Szabó Zoltán

CPM/TIME-algoritmusok korlátozott kapacitások esetén

Bevezetés

A hálótervezési módszerek közül legismertebb és legelterjedtebb a kritikus út módszere, a CPM/TIME. Tekintsük át röviden e módszert.

Az egy kezdő- és egy végponttal rendelkező, irányított, körútmentes, véges gráf (hálózat) minden ívéhez (tevékenységéhez) hozzárendelünk egy nem negatív számot, a tevékenység időtartamát. Ezek segítségével a csomópontokhoz (eseményekhez) hozzárendelhetjük a legkorábbi és a legkésőbbi bekövetkezési időket a következőképpen.

Az általánosság megszorítása nélkül feltehetjük, hogy a kezdő eseménynek a legkorábbi bekövetkezési ideje zérus. Tetszőleges i esemény legkorábbi bekövetkezési ideje a kezdő eseményből az i-be érkező utak hálótervezési értelemben vett (azaz a tevékenységek időtartamainak összegeként képzett) hosszainak maximuma. A hálózat befejező eseményének legkorábbi bekövetkezési idejét a hálózat teljes tervezési idejének nevezzük, s λ-val jelöljük. Tetszőleges j esemény legkésőbbi bekövetkezési ideje a λ-nak és a j-ből a befejező eseménybe érkező leghosszabb út hosszának a különbsége.

Minden tevékenységhez két esemény (a tevékenység kezdő és befejező eseménye), így négy időpont tartozik: a legkorábbi és legkésőbbi kezdés, valamint a legkorábbi és legkésőbbi befejezés időpontjai. A tevékenység időtartaléka a legkorábbi és a legkésőbbi kezdési időpontok különbsége. A kezdő eseményből a végeseménybe vezető utak leghosszabbikát (ha több van, úgy ezek mindegyikét) kritikus útnak nevezzük. A kritikus út hossza nyilván λ. Az is ismeretes, hogy a kritikus út tevékenységeinek és csak ezeknek az időtartalékuk zérus. A CPM/TIME algoritmusok kiszámítják a tevékenységekhez tartozó négy időpontot, és megjelölik a kritikus út tevékenységeit.

A túlterhelési probléma megfogalmazása

A termelésben gyakran előfordul, hogy több azonos típusú műveletet (tevékenységet) kell egyidőben elvégezni, mint amennyi a rendelkezésre álló termelési erőforrások (kapacitások) segítségével elvégezhető. Állapítsuk meg, mely műveleteket hagyjuk későbbre és mennyivel, hogy a végtermék előállítása minél kisebb mértékben késsen.

Operációkutatási terminológiában a következőképpen fogalmazzuk meg a problémát. Legyen adva egy fent definiált hálózat, tevékenységi időtartamokkal ellátva. A tevékenységek osztályát diszjunkt csoportokra, ún. azonos kapacitást terhelő (röviden azonos) tevékenység-csoportokra osztjuk. Minden csoporthoz tartozzon egy $c \geq 1$ természetes szám, a kapacitás mértéke, mely azt mutatja meg, hogy legfeljebb hány tevékenység végezhető egyidőben. Amennyiben van olyan tevékenység-csoport, melyből kiválasztható $c + p$ ($p > 0$) számú tevékenység úgy, hogy legkorábbi kezdési és legkésőbbi befejezési időpontjaik által alkotott időintervallumaik **I** metszetének mértéke pozitív, akkor túlterhelésről beszélünk. Nem okoz zavart, ha a tevékenység időintervallumát röviden tevékenységnek is nevezzük. Az **I** intervallumot túlterhelési helynek nevezzük. A dolgozatban olyan algoritmusokat adunk meg, melyek a hálózat túlterheléseit megszüntetik.

A későbbiekben a következő fogalmakra lesz szükségünk.

A tevékenység felezőpontja a legkorábbi kezdés és a legkésőbbi befejezés számtani közepe.

A tevékenység lényeges része az időintervallumnak és az időtartaléknak a különbsége.

Két tevékenység időtartalékaival metsz egymásba, ha a közös metszet nem nagyobb egyik tevékenység időtartalékánál sem.

