

Kosztján Zsolt Tibor

Optimális erőforrás-tervezés

DOKTORI (PhD) ÉRTEKEZÉS

Témavezető: Dr. Bencsik Andrea



**Gazdálkodás- és Szervezéstudományok Doktori
Iskola**

Veszprém

2005

OPTIMÁLIS ERŐFORRÁS-TERVEZÉS

Értekezés doktori (PhD) fokozat elnyerése érdekében

*a Veszprémi Egyetem.....

Doktori Iskolájához tartozóan*.

Írta:

Kosztván Zsolt Tibor

****Készült a Veszprémi Egyetem Gazdálkodás és Szervezéstudományok Doktori iskolája
keretében**

Témavezető: Dr.Bencsik Andrea

Elfogadásra javaslom (igen / nem)

(aláírás)**

A jelölt a doktori szigorlaton % -ot ért el,

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: igen /nem

.....
(aláírás)

Bíráló neve: igen /nem

.....
(aláírás)

***Bíráló neve: igen /nem

.....
(aláírás)

A jelölt az értekezés nyilvános vitáján% - ot ért el

Veszprém/Keszthely,

.....
a Bíráló Bizottság elnöke

A doktori (PhD) oklevél minősítése.....

.....
Az EDT elnöke

Megjegyzés: a * közötti részt az egyéni felkészülők, a ** közötti részt a szervezett
képzésben résztvevők használják, *** esetleges

Tartalmi kivonat

Optimális erőforrás-tervezés

A hálótervezési és erőforrástervezési technikák különböző területeken és sokféle formában jelenhetnek meg. Széles körben alkalmazzák a termelés, tervezés, elosztás, telepítés, erőforrás-gazdálkodás, pénzügyi tervezés, projekt-menedzsment stb. területén különböző problémák megoldására.

A szerző a doktori értekezésében olyan algoritmusokat mutat be, melyeket széles körben lehet majd alkalmazni a projektmenedzsmentben, erőforrás-tervezésben, logisztikában egyedi és kissorozatgyártás termelésirányítására.

Olyan módszereket fejlesztett ki, amely:

- garantáltan véges lépésben megadja az adott célfüggvényre optimális megoldást;
- használható olyan esetekben, amikor az erőforráskorlát időben nem állandó;
- alkalmazható a már elkezdett projektekben lévő megváltozott lefutási idejű, erőforrás-szükségletű tevékenységek újraütemezésére;
- használható ott is, ahol az idő-, költség- és erőforrásigény együttes optimalása a cél;
- képes többfajta erőforrás egyidejű, illetve párhuzamosan működő projektek közötti erőforrás-megosztás kezelésére;
- a fel nem használt erőforrások kezelését is biztosítja;
- használhatók bizonytalan átfutási idejű projektek illetve termelési programok tervezésére is.

Summary of contents

on Optimal Resource Allocation

Network planning and resource allocating techniques can be used in a variety of different fields of logistics and project management. These techniques are widely used in production, planning, distribution, installation, resource management, financial planning and project management to solve different kinds of problems.

In this dissertation some new algorithms are introduced which could hopefully be widely used in project management, resource planning and in methodology of small-scale series production management.

Some new algorithms have been developed which

- result the optimal solution in finite steps, where target function is given;
- can be used with when the availability(s) of resource(s) is/are constant by sections;
- can be used when the project in progress and the duration times and/or resources of activities are changing;
- can be used to determine a resource allocation with minimal total project time (TPT) and minimal total project cost;
- are capable of using different resources and can be applied in parallel projects;
- ensure the management of the resources not in use;
- can also be used in stochastic resource allocation.

Zusammenfassung

über Optimal Ressourcen-Verteilung

Die Netz-Plantechiken und die Techniken der Ressourcenplanung erscheinen auf verschiedenen Gebieten und in verschiedenen Formen. Diese Techniken werden weitverbreitet angewendet, Probleme bei der Produktion, Planung, Zuteilung, Installierung, Ressourcenwirtschaft, finanziellen Planung, bei dem Projekt-Management usw. zu lösen.

Im Artikel werden solche Algorithmen vorgestellt, die bei dem Projekt-Management, der Ressourcenplanung und Steuerung von Einzelherstellung oder von kleinen Serien in der Logistik hoffentlich weitverbreitet angewendet werden könnten.

Neue Algorithmen wurden entwickelt, die

- die optimale Lösung von einer angegebenen Zielfunktion nach endlicher Schrittzahl ergeben;
- auch in Fällen, wobei die Grenzen der Ressourcen keine Festwerte sind, angewendet werden können;
- auch für die Neutaktierung von Tätigkeiten, bei denen die Ressourcenanforderungen oder die Ablaufszeit während des Prozesses geändert werden, geeignet sind;
- für solche Probleme angewendet werden können, wo das Ziel die Gesamtoptimierung von Zeit-, Kosten- und Mittelanforderungen ist;
- gleichzeitige Behandlung von verschiedenen Ressourcen bzw. Ressourcenverteilung zwischen parallelen Prozessen ermöglichen;
- die Behandlung von den nicht verwendeten Ressourcen versichern;
- für Projekte mit unsicher durchlaufene Zeit und für Konstruktion der Produktionsprogramme benutzbar sind.

A témaválasztás indoklása

Az erőforrás-elosztás problémájával először egyetemistaként Szervezés technika című órán találkoztam. Az itt tanult módszerrel egy projekt tevékenységeit úgy kellett beütemezni, hogy a tevékenységek erőforrásigényei ne lépjék túl a rendelkezésre álló erőforráskorlátot. Már az órán felfigyeltem, hogy ez a módszer csak egy megengedett megoldást szolgáltat.

Ennek a módszernek a kiegészítését, mellyel a tevékenységek lehető legkorábbi kezdését lehetett meghatározni (a rendelkezésre álló tevékenységek erőforráskorlátainak figyelembevételével) egyetemistaként készítettem el egy hétoldalas beadandó feladat formájában.

Az eljárást továbbfejlesztettem; több hazai és nemzetközi konferencián is bemutattam, és még 2001-ben szabadalmaztattam. Többször indultam intézményi és országos Tudományos Diákköri versenyen, melyeken valamennyiszer díjazásban részesültem. 2001-ben pedig dolgozatommal megnyertem a Magyar Logisztikai Beszerzési és Készletezési Társaság éves diákpályázatát.

Ekkor döntöttem el, hogy ezzel a témával szeretnék tovább foglalkozni doktoranduszként is. Témavezetőm egykori szervezés technika-tanárom lett, akinek ezúton is köszönetet szeretnék mondani azért a sok segítségért, amivel egyetemi, doktoranduszi munkámat segítette.

Köszönetnyilvánítás

Ezúton is szeretnék köszönetet mondani *szüleimnek*, hogy segítettek az egyetemi éveim alatt és ösztöndíjas PhD hallgatóként abban, hogy munkámat a lehető legjobb tudásom szerint végezhessem el.

Köszönetet szeretnék mondani témavezetőmnek, *Dr. Bencsik Andreának*, aki mindenfajta (tárgyi és szellemi) segítséget biztosított munkám elvégzéséhez.

Külön szeretnék köszönetet mondani feleségemnek, *Mátrai Ritának*, aki munkámban mindvégig segítségemre volt, valamint *Póta Szabolcsnak* és *Hogyor Andrásnak*, akik az elmélet informatikai implementálásában segédkeztek.

A módszer gyakorlati vizsgálatát valós projekteken tesztelhettem, mely lehetőségért köszönettel tartozom *Perjés Zoltánnak*, az INTEGRÁL-HEXA Rt ügyvezető igazgatójának.

Tartalomjegyzék

1.	Irodalmi áttekintés, alapfogalmak.....	1
1.1	Bevezetés	1
1.2	Projekt, projektmenedzsment	4
1.2.1	A projektek közös elemei	6
1.2.1	A projekttervezés elemei	7
1.2.2	A munkalebontási szerkezet	8
1.3	Hálótervezés.....	10
1.3.1	A hálótervezési módszerek csoportosítása	13
1.3.2	Alapfogalmak	16
1.3.2.1	Gráfelméleti alapfogalmak	16
1.3.3	Tevékenységidő- és eseményidő-adatok	17
1.3.3.1	Tartalékidők	18
1.3.4	Az MPM-háló.....	19
1.3.5	Véletlen időtartamú tevékenységek.....	21
1.3.6	Költségoptimalizálás	25
1.3.6.1	Determinisztikus költségoptimalás tevékenység-nyíl típusú hálók esetén	27
1.3.6.2	Determinisztikus költségoptimalás tevékenység-csomópontú hálók esetén.....	27
1.3.6.3	Determinisztikus költségoptimalás – további módszerek.....	27
1.3.7	Bizonytalanság kezelése hálótervezési technikák segítségével különböző logisztikai feladatok esetén	28
1.3.7.1	Bizonytalanság	30
1.3.7.2	Standard bizonytalanság	30
1.3.7.3	Összetett bizonytalanság	31
1.3.7.4	Kiterjesztett bizonytalanság	31
1.3.8	Erőforrás-tervezés	32
1.4	Algoritmusok csoportosítása	33
1.5	Ütemezés, erőforrás-allokáció	35
1.5.1	Allokálás	39
1.5.1.1	A soros allokálási eljárás.....	39
1.5.1.2	A párhuzamos allokálási eljárások	40
1.5.2	Erőforrás-allokáció (ERALL-módszer).....	40
1.6	Optimumkeresési eljárások	41
1.6.1	Kiegyenlítés.....	41
1.6.2	Heurisztikus és algoritmikus eljárások összehasonlítása.....	42
1.6.2.1	A Branch and Bound algoritmus	43
1.6.2.2	Dinamikus programozás.....	46
1.7	Az ütemezés, erőforrásallokáció alkalmazási területei	48
1.7.1	A módszerek alkalmazási feltételei, lehetőségei	49
1.8	Számítógépes projekttervező szoftverek	50
1.9	Módszerek, szoftveres alkalmazások értékelése	57
1.10	Hipotézisek.....	59
2.	AZ OPTIMÁLIS ERŐFORRÁS-ALLOKÁCIÓ KERESÉSÉNEK MÓDSZERTANI BEMUTATÁSA	60
2.1	Erőforrás-allokáció, megengedett megoldásból optimális megoldás keresése (ERALL-OPT, OPT-RALL)	60
2.1.1	A feladat megfogalmazása	60
2.1.2	Az algoritmus leírása.....	63
2.1.3	Példák	71
2.2	Erőforrás-allokáció szakaszonként konstans erőforráskorlátozás esetén (ERALL-OPT/SZK, OPT-RALL/VRA).....	75
2.2.1	A feladat és a megoldás leírása	75
2.2.2	Példa.....	77
2.3	További alkalmazások (tevékenységek, erőforrásigények követése, megszakítható tevékenységek kezelése)	79

2.3.1	Erőforráskorlát változása (ERALL-OPT/ON-LINE/VK, OPT-RALL/ON-LINE/VRA).....	79
2.3.1.1	Példa.....	81
2.3.2	Tevékenységek erőforrás-szükségletének változása (ERALL-OPT/ON-LINE/EV, OPT-RALL/ON-LINE/VR).....	84
2.3.2.1	Példa.....	85
2.3.3	Tevékenységekhez szükséges idő változása (on-line módon) (ERALL-OPT/ON-LINE/TLV, OPT-RALL/ON-LINE/VRT).....	87
2.3.3.1	Példa.....	89
2.3.4	Tevékenységek megszakíthatósága (ERALL-OPT/MSZT, OPT-RALL/IA).....	90
2.4	Költség-, idő-, erőforrás-optimalás egyidejű megvalósítása (ERALL-OPT/KLTG, OPT-RALL/COST)	92
2.4.1	A feladat megfogalmazása	93
2.4.2	Az algoritmus leírása.....	96
2.4.3	Példa.....	99
2.5	Költségcsökkentés alternatív megoldások segítségével	102
2.5.1	Alternatív megoldások	103
2.5.1.1	Alternatív megoldások keresése	104
2.5.2	Példa.....	105
2.6	Többféle erőforrás egyidejű kezelése, párhuzamos projektek közötti erőforrás-elosztás	110
2.6.1	Párhuzamos projektek közötti erőforrás-elosztás (ERALL-OPT/PP, OPT-RALL/PP)	111
2.6.1.1	Példa.....	113
2.6.2	Többfajta erőforrás egyidejű kezelése (ERALL-OPT/TE, OPT-RALL/MR)	117
2.6.2.1	Példa.....	118
2.7	Erőforráscsoport – csoportos erőforrás-tervezés	119
2.7.1	Erőforráscsoportok	120
2.7.2	Csoportos erőforrás-tervezés	121
2.7.3	Kompetencia alapú kiválasztás informatikai támogatása	122
2.7.4	Tökéletes helyettesítéstől a tökéletes kiegészítésig	126
2.8	Bizonytalan átfutási idejű projektek optimális erőforrás-elosztása.....	127
2.8.1	Sztochasztikus időtervezés	128
2.8.1.1	Példa.....	130
2.8.2	Költségtervezés bizonytalan átfutási idő esetén	131
2.8.2.1	Determinisztikus költségtervezés sztochasztikus időtervezés esetén	131
2.8.2.2	Sztochasztikus költségtervezés sztochasztikus időtervezés esetén	135
2.8.3	Bizonytalanság kezelése kissorozatgyártás termelésirányításában.....	138
2.8.3.1	Paraméterek becslése.....	138
2.8.3.1.1	Példa az időtartamok becslésére	138
2.8.3.1.2	Változó költség és az időtartam kapcsolatának meghatározása.....	142
2.8.4	Példa a bizonytalan átfutási idejű projektek erőforrás-tervezésére.....	146
2.9	Összefoglalás.....	161
3.	A MÓDSZER EGY GYAKORLATI ALKALMAZÁSA AZ INTEGRAL-HEXA Rt-NÉL	163
3.1	Meglévő adatok feldolgozása	163
3.2	Alkalmazott informatikai technológiák bemutatása	173
3.3	Eredmények	183
4.	TÉZISEK	191
5.	FÜGGELÉK	I
5.1	Felhasznált irodalom:	i
5.2	A dolgozatban használt jelölések	xl
5.3	Az algoritmusok szerkezete, tulajdonságok öröklődése.....	xl

