EGY IMPLICIT LESZÁMLÁLÁSON ALAPULÓ ÚJ ERŐFORRÁS KIEGYENLÍTŐ ELJÁRÁS[1]

CSÉBFALVI GYÖRGY – PETROS KONSTANTINIDIS
Janus Pannonius Tudományegyetem – Technological Educational Institution of Athens

Dolgozatunkban egy új implicit leszámláláson alapuló egzakt erőforrás kiegyenlítő eljárást ismertetünk. Az eljárás tevékenységek tartalékidőn belüli késleltetésével olyan ütemtervet keres, amelyben az erőforrás hisztogramok alakja megfelel a kívánatos konkáv alaknak. Az eljárás kiindulási adatait a kritikus út módszere szolgáltatja. Tapasztalataink az mutatják, hogy kis és közepes méretű problémák esetében az egzakt megoldások előállítása nem igényel komolyabb számítástechnikai hátteret.

1. Bevezetés

Projektek erőforrás felhasználásával kapcsolatos ütemezési problémák két alapvető csoportba oszthatók, ennek megfelelően erőforrás hozzárendelési, illetve erőforrás kiegyenlítési problémáról beszélhetünk. Mindkét probléma kiindulási alapját a kritikus út módszerével kapott legkorábbi tevékenység ütemezésnek megfelelő ütemterv alkotja.

Hozzárendelési problémáról van szó, ha a rendelkezésre álló erőforrások korlátozottsága miatt a kritikus út módszerével kapott legkorábbi ütemterv nem valósítható meg. Ekkor az erőforrás felhasználási konfliktusok feloldásával, vagyis a kritikus és a nem kritikus tevékenységek ütemezésének késleltetésével olyan ütemtervet kell keresünk, amely kielégíti az erőforrás korlátokat és amely a kritikus út módszerével kapott befejezési időpontot minimális mértékben növeli. A jól ismert nehéz probléma szakirodalma igen széles körű, számtalan heurisztikus és jó néhány egzakt módszert dolgoztak ki a probléma megoldására.

Az erőforrás kiegyenlítési probléma esetében a kritikus út módszerével kapott ütemtervek megvalósítását erőforrás korlátok nem akadályozzák, de megoldandó problémát jelentenek az erőforrás felhasználásban mutatkozó ingadozások. Ekkor kiindulva a legkorábbi ütemtervből, a nem kritikus tevékenységek ütemezésének késleltetésével egy olyan ütemtervet keresünk,

[1] Beérkezett: 1998. február 4.

amelyben az erőforrás felhasználási hisztogramok alakja a lehető legjobban megközelíti a kívánatos alakot. Egy erőforrás esetében a feladat célfüggvényét hagyományosan az erőforrás felhasználás szórásnégyzeteként (az átlag felhasználástól való négyzetes eltérésként) definiáljuk.

A probléma megoldására számos heurisztikus eljárást dolgoztak ki. Az eljárások közös vonása, hogy kiindulva a legkorábbi ütemtervből, valamilyen ökölszabály alkalmazásával bizonyos tevékenységek ütemezését késleltetik. Két jellegzetes példaként Harris (1990) és Konstantinidis (1998) módszerét említhetnénk.

A probléma egzakt megoldására szolgáló módszer szinte alig van. Jellegzetes irányzatokat képvisel Ahuja (1976) explicit leszámlálást alkalmazó módszere, Valadares Taveres (1987) eljárása amely a kívánatos hisztogram alakot folytonos időfüggvényként kezeli, Easa (1989) egész értékű optimalizáláson alapuló modellje, illetve a Bandeloni és társai (1994) által kifejlesztett egy erőforrást kezelő, dinamikus programozáson alapuló módszer.