Egyik tevékenység időtartalékával metsz a másik tevékenység lényeges részébe, ha a közös metszet kisebb az egyik tevékenység időtartalékánál, de nagyobb a másikénál.

Két tevékenység lényeges részeivel metsz egymásba, ha a közös metszet mindkét tevékenység időtartalékánál nagyobb.

Egyik tevékenység balról (jobbról) metsz a másik tevékenységbe, ha a két tevékenység egymásba metsz, és az előbbi tevékenység felezőpontja nem nagyobb (nem kisebb) az utóbbiénál.

Az **A** tevékenység megelőzi a **B** tevékenységet (**A** balra van **B**-től), ha **A** végpontjából a hálózat élei mentén az irányításnak megfelelően haladva, elérhetünk **B** kezdőpontjába. Ekkor **B** követi az **A**-t (**B** jobbra van **A**-tól).

Az **A** tevékenységet jobbra (balra) toljuk, ha legkorábbi vagy legkésőbbi időpontjait egyenlő mértékben növeljük (csökkentjük).

Az algoritmus

1. A hálózatra végezzük el a CPM/TIME algoritmust, azaz a tevékenységekhez tartozó négy időpontot számítsuk ki. Menjünk a 2. ponthoz.

2. Keressük meg a túlterhelési helyeket. Ha ilyen nincs, menjünk a 8. ponthoz, különben menjünk a 3. ponthoz.

3. A túlterhelési helyek baloldali végpontjaik közül a legkisebbikhez (ha több van, ezek bármelyikéhez) tartozó túlterhelési hellyel menjünk a 4. ponthoz.

4. A túlterhelésbe belejátszó tevékenységeket állítsuk legkorábbi kezdési időpontjaik szerint növekvő nagyság szerinti sorrendbe, és az első $c + 1$ számú tevékenységgel menjünk az 5. ponthoz.

5. Ha van két olyan tevékenység, melyek csak időtartalékaikkal metszenek egymásba, menjünk a 7. ponthoz.

Ha van két olyan tevékenység, melyek egyike időtartalékával metsz a másik tevékenység lényeges részébe, menjünk a 6. ponthoz.

Válasszuk ki a tevékenységek legkésőbbi kezdései közül a legnagyobbikat (t_1) és a második legnagyobbat (t_2), továbbá a legkorábbi befejezési időpontok közül a legkisebbet (T_1) és a második legkisebbet (T_2). Ha t_1 és T_1 ugyanazon tevékenységhez tartoznak, akkor vegyük

$$m = \min (T_1 - t_2, T_2 - t_1)\text{-hoz tartozó } T_i, \text{ ill. } t_i$$

időpontokat, s jelöljük T, ill. t-vel; különben $t_1 = t$, $T_1 = T$. A t-hez tartozó **A** tevékenység legkorábbi és legkésőbbi kezdési időpontját toljuk el T időpontba. Ezáltal az **A**-val közös úton, **A**-tól jobbra eső tevékenységek kezdési időpontjai rendre egy-egy közös pontba, jobbra eltolódnak, és keletkezik egy **A**-t közvetlenül megelőző ún. várakozási (vagy idő-) tevékenység (time-activity). A T-hez tartozó **B** tevékenység legkésőbbi időpontjait pedig toljuk el balra az időtartalék mértékével, miáltal a **B**-től balra levő, vele azonos úton fekvő tevékenységek legkésőbbi időpontjai is balra tolódnak. Keletkezik a **B** tevékenységet közvetlenül követő várakozási tevékenység. Menjünk a 2. ponthoz.

6. Ha van olyan **A** tevékenység, melynek időtartaléka balról metsz egy tevékenység lényeges részébe, akkor a lényeges részbe történő metszés mértékével balra toljuk az **A** tevékenység legkésőbbi időpontjait (ezáltal **A**-tól balra levő tevékenységeknél is ugyanez történik), és az **A**-tól jobbra most is fellép egy várakozási tevékenység. Menjünk a 7. ponthoz.