1. Irodalmi áttekintés, alapfogalmak

1.1 Bevezetés

Egy szervezet, legyen az profitorientált vállalkozás vagy közszolgáltatást megvalósító kormányzati szerv, minden esetben valamilyen - többé-kevésbé jól körülhatárolható - feladat ellátása érdekében működik. Ilyen közvetlenül teljesítendő cél lehet bizonyos termékek előállítása és értékesítése, szolgáltatások teljesítése stb. [126, 158, 159]

Egy működő szervezet napi feladatait, valamint a feladatok teljesítésének körülményeit és feltételeit hosszabb-rövidebb távon a szervezet belső adottságai és az azokon keresztül érvényesülő külső környezet alakítja ki, illetve határozza meg. A belső és a külső körülmények változásával szükségszerűen együtt jár egy szervezet napi feladatainak változása is, mert szükségszerűvé, és gyakran ezzel együtt lehetővé is válik például a korábban gyártott termékek módosítása, esetleg új, korábban nem gyártott termékek vagy szolgáltatások bevezetése. Az esetek többségében azonban nemcsak a teljesítendő napi feladatok módosulnak, hanem ezek teljesítésének szervezeten belüli körülményei is megváltoznak, így például új technológiák kerülnek alkalmazásra, új gyártókapacitások jönnek létre, új piacok kerülnek megszerzésre, új szervezeti struktúra és új tulajdonosi összetétel, illetve tulajdonforma alakul ki stb. Az ilyen változások közben természetesen az adott szervezet folyamatosan teljesíti és megvalósítja egy adott időszak napi feladatait és közvetlen céljait. [158, 159]

Ez a szervezetek működésében megnyilvánuló kettősség - rövid távon a napi feladatok folyamatos teljesítése viszonylag azonos belső körülmények között, hosszú távon pedig a napi szinten ellátandó tevékenységek változása vagy azok teljesítési körülményeinek az átalakulása - nyilvánvalóvá teszi, hogy egy szervezet vezetése többdimenziós feladat.

A vezetés egyik dimenziója a szervezetben, hogy biztosítsa az éppen aktuális közvetlen célok és napi feladatok folyamatos és eredményes teljesítését, így például egy szériatermékeket előállító vállalatnál a beszerzést és készletezést; a gyártást, összeszerelést és minőségellenőrzést; a raktározást és értékesítést; valamint a reálfolyamatokkal összefüggő pénzügyi, számviteli és egyéb adminisztratív tevékenységeket. A vezetés ebben a dimenziójában ún. *operatív menedzsment*. [158, 159]

Az éppen aktuális közvetlen célok és napi feladatok folyamatos teljesítése mellett egy szervezet vezetőségének szembe kell néznie a változásokkal is. A vezetésnek ebben a minőségében a külső körülmények várható változásaiból, valamint a belső adottságok sajátosságaiból meg kell határoznia a szervezet jövőbeni változási pályáját, ki kell jelölnie azt az állapotot, amelyet fennmaradása érdekében a jövőben el kell érnie. Természetesen ahogy maga a környezet és a szervezet belső adottságai is változnak, annak megfelelően válik többé-kevésbé folyamatossá a szervezet változási pályájának alakítása is, és így újabb és újabb - egymást időben követő, de egymásnak nem szükségszerűen ellentmondó - célállapotok jelennek meg a szervezet jövőképében. A vezetés ebben a dimenziójában *stratégiai menedzsment*. [158, 159]

Reális jövőkép kialakítása esetén - és az annak megfelelő reális stratégiai célok megfogalmazását is feltételezve - a szervezet jövőbeni működőképessége nagymértékben a stratégiai célok realizálásának mikéntjén múlik, ugyanis a stratégiai célok megvalósításának eredményei egy bizonyos időszakra a napi operatív tevékenységek részévé válnak, és jelentősen befolyásolják a szervezet működésének eredményességét. Egy szervezetben a jövőkép elérése érdekében egyidejűleg több stratégiai cél és rész cél is megfogalmazásra kerül, és ezek megvalósítása sokszor párhuzamosan, illetve egymással bizonyos átfedésben vagy egymásutániségben történik. A stratégiai célok és rész célok realizálásának folyamatai nemcsak időben különülhetnek el egymástól, hanem a szakmai tartalom tekintetében is (pl. új termék kifejlesztése, a gyártókapacitás létrehozása, a termék piaci bevezetése), így e vonatkozásban a stratégia realizálásának egy-egy jól körülhatárolható, komplex és egyszeri feladatait is képezik. Ezek a feladatok mind az operatív menedzsmenttől, mind pedig a stratégiai menedzsmenttől eltérő vezetési szemléletet, továbbá eltérő módszereket és technikákat hoztak magukkal a szervezetek vezetésében. Ezért ebben a dimenziójában a vezetés elsősorban *projektmenedzsment*. [158, 159]

A szervezetek vezetésének ez a hármas megnyilvánulási módja gyakorlatilag minden szervezetben tetten érhető, noha sok esetben mindez nem tükröződik a szervezeti struktúrában. Természetes, hogy kisméretű vállalkozásban ezek a funkciók nem különülnek el, és ugyanígy természetes, hogy nagyméretű szervezetekben a stratégiaalkotás, a projektirányítás és az operatív vezetés funkciói mintegy látható módon visszatükröződnek a szervezeti struktúrában is. [158, 159]

A projektmenedzsment tehát egyfajta köztes kategória a vezetés stratégiai és operatív szintjei között, és mint ilyen, a stratégiai célok realizálását valósítja meg, amelynek

következtében a stratégiai célok a napi operatív működés szintjére transzformálódnak, vagyis az elkészült projekteredmény integrálódik a szervezet napi operatív folyamataiba. Gyakorlatilag ezzel valósul meg a stratégiában megfogalmazott változás. Ebben az értelemben a projektmenedzsment a stratégia megvalósításának eszköze, maga a projekt pedig egy-egy konkrét stratégiai program vagy részprogram, illetve stratégiai akció vagy annak egy jól körülhatárolható része. [158, 159]

A hálótervezési technikák különböző területeken és sokféle formában jelenhetnek meg. A termelés, tervezés, elosztás, telepítés, erőforrás-gazdálkodás, pénzügyi tervezés, projektmenedzsment stb. területén különböző problémák megoldására alkalmazzák. [25, 77, 84, 137, 191, 197]

Egy beruházás, vagy egy innovációs projekt megvalósításánál három fontos szempontot kell szem előtt tartanunk: a lehető legrövidebb idő alatt, a lehető legkisebb költséggel kell a projektet megvalósítanunk úgy, hogy a rendelkezésre álló erőforrásainkat (munkaerő, anyagok, gépek stb.) ne lépjük túl. [4, 158, 159, 306-309]

Miért is fontos, hogy a lehető legrövidebb idő alatt és a lehető legkisebb költséggel valósítsuk meg a projektet? Ha egy beruházás megvalósítására több cég vagy szervezet pályázik, akkor általában annak a pályázónak van nagyobb esélye a kiírt tender elnyerésére, aki hamarabb és kevesebb költséggel tudja a beruházást megvalósítani. Ezt a problémát már az ötvenes-hatvanas években hálótervezés (pl. CPM, MPM, PERT), ütemezés [184-186, 389] (pl. Gantt-diagramok, LOB) és ezekhez tartozó költség-optimalizációs eljárásokkal (pl. **CPM/COST-, MPM/COST-, PERT/COST-módszerek**) kezelni tudták. A legnagyobb problémát az erőforrások kezelése jelentette. A megvalósítás során az erőforrások kezelésétől nem tekinthetünk el, hiszen egy projekt esetén a rendelkezésre álló erőforrásaink szűkösek. Meghatározott létszámú munkaerővel, géppel stb. dolgozhatunk. [25, 134, 158, 159]

Ha azt szeretnénk, hogy a lehető legrövidebb idő alatt, a lehető legkisebb költséggel valósítsuk meg a projektet vagy a beruházást úgy, hogy a rendelkezésre álló erőforrásokat (munkaerő, anyagok, gépek stb.) ne lépjük túl, és az erőforrásokat a lehető legjobban használjuk fel, akkor könnyen (akár már 5 000 elvégzendő tevékenység esetén is) olyan komplex problémához juthatunk, amelyet a mai számítástechnikai programok csak nagyon hosszú számítási idővel tudnak megoldani. [202, 298]

A megoldandó feladatot tovább bonyolítja, hogy pl. egy beruházás megvalósítása során az elvégzendő tevékenységek megvalósítási idejét, (változó)költség-igényét, erőforrás-szükségletét csak becsülni tudjuk. [182, 189, 241, 275, 298]

Értekezésemben olyan módszereket mutatok be, melyek kiváló lehetőségként állnak rendelkezésre a menedzserek számára. Az eljárással tetszőleges projekt erőforráskorlátos (költség-, erőforrásigény-, idő)optimális erőforrás-allokációját lehet meghatározni, figyelembe véve az egyes paraméterek becslésének bizonytalanságát is. A bemutatandó módszereket a menedzsment területén minden olyan esetben lehet alkalmazni, ahol a lehető legrövidebb megoldási idő, a lehető legkisebb költség és optimális erőforrásfelhasználás a cél. Ezen belül is a módszert leginkább a projektmenedzsmentben, erőforrás-tervezésben, logisztikában, egyedi termékek gyártásában lehet alkalmazni.

1.2 Projekt, projektmenedzsment

Az általam használt módszereket elsősorban a projektek tervezésénél, szervezésénél lehet alkalmazni, ezért szükséges néhány fogalmat definiálni, mielőtt a hálótervezési és erőforrás-allokációs módszereket áttekinteném.

Aggteleky Béla: „*A projektek időben lehatárolt, gyakorlati vonatkozású vagy absztrakt tervek, amelyek méretük, bonyolultságuk, jelentőségük és egyediségük miatt a menedzsment rutinszerű terv- és vezetői feladatainak keretei között általában nem oldhatók meg kielégítően.*” [4]

Bajna Miklós: „*A projektek egyedi, egyszer előforduló, nagyobb, igényes és komplex problémák, amelyek nagyobb tervezési ráfordítást, speciális szakismereteket és esetenként különleges tervezési eljárást tesznek szükségessé, ezenkívül pontosan meghatározható kezdési és befejezési időponttal, terjedelemmel rendelkeznek.*” [4]

Görög Mihály: „*Projektnek tekinthetünk minden olyan feladatot, illetve annak végrehajtását, amely eltér egy szervezet szokásos, s így rutinjellegűnek nevezhető napi tevékenységétől, és valamilyen egyszeri, komplex feladatot jelent a szervezet számára.*” [158-159]

Robert J. Graham: „*A projekt konkrét, általában egy meghatározott költségvetési és időkereten belül elérendő cél megvalósítására ideiglenes jelleggel összeválogatott emberek és egyéb erőforrások csoportja.*” [159]

A „projekt” egyetlen definíciója sem felel meg minden projekthelyzetnek, de az ISO szabványokban leírt definíciót a felhasználók elfogadhatónak tartják. Az ISO 8402 (1994) a következőket állapítja meg:

Projekt: egyedi folyamatrendszer, amely kezdési és befejezési dátumokkal megjelölt, specifikus követelményeknek - beleértve az idő-, költség- és erőforrás-korlátokat - megfelelő célkitűzés elérése érdekében vállalt, koordinált és kontrollált tevékenységek csoportja. [202]

A fenti definíció mind a költségek, mind a tevékenységek időtartamait, mind pedig az erőforrások kezelését szem előtt tartja. Ezért ezt a definíciót alkalmazom a továbbiakban értekezésem során.

Ehhez a definícióhoz az ISO-dokumentumok - a „projekt” kifejezés pontosítására - további magyarázatokat is fűznek. Az ISO 10006 szabvány - *Irányelvek a minőségi projektmenedzsmenthez* - a projektek alábbi jellemző ismérveit közli: [202]

Ismérvek:

- 1. A szervezet ideiglenes, és a projekt élettartamára alakították.**
- 2. A projekt számos esetben egy nagyobb projektstruktúra részét képezi.**
- 3. A projektcélkitűzéseket és termékjellemzőket folyamatosan lehet meghatározni és elérni a projekt időtartama alatt.**
- 4. A projekt eredménye lehet egy termék egy vagy több egységének megteremtése.**
- 5. A projekttevékenységek közötti viszony összetett is lehet.**

A projektmenedzsment elsődlegesen a változások bevezetéséhez és menedzsmentjéhez kapcsolódik. Bizonyos szempontból minden projekt egyedi, és különbözhet attól a szokásos üzletviteltől, amelyre az anyacéget létrehozták. A projektszervezetet - amelyre gyakran projekt-teamként hivatkoznak, bár ez a teljes projektszervezetnek csak kis része – azért alakítják ki, hogy egy meghatározott célkitűzést – a projektterméket – valósítson meg. [202]

A projektmenedzsment: A vezetéstudomány önálló, független ága, amely egy „totális rendszerbe” integrálja azokat a technikákat, amelyek elősegítik a projekt céljainak hatékony és eredményes realizálását. [202]

1.2.1 A projektek közös elemei

A projektterméknek számos formája lehet, az egészen fizikaitól (egy új város létrehozása vagy egy új mozdony megépítése) a virtuálisan absztraktig bezárólag (folyamat egy lehetséges vészhelyzet kezelésére). E két szélsőséges eset között számos eltérő termék van, mindegyik a maga sajátos követelményeivel, amelyek eltérő projektmenedzsmentet igényelnek. Ez a - cégek és iparágak között elterjedt - sokszínűség gátolta annak a korai felismerését, miszerint minden projektnek vannak közös elemei. [158, 159, 202]

Ennek ellenére mindegyik projekt rendelkezik az összes alábbi tulajdonsággal vagy közülük legalább néhányval:

- a projektnek van terméke, azaz megvalósítandó célja;
- létezik a megvalósítást leíró projektterv;
- adott a projekt megvalósítására szánt időkeret;
- adott a megvalósításra szánt költség;
- létezik a projektidőre bontott költségterv;
- léteznek a projekttermékkel szemben megfogalmazott minőségi elvárások, követelmények;
- létezik a projekt megvalósítását gátló bizonytalansági területek megjelölése, valamint
- az esetleges kockázatok értékelése és a megfelelő reakciók.