A cikk egy új, az implicit leszámlálás módszerén alapuló egzakt algoritmust ismertet az erőforrás kiegyenlítési probléma megoldására. Az eljárás az előzőekben említett eljárásokhoz képest továbblépést jelent azáltal, hogy az explicit leszámlálásnál hatékonyabb, nem követeli meg a kívánatos alak folytonos időfüggvényként történő megadását, nem igényli számításigényes egész értékű programozási feladat megoldását és semmilyen formában nem korlátozza az ütemtervben szereplő erőforrások számát.

Az algoritmus új eleme, hogy a kívánatos hisztogram alaknak a lapos konkáv alakot tekinti, tehát szakít a négyzetes eltérés típusú célfüggvények hagyományával. A váltás mögött az a felismerés áll, hogy egyrészt

- a gazdaságilag nehezen megfogható ingadozás minimalizálás mögött lényegében mindig a jól számszerűsíthető állásidő minimalizálási törekvés húzódik meg, annak ellenére, hogy a minimális ingadozású és a minimális állásidejű ütemezés nem azonos, másrészt
- gazdasági szempontból a mesterkélt átlagos erőforrás felhasználásnál a maximális felhasználás lényegesen többet mond, a menedzsment a maximális felhasználás minimalizálásában, vagyis az erőforrás hisztogramok összelapításában érdekelt.

A módszertani felvetés lényegében a hozzárendelési probléma és kiegyenlítési probléma összekapcsolását jelenti. Rögzítve a projekt befejezési időpontját, erre elméletileg három lehetőségünk van:

1. Megoldjuk a kiegyenlítési problémát, majd a megoldások halmazából kiválasztjuk azokat, amelyek kielégítik az erőforrás korlátokat.

2. Megoldjuk az erőforrás hozzárendelési problémát, majd a megoldások halmazán megkeresünk konkáv hisztogramokkal jellemezhető megoldásokat.

3. A két problémát egyesítjük, vagyis olyan ütemtervet keresünk, amelyben minden erőforrás hisztogram konkáv és a megadott korláton belül marad.

Azt, hogy a három megközelítés közül melyik a legjobb, szinte lehetetlen megválaszolni, mivel konkrét feladatok esetében, a megoldáshalmazok függvényében, igen eltérő feleleteket kaphatunk (Csébfalvi és Konstantinidis (1998)). Az értékelést tovább nehezíti, ha a kiegyenlítési problémát *liberalizáljuk*, vagyis közelítőleg konkáv alakokat is megengedünk.

Jelen keretek között csak az első esettel, vagyis a kiegyenlítési probléma megoldásával foglalkozunk. A második eset az erőforrás hozzárendelési probléma összes lehetséges megoldásának ismeretét igényli, így át kell fogalmaznunk az erőforrás hozzárendelési problémát, hiszen ezen hagyományosan csak az első megoldás megkeresését értjük (Csébfalvi (1998)). A harmadik eset, azaz a két szempont szerinti egyidejű keresés, módszertanilag teljes egészében új kutatási területet jelent, azzal a kecsegtető lehetőséggel, hogy a keresési szempontok összekapcsolása a keresési fa méreteit csökkenti.

2. Az erőforrás kiegyenlítés módszere

Álljon a projekt $N + 2$ tevékenységből, és sorszámozzuk a tevékenységeket 0-tól $N + 1$-ig. A szokásos módon jelölje a 0-adik és az $N + 1$-edik ún. mesterséges tevékenység a projekt egyedi kezdetét és végét. Jelölje D_i az i-edik ($i = 1, 2, \ldots, N$) tevékenység időtartamát. A tevékenységek közötti függőségi viszonyokat vég-kezdet típusú logikai kapcsolatokkal jellemezzük. A projekt ábrázolásakor a *"tevékenységek csúcspontok, a kapcsolatokat élek"* konvenciót alkalmazzuk. Feltételezzük, hogy az ütemezési probléma minden adata egész értékű.