Kiválasztjuk azokat a tevékenységeket, melyeknek időtartalékai metszenek más tevékenységek lényeges részeibe. Ezek közül kiválasztjuk azt a tevékenységet, melynek időtartalékából a metszetet elvéve, a legnagyobb maradékidőtartalékhoz jutunk. Ezen tevékenység legkorábbi időpontjait jobbra toljuk a metszet mértékével (miáltal előtte egy várakozási tevékenység keletkezik), és a tőle jobbra álló tevékenységek is jobbra tolódnak. Menjünk a 7. ponthoz.

7. A balról metsző tevékenységek közül kiválasztjuk azt, amelyiknek időtartalékából kivonva a metszet-intervallumot, a legnagyobb időtartalék marad. Ezen tevékenységnek a legkésőbbi időpontjait a metszet mértékével balra toljuk (miáltal a tőle balra lévő tevékenységekkel is ugyanez történik). Itt is keletkezett egy várakozási tevékenység. Menjünk a 2. ponthoz.

8. Lépjünk ki az algoritmusból.

1. Megjegyzés

Nyilvánvaló, hogy a túlterhelések megszüntetésének egyetlen módja a tevékenységek időbeli eltolása. Szimmetriai okokból azonban világos, hogy két tevékenység vizsgálata esetén az egyik tevékenységet ugyanannyival kell előretolni, mint a másikat hátratolni, azaz az előre- és hátratolás a teljes tervezési idő szempontjából equivalens. Lényeges részek metszése esetén azonban csak a hátratolás alkalmazható, ugyanis előretolás alkalmával a túlterhelési helytől balra felléphetne újabb túlterhelés, ami azt eredményezheti, hogy az eljárás végtelen ciklusba ugrik. Időtartalékok csökkentése esetén természetesen mindkét eltolás alkalmazható, hiszen ekkor újabb túlterhelés nem léphet fel a tevékenységek időintervallumainak csökkenése miatt.

1. Tétel: A túlterhelési hely jobb oldali végpontjától balra az algoritmus folyamán nem lép fel újabb túlterhelés.

Bizonyítás:

A hálózattal, a tevékenységekkel az algoritmus folyamán a következők történhetnek:

a) A túlterhelésbe belejátszó egyik tevékenységnek és a tőle balra (következésképpen a túlterhelési helytől még inkább balra) levő néhány tevékenységnek a legkésőbbi időpontja balra tolódik (legfeljebb a megfelelő legkorábbi időpontig), miáltal az említett tevékenységek időintervallumainak jobboldali végpontjai nem növekednek. Tehát ekkor nem lép fel újabb túlterhelés.

b) A túlterhelésbe belejátszó egyik tevékenység legkorábbi időpontjai vagy legkésőbbi időpontjai (esetleg minden időpontja) jobbra tolódik, miáltal fellépő újabb túlterhelési hely baloldali végpontja nem lehet balra az említett tevékenység eredeti jobboldali végpontjától, annál inkább nem lehet balra az eredeti túlterhelési hely jobboldali végpontjától.

c) A túlterhelési helybe belejátszó egyik tevékenységtől jobbra (tehát a túlterhelési hely jobboldali **P** végpontjától még inkább jobbra) levő tevékenységek jobbra eltolódnak. Ezáltal a **P** ponttól balra természetesen nem lép fel túlterhelés.

d) Várakozási tevékenységek keletkeznek, melyek egyetlen azonos kapacitást terhelő tevékenység-csoportba sem tartoznak, következésképpen nem okoznak újabb túlterhelést.

A fenti meggondolásokból a tétel igazsága következik.

1. Következmény

Eljárásunk lépéseinek véges voltából, a hálózat, a túlterhelések végességéből az 1. tétel felhasználásával azt kapjuk, hogy az eljárás minden túlterhelést megszüntet (balról jobbra haladva, amit az eljárás 4. pontja biztosít). Következésképpen eljárásunk nem ugorhat végtelen ciklusba, tehát algoritmus.

