^{(2)} + c_{61} = 24$$

$$a_2^{(3)} = \min\{ 18+4, M, M+6, M+6, M, M\} = a_1^{(2)} + c_{12} = 22$$

$$a_3^{(3)} = \min\{ 18+5, M+6, M, M, M+7, M\} = a_1^{(2)} + c_{13} = 23$$

$$a_4^{(3)} = \min\{ M, 11+6, M, M, 12+10, M+11\} = a_2^{(2)} + c_{24} = 17$$

$$a_5^{(3)} = \min\{ M, M, 9+7, 10+10, M, 17+12\} = a_3^{(2)} + c_{35} = 16$$

$$a_6^{(3)} = \min\{ M, M, M, 10+11, 12+12, M\} = a_4^{(2)} + c_{46} = 21$$

$$a_1^{(3)} \rightarrow \{24, 46, 61\}$$

$$a_2^{(3)} \rightarrow \{46, 61, 12\}$$

$$a_3^{(3)} \rightarrow \{46, 61, 13\}$$

$$a_4^{(3)} \rightarrow \{13, 32, 24\}$$

$$a_5^{(3)} \rightarrow \{21, 13, 35\}$$

$$a_6^{(3)} \rightarrow \{12, 24, 46\}$$

$$a_1^{(4)} = \min\{ M, M+4, M+5, M, M, M+7\} = M$$

$$a_2^{(4)} = \min\{ M+4, M, 23+6, M+6, M, M\} = a_3^{(3)} + c_{32} = 29$$

$$a_3^{(4)} = \min\{ 24+5, 22+6, M, M, M+7, M\} = a_2^{(3)} + c_{23} = 28$$

$$a_4^{(4)} = \min\{ M, M+6, M, M, 16+10, M+11\} = a_5^{(3)} + c_{54} = 26$$

$$a_5^{(4)} = \min\{ M, M, 23+7, 17+10, M, 17+12\} = a_4^{(3)} + c_{45} = 27$$

$$a_6^{(4)} = \min\{ M, M, M, 17+11, 16+12, M\} = a_4^{(3)} + c_{46} = 28 \\ = a_5^{(3)} + c_{56} = 28$$

$$a_1^{(4)} \rightarrow M$$

$$a_2^{(4)} \rightarrow \{46, 61, 13, 32\}$$

$$a_3^{(4)} \rightarrow \{46, 61, 12, 23\}$$

$$a_4^{(4)} \rightarrow \{21, 13, 35, 54\}$$

$$a_5^{(4)} \rightarrow \{13, 32, 24, 45\}$$

$$a_6^{(4)} \rightarrow \{13, 32, 24, 45\} \\ \{21, 13, 35, 56\}$$

$$a_1^{(5)} = \min\{ M, M+4, M+5, M, M, M+7\} = M$$

$$a_2^{(5)} = \min\{ M+4, M, M+6, M+6, M, M\} = M$$

$$a_3^{(5)} = \min\{ M+5, M+6, M, M, M+7, M\} = M$$

$$a_4^{(5)} = \min\{ M, M+6, M, M, M+10, 28+11\} = a_6^{(4)} + c_{64} = 39$$

$$a_5^{(5)} = \min\{ M, M, 28+7, M+10, M, 28+12\} = a_3^{(4)} + c_{35} = 35$$

$$a_6^{(5)} = \min\{ M, M, M, 26+11, 27+12, M\} = a_4^{(4)} + c_{46} = 37$$

$$\begin{aligned}
 a_1^{(5)} &\rightarrow M \\
 a_2^{(5)} &\rightarrow M \\
 a_3^{(5)} &\rightarrow M \\
 a_4^{(5)} &\rightarrow \{21, 13, 35, 56, 64\} \\
 a_5^{(5)} &\rightarrow \{46, 61, 12, 23, 35\} \\
 a_6^{(5)} &\rightarrow \{21, 13, 35, 54, 46\}
 \end{aligned}$$

Amint látható, három  $n-1$  élt tartalmazó utat kaptunk, ezeket az utakat az  $n$ -edik fázisban az

$$a_i^{(n)} = a_j^{(n-1)} + c_{ji}.$$

formulával zárhatjuk le:

$$\begin{aligned}
 a_2^{(6)} &= a_4^{(5)} + c_{42} = 39 + 6 = 45, \\
 a_4^{(6)} &= a_5^{(5)} + c_{54} = 35 + 10 = 45, \\
 a_6^{(6)} &= a_6^{(5)} + c_{62} = 37 + M = M.
 \end{aligned}$$

A körutakhoz tartozó indexláncok:

$$\begin{aligned}
 a_2^{(6)} &= 45 \rightarrow \{21, 13, 35, 56, 64, 42\}, \\
 a_4^{(6)} &= 45 \rightarrow \{46, 61, 12, 23, 35, 54\}.
 \end{aligned}$$

Mind a két körút értéke 45, így mindkettő a feladat megoldása.

A bemutatott eljárás jó példája a dinamikus programozás rendkívül sokrétű gyakorlati alkalmazhatóságának. A mintapélda alapján meggyőződhattünk arról is, hogy az algoritmus alkalmazása nem igényel mélyebb matematikai ismereteket, az alapprobléma megoldásához elegendő a közölt képletek mechanikus használata. Egyéb járulékos feltételek esetén pedig úgy járunk el, hogy a feltételeknek meg nem felelő utakat tiltótarifával lezárjuk. Az eljárás további Előnyeként említhető: alternatív optimumok létezésekor valamennyi optimális körutat egyidejűleg megkapjuk.

## IRODALOM

1. **Bellman, R.:** Dynamic Programming. Princeton, 1957.
2. **Benkő J.:** Anyagmozgatási folyamatok tervezése. GATE egyetemi jegyzet, Gödöllő, 1989.
3. **Felföldi L.:** Anyagmozgatási folyamatok tervezése. Műszaki Könyvkiadó, Budapest, 1976.
4. **Forgó F.:** Nemkonvex és diszkrét programozás. Közgazdasági és Jogi Könyvkiadó, Budapest, 1978.
5. **Held, M.-Karp, R. M.:** The Travelling Salesman Problem and Minimum Spanning Trees. Operations Research, 18. 1970.
6. **Kaufmann, A.:** Az operációkutatás módszerei és modelljei. Műszaki Könyvkiadó, Budapest. 1968.
7. **Krekó B.:** Optimumszámítás. Közgazdasági és Jogi Könyvkiadó, Budapest, 1972.
8. **Little, J.-Murty, K.-Sweeney, D.-Karel, O.:** An Algorithm for the Travelling Salesman Problem. Operations Research, 11. 1963.

**Publikálva:**

Járművek, Mezőgazdasági Gépek, 37. évfolyam, 1990. 11. szám