Jelölje E_i, illetve L_i az i-edik ($i = 1, \ldots, N$) tevékenység legkorábbi, illetve legkésőbbi kezdési időpontját. Legyen $F_i = L_i - E_i$ az i-edik ($i = 1, \ldots, N$) tevékenység tartalékideje. Jelölje R az erőforrások számát, Q_r az r-edik ($r = 1, 2, \ldots, R$) erőforrásból időegységenként rendelkezésre álló mennyiséget, illetve U_{ir} ($i = 1, 2, \ldots, N$; $r = 1, 2, \ldots, R$) az i-edik tevékenység időegységre eső erőforrás igényét az r-edik erőforrásból. Legyen T a projekt rögzített befejezési időpontja. Jelölje

$$S = \{\{t_1, s_1\}, \{t_2, s_2\}, \ldots, \{t_n, s_n\}\},\ t_i \in \{1, 2, \ldots, N\}, \quad n \leq N \tag{1}$$

egy adott ütemtervhez tartozó késleltetések halmazát, ahol t_i a tevékenység sorszáma, s_i a tevékenység késleltetése időegységekben mérve, amelyre igaz, hogy

$$s_i > 0,\ s_i \leq F_{t_i}\ . \tag{2}$$

A késleltetés a tevékenység *legkorábbi kezdési időpontjához képest* jelent késleltetést, így a legkorábbi ütemtervre $S = \{\}$.

Nevezzük az S halmazban szereplő tevékenységeket *kötött* tevékenységeknek. Egy adott ütemtervből csak a *szabad* (nem kötött) tevékenységek késleltetésével konstruálhatunk újabb ütemterveket.

Jelölje $C_{tr} = C_{tr}(S)$ egy adott S késleltetés halmazzal jellemezhető ütemterv esetében a t-edik ($t = 1, 2, \ldots, T$) időpontban a tevékenységek együttes erőforrás felhasználását az r-edik ($r = 1, 2, \ldots, R$) erőforrásból.

1. Definició. *Nevezzük az r-edik erőforrás hisztogram* $[a, b]$ *időintervallumba eső részhalmazát* völgynek, *ha* $b - a > 1$,

$$\mathrm{Min}(C_{ar}, C_{br}) > C_{kr} \quad \text{minden} \quad k \in [a+1, b-1] \tag{3}$$

értékre és ahol az $[a, b]$ *időintervallum olyan nagy, amilyen csak lehetséges.*

Jellemezzük a völgyeket az$\{a, b, r, p\}$ adatokkal, ahol

$$p = \sum_{k=a+1}^{b-1} \mathrm{Max}(\mathrm{Min}(C_{ar}, C_{br}) - C_{kr}, 0) \tag{4}$$

a völgy nagysága, vagyis az állásidők összege.

Jellemezzük magát az ütemtervet az időtengelyen balról jobbra, illetve azon belül az erőforrásokon növekvő sorrendben haladva a völgyek

$$V = \{\{a_1, b_1, r_1, p_1\}, \{a_2, b_2, r_2, p_2\}, \ldots, \{a_v, b_v, r_v, p_v\}\},\ v = |V| \tag{5}$$

halmazával, illetve a

$$P = \begin{cases} \sum_{i=1}^{v} p_i, & \text{ha } v > 0; \\ 0, & \text{ha } v = 0 \end{cases} \tag{6}$$

mennyiséggel, mint *értékelési kritériummal.*

Ha egy erőforrás hisztogram nem tartalmaz völgyeket, vagyis alakja konkáv, akkor az adott erőforrás igénybevételét leállások nem szakítják meg. Ha a kivánatos alak a tevékenységek késleltetésével elérhető, akkor a projekt által nem hasznosított erőforrás kapacitás a projekt *elején és végén* jelentkezik, megteremtve a másirányú hasznosítás lehetőségét. *Több projektes* alkalmazási környezetben például, az így adódó kapacitást más projektekhez rendelhetjük.