A projektkövető rendszereket úgy kell kialakítani, hogy folyamatosan (*real time*) gyűjtsék össze a megvalósításra és a költségekre vonatkozó adatokat. Az adatelemzéshez és az elemzés eredményeinek lehető legrövidebb időn belüli terjesztéséhez megfelelő technikák és eszközök szükségesek. Ezek az eszközök minden projekt esetében jellemzőek, s a terméktől, a projekt méretétől és attól az iparágtól függnék, amelyben használják őket. [202, 307]

Amikor egy hagyományos funkcionális alapon működő cég rájön, hogy kényelmetlenül magas azoknak a projekteknek a száma, amelyekkel foglalkoznia kell, és elég sok olyan projekt van, ami rendszeres igazgatósági konzultációt és jóváhagyást igényel, akkor

(igazgatósági szinten) meg kell fontolni a szervezeti felépítés módosítását. Gyakran nem érzékelik, hogy az ilyen változtatáshoz sok idő szükséges. Az egyik szerző a tapasztalatok alapján úgy véli, hogy bármilyen méretű cégnél 3-5 év kell ahhoz, hogy az új szervezet zökkenőmentesen működjön, illetve kényelmes legyen az emberek számára. Közben kérdések merülnek fel, egyesek tiltakoznak a nyilvánvaló státuszvesztés ellen, politikai mozgások és átrendeződések történnek a projekt ellenében vagy mellett, és kifejlődik a munkát nagymértékben megkönnyítő nem hivatalos struktúra. Kezdetben az egyes projekteket nemcsak megoldandó problémaként látják, hanem valami olyan dologként, amivel szemben ellenállást kell tanúsítani. [158, 202]

1.2.1 A projekttervezés elemei

A hivatalos tervezési rendszerre vonatkozó legkorábbi kísérlet a Gantt-diagram volt, amelyet az egyszerű sávok diagramból származtattak. A „sávok diagram” vagy „sávok ütemterv” kifejezés használata mostanra már általánossá vált. [202]

A projektterv legegyszerűbb formája a projektet alkotó tevékenységek időtáblázatba foglalása. Ez a projektmenedzsment folyamat első fő lépése, kijelöli, hogy „mikor”, „ki és mit” „miként” csinál. A finomítás következő fázisában azt is meghatározza, hogy „milyen teljesítmény- és minőség szinten” és „mekkora költség” mellett. A finomítás egy még magasabb fokán a tevékenységek elvégzéséhez szükséges erőforrások hozzárendelése is megtörténik. Ahogy az egyes fázisokat átgondoljuk, szükség lehet az előzők újraértékelésére és módosítására. [202]

A projektterv elkészítésének folyamata a projektmenedzser és a projekt-team felelőssége. A projektterv elkészítése iterációs és ismétlődő folyamat. A tervet folyamatosan felül kell vizsgálni és frissíteni, ahogy a projekt kifejlődése során az újabb információk hozzáférhetővé válnak. A projektterv sohasem végleges dokumentum; tervezésre – igaz, egyre csökkenő mértékben, de – egészen a projekt befejezéséig sor kerülhet. A projektterv képezi az összes projektbeli előrehaladás viszonyítási alapját. Ezért a lehető legpontosabbnak és naprakésznek kell lennie. *Egyetlen tervet sem szabad megváltoztathatatatlannak tekinteni — hiszen ezek munkaeszközök.* [157, 158, 202]

A projektterven belül mindig számos, ún. projekttervezési szint lesz. A legmagasabb szinten helyezkedik el az összegzés, amely a felső vezetésnek szánt áttekintésként mutatja be a projekt kulcseményeit és mérföldköveit. Konzorcium-projekt esetén a terv jelzi a konzorcium tagjai közötti munka megállapodás szerinti megosztását. Az alacsonyabb szintek

egyre több részletet tartalmaznak, míg a legalacsonyabbak megadják azon feladatgazdák tevékenységlistáját, akik egy meghatározott költségközpontban vagy funkcióban felelősek a tevékenységekért. [202]

A tervben mindig szerepel az elvégzendő munka megvalósítási ütemterve. Ez az ütemterv általában sávos ütemterv formátumú, de célszerűbb, ha a tevékenységek kezdési és befejezési időpontjainak kiszámítására az ún. hálótervezési technikák valamelyikét használjuk. A projekt előrehaladtával a terveket aktualizálni kell, így lehetőség nyílik a végrehajtásra (előrehaladásra) vonatkozó információknak a tervezett megvalósulással való összevetésére. Hosszú átfutási idejű projekt esetén a tervbe foglalt részletek mennyiségének az idővel összhangban kell változnia. Rövid távnál - amely háromtól hat hónapig terjedhet vagy néhány esetben tovább - a részletek teljes kiterjesztésének meg kell jelennie, de a részletezettség az előrehaladással párhuzamosan egyre csökken. A részletek száma minden egyes frissítésnél vagy újratervezésnél nő. [202]

A terv tartalmazza a projekt költségvetését és pénzügyi kimutatásait, amelyekből látható a nettó jelenértékhez igazított kiadás, bevétel és a nettó cash flow. A projekttervnek ezt a részét, ha – a többi részhez hasonlóan – bizalmas információt tartalmaz, a vezetők bizonyos szintjeire lehet korlátozni. [121, 178, 202, 393]

A projekttervben szerepelni fognak a változás- és konfiguráció-menedzsmentre, a minőségre és beszerzésre, valamint a kockázatértékelésre vonatkozó tervek is. Ezek bemutatják mind a pénzügyi, mind pedig a szakmai kockázatokat és azok lehetséges hatását a projektre. Ez lehet az anyacégen belüli különálló csoportok munkája, amelyek a megfelelő technikára szakosodtak, de a projektet illetően a projektmenedzsernek jelentenek, akinek egyetértési és jóváhagyási joga van. Ha léteznek különálló csoportok, akkor elképzelhető, hogy delegálnak egy tagot a projekt-teambe a megfelelő tevékenységek támogatására. Ha nagy projektről van szó, vagy a projekt a kezdeményező cégtől fizikailag távol valósul meg, akkor a projekt-team részeként létrejöhetnek támogató teamek a projektterv néhány vagy összes aspektusára vonatkozóan. [202]

1.2.2 A munkalebontási szerkezet

Az idő alapú terv elkészítésével egyidejűleg a *munkalebontási szerkezetet* (*work breakdown structure* - WBS) is el kell készíteni, ami a projekt összes tevékenységét ábrázolja annak hierarchikusan felépített felosztásával. A WBS-t – alkalmazásától függően – számos alternatív módon lehet felépíteni. Megszokott módszer a termék főbb összetevőkre való felosztása,

amelyeket aztán funkcionális lebontással vagy költségközpontkóddal alcsoportokra és ismét összetevőkre bontanak. A kiválasztott mód általában a projekt típusával, illetve azzal az ipari vagy közszektorral van összefüggésben, amelyben elhelyezkedik. [202]

A WBS legalacsonyabb szintje minden esetben tevékenységcsoportokból áll – vagy egyedülálló tevékenységekből, ha elég nagyok – amelyek egy megnevezett egyén – a feladatgazda – felelősségét képezik. A feladatgazda feladata a munka előrehaladásának beazonosítása, becslése, tervezése, végrehajtása és jelentése a projekttervnek megfelelően. A feladatgazda felelős továbbá a projektterv inputjainak minőségéért, illetve annak biztosításáért, hogy az előrehaladási adatokat pontosan szerezzék meg és a terv szerinti időben adják át. Minden egyes feladathoz szükség van egy „munkakimutatásra” (*statements of work - SOW*), ami kielégítő részletességgel írja le a feladatban szereplő tevékenységeket. Ez félreérthetetlenül kimutatja a feladatgazda projekt iránti elkötelezettségét, és ezáltal biztosítja a tevékenységi lista adatainak teljességét. A szükséges adatok magukba foglalják majd a becsült időtartamot az igényelt erőforrásokkal, költségekkel, teljesítménymérésekkel együtt, beleértve az egyes megállapított minőségi követelmények értékelésének mikéntjét, az esetleges kockázatokat és bizonytalanságokat, illetve a jelentési folyamatok részleteit. [202]

Amikor a feladat saját jogán is alprojekt, tehát a feladatgazda a projekt beszállítójává válik, akkor a munkakimutatásnak lehet, hogy egy jogi szerződés formáját kell magára öltetnie. Abban az esetben ez természetesen így lesz, ha a feladatgazda az anyacégen kívüli. [202, 310]

A projektmenedzsernek a munkamegbízás elfogadása előtt értékelnie kell és meg kell vitatnia a feladatgazdák által biztosított munkakimutatásokat. A megállapodás szerves része, hogy a feladatgazda képes a feladat teljesítésére a projekttervben kijelölt időpontban és a kimutatásban meghatározottak szerint. Tehát a feladatgazda köteles a megfelelő időben megadni az igényelt erőforrásokat anélkül, hogy más munkájába beavatkozna. [202]

A WBS-el egyidejűleg meg kell tervezni a projekttevékenységek közötti logikai kapcsolatokat sávós ütemtervek alkalmazásával (ahol a tevékenységek közötti logikai kapcsolatokat bonyolult kifejezni), vagy inkább hálós ütemtervek használatával (amelyekben a logikai kapcsolatok használata a technika részét képezi). A hálós ütemterv két alapvető megjelenési módja közül – tevékenység a nyilakon ábrázolva (*activity-on-arrow — AoA*) vagy tevékenység a csomópontokon ábrázolva (*activity-on-node - AoN*) – bármelyiket használhatjuk. [202]

1.3 Hálótervezés

Az előző fejezetben említett nehézségeket a *hálótervezési technikák* családjának kifejlesztésével oldották meg.

A hálótervezési technikák (*Project Network Techniques* - PNT) az 1950-es évek közepén és az 1960-as évek elején keletkeztek, amikor ezen technikák két alaptípusát fejlesztették ki. Ezek a javasolt projektet lényegében az *élek* és az azokat összekötő *csomópontok* sorozatából felépülő gráfként ábrázolják. A diagram (projektmodell) szerkezete csak a projekt javasolt megvalósítási módszerétől függ, és úgy ábrázolják, hogy a terv logikai struktúrája könnyen megfigyelhető és ellenőrizhető legyen. Nem szabad azonban elfelejtenünk, hogy az elkészített hálóterv – bármilyen más tervhez hasonlóan – bármilyen részletezettséggel készüljön is, nem a projekt legjobb megvalósítási lehetőségét mutatja be. A terv csupán az alkotó elképzeléseit önti modell formába, s ez a modell lehetőséget ad a tervező elképzeléseinek megértéséhez, illetve ha szükséges, a modell módosításához. [25, 54, 70, 77, 84, 116, 123, 127, 202, 203, 205, 247]

Miután a diagram vélhetően elfogadható logikát mutat, meghatározhatják a különböző tevékenységek időtartamát. Ezután számításokkal meghatározzák a projekt megvalósítási idejét. Ha az eredmények kielégítőek, akkor a projektterv készítését felfüggeszthetjük. A tervezés folytatására akkor kerülhet sor, ha újabb információk merülnek fel a projekttel kapcsolatban. Ha csökkenteni kell a projekt átfutási idejét, akkor megvizsgálják a megvalósulási időtartamot meghatározó tevékenységeket – a „kritikus tevékenységeket” –, hogy lássák, van-e lehetőség az átfutási idő megfelelő mértékű rövidítésére a tevékenység-időtartamok csökkentésével vagy a terv logikájának megváltoztatásával. Egy tevékenység időtartamát megváltoztatni tulajdonképpen lehetetlen, kivéve ha azt technológiai változtatásokkal vagy a tevékenységhez rendelt erőforrás nagyságának megváltoztatásával érik el. Sajnos ez gyakran elfogadhatatlan a felső vezetés számára, melynek tagjai azt hiszik, hogy egy rendelet kiadásával már meg is oldották a problémát. „Nem érdekel, hogy mi áll a tervében, a feladatot a rendelkezésre álló erőforrásokkal rövidebb idő alatt kell elvégezni.” Ezt a kijelentést minden tervező számos alkalommal hallja. A javaslatok változtatása addig folytatódik, míg elfogadható megoldást nem érnek el. [202, 239, 259, 315, 320-321]

Miután az időt és a logikát átgondoltuk, néhány eset kivételével szükséges lehet a terv jelenlegi állása szerint igényelt és a rendelkezésre álló erőforrások összevetése. Ezt a hálóterven történő áthaladással és az egyes időperiódusokban igényelt erőforrások

összeadásával („aggregálás”) lehet megtenni. Az összesített erőforrásokat („erőforrás terhelés”) össze kell hasonlítani a rendelkezésre álló erőforrásokkal („kapacitás”), és ha az igény meghaladja a hozzáférhetőséget, a hálótervet újra meg kell vizsgálni, hogy lássuk, valamilyen módosítással van-e lehetőség az erőforrásigények kielégítő „elosztására”. Ha nincs, akkor világos, hogy vagy a rendelkezésre álló erőforrás-kapacitásokat, vagy a projekt megvalósítási idejét (*total project time* - TPT), vagy mindkettőt növelni kell. [202, 297]

Tudatában kell lenni annak, hogy a hálóterv „időelemzése” minden tevékenységet a legkorábbi megvalósulásra jelöl ki, ezért az erőforrások összesítése a tevékenységek korai kezdésére vonatkozó összesítést jelenti. A hálónak tehát a teher csökkentésére irányuló valamennyi módosítása elkerülhetetlenül késlelteti ugyan az egyes tevékenységeket, de a projekt befejezési dátumát nem szükségszerűen. Az eredményként kapott, más tevékenységekkel való kölcsönhatások és azok erőforrásigényei rendkívül összetetté válhatnak, és a nagyon kis projektek vagy egyszerű szituációk kivételével az ilyen módosításokat megfelelő számítógépes program nélkül nem szabad megkísérelni. [202, 297]