2. Megjegyzés

Az algoritmus folyamán keletkező várakozási tevékenységeket nem adhatjuk hozzá sem az előtte, sem az utána álló tevékenységek időtartalékaihoz, ugyanis ellenkező esetben az eljárás végtelen ciklusba ugorhatna. Ugyanezen ok miatt nem végezhető el egyik tevékenységeltolás után sem a CPM/TIME algoritmus. Pontosabban mondva, ha egy tevékenységet jobbra (balra) eltolunk, úgy az eltolást csak az őt követő (megelőző) tevékenységekre kell elvégezni. A vég- (kezdő-) esemény megváltozott időpontjai szerint nem szabad a fennmaradó tevékenységek időpontjait is megváltoztatni, mert különben ugyanaz a túlterhelés újra felléphet, vagy más, újra fellépő túlterhelés miatt ugrik az eljárás végtelen ciklusba.

2. Tétel

Az algoritmus 5. pontjának egyszeri alkalmazása a teljes tervezési időt a lehető legkisebb mértékben növeli meg.

Bizonyítás

Világos, hogy az 5. pont alatti eljárás egyszeri elvégzésekor a teljes tervezési idő a $(T - t)$ időtartam mértékével növekszik meg. A $(T - t)$ konstrukciójából következik, hogy a $(c + 1)$ számú tevékenység közül bármelyik kettő lényeges részeinek metszete nem kisebb $(T - t)$-nél. Tehát a túlterhelés megszüntetésére irányuló, 5. pontban leírt eltolás minimális, $(T - t)$ mértékben növeli meg a teljes tervezési időt.

(Az 1. megjegyzés alapján a teljes tervezési idő növekedése a hálózat befejező eseménye bekövetkezési idejének $(T - t)$ mértékben történő késése útján valósul meg az algoritmus minden olyan szakaszában, amikor az 5. pont alatti eljárást alkalmazzuk.)

3. Megjegyzés

Hálózatunk általánosítása a CPM rendszerben tárgyalt hálózatoknak. Utóbbiak ugyanis az előbbieknek olyan speciális esetei, amikor nincs túlterhelés, vagy egész egyszerűen minden tevékenység-csoportnál c majorálja a tevékenységek számát.

4. Megjegyzés

Bár algoritmusunk a túlterheléseket oly módon szünteti meg, hogy az 5. pont egyszeri alkalmazása esetén λ minimális mértékben növekszik meg, mégsem tarthat igényt az „optimális" elnevezésre. Ennek oka az, hogy az egyik túlterhelés megszüntetése hatással lehet az illető túlterhelési helytől jobbra levő túlterhelésekre.

Ezt a hiányosságot igyekszik csökkenteni az algoritmus következő módosítása.

Mivel a teljes tervezési idő csak olyan túlterhelések megszüntetésekor növekszik, melyek a tevékenységek lényeges részeinek egymásba metszése útján állnak elő, az algoritmust csak ilyen esetben módosítjuk. E célból szükségünk lesz a csökkenési és a növekedési mérték fogalmára.

Ha egy **A** tevékenységet lényeges részével jobbra eltolunk, az **A**-tól jobbra lévő tevékenységek is eltolódnak. Ezek az eltolódások hatással lehetnek a túlterhelésekre: kevesebb vagy több azonos időben ütemezett tevékenység kerülhet a túlterhelésekbe. Túlterhelési csökkenési (ill. növekedési) mértéknek nevezzük valamely **A** tevékenységnek, és ezáltal az **A**-tól jobbra lévő tevékenységeknek lényeges résszel történő eltolásakor fellépő túlterhelés-csökkenések (ill. túlterhelés-növekedések) hossz-összegét az összes tevékenység-csoportokra vonatkozóan.

Az algoritmus módosítása

Az olyan túlterheléseket, melyeket időtartalékok metszeteként kaptunk, eredeti algoritmusunk szerint szüntetjük meg. A lényeges részek egymásba metszése által keletkezett túlterhelési helyeket baloldali végpontjaik növekvő nagysági sorrendje szerint rendezzük sorba. Az első túlterhelést (ha több ilyen van, úgy mindegyiket) vizsgálat alá vesszük. Megnézzük, hogy ezen túlterhelésbe (ill. ezen túlterhelésekbe) belejátszó tevékenységek olyan jobbra tolásai, melyek a túlterhelésből a tevékenységeket „kiszabadítják", milyen mértékű túlterhelési csökkenést és növekedést idéznek elő a tevékenység-csoportok összességében. Azt a tevékenységeltolást alkalmazzuk, melynél a

$$(1) \qquad \frac{\text{túlterhelési csökkenési mérték} - \text{túlterhelési növekedési mérték}}{\text{az eltolás mértéke}}$$

kifejezés maximális. (Ha több ilyen van, úgy ezek közül azt alkalmazzuk, melynél az eltolás mértéke a legkisebb.)