Hangsúlyoznunk kell, hogy az $\{r_i, p_i\}$ $, i = 1, 2, \ldots, v$ értékekből kiindulva sokféle értékelési kritérium konstruálható, azonban ezek taglalásától a jelen keretek között eltekintünk. A $P = P(S)$ választás, mint az állásidők összege, a legegyszerűbb esetet jelenti, mivel a különböző erőforrásokat összemossa. Az erőforrások költségfüggvényének bevezetésével a kívánatos cél sokkal *árnyaltabban* is megfogalmazható.

Ezen túlmenően meg kell jegyeznünk, hogy különösen több erőforrásos projektek esetén a *minden erőforrás dimenzióban kiegyenlített* ütemezés — a technológiai kötöttségek miatt — nem mindig érhető el. A közölt algoritmus, az érdemi elemek megtartásával, illetve egy megengedett $P_{tűrés}$ bevezetésével továbbfejleszthető. Az általánosított algoritmus kidolgozása folyamatban van és a későbbiekben közöljük.

Az ismertetendő keresési eljárásban a *legelső völgy* kiemelt szerepet játszik, mivel az algoritmus minden lépésben a kiválasztott *legígéretesebb* ütemterv *legelső* völgyének feltöltésére törekszik. A legígéretesebb ütemterv alatt azt az ütemtervet értjük, amelynek az *állásideje a legkisebb* és ezen belül azt, amelyben *legelső völgy a lehető legkésőbb* található. A másodlagos értékelési kritérium mögött az a feltételezés áll, hogy a minimális állásidővel rendelkező ütemtervek közül valószinüleg az áll *"legközelebb"* a probléma megoldásához, amelyre vonatkozóan a másodlagos értékelési kritérium fennáll.

2. Definició. *Akkor mondjuk, hogy egy $S_{régi}$ késleltetés halmazzal jellemzett ütemterv* legelső, *az r-edik hisztogram $[a, b]$ intervallumába eső völgye* feltölthető, *ha van olyan*

$$S_{új} = S_{régi} \cup S_{feltöltő} \tag{7}$$

késleltetés halmaz, hogy az ehhez tartozó ütemterv minden *erőforrás hisztogramja* konkáv *az $[1, b]$ intervallumban.*

Az *implicit leszámlálás* logikájának megfelelően az eljárás egy keresési fát épít fel, amelynek gyökerét a kritikus út módszerével kapott legkorábbi ütemterv alkotja, amelyről feltesszük, hogy nem konkáv az $[1, T]$ időintervallumban. Az algoritmust az *összes* kiegyenlített megoldás keresésére alkalmas formában fogalmazzuk meg.

Köztudott, hogy az implicit leszámlálás módszerének számítástechnikai hatékonyságát (a keresési fa méreteit) alapvetően az befolyásolja, hogy viszonylag egyszerű és gyors próbák segítségével milyen mértékben sikerül a fa bizonyos részeit a további vizsgálatokból kizárni. Hatékony *metszési szabályok* nélkül az implicit leszámlálás hatékonysága az explicit leszámlálás szintjén mozog.

A javasolt eljárás a *kiegyenlítési problémát feltöltési problémák sorozatára* vezeti vissza. A lényeget tekintve, a feltöltési probléma és az eredeti probléma azonos. A feltöltési probléma azonban, extrém esetektől eltekintve, jóval *kisebb méretű*, megoldása gyors, ezért alkalmas arra, hogy segítségével a keresési fa felesleges ágait levágjuk.

Ha egy ütemtervben egy völgyről bizonyítható, hogy nem tölthető fel, akkor az ütemtervet a további vizsgálatokból kizárhatjuk. Meg kell jegyeznünk, hogy a bizonyítás nem feltétlenül jelenti az összes feltöltési lehetőség explicit leszámlálását, mivel az esetek egy részében az explicit leszámlálás egy egyszerű és gyors próbával helyettesíthető.