A háló megrajzolásának egyik nagyon fontos velejárója, hogy a diagram hasznos kommunikációs lehetőséget biztosít. Bemutatja, hogy a projektet hogyan lehet végrehajtani, és lehetővé teszi a vezető számára, hogy információt továbbítson egy felettesének vagy alárendeltjének. Valószínű, hogy a hálót nem azok az emberek használják majd, akik elkészítették. Ezért fontos, hogy a tevékenységek a megvalósítandó projektfeladatban *egyértelműen* legyenek definiálva. Ezenkívül, ahol lehetséges, az összetevőket oly módon kell kiválasztani, hogy a tevékenység elvégzésének felelősségét egyértelműen át lehessen ruházni. Ha a helyzet úgy kívánja, a hálótervet könnyedén át lehet alakítani sávos ütemtervekké, amelyekben szükség esetén lehetőség van a változtatás beépítésére vagy módosításra. [202, 310]

A hálótervezési technikákat vizsgálva két hálótervezési család ismeretes: a tevékenységeket a gráf élein ábrázolt (*activity on arrow* - AoA) család, ahol a tevékenységet a gráf éle jelöli; és a tevékenység-csomópont ábrázolása (*activity on node* — AoN) család, ahol a tevékenységet a gráf csomópontja jelöli. Mindkét technikának vannak hívei. Egyik sem mutat túlzott előnyt a másikkal szemben. Néhány szervezetben szükséges, hogy a vevők/ügyfelek igényeinek kielégítése miatt mindkét technika alkalmazható legyen. Körültekintően kell eljárni, nehogy összekeverjük őket. Jelenleg számos kitűnő számítógépes program áll rendelkezésre mind az AoA, mind pedig az AoN családhoz. [25, 54, 77, 84, 116, 123, 127, 202, 203, 205, 247]

A hálótervezési technikákat olyan helyzetekben lehet alkalmazni, amikor a feladat kezdetét és befejezését meg lehet határozni; a folytonos vagy folyamatos nagyüzemi termelést nem érdemes hálótervezési technikákkal modellezni. A projekt mérete nem fontos; a hálótervezést ugyanolyan sikeresen használták már egyszerű tesztfolyamat megtervezésére, mint egy új város konstrukciójának kialakítására vagy egy űrhajó indítására. [157, 158, 202]

Mint minden más új menedzsmenteszközt, a hálótervezési technikák alkalmazását is körültekintően kell bevezetni a szervezetbe. [158, 202]

Minden ütemezési technikánál, így a tevékenységekhez rendelt időnek itt is reálisnak kell lennie, azaz számításba kell venni minden jelenlegi helyi körülményt. Elméleti becslések helyett jobban alkalmazhatók a tapasztalati adatokon alapuló becslések, bár a munka elvégzésének feltételeit össze kell vetni a tapasztalati adatokkal. Rendkívül hasznos összegyűjteni az erőforrásokra és feladatokra vonatkozó adatokat, hiszen az idő- és erőforrás-teljesítmény előre nehezen látható módon hathat egymásra. Rögzítés után ezek segítenek a becslést végzőnek a jelenlegi körülmények átgondolásában, és jobb időtartambecslésekhez vezetnek. Az időtartamoknak inkább *reálisnak*, mint *elvárhatónak* kell lenniük, és el kell őket fogadtatni azokkal, akik megvalósításukért felelősek. [202]

A hálós irányítási rendszerek két ismert alapváltozatát, a PERT- és a CPM-módszert közel egy időben dolgozták ki és publikálták. 1957-ben az USA haditengerészetének különleges tervezési hivatala megbízást kapott a POLARIS rakéták kifejlesztésével kapcsolatos sok száz tevékenység irányítására. [25, 63-66, 71, 199-203, 205, 252]

Az E. I. DuPont de Hemonds and Co. 1956-ban átfogó kutatást indított olyan módszer kifejlesztésére, mely lehetővé teszi számítógép felhasználását a műszaki feladatok megtervezésében és ütemezésében. Walker és Kelley 1957-ben jutott el egy nyíldiagramos, hálós módszert alkalmazó és később CPM néven közismertté váló rendszer kipróbálásáig. A módszert 1959-ben publikálták. [25, 63-66, 71, 199-203, 205, 252]

A fejlődés azóta sem állt meg, a hálós eljárások egész sora jött létre, és számuk óvatos becslések szerint is több mint százra tehető. [25]

Egy (beruházás, innovációs stb.) projekt megvalósításánál három fontos szempontot kell szem előtt tartanunk: a lehető legrövidebb idő alatt, a lehető legkisebb költséggel kell a projektet megvalósítanunk úgy, hogy a rendelkezésre álló erőforrásainkat (munkaerő, anyagok, gépek stb.) ne lépjük túl. [25, 202, 262]

1.3.1 A hálótervezési módszerek csoportosítása

A hálótervezési módszerek felosztása sokféleképpen elvégezhető, ezek közül néhány gyakori csoportosítási szempont: [25, 116]

1. Időtervezés jellege: sztochasztikus, determinisztikus
2. Felhasználási céljuk alapján: idő-, költség- és erőforrásoptimáló technikák
3. A hálók irányultságuk alapján: tevékenységorientáltak vagy eseményorientáltak
4. Megjelenési formájuk szerint: tevékenység-nyíl, tevékenység-csomópontú és esemény-csomópontú hálók

Természetesen még további csoportosítási lehetőségeket sorolhatnánk fel, azonban az általunk kifejlesztett algoritmus(ok) szempontjából ezek ismerete elegendő.

Determinisztikus hálótervezési módszerek: Olyan hálótervezési módszerek, melyeknél a tevékenységidők jól meghatározott értékek. (Ilyen, pl. a CPM-, MPM-, DCPM- stb. háló.) [71, 78, 159, 193] Ezeket a módszereket elsősorban akkor lehet alkalmazni, ha viszonylag pontos becsléssel rendelkezünk a tevékenységek időtartamait illetően. Ha a tervezett és a tényleges időtartamok eltérnek, akkor a hálót aktualizálni kell, illetve ki kell számolni újra a tevékenységek időadatait. Determinisztikus hálótervezési módszerek nem csak kevés, hanem sok tevékenységet tartalmazó, illetve bonyolult hálók esetén is alkalmazhatók. [160, 190, 232]

Sztochasztikus hálótervezési módszerek: Olyan hálótervezési módszerek, melyeknél a tevékenységidőt valamilyen valószínűségi eloszlás sűrűségfüggvénye határozza meg. (Ilyen pl. a PERT- vagy a GERT-háló.) [50, 115] Ezzel a hálótervezési technikával már a tevékenység időtartamának bizonytalanságát is tervezhetjük. [195] Előnye, hogy már a tervezés fázisában meg lehet határozni a projekt egy adott valószínűségi szinthez tartozó várható átfutási idejét. [67, 68, 103-107] Előnye, hogy a „betervezett bizonytalanságnak” köszönhetően előre fel tudunk készülni az esetleges nehézségekre. [201] Szakirodalom szerint ezzel a tervezési technikával akár 10-12% költség is megtakarítható. [217, 219, 245] Hátránya, hogy a tevékenységek időtartamának eloszlása általában nem ismertek, a projekt során sok esetben nincs előzetes információnk a tevékenységek időtartamát illetően, így nagyon nehéz meghatározni a projekt átfutási idejét. [202, 260]

Az időtervezési eljárásoknál cél a projekt átfutási idejét megtalálni. (Ilyen pl. PERT, CPM, MPM stb.). [265] Az időtervezési eljárásoknál determinisztikus esetben viszonylag

könnyű meghatározni a projekt átfutási idejét, míg sztochasztikus hálótervezési eljárásoknál nagyon sok számítási módszer létezik. Ezek közül a legismertebbek az ún. PCI (Path Critical Index) módszer [107] és az ACI (Activity Critical Index) módszer. [131] A legújabb kutatások más sztochasztikus hálótervezési módszereket is vizsgálnak [146, 153-155, 172, 271, 277-279], ezek közül az egyik az ún. Fuzzy-logikán [45, 268-269, 388] alapuló sztochasztikus módszerek. Néhány cikk foglalkozik a hagyományos – projektmenedzsmentben használt – PERT-módszer és a Fuzzy-logikán alapuló becslések összevetésével. [45, 50, 147, 270, 326] A sztochasztikus időtervezési módszerekről elmondható, hogy a projekt átfutási idejére jellemző bizonytalanság meghatározásának módszereiben jelentősen eltérnek. Ennek következtében egy adott valószínűségi szinthez tartozó átfutási idő is jelentősen eltérhet. Ennek a problémának jelentős szakirodalma van. A módszerek mind a kiszámítás komplexitásában, mind pedig a becslés pontosságában jelentősen eltérnek. [117, 134, 136, 142, 162, 163, 168, 189, 191, 203, 234, 247-249, 252-253] A PERT-módszernél tehát a tevékenységek időtartamának bizonytalanságát modellezzük. Egy kutatási, fejlesztési projektnek azonban a kimenetele is bizonytalan lehet attól függően, hogy az egyes tevékenységek (pl. kísérletek) milyen eredményhez vezettek. Ennek megfelelően a háló további tevékenységei is ennek függvényében alakulnak. Ezt a problémát pl. az általánosított **PERT-módszer (GERT-módszer)** modellezi. [15, 293, 327, 329, 386]

A költség- és erőforrás-optimalizáló eljárásoknál az átfutási idő meghatározása mellett a költség-, erőforrás-optimalizálás, kiegyenlítés is fontos szempont. (Ilyen pl. CPM/COST, MPM/COST, CPA stb. RAMPS, RAPP, ERALL stb.). [8-9, 15, 25, 35, 88-91, 94, 102, 149, 156, 214, 215, 272, 278, 279, 352-353] A költségoptimalizáló módszereknél feltételezzük, hogy létezik a tevékenységek időtartamai és a közvetlen költségek között egy determinisztikus vagy egy sztochasztikus függvénykapcsolat. Determinisztikus függvénykapcsolat esetén feltételezzük továbbá, hogy ez a kapcsolat egy konvex függvénnyel írható le. [8, 96, 123-125, 128, 135, 198, 239, 259, 351, 389] Ebben az esetben viszonylag gyorsan (polinomiális időben) meg lehet határozni egy minimális átfutási idővel/minimális összköltséggel járó projektet. [8, 9, 82, 90] Amennyiben a költséggörbék nem konvexek, úgy a feladat visszavezethető egy ún. szeparálási feladatra, melyben a nemkonvex költséggörbék konvex burkainak meghatározásával lehet a minimális közvetlenköltségű időtartamot megtalálni. [195, 356] Az erőforrás-optimalizáló módszereknél a tevékenységek időtartamán kívül azok erőforrásszükségleteit is nyilván kell tartani. Az erőforrás-szükségletek alapján a

rendelkezésre álló erőforrásainkat kell optimálisan elosztani úgy, hogy adott tevékenységeket későbbi időpontra ütemezzünk be. [25, 183] Ahhoz, hogy költség- vagy erőforrásoptimalást végezzünk, járulékos információkra van szükségünk (közvetlen költségigény, erőforrásigény, erőforráskorlát). Ha ezek nem állnak rendelkezésre, akkor ezeket az optimalizálásokat nem lehet elvégezni. [183] Mai napig nyitott kérdés, hogy hogyan kezeljük az egyes költségek, erőforrásigények bizonytalanságát.

A tevékenységorientált hálónál a tevékenységek, míg az eseményorientált hálónál az események hangsúlyozása kerül előtérbe.

Mindegyik módszernek megvan a maga létjogosultsága. Eseményorientált hálónál inkább az eseményekre (határidőkre, egy adott részmunka (várható) lezárására stb.) koncentrálunk, míg a tevékenységorientált hálónál a tevékenységek lehető legkorábbi vagy legkésőbbi megkezdése, illetve befejezése kerül a számítás előterébe. [25, 202]

A tevékenység-nyíl hálónál az élek reprezentálják a tevékenységeket, a csomópontok az eseményeket.

A tevékenység-csomópontú hálónál az élek reprezentálják a tevékenységek közötti kapcsolatokat, a csomópontok a tevékenységeket.

Az esemény-csomópontú hálónál is az élek reprezentálják a tevékenységeket, a csomópontok pedig az eseményeket, de itt az események hangsúlyozása lényeges, míg a tevékenység-nyíl hálónál az események ábrázolását el is hagyhatjuk. [25]

Az angolszász szakirodalomban csak kétfajta ábrázolást különböztetnek meg attól függően, hogy a tevékenységeket a csomópontokon, vagy az éleken reprezentáljuk. Tulajdonképpen mindkét hálótervezési technikának megvannak az előnyei és a hátrányai. [202]

A tevékenység-nyíl hálók mind az események (legkorábbi/legkésőbbi) bekövetkezése, mind pedig a tevékenységek legkorábbi/legkésőbbi kezdése, illetve befejezése kiszámítható. Hátrányuk viszont, hogy általában csak vég-kezdet kapcsolatot lehet velük reprezentálni. Sok kellemetlenséget okozhat továbbá az ún. látszattevékenységek helytelen használata. Tipikus képviselőjük a CPM-, illetve a klasszikus PERT-hálók. [115, 202]

A tevékenység-csomópontú hálók esetén általában többfajta kapcsolatot (kezd-kezd, vég-vég, kezd-vég stb.) is használhatunk. Előnyük továbbá, hogy nincs szükség

látszattervékenységek alkalmazására. Jellemzőjük, hogy eseményeket nem használnak. Tipikus képviselőjük az MPM/PDM. [78, 267]

1.3.2 Alapfogalmak

A legfontosabb gráfelméleti és hálótervezési alapfogalmakat tekintem át a következőkben. A hálótervezési alapfogalmak egy része az egyes hálókra specifikus.