Ezt az eljárást addig folytatjuk, míg a túlterhelések meg nem szűnnek.

5. *Megjegyzés*

Könnyen belátható, hogy a módosított eljárás is algoritmus, hiszen a tevékenységek, így a túlterhelések száma is véges, továbbá az 1. tétel ebben az esetben is érvényes.

6. *Megjegyzés*

Módosított eljárásunknak az eredeti, általunk megadott algoritmus speciális esete abban az értelemben, hogy ott az (1) kifejezés maximalizálása csupán a pillanatnyilag tekintetbe vett túlterhelésbe belejátszó tevékenységekre vonatkozott.

7. *Megjegyzés*

A módosított modellben lehetőség van arra is, hogy a különböző tevékenység-csoportok esetén a túlterhelési növekedési (és csökkenési) mértékeket az (1) kifejezésben súlyozva vegyük figyelembe.

Erre szükség lehet, ha

a) a túlterhelések közül néhányat elhanyagolunk, másokat a leggyorsabban meg kell szüntetnünk (ugyanannak a hálózatnak más-más időszakban történő ütemezése esetén);

b) a túlterheléseket fontossági sorrendjük szerint vesszük figyelembe;

c) a kapacitások sztochasztikus jellegűek.

Ez nagymértékű általánosítása a módosított modellnek is.

Lehetőség van továbbá arra is, hogy a növekedési (csökkenési) mértékek az idő függvényében is súlyozva legyenek. Erre több, egymástól független hálózat esetén lehet szükség, ha néhány hálózat határidős termékelőállításra vonatkozik.

(Beérkezett: 1969. I. 7.)

IRODALOM

[1] ABRAMOV, SZ. A.—MARINCSEV, M. I.—POLJAKOV, P. D.: Hálódiagramos tervezési és irányítási módszerek. Budapest, 1966.
[2] BERGE, C.: The theory of graphs and its applications. London, 1962.
[3] KELLEY, J. E., Jr.: Critical-path planning and schedulding: mathematicalbasis. Operations Research, 1961, Vol. 9. pp 296–320
[4] MARTIN, J. J.: Distribution on the time through a directed, acyclic network. Operations Research, 1965. Vol. 13., No. 1.

THE CPM/TIME ALGORITHM IN THE CASE OF LIMITED SOURCES

The paper generalizes the CPM/TIME network planning (i.e. critical path) method in the following manner:

It is assumed that in the case of the various types of activity groups the sources and capacities are limited. The author gives an algorithm which carries out scheduling, with

due regard to the source constraints, in an optimal manner, in the sense that the final event occurs at the earliest point in time. Following the CPM/TIME scheduling carried out without taking into account the source constraints, the algorithm modifies the time points of the events in a way that the number of activities scheduled for the same time should not exceed the given constraint in the case of any activity group (relying on the same source). The author then proves the finite and optimal character of the algorithm

АЛГОРИТМ СЕТЕВОГО ПЛАНИРОВАНИЯ (CPM/TIME) В СЛУЧАЕ ОГРАНИЧЕННЫХ РЕСУРСОВ

В статье дается общее применение метода сетевого планирования в системе CPM/TIME (то есть метода критического пути). Автор предполагает при этом, что в отношении деятельностей различного типа имеются ограниченные ресурсы, мощности. Он представляет алгоритм, при помощи которого с учетом ресурсов как ограничений можно получить оптимальный сетевой график с наиболее близким сроком конечного события. Алгоритмом сроки событий сетевого графика, получаемого при помощи метода CPM/TIME без учета ресурсов как ограничений, корректируются таким образом, чтобы количество деятельностей, приходящихся на тот же отрезок времени, не превышало заданные ограничения ни по одной группе деятельностей (использующих тождественные ресурсы). После этого автор доказывает конечномерность и оптимальность алгоритма.