Jelölje egy adott S késletetés halmaz esetében K_{tr} illetve U_{tr}, ahol $t = 1, 2, \ldots, T$ és $r = 1, 2, \ldots, R$ az adott t időpontban az r-edik erőforrásra vonatkozóan a *minimális* illetve a *maximális* igényt, ahol a minimális (maximális) igényt az adott időpontban lehetséges tevékenység kombinációk összesített igényének *minimuma (maximuma)* adja.

Próba. *Ha egy* $\{a, b, r\}$ *adatokkal jellemzett völgy esetében van olyan*

$$h \in \{a+1, a+2, \ldots, b-1\} \tag{8}$$

időpont, hogy erre az időpontra vonatkozóan

$$U_{hr} < \mathrm{Min}(K_{ar}, K_{br}), \tag{9}$$

akkor a völgy nem tölthető fel.

A keresési algoritmus lényegét az alábbiakban foglaljuk össze. A keresési fa csúcspontjait a $k = 0$ értékből kiindulva sorszámozzuk és az

$$\{S^{(k)}, P^{(k)}, V^{(k)}, I^{(k)}\} \tag{10}$$

értékekkel jellemezzük, ahol $S^{(k)}$ a k-adik csúcsponthoz tartozó késleltetés halmaz, $P^{(k)}$ az állásidő nagysága, $V^{(k)}$ a völgyek halmaza, $I^{(k)}$ pedig a csúcspont indikátor változója,

$$I^{(k)} = \begin{cases} 0, & \text{ha a } k\text{-adik csúcspont még nincs kifejtve;} \\ 1, & \text{ha a } k\text{-adik csúcspont kifejtése már megtörtént;} \\ 2, & \text{ha a csúcspont optimális.} \end{cases}$$

Jelölje $k_{\max}$ a legnagyobb csúcspont sorszámot (induláskor $k_{\max} = 0$). A keresési algoritmus lépései:

1. Minden egyes lépés a legígéretesebb, $k_{\min}$ sorszámú csúcspont kiválasztásával kezdődik, amelynek keresési feltétele a következő:

$$P^{(k_{\min})} \to \mathrm{Min}, \quad I^{(k_{\min})} = 0, \quad k_{\min} \in \{0, 1, \ldots, k_{\max}\}. \tag{11}$$

Ha a keresés sikertelen, vagyis nincs olyan csúcspont, amelyet még nem vizsgáltunk meg, az eljárás befejeződik és az

$$O = \left\{\{S^{(k)}, P^{(k)}, V^{(k)}, I^{(k)}\} \mid I^{(k)} = 2, \quad k \in \{0, 1, \ldots, k_{\max}\}\right\} \tag{12}$$

halmaz adja a feladat optimális megoldásait. Természetesen az $O = \{\}$ előállhat.

Ha a keresés eredményeképpen több csúcspontot kapunk, akkor ezek közül vesszük azt, amelyben legelső völgy a lehető legkésőbb található, mivel feltételezzük, hogy ez a csúcspont van a legközelebb a kiegyenlítési probléma megoldáshoz.

2. A kiválasztott csúcspont indikátor változójának értékét egyre változtatjuk, vagyis $I^{(k_{\min})} = 1$ és a csúcspontot kifejtjük.

   A kifejtés során a kiválasztott csúcspont *legelső völgyének* feltöltési lehetőségei adják az

$$\{S^{(k)}, P^{(k)}, V^{(k)}, I^{(k)}\}, \quad k = k_{\max} + 1, k_{\max} + 2, \ldots, k_{\max} + k_{uj} \tag{13}$$

   származtatott csúcspontokat, ahol

$$I^{(k)} = \begin{cases} 0, & \text{ha } P^{(k)} > 0; \\ 2, & \text{ha } P^{(k)} = 0. \end{cases} \tag{14}$$

   A feltöltési lehetőségek közül *azonnal töröljük* azokat, amelyek esetében bizonyítható (lásd *Próba*), hogy valamelyik erőforrás dimenzióban olyan hisztogramot eredményeznének, amelynek van olyan völgye, amely nem tölthető fel.