1.3.2.1 Gráfelméleti alapfogalmak

Gráf: $G = (N, A)$ egy véges ponthalmaz (csúcsok), és egy véges pontpárhalmaz (élek) együttese. N ponthalmaz a csúcsok halmaza $N = \{N_1, N_2, \dots, N_n\}$. A pontpárhalmaz az élek halmaza $A = \{A_1, A_2, \dots, A_m\}$, ahol $A_k = (N_i, N_j)$. Irányított gráf esetén a pontpárok rendezettek, ekkor N_i az A_k él kezdőpontja, N_j pedig a végpontja. Irányítatlan gráf esetén a pontpárok nem rendezettek, vagyis $(N_i, N_j) = (N_j, N_i)$. [6, 13, 62, 65, 180, 200, 376]

Hurokél: Ha $A_j = (N_i, N_i) \in A$, akkor azt mondjuk, hogy A_j egy hurokél.

Többszörös él: Ha $\exists m, n$ melyre $(N_i, N_j) = A_m = A_n = (N_i, N_j)$, és $A_m, A_n \in A$; $N_i, N_j \in N$, akkor a gráfban N_i és N_j között többszörös él van. [6, 13, 62, 65, 180, 200, 376]

(Irányítatlan) út: Az élek olyan sorozata, melyben bármely két szomszédos élnek van közös pontja. [6, 13, 62, 65, 180, 200, 376]

A továbbiakban hurok- és többszörösél-mentes irányított gráfokkal foglalkozom.

(Irányított) út: Élek olyan sorozata, amelyben bármely él végpontja azonos a következő él kezdőpontjával (kivéve az utolsót). [6, 13, 62, 65, 180, 200, 376]

(Irányított) egyszerű út: Olyan (irányított) út, ahol minden él csak egyszer szerepel. [6, 13, 62, 65, 180, 200, 376]

(Irányított) kör: Olyan (irányított) út, amelyben az első él kezdőpontja azonos az utolsó él végpontjával. [6, 13, 62, 65, 180, 200, 376]

(Irányított) egyszerű kör: Olyan (irányított) kör, amelyben egy él csak egyszer szerepel. [6, 13, 62, 65, 180, 200, 376]

Erdő: (Irányított) körmentes gráf. [6, 13, 62, 65, 180, 200, 376]

Összefüggő gráf: Egy gráfot összefüggőnek nevezünk, ha bármely két pontja között létezik egy (irányítatlan) út. [6, 13, 62, 65, 180, 200, 376]

Erősen összefüggő gráf: Egy gráfot erősen összefüggőnek nevezünk, ha bármely két pontja között létezik egy (irányított) út. [6, 13, 62, 65, 180, 200, 376]

Fa: Összefüggő, kört nem tartalmazó gráf. [6, 13, 62, 65, 180, 200, 376]

Aciklikus gráf: Kört nem tartalmazó gráf. [6, 13, 62, 65, 180, 200, 376]

Súlyozott gráf: irányított vagy irányítatlan gráf akkor, ha minden élhez egy vagy több számot rendelünk. [6, 13, 62, 65, 180, 200, 376]

Háló: Olyan súlyozott körmentes, irányított gráf, amelynek egy kezdő és egy végpontja van. [25, 376]

A továbbiakban tárgyalt hálónál a többszörös és hurokél nem megengedett, a súlyok időadatokat reprezentálnak.

Megjegyzés: a CPM-hálónál az alábbi esetek mindig fennállnak:

1. A hálóban nem lehet irányított kör.
2. A hálóban csak egy forrás (kezdő esemény) és egy nyelő (záró esemény) van.

Ha ez nem teljesül, akkor a háló felrajzolásakor logikai hibát vétettünk. [203]

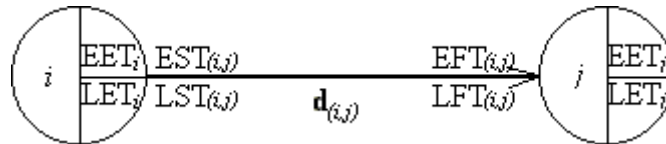
A továbbiakban a hálót adottnak tekintjük, valamint ismertnek tételezzük fel a tevékenységek idejét, erőforrásigényét.

1.3.3 Tevékenységidő- és eseményidő-adatok

A *TPT* (Total Project Time = teljes projekt átfutási ideje) kiszámításához el kell végeznünk a progresszív időelemzést, amivel az egyes tevékenységek legkorábbi kezdési időpontját ($EST_{(i,j)}$) számítjuk ki. Ebből meghatározhatjuk a legkorábbi befejezési időpontot, ahol a legkorábbi befejezési időpont ($EFT_{(i,j)}$) = a legkorábbi kezdési időpont ($EST_{(i,j)}$) + a tevékenység időtartama ($d_{(i,j)}$). A teljes projektidő (*TPT*) tehát az a legrövidebb időtartam, ami alatt a projekt befejezhető, és ezt a tevékenységek sorrendje (vagy sorrendjei) kritikus útként (vagy utakként) határozza (határozzák) meg. [200, 203]

A kritikus út meghatározására a retrográd számítás elvégzése után kerülhet sor, így a tevékenység legkésőbbi kezdési pontját ($LST_{(i,j)}$), valamint a hozzá tartozó legkésőbbi befejezési időpontot ($LFT_{(i,j)}$) határozhatjuk meg a következőképpen: Legkésőbbi kezdési időpont ($LFT_{(i,j)}$) = legkésőbbi befejezési időpont ($LST_{(i,j)}$) – tevékenység időtartama ($d_{(i,j)}$).

Egy csomóponthoz (eseményhez) két idő tartozik. A progresszív elemzésből az esemény legkorábbi bekövetkezésének időpontja (EET_i), vagyis az a legkorábbi időpont, amelyre az eseményt realizálni lehet. Számítása: $EET_j = \max_i EFT_{(i,j)}$. A retrográd elemzésből az esemény legkésőbbi bekövetkezésének időpontja (LET_i), vagyis az a legkésőbbi időpont, amelyre az eseményt realizálni kell $LET_i = \min_j LST_{(i,j)}$. [200, 203]



1.3.3.-1 ábra: esemény- és tevékenységidők

1.3.3.1 Tartalékidők

A szakirodalomban tevékenységek esetében általában négy tartalékidőt különböztetnek meg. Ezek az alábbiak: [200, 203]

- Teljes tartalékidő (Total Float): az a teljes időtartam, amivel egy tevékenység kiterjedhet vagy késhet a teljes projektidőre (TPT) gyakorolt hatás nélkül. Számítása: $TF_{(i,j)} = LST_{(i,j)} - EST_{(i,j)} = LFT_{(i,j)} - EFT_{(i,j)}$.

A gyakorlatban ezt a tartalékidőt alkalmazzák a leggyakrabban, hiszen ez adja meg egy tevékenység maximális csúszásának mértékét, mellyel még nem változik meg a projekt átfutási ideje. Fontos megjegyezni ezzel a típusú tartalékidővel kapcsolatban, hogy egy tevékenység teljes tartalékidején általában több azonos ágban lévő tevékenység osztozik.

- Szabad tartalékidő (Free Float): az a teljes mennyiség, amivel egy tevékenységidő megnőhet, vagy a tevékenység csúszhat anélkül, hogy hatással lenne bármely, soron következő tevékenység legkorábbi kezdetére. Számítása: $FF_{(i,j)} = EET_j - EFT_{(i,j)}$.

Ennek a típusú tartalékidőnek kulcsszerepe van az erőforrástervezés során. A szabad tartalékidő mondja meg ugyanis, hogy mennyivel csúszhat maximálisan egy tevékenység úgy, hogy az őt követő tevékenységek még időben el tudjanak indulni. A megvalósítás során a tevékenység ezen tartalékidejével „szabadon” gazdálkodhatunk, ezen más tevékenységgel nem osztozik, vagyis az adott tevékenység csúszása, ha kisebb, mint ez a szabad tartalékidő, akkor az más tevékenység kezdési idejére nincs hatással. A szabad tartalékidő e fontos tulajdonságát használják ki a heurisztikus erőforrástervezésben, hiszen ezen tartalékidővel rendelkező tevékenységeket későbbi időpontra mozgatva egyrészt a többi tevékenység kezdési ideje nem módosul, másrészt sok esetben a szabad

tartalékidővel rendelkező tevékenységek későbbi időpontra történő mozgatásával meg lehet határozni egy erőforráskorlátos erőforrásallokáció megengedett megoldását. Általánosságban elmondható, hogy a szabad tartalékidő nem lehet negatív szám, és nem lehet nagyobb, mint a teljes tartalékidő. Szabad tartalékidő csak a csomópontokban jöhet létre.

- Feltételes tartalékidő: a teljes és a szabad tartalékidő különbsége.
- Független tartalékidő (Independent Float): azt az időmennyiséget adja meg, amennyivel az adott tevékenység eltolható, ha az őt közvetlenül megelőző tevékenység a lehető legkésőbbi időpontban fejeződik be, és a közvetlenül következő tevékenység a legkorábbi időpontban kezdődik.

Számítása: $IF_{(i,j)} = EET_j - LET_i - d_{(i,j)}$. Ha $IF > 0$, akkor belefér a tevékenység megvalósítása.

Ha $IF < 0$, akkor $|IF|$ -el csúszhat az egész program megvalósítása. [25, 200, 203]

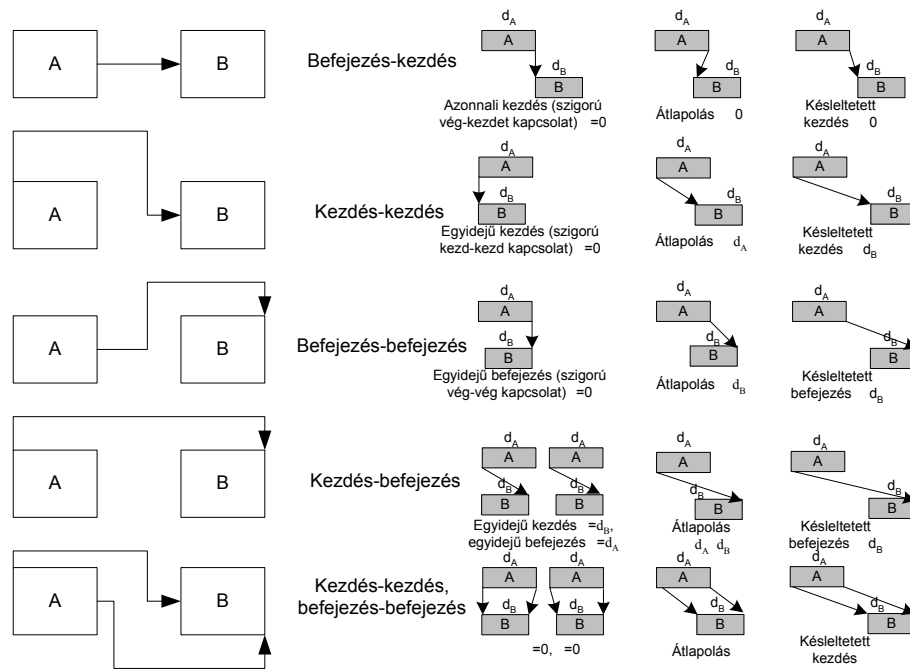
1.3.4 Az MPM-háló

Az MPM (Metra Potenciális Módszer, az angolszász országokban Precedence Diagramming Method)-technika a francia Roy nevéhez kötődik. A kézi ábrázolású technika a tevékenységeket a gráf csomópontjaiként ábrázolja, a gráf élei pedig a tevékenységek közötti logikai kapcsolatokat szimbolizálják. [203, 265]

Az MPM-háló a logikai kapcsolatoknál kezeli az ún. minimális és maximális kapcsolatokat, kezeli a vég-kezdet, kezdet-vég kapcsolatok minden kombinációját. [203, 265]

Az MPM-technikával megszakítható tevékenységek is tervezhetők.

1.3.4-1 ábra: egy tevékenység az MPM-hálóban



1.3.4-2 ábra: tevékenységek lehetséges kapcsolatai az MPM-hálóban

 α β

Bizonyos megszorításokkal a különböző tevékenységkapcsolatok egymásba is konvertálhatók.

- Irányelvek:
 - A hálótérvezés során a kiértékelésnél egy minimális illetve egy maximális kapcsolatot használunk.
 - A különböző kapcsolatok egymásba csak bizonyos megszorításokkal konvertálhatók (különösen figyelni kell ezekre a megszorításokra költségtervezésnél), így ezeket a konverziókat célszerű jelezni.

Példa: minimális kapcsolatok konvertálása kezd-kezd kapcsolatokká:

- Befejezés – kezdés kapcsolat konverziója:



$$\beta = d_A + \alpha$$

- Befejezés – befejezés kapcsolat konverziója:



$$\beta = d_A + \alpha - d_B$$

- Kezdés – befejezés kapcsolat konverziója:

 γ δ



$$\beta = \alpha - d_B$$

Számos programcsomag tevékenység-csomópontú hálót kezel, ezért gyakran előfordulhat, hogy egy tevékenység-nyíl hálót át kell alakítani tevékenység-csomópontú hálóvá. [203, 265]

A CPM =>MPM átalakítás menete:

1. Minden tevékenységből (kivéve a látszattevékenységet), melyet a tevékenység-nyíl hálóban a nyílakon szerepeltetünk, most csomópontokként reprezentáljuk.
2. A tevékenységeket a logikai kapcsolataik szerint kapcsoljuk össze.
3. A tevékenységek legkorábbi, illetve legkésőbbi kezdési illetve befejezési idejei, a projekt átfutási ideje, a tevékenységek tartálásideje meg kell hogy egyezzenek a két hálóban.
4. MPM-ben az eseményidőket nem használjuk!

1.3.5 Véletlen időtartamú tevékenységek

Főleg kutatási és fejlesztési programokban, illetve nagy rizikójú projektek esetén a tevékenységek tartamai kevésbé ismertek, és nem determinisztikusan meghatározottak. Ilyenkor két eset fordulhat elő: [55, 140, 200, 203, 281, 311]

1. a szóban forgó tevékenységek vagy nem teljesen ismeretlenek és mindegyikükre közelítőleg ismerjük az időtartamuk valószínűség-eloszlását (pl. ipar),
2. vagy pedig teljesen ismeretlenek és/vagy nem ismerjük minden időtartam valószínűség-eloszlását (kutatás).