   Az új csúcspontok számának megfelelően módosítjuk $k_{\max}$ értékét, vagyis $k_{\max} = k_{\max} + k_{uj}$ és visszatérünk az 1. lépésre.

3. Alkalmazási példa

A javasolt új erőforrás kiegyenlítő eljárás működését egy egyszerű alkalmazási példa bemutatásával szemléltetjük. A példa kiindulási adatait, illetve a kritikus út módszerének eredményeit az *1. táblázat* tartalmazza. A feladat kapcsolatokkal kiegészített Gantt diagramját, illetve erőforrás hisztogramját a legkorábbi ütemezésnek megfelelően az *1. ábra* szemlélteti. A feladatban csak egy erőforrás szerepel, amellyel kapcsolatban feltesszük, hogy igénybevétele nincs korlátozva. A projekt rögzített befejezési időpontja $T = 17$ időegység.

A *nem-kritikus*, vagyis késleltethető tevékenységek száma 5. A legkorábbi ütemezésnek megfelelő erőforrás hisztogram három völgyet tartalmaz. A kiindulási ütemterv esetében az értékelési kritérium értéke $P = 3 + 2 + 2 = 7$.

A lehetséges ütemezések száma 1860. *Explicit leszámlálás* esetén ennyi esetet kellene megvizsgálnunk, hogy a konkáv hisztogrammal rendelkező eseteket kiválaszthassuk.

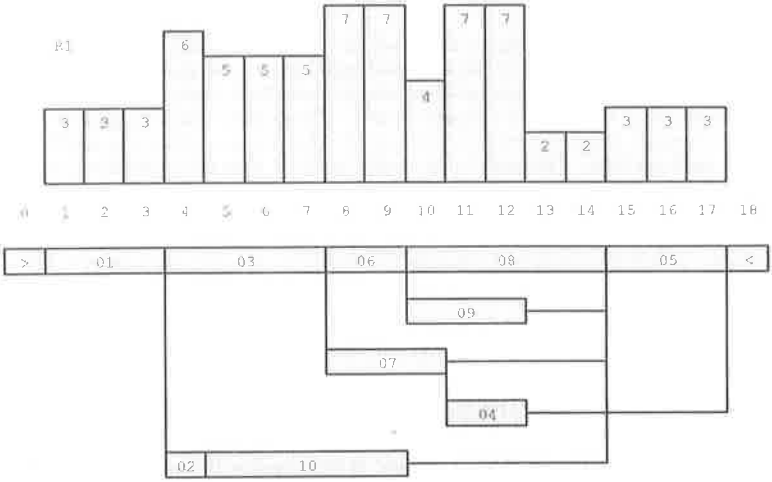
Az első völgy *"feltöltésében"* három változó játszik szerepet: $\{A02,\ A07,\ A10\}$. Az előzőekben ismertetett *vágási szabály alkalmazása nélkül* az első völgy feltöltéséhez 105 esetet kell megvizsgálnunk, amelyek közül 59 esetben kapunk a feltöltésre alkamas késleltetés kombinációt. A *vágási szabály alkalmazásakor* a lehetséges késleltetési kombinációk száma 17, ezek közül 10 alkalmas feltöltésre. Mindkét esetben a megoldások között pontosan egy olyan késleltetés kombináció található, amely optimális megoldása a kiindulási problémának (*2. ábra*).