Ha nem ismerjük az időtartamok eloszlását, akkor a projekt tervezésekor a számítások megkönnyítése érdekében sok esetben - a hálótervezésben használt - **PERT-módszert** használjuk, és feltesszük, hogy az időtartamok ún. β -eloszlást követnek. [98, 200, 203, 324-325, 358-359, 370] (Természetesen más módszerrel is lehet modellezni az időtartamok bizonytalanságát. [14, 72, 231, 240-241, 256, 275, 369, 390] Mégis a projektmenedzsmentben sztochasztikus időtartamok tervezésére a legtöbb esetben **PERT-módszert** alkalmaznak.) A β -eloszlás sűrűségfüggvénye a normális eloszlásra hasonlít, de mindkét irányban korlátos (egy-egy A , illetve B ponttól kezdve „kifelé” azonosan nulla). [129, 140, 203, 206, 267]

Definíció: A ξ valószínűségi változót α és β paraméterű béta-eloszlásúnak nevezzük, ha sűrűségfüggvénye:

$$f(x) = \begin{cases} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}, & x \in (0,1) \\ 0, & x \notin (0,1) \end{cases}, \quad (1.3.5-1)$$

ahol $\alpha > 0$ és $\beta > 0$ paraméter rögzített, és az ún. gamma függvény a következőképpen

$$\text{számítható: } \Gamma(\alpha) = \int_0^{\infty} u^{\alpha-1} e^{-u} du, \quad \alpha > 0. \quad (1.3.5-2)$$

Az eloszlás momentumai: [3, 109, 129, 188, 203, 267, 275]

$$M^r = \frac{\Gamma(\alpha + r)\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\alpha + \beta + r)}, \quad r > 0. \quad (1.3.5-3)$$

Ebből a várható érték és a variancia (szórásnégyzet): [109, 203]

$$E(x) = \frac{\alpha}{\alpha + \beta} \quad (1.3.5-4)$$

$$D^2(x) = \frac{\alpha\beta}{(\alpha + \beta + 1)(\alpha + \beta)^2}$$

A módusz:

$$M = \frac{\alpha - 1}{\alpha + \beta - 2}. \quad (1.3.5-5)$$

A **PERT-módszerben** a gyakorlatban legtöbbször olyan β -eloszlást választunk, amelyre a tevékenységek várható időtartama, varianciája (szórásnégyzete), illetve szórása az alábbi módon számítható: [263, 266]

$$E(t_{(i,j)}) = \bar{t}_{(i,j)} = \frac{1}{k+2} (A_{(i,j)} + kM_{(i,j)} + B_{(i,j)}) \stackrel{k=4}{=} \frac{1}{6} (A_{(i,j)} + 4M_{(i,j)} + B_{(i,j)}) \quad (1.3.5-6)$$

$$D^2(t_{(i,j)}) = \sigma_{(i,j)}^2 = u_{(i,j)}^2(t_{(i,j)}) \stackrel{k=4}{=} \left[\frac{1}{6} (B_{(i,j)} - A_{(i,j)}) \right]^2$$

ahol $A_{(i,j)}$, $B_{(i,j)}$ és $t_{(i,j)}$ valószínűségi változó alsó, illetve felső korláta, $M_{(i,j)}$ a legvalószínűbb értéke (módusza), $E(t_{(i,j)})$ a várható értéke, $D^2(t_{(i,j)})$ pedig a varianciája (szórásnégyzete). [129, 140, 188, 195, 200, 202, 203]

A tevékenység időtartamának becslésekor minden egyes tevékenységről az azzal foglalkozó szakemberekhez a következő három kérdést intézzük: [261]

1. Mennyire becsüli (i,j) tevékenység $A_{(i,j)}$ minimális időtartamát (optimista becslés)? Legyen a_{ij} a minimális időtartam becsült értéke.

2. Mennyire becsüli (i,j) tevékenység $B_{(i,j)}$ maximális időtartamát (pesszimista becslés)?

Legyen $b_{(i,j)}$ a maximális időtartam becsült értéke.

3. Véleménye szerint mennyi (i,j) tevékenység $M_{(i,j)}$ legvalószínűbb időtartama (módusza)?

Legyen a $m_{(i,j)}$ legvalószínűbb időtartam becsült értéke. [203]

Minthogy más ismeretekkel nem rendelkezünk, azért feltesszük, hogy ezek az információk $A_{(i,j)}$, $B_{(i,j)}$, $M_{(i,j)}$ torzítatlan becslését adják, amit a következőképpen jelölünk:

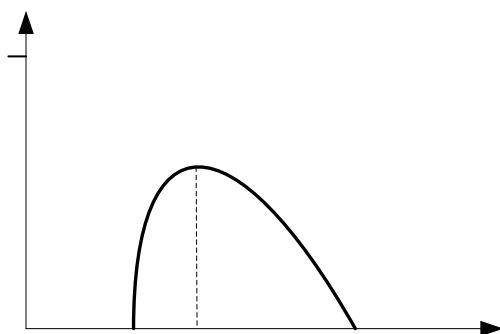
$$a_{(i,j)} = \hat{A}_{(i,j)}, b_{(i,j)} = \hat{B}_{(i,j)}, m_{(i,j)} = \hat{M}_{(i,j)} \quad (1.3.5-7)$$

Ekkor felírható a várható értékre:

$$\hat{t}_{(i,j)} = \frac{1}{6} [\hat{A}_{(i,j)} + 4\hat{M}_{(i,j)} + \hat{B}_{(i,j)}] = \frac{1}{6} [a_{(i,j)} + 4m_{(i,j)} + b_{(i,j)}] \quad (1.3.5-8)$$

A szórásnégyzetre már nem írható fel hasonló összefüggés, de közelítésként elfogadjuk az alábbi összefüggéseket:

$$\sigma_{(i,j)}^2 \approx \left[\frac{\hat{B}_{(i,j)} - \hat{A}_{(i,j)}}{6} \right]^2 \approx \left[\frac{b_{(i,j)} - a_{(i,j)}}{6} \right]^2 \quad (1.3.5-9)$$



1.3.5-1 ábra: a β -eloszlás sűrűségfüggvénye

Ekkor felhasználjuk a független valószínűségi változók várható értékeire, illetve varianciáira vonatkozó additivitási összefüggéseket, hiszen elegendően sok független valószínűségi változó esetén az összeg közelítőleg normális eloszlásúnak mondható.

[140, 195, 200, 202, 203]

$$E\left(\sum_{i=1}^n t_i\right) = \sum_{i=1}^n E(t_i) \quad (1.3.5-10)$$

Tehát a projekt várható átfutási ideje meg fog egyezni a kritikus úton lévő tevékenységek várható időtartamainak összegével. [23, 192, 195, 200, 202, 203] A projekt átfutási idejének szórásának meghatározására több módszer is létezik. Az egyik ilyen az ún.

PCI/ACI-értékek kiszámítása, mely érték megmondja, hogy egy tevékenység mikor lehet kritikus úton. [108, 131] Ha egy tevékenység kritikussá válhat, akkor azt valamilyen módon figyelembe kell venni az átfutási idő bizonytalanságának kiszámításánál is. Az alkalmazott módszerek többségében ebben térnek el egymástól. Talán a legegyszerűbb módszer, amikor csak a kritikus úton lévő tevékenységek szórásnégyzeteinek összegét választjuk az átfutási idő szórásnégyzetének. Ez tulajdonképpen egy alsó becslése az átfutási idő varianciájának, hiszen egyáltalán nem veszi figyelembe, hogy bizonyos valószínűséggel az alternatív utakon lévő tevékenységek is válhatnak kritikussá és ekkor e tevékenységek időtartamainak szórását nem lehet elhanyagolni. Tehát ezt a módszert csak bizonyos fenntartásokkal lehet alkalmazni. [115] A másik igen egyszerű módszer azt az esetet tekinti, hogy adott valószínűséggel bármelyik tevékenység válhat kritikus tevékenységgé, így a leghelyesebb, ha valamennyi tevékenységet számításba vesszük. [191] Nyilvánvaló, hogy ebben az esetben egy felső becslést kapunk. Az „arany középút” megtalálása igen komoly matematikai feladat. Ennek megfelelően a problémára számos különféle megoldás született a heurisztikus megoldásoktól Monte Carlo analízisen át az egzakt megoldásokig. [36, 46-49, 56, 68, 104-108, 115-117, 129-131, 133, 142, 143, 150, 151, 162, 163, 191, 201, 209, 230, 233-236, 247-249, 283-285, 300, 331-334, 342, 348-350, 353, 360-361, 371, 375, 381, 385, 387]

PERT-háló felrajzolása, időtartamok, és azok szórásainak kiszámítása

A PERT-háló felrajzolásakor, a tevékenységek időtartamainak kiszámításához az alábbi teendőket kell elvégeznünk: [349]

1. Logikai háló elkészítése.
2. $A_{(i,j)}$, $B_{(i,j)}$, $M_{(i,j)}$, $t_{(i,j)}$, $\sigma_{(i,j)}$ meghatározása.
3. Megfelelő hálós modell kiválasztása (tevékenység-nyíl, tevékenység-csomópontú). (A **PERT-módszert** mindkét hálótervezési eljárásnál lehet alkalmazni)
4. Az előző fejezetekben tárgyalt módszerekkel (lásd: **CPM**, **MPM**) a kritikus út kiszámítása.
5. A megvalósítási idő szórásának kiszámítása.
6. Alternatív úton lévő tevékenységek, illetve a tartalékidők szórásainak kiszámítása

A PERT-háló aktualizálása

Annak ellenére, hogy a **PERT-módszer** segítségével a projekt átfutási idejének bizonytalanságát közelítőleg becsülni tudjuk, nem várt események miatt – amelyek jelentősen befolyásolhatják a projekt átfutását – szükség lehet a PERT-háló aktualizálására. A háló aktualizálásakor a következőképpen járunk el: [349, 357]

1. A már lefutott tevékenységek tényleges időadatait írjuk be a hálóba a várható értékük helyett. Ekkor ezen tevékenységek szórása nulla. (Ha semmilyen változás sem történik a becsült időhöz képest, akkor is az átfutási időre egy pontosabb értéket kapunk, hiszen az átfutási idő varianciája csökken).
2. A még le nem futott tevékenységek időtartamait módosít(hat)juk aszerint, hogy a lefutott tevékenységek mennyire térnek el a becsülttől.
3. A változtatott adatokkal a hálót (vagy a részhálót) újra számítjuk.

1.3.6 Költségoptimalizálás

A pénzügyi költségvetés meghatározása a következő:

Pénzügyi és/vagy mennyiségi kimutatás, amely egy meghatározott időszak előtt készül, az időszak alatt követendő politika szerint, azzal a céllal, hogy egy adott célkitűzést elérjen.

Ez a hálóterv jellemzője is – sőt, bármely átfogó projekttervé –, és gyakran sokat segíthet, ha a hálót idő-költségvetésként fogjuk fel, azaz meghatározzuk, hogy adott időszakokra milyen költségek merülnek fel. A gazdasági szakemberek és a könyvelők sok ismeretet szereztek a pénzügyi költségvetések összeállításában és alkalmazásában. A hálóttervek felvázolásakor és alkalmazásakor hasznos lehet figyelembe venni ezt a tapasztalatot. [202]

A hálók hierarchiája olyan, mint a költségvetések hierarchiája: lehetővé teszi a problémák felesleges részletek nélküli beazonosítását, és a felelősséget a cég szerkezetének megfelelő pontjára képesek helyezni. [202, 309]

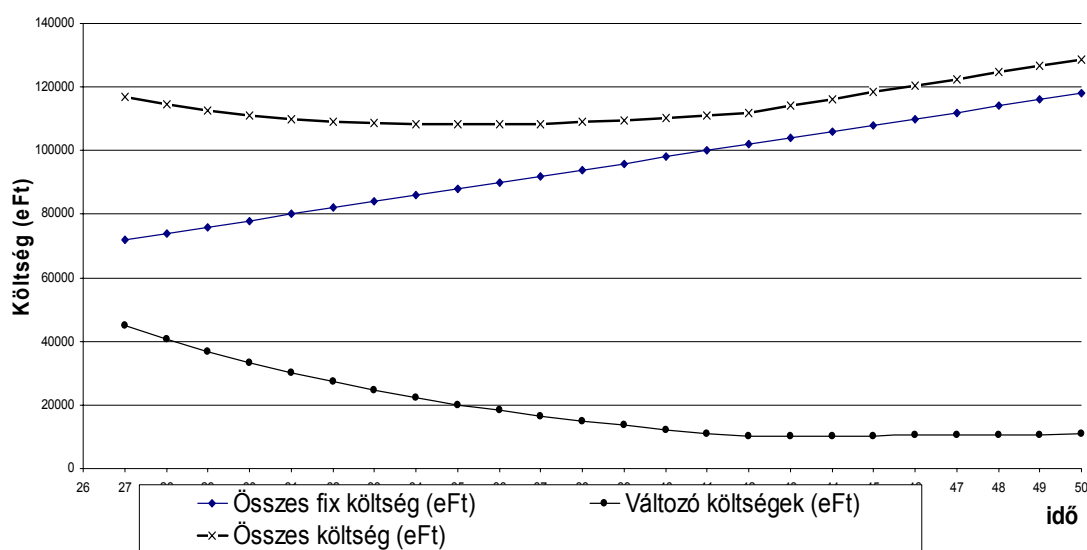
Nem lehet eléggé hangsúlyozni, hogy az időtartamok, erőforrások és költségek értékelése a tervezés nagy fontosságú aspektusa. A hálótervből levont következtetések nem lehetnek jobbak, mint a kalkulációkban használt információk; következésképpen fontos, hogy a tevékenységek végrehajtásáért felelős menedzserek feltétlenül vegyenek részt az

értékelésben. Ha nem konzultálnak velük, akkor esetleg figyelmen kívül hagyják a tervet, mert teljesen irreálisnak tartják, s ez katasztrofális következményekkel járhat a projekt számára. [202]

Az idő- és erőforrásadatok elengedhetetlenek az ütemezéshez és az erőforrás-allokációhoz. Ezenkívül közöttük alakul ki a tevékenységek költségalapja, ami azután a projekt költségtervét képezi. Sok – különösen a feldolgozó iparágakon belüli – szervezet számára bonyolult a projekt erőforráskontrollja, mivel a projekthez felhasználható erőforráskapacitások előzetesen gyakorlatilag ismeretlenek, így a projektköltségterv a projektkontroll legfőbb módja. [182, 202]

A költségoptimalizálás során vagy a minimális összköltséggel rendelkező termelési programot, vagy a minimális átfutási idejű, lehető legkisebb változó költséggel járó termelési programot keressük úgy, hogy figyelembe vesszük, hogy egy átlagos projekt, illetve termelési program esetén a költség-idő függvények általában a következőképpen alakulnak:

Költségek alakulása az idő függvényében



1.3.6-1 Költségek alakulása az idő függvényében

- A változó költség-idő függvény általában monoton csökkenő a minimális és a projekt normál átfutási ideje alatt, hasonlóan igaz ez az egyes tevékenységekre is. A normál átfutási, illetve a tevékenységeknél a normál időtartam után a függvény általában monoton nő.
- A fix költség-idő függvény általában monoton nő a teljes projekt átfutási idejére nézve.
- Valamennyi költség-idő függvény általában konvex.

[8, 96, 123-125, 128, 135, 198, 239, 259, 389]

1.3.6.1 Determinisztikus költségoptimalás tevékenység-nyíl típusú hálók esetén

A determinisztikus költségoptimalás mind tevékenység-csomópontú, mind pedig tevékenység-nyíl típusú hálók esetén használható. (A továbbiakban a determinisztikus tevékenység-nyíl hálók költségoptimalását **CPM/COST-módszereknek** nevezem, noha az alkalmazott módszerek eredményeinek kiszámítási módjában eltérhetnek.) A **CPM/COST-módszerek** elsősorban projektmenedzsmentben alkalmazott hálótervezési és költségtervezési technikák. Az algoritmusok során először egy CPM-hálót kell felrajzolni, majd a kritikus úton lévő minimális költségnövekedéssel járó tevékenységek időtartamait csökkentjük. A módszerekre jellemző, hogy algoritmikus, mohó (greedy) eljárások, tehát egyben költség- és időoptimalis megoldásokat találnak, valamint ha adott egy költségkorlát, amelyet nem léphet túl, akkor meghatározható a költségkorlátot nem túllépő költség- és időoptimalis megoldás. [29, 96, 126, 129, 137, 156, 239, 259, 352-353, 389] Determinisztikus költségsökkentésre esettanulmány található Dr. Papp Ottó: Projektmenedzsment a gyakorlatban című könyvének 13. fejezetében: „Számítógépes információs rendszer kiépítése és a rendszerfejlesztési projekt átfutási idejének csökkentését célzó döntések megalapozása – a hálós projekttervezési technikák felhasználásával; egy innovációs (fejlesztési) projekt esetén.” című esettanulmányban. [310]

1.3.6.2 Determinisztikus költségoptimalás tevékenység-csomópontú hálók esetén

Ezek a módszerek (továbbiakban **MPM/COST-módszerek**) is hasonlóan a **CPM/COST-módszerekhez** a kritikus ut(ak)on lévő tevékenységek közül azok időtartamait csökkenti, amelyek (egységnyi) költségnövekedési tényezője a legkisebb, és a csökkentés révén az átfutási idő is csökken. [8, 9, 356, 382]

1.3.6.3 Determinisztikus költségoptimalás – további módszerek

A **CPM/COST-**, **MPM/COST-módszert** elsősorban lineáris költségfüggvények esetén lehet alkalmazni, bár létezik konvex függvényekre kialakított változatuk is. Determinisztikus tevékenység-nyíl hálók költségminimalizálására számos módszer született. Ezen módszerek elsősorban a futási sebességben térnek el egymástól. [83, 88-90, 94, 96, 100, 112, 124-126,

129, 167, 197, 200, 274, 315, 320-321] Kamburowski költségminimalizáló algoritmusával pl. lineáris, illetve közel lineáris futási időt is el lehet érni. [194]

1.3.7 Bizonytalanság kezelése hálótervezési technikák segítségével különböző logisztikai feladatok esetén

Gyakran előfordul, hogy egy logisztikai feladat során nemcsak arra vagyunk kíváncsiak, hogy az adott feladat várhatóan mikor hajtható végre, milyen költséggel jár, hanem arra is, hogy egy termelés során ezt az előírt időt, költséget milyen valószínűséggel fogjuk tudni ténylegesen betartani. [224]

Sztocasztikus folyamatok jellemzése – stacioner folyamatok:

A sztochasztikus folyamatok egyik legfontosabb alosztályát a stacionárius folyamatok képezik. A stacionaritás a statisztikai jellemzők időfüggetlenségével (időbeli eltolásra való invarianciájával) van kapcsolatban. Pl. ha valaki egy adott időpillanatban ismeri a folyamat egydimenziós projekciójának az eloszlását, akkor vajon milyen hosszú ideig marad "érvényes" ez az ismeret? Időben ugyanaz marad-e, vagy megváltozik a folyamat viselkedését meghatározó "véletlen sorsolási mechanizmus"? Egy stacioner folyamatnál érvényes, hogy amit statisztikailag ismerünk a folyamatról egy adott időpillanatban, az igaz lesz bármilyen jövőbeli időpillanatban. (Pl. egy kockadobás eredménye mindig egyenletes eloszlást követ, függetlenül attól, hogy vasárnap 5 órakor vagy szerdán 10 órakor végezzük el a kísérletet, feltehető továbbá, hogy a tevékenységek lefutási ideje is minden esetben β -eloszlást követ, melyet a későbbiekben tárgyalunk.) [48, 72, 92]

A stacionaritás fogalma attól is függ, hogy a folyamatot milyen mélységig jellemző statisztikai jellemzőre teljesül az időbeli eltolással szembeni invariancia. Ebben két szintet különböztetünk meg:

- gyenge stacionaritás, amikor csak a várható érték és a korrelációs függvény időeltolással szembeni invarianciája teljesül;
- erős stacionaritás, amikor az összes véges dimenziós projekció valószínűségeloszlás-függvénye invariáns az időbeli eltolásra.

Az erősen stacionáris folyamatokon belül fontos alosztályt képeznek az ergodikus folyamatok. Egy folyamat statisztikájának a felderítéshez valaki megfigyeléseket végezhet az idő folyamán (különböző időpillanatokban mintavételezve a folyamatot), majd ezen minták alapján időbeli átlagokkal próbálja közelíteni a folyamat statisztikai jellemzőit. Pl. az egydimenziós projekció alapján a várható értékét úgy próbálja megállapítani, hogy a folyamat tíz különböző időpillanatban megfigyelt értékét átlagolja. Ez az eljárás a következő fundamentális kérdést veti fel: mennyire megbízhatók az időátlagolással kapott becslései a folyamat valódi statisztikájának? A kérdés hasonló ahhoz a kísérlethez, amikor valaki egyszerre dob fel tízezer kétforintost, vagy időben egymásután tízezerszer dob fel egy darab kétforintost azért, hogy empirikusan határozza meg a fej vagy írás valószínűségi változó várható értékét (vagy bármilyen más statisztikai jellemzőjét, pl. eloszlását). Van-e bármilyen különbség a két kísérlet között? (Azt leszámítva, hogy az első kísérlethez jóval nagyobb "tőkeberuházás" kell, mint a másodikhoz.) A választ az ergodicitás fogalma adja meg. Egy sztochasztikus folyamat ergodikus, ha szinte bármilyen g függvényre (pontosabban bármilyen g Borel-mérhető függvényre) az időbeli és statisztikai átlagok azonosak, azaz:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} g(x(t)) dt = E(g(\xi_t)). \quad (1.3.7-1)$$

A tevékenységek lefutási idejének további statisztikai vizsgálatának szempontjából a g függvénynek két konkrét választása érdekes. Mikor g az identitásfüggvény, az maga után vonja, hogy

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} x(t) dt = E(\xi_t). \quad (1.3.7-2)$$

Mikor g a négyzetes függvény, az maga után vonja, hogy

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} x^2(t) dt = E(\xi_t^2). \quad (1.3.7-3)$$

Ez annyit jelent, hogy a folyamat első- és másodrendű statisztikája időben egymásután megfigyelésekből és átlagolásokkal rekonstruálható. [72, 92]

Ezeket a tulajdonságokat feltesszük a tevékenységek lefutásainak, költségeinek, erőforrásszükségleteinek becslésekor. (Lásd 2.8.1. fejezet) (Tehát a továbbiakban a vizsgált valószínűségi változókról feltesszük, hogy ergodikus folyamatok.)

1.3.7.1 Bizonytalanság

A mérés technikában az **ISO GUIDE TO THE EXPRESSION OF UNCERTAINTY IN MEASUREMENT 1993** – as ajánlásában 3 bizonytalansági fogalmat különböztet meg. Ezek a következők:

- Standard bizonytalanság (standard uncertainty)
- Összetett bizonytalanság (combined standard uncertainty)
- Kiterjesztett bizonytalanság (expanded uncertainty)

[397]

1.3.7.2 Standard bizonytalanság

A standard bizonytalanság meghatározásához először fel kell állítanunk egy modellfüggvényt.

$$Y=f(X_1, X_2, \dots, X_N) \quad (1.3.7.2-1)$$

Y a modellfüggvényünk kimenete, $X_1, X_2, \dots, X_N$ a függvényünk paraméterei. Legyen y Y

becslése (általában tekinthető y Y_i mért értékek átlagának: $y = \frac{1}{N} \sum_{i=1}^N Y_i = \bar{Y}$). (Vagy súlyozott

átlagának) [397]

A várható érték standard bizonytalanságát úgy számíthatjuk, hogy a várható értéktől való átlagos négyzetes eltérések összegét osztjuk a szabadságfokkal, és ebből négyzetgyököt vonunk. Képlettel:

$$u^2(x_i) = s^2(x_i) = \frac{1}{(N-1)} \sum_{j=1}^N (x_{i,j} - \hat{E}(X_i))^2, \text{ ahol } i=1, 2, \dots, M. \quad (1.3.7.2-2)$$

A képletben u az ún. standard bizonytalanság. M db paramétert (itt az utak költségét) kell becsülnünk. $x_{i,j}$ i . paraméter j . becslése (pl. egy adott út költsége). Ha az adott paraméter közvetlenül megfigyelhető (itt a költség közvetlenül megmérhető), akkor $x_{i,j}=X_{i,j}$, vagyis i . paraméter j . becslése megegyezik i . paraméter j . mérésével. $\hat{E}(X_i)$ az i . paraméter várható értékének becslése. A várható érték becslésére választhatunk egyszerű átlagot, mozgó átlagot, exponenciális illesztést stb. Az egyszerű átlag sokszor azért nem jó becslési módszer, mert valamennyi mérést ugyanannyi súllyal veszi figyelembe, ezzel a módszerrel pl. nagyon nehéz

a költségnövekedéseket (pl. infláció) megfelelően figyelembe venni. Ezért célszerűbb egyéb más módszerrel becsülni a várható értéket, amely a régebbi méréseket csökkentett súllyal veszi figyelembe. A standard bizonytalanságunk akkor lesz minimális, ha a várható érték becslésére az egyszerű átlagot választottuk. Ez egyrészt az egyszerű átlag becslési tulajdonságaiból adódik, másrészt a standard bizonytalanság a várható értéktől történő eltérést ugyanolyan súllyal kezeli. Ezért ha figyelembe akarjuk venni a súlyokat is, akkor a bizonytalanság kiszámításánál a tapasztalati szórás helyett súlyozott tapasztalati szórást alkalmazhatunk. [397]

$$u_w^2(x_i) = s_w^2(x_i) = \frac{1}{(N-1)} \sum_{j=1}^N w_j (x_{i,j} - \hat{E}(X_i))^2, \text{ ahol } i=1,2,\dots,M. \quad (1.3.7.2-3)$$

Ekkor a standard bizonytalanság helyett súlyozott standard bizonytalanságot (u_w) számítunk. Ha a bizonytalanságokat össze akarjuk hasonlítani, akkor a relatív standard bizonytalanságokat kell meghatározni ($\Delta u, \Delta u_w$). Ezeket úgy számíthatjuk ki, hogy az adott (súlyozott) standard bizonytalanságot elosztjuk a várható érték becslésével. [397]

1.3.7.3 Összetett bizonytalanság

Az összetett bizonytalanság a következőképpen számítható ki általános esetben:

$$\begin{aligned} u_c^2(y) &= \sum_{i=1}^M \sum_{j=1}^M \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) = \\ &= \sum_{i=1}^M \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + 2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) \end{aligned} \quad (1.3.7.3-1)$$

ahol $u_c(y)$ az ún. összetett standard bizonytalanság, $u(x_i, x_j)$ x_i és x_j paraméter közötti kovariancia. A képlet korrelációkkal is felírható, amely a következőképpen módosul: [397]

$$u_c^2(y) = \sum_{i=1}^N \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i) u(x_j) r(x_i, x_j), \quad (1.3.7.3-2)$$

ahol $-1 \leq r(x_i, x_j) \leq 1$ x_i és x_j paraméter közötti korreláció.