Az alkalmazott keresési startégiának — *depth-first solution strategy* — megfelelően viszonylag kevés esetet kell megvizsgálnunk ahhoz, hogy az optimális megoldást megkapjuk (31 illetve 10). A további lépések (170 illetve 151) ahhoz szükségesek, hogy bizonyítsák, hogy az adott feladatnak nincsen más optimális megoldása. Az *összes* kiértékelt eset száma $31 + 170 = 201$ vágás nélkül, illetve $10 + 151 = 161$ vágással, ami jól szemlélteti az implicit leszámlás, illetve azon belül a javasolt vágási szabály előnyeit az explicit leszámláláshoz képest.

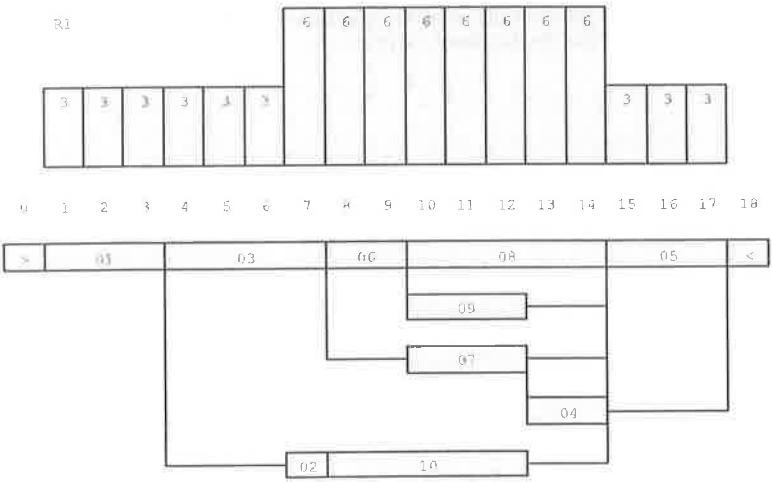
Megjegyezzük, hogy erőforrás korlátok beépítésével a kiértékelésre kerülő esetek száma tovább csökkenthető.

tev.	megelőző tevékenység	időtartam	erőforrás igény	legkorábbi ütemezés	legkésőbbi ütemezés
D00		1		0	0
A01	D00	3	3	1	1
A02	A01	1	3	4	9
A03	A01	4	3	4	4
A04	A07	2	4	11	16
A05	A07,A08,A09,A10	3	3	15	15
A06	A03	2	4	8	8
A07	A03	3	1	8	12
A08	A06	5	2	10	10
A09	A06	3	1	10	12
A10	A02	5	2	5	10
D11	A04,A05	1		18	18

1. táblázat. Kiindulási adatok és a kirikus út módszerének eredményei



1. ábra: A kritikus út módszerével adódó legkorábbi ütemezés



2. ábra: A kiegyenlítés eredményeképpen adódó ütemezés

OPTIMAL RESOURCE LEVELING USING IMPLICIT ENUMERATION

This paper presents a new exact approach for resource leveling based on implicit enumeration. The objective of the proposed approach is to shedule all the activities in the project to specific time periods within their float, such that the resource histograms meet the desired concave shape. The model, which is applied to an activity-on-node network, requires as input the sheduling results computed by the critical path method. Computational results indicate that the procedure provides exact solutions for small to medium size problems, requiring only modest computing facilities.

Irodalom

1. Valadares Tavares, L. (1987), Optimal Resource profiles for program scheduling, EJOR 29, 83–90
2. Bandelloni, M. et. al. (1994), Optimal resource leveling using non-serial dynamic programming, EJOR 78, 162–177
3. Easa, S. M. (1989), Resource leveling in construction by optimization, JCEM 115 2, 302–316
4. Konstantinidis, P. (1998), A model to optimize project resource allocation by construction of a balanced histogram, EJOR 104, 559–571
5. Csébfalvi, G. (1998), A fast exact solution procedure for multiple resource constrained project scheduling, PROC. APMOD '98, Limassol, Cyprus
6. Csébfalvi, G. and Konstantinidis, P. (1998), A new exact resource balancing procedure for multiple resource constrained project scheduling problem, PROC. APMOD '98, Limassol, Cyprus.