Az összetett bizonytalanság kiszámításánál nem feledkezhetünk meg a korrelációról, mely megmutatja a különböző paraméterek közötti kapcsolatok erősségét. [3, 397]

1.3.7.4 Kiterjesztett bizonytalanság

A (súlyozott) kiterjesztett bizonytalanságot a következőképpen írhatjuk fel:

$$U(y) = k u_c(y) \quad (1.3.7.4-1)$$

$$U_w(y) = k u_{c,w}(y) \quad (1.3.7.4-2)$$

Itt U az ún. kiterjesztett bizonytalanság, U_w pedig a súlyozott kiterjesztett bizonytalanság. k az ún. átfedési tényező (coverage factor). Ez az érték egy eloszlásfüggő paraméter (pl. itt $\nu=N-1=9$ szabadságfokú egyoldali t -eloszlás esetén $k=2,26$). A központi határeloszlás tétele miatt $N \rightarrow \infty$ esetén t -eloszlás helyett használhatunk normális eloszlást is. A mintaelemek számának növelésével (legalább 50, 100) ezt a közelítést megtehetjük, anélkül hogy számottevő hibát elkövetnénk. [397]

Az ISO által bevezetett ajánlott mérési kiértékelési módszereket már számos területen különböző méréseknél széles körben alkalmazzák. Ez a módszer azonban alkalmazható különböző logisztikai problémák megoldására, tervezésre, előrejelzésre is. [397]

1.3.8 Erőforrás-tervezés

A számítástechnika fejlődésével lehetőség nyílt olyan problémák megoldására is, amelyek megoldása korábban elképzelhetetlennek tűnt. Ilyen probléma volt az erőforrás-elosztás kérdése is. A gyakorlatban az erőforrás-elosztás problémája sok helyen előkerül. [362, 364-367, 392] Hatékony erőforrás-elosztásra van szükség például egy hálózati operációs rendszer, vagy egy telekommunikációs rendszer megvalósítása során. Itt például erőforrás lehet a felhasznált memória, különböző háttértárolók stb. Mivel ezeknek az eszközöknek különböző az elérési sebességük, a feladatok helyes ütemezése kulcskérdés lehet a gyors feldolgozás szempontjából. Ilyen jellegű problémákra elsősorban heurisztikus algoritmusok készültek melyek ezt a fajta erőforrás-tervezési problémát viszonylag gyorsan igen nagy hatékonysággal oldják meg. [16, 17]

A projektmenedzsmentben fellelhető erőforrás-tervezési problémák általában azonban más jellegűek. A feladat megoldására készültek heurisztikus, evolúciós-szimulációs és egzakt algoritmikus eljárások is. A menedzsment területén kétfajta erőforrás-tervezéssel, erőforrás-allokációval foglalkozunk attól függően, hogy mit tekintünk „erősebb” korlátnak. [39, 343, 373]

Időkorlátos erőforrás-allokációnak nevezzük azt az erőforrás-tervezést, amikor az erőforrásainkat úgy kell csoportosítani, hogy az adott projektet vagy beruházást egy adott időkorlát alatt végre lehessen hajtani. Itt tehát a korlátot elsődlegesen az időkeret adja, a cél pedig: minél kevesebb vagy minél egyenletesebb erőforrás-felhasználás. [39, 98, 110, 111, 202]

Olyan projektek esetén célszerű ezt a módszert használni, ahol lehetőség van pótlólagos erőforrások bevonására, de a projekt csúszása nem megengedhető (pl. reaktorcsere). [98, 202, 297]

Ezzel szemben az *erőforrás-korlátos erőforrás-allokáció*nál a korlátot a meglévő erőforrásaink szolgáltatják. Ezt az erőforrás-korlátot tehát nem léphetjük túl, a célunk pedig: a program végrehajtása a lehető legrövidebb idő alatt. [31, 33, 39-44, 133, 202, 243, 244, 287, 297, 377-380]

Ebben az esetben tehát az erőforrásaink jelentik a szűk keresztmetszetet. Természetesen itt is az a célunk, hogy a projektet minél hamarabb be tudjuk fejezni, de a meglévő erőforrásainkból kell gazdálkodni. Pótlólagos erőforrások igénybevételére általában nincs mód. A legtöbb projekt esetén ezt a módszert alkalmazzák, hiszen a projekt vezetője pontos képet kap, hogy a rendelkezésére álló erőforrásokból adott időpontra meg lehet-e valósítani a projektet vagy sem. Ha nem, akkor még mindig elgondolkodhat azon, hogy pótlólagos erőforrásokat vesz-e igénybe, illetve kiadja-e a részmunkákat alvállalkozóknak. [39, 69, 79-81, 85-87, 97-99, 110, 111, 118, 246, 250, 273, 297, 354-355]

Bármelyik erőforrás-allokációs módszert alkalmazzuk is, legtöbb esetben a beruházó cégek megelégszenek egy ún. megengedett megoldás megtalálásával, vagyis egy olyan ütemtervvel, ahol a megvalósítás során a rendelkezésre álló erőforrásainkat vagy időkeretünket nem lépjük túl. Egy ilyen megengedett megoldás néhány tevékenység esetén kézzel, több tevékenység esetén számítástechnikai szoftverekkel (pl. Microsoft Project, CA-SuperProject, Primavera stb.) viszonylag könnyen megtalálható. Egy ilyen ún. megengedett erőforrás-allokáció esetén sokszor számos probléma merülhet fel. [111, 202, 399-401]

Több mint 200 tevékenységet tartalmazó projektek esetén fontos lehet például, hogy az erőforrásainkat viszonylag egyenletesen használjuk fel, vagy a tevékenységeket a lehető legkorábban végrehajtsuk stb. Ezeket a kívánalmakat célként (célfüggvényként) fogalmazhatjuk meg. Egy *erőforrás-allokációs probléma (adott célfüggvényre nézve) optimális megoldása* egy olyan megengedett ütemterv lenne, amely az adott célt legjobban kielégíti (vagy másképpen a célfüggvény értéke ebben az esetben minimális/maximális). [202, 220-221, 296, 328]

1.4 Algoritmusok csoportosítása

Általában egy probléma megoldására használt módszereket három csoportba oszthatjuk. [6, 20, 101, 144, 187, 242, 251, 306, 336, 344, 376]

- Heurisztikus módszerek
- Evolúciós módszerek
- Algoritmikus módszerek

A **heurisztikus módszerek**re jellemző, hogy hamar adnak gyors megoldást, melyek viszonylag jól használhatók, de legtöbbször nem garantálják az optimális megoldás megtalálását. Az ilyen algoritmusok valamilyen jól bevált tapasztalati módszereket követnek, melyek néha egymásnak is ellentmondanak. A heurisztikus módszerek igen közkedveltek. Ennek egyik oka, hogy viszonylag kevés számítási időt igényelnek, másik nagy előnye, hogy a fejlett heurisztikus módszerek (pl. genetikus algoritmusok) viszonylag gyorsan optimumközeli megoldást szolgáltatnak. [10-12, 24, 27, 32, 37, 41, 204, 211, 212, 216, 254, 290, 301-305, 313, 338, 341, 374, 393-395] Legújabb kutatások a mesterséges intelligencia eszköztárát is kihasználják. Ennek köszönhetően az algoritmusok képesek tanulni. Korábbi megoldáskereséseket elemezve a következőkben esetleg gyorsabban találnak egy elfogadható megengedett megoldást. Másik fontos kutatási terület az ún. genetikus algoritmusok alkalmazása. Ezt a módszert az evolúciós módszereknél is előszeretettel alkalmazzák. A módszer lényege, hogy a különböző egyedek – melyek pl. maximalizálási/minimalizálási feladatban egy adott pontban helyezkednek el, és a maximalizálandó, vagy minimalizálandó függvény abban az adott pontban lévő értékével rendelkeznek – egymással rekombinálandódnak és új egyedeket hozhatnak létre. Kutatási eredmények azt mutatják, hogy ezekkel a módszerekkel viszonylag gyorsan lehet megengedett megoldást találni a legtöbb optimalizációs feladatra. [154-156, 301, 321, 323] Sajnos azonban, ha az optimalitás számunkra fontos, akkor ezek a módszerek kevésbé használhatók, hiszen optimális megoldás megtalálását egyik módszer sem garantálja.

Az **algoritmikus módszerek** ezzel szemben garantálják az optimális megoldást, ennek ára viszont az, hogy általában jóval lassabbak a heurisztikus módszereknél. Vízonylag gyors (polinomiális) lefutási idejű problémák esetén szinte kizárólag algoritmikus módszereket alkalmaznak, hiszen ebben az esetben a probléma számítási igénye még kezelhető, és valóban optimális megoldást el tudunk érni véges (és még belátható) időn belül. Heurisztikus, vagy algoritmikus módszerek alkalmazásának kérdése kombinatorikus feladatok esetén vethető fel elsősorban. [242, 306] Itt a feladatok számítási igénye igen nehéz. NP-teljes vagy NP-nehéz

feladatok megoldása sok esetben egyszerű algoritmikus eljárásokkal szinte reménytelen. [144, 181, 306] A feladatmegoldás gyorsítására általában két irányba lehet elindulni. Az egyik módszer az ún. korlátozás és szétválasztás (angolul: Branch and Bound), ahol a kiértékelés során olyan allokációkat, melyek nem vezetnek optimális megoldásra, kiejtjük a döntési fából. Ezeket a részproblémákat tovább nem értékeljük ki. Ezáltal jelentősen csökkenthetjük a probléma nagyságát. [61, 169, 170, 242, 298, 306] A másik gyakran alkalmazott módszer a dinamikus programozás, melyben szintén felbontjuk a problémákat további részproblémákra. Ezeket akár egyidejűleg is kiértékelhetjük. [26, 101, 118, 335] Több számítógépen egyidejűleg futtatva egy ilyen problémát akár töredékére csökkenthetjük a szükséges számítási időt.

Az *evolúciós módszerek* a kettő közti átmenetet képviselik. Egy heurisztikus módszer által megadott megengedett megoldásból indulnak, amelyet fokozatosan javítanak. Hátrányuk ugyanaz, mint a heurisztikus megoldásoknak, vagyis nem garantált az optimális megoldás megtalálása véges lépésben. Ide tartoznak például a különböző szimulációs módszerek, melyek minden lépésben egyre jobb megoldást szolgáltatnak. [34, 57, 76, 242]

1.5 Ütemezés, erőforrás-allokáció

A hálótervezés egyik legfontosabb feladata, hogy a mai gazdasági életben az egyes folyamatokat leírjuk, tevékenységeiket megadott sorrendben és adott időtartammal szemléltessük, valamint az így kapott gráfból matematikai segédletekkel, algoritmusokkal a legmegfelelőbb sorrendet állítsuk fel a folyamatok közt, figyelembe véve a tervezés során a technológiai, gazdasági összefüggéseket. [38, 119, 202, 212]

Egy projekt tervezésekor adottak bizonyos tevékenységek (pl. egy építés esetén alapítása, betonozás, a tetőszerkezet kialakítása stb.) A tevékenységekhez rendelhetők erőforrások, amelyek megmutatják, hogy az adott tevékenység elvégzéséhez mennyi erőforrásra van szükség (pl. hány emberre, berendezésre stb.). [28, 161, 162, 202]

A tevékenységek egymásutániségát (rákövetkezési relációit) sokféle módon ábrázolhatjuk. Talán a legismertebb módszer a CPM-háló vagy MPM-háló alkalmazása.

Az időegység lehet bármilyen (nap, hét, hónap stb.). A tervezéskor mindig azt az időtartamot rendeljük a szóban forgó tevékenységhez, amely alatt a tevékenység normális körülmények között elvégezhető. Ezen tevékenységekhez (pl. CPM-háló esetén) nyilatkat

rendelünk, ezzel jelöljük őket a hálóban. Ennek az időtervezésnek a célja, hogy az egész hálóterv átfutási idejét meghatározzuk. **[4, 25, 202]**

A logikai háló tervezése után a kezdő és befejező esemény között az egyes tevékenységeket jelölő nyilak mentén haladva több utat találhatunk. Az egyes utak az időtervezés szempontjából akkor nyernek értelmet, ha hozzájuk a végrehajtáshoz szükséges időtartamot rendeljük. A hálóban a kezdő és befejező esemény között több út is létezhet, és ezek különböző összidőt igényelnek. Ezek közül kell kiválasztanunk azt az utat, mely a leghosszabb idejű, és ezt nevezzük kritikus útnak. Ez azért kritikus, mert ha ezeknek a tevékenységeknek – melyek ezen az úton találhatóak – a kezdése csúszik, akkor az egész program ideje elcsúszhat, mely egy nagy projekt esetében hatalmas veszteséget jelenthet. **[4, 25, 122, 202, 307]**

A módszer során különféle időmutatókat használhatunk fel az elemzésre, mint például legkorábbi kezdési időpont, legkorábbi befejezés, legkésőbbi kezdés és befejezés stb.

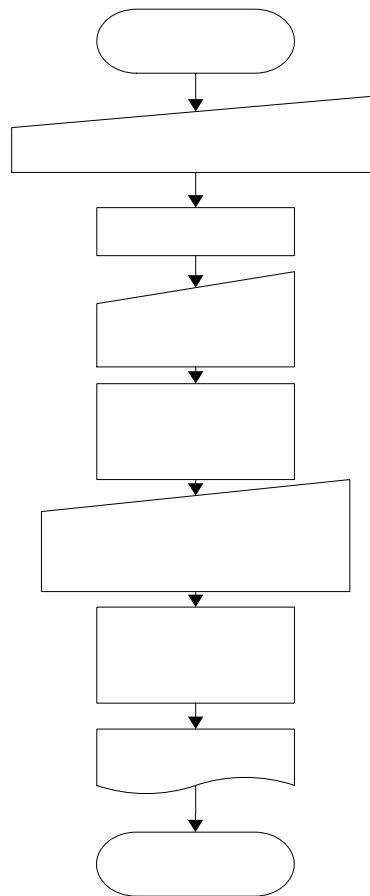
Ezek segítségével a tartalékidők kifejezhetők, melyek a feltételektől függően különbözőek. (Lásd. 1.3.4. fejezet) **[4, 25, 202, 307]**

A CPM- és az MPM-hálótervezés és ezek elemzése messzebb mutat, mint azt elsőre felismerjük. Ezen eljárással költségoptimalizálást is végezhetünk, mert az élet legtöbb területén a tevékenységek bizonyos költséggel hajthatók végre, viszont a végrehajtási idő rövidíthető is megfelelő költségnövekmény ellenében. **[8, 9, 18, 25, 137, 218-219, 239, 352]** Erre a problémára a Kelley-Fulkerson-féle algoritmus meghatározza az egész háló közvetlen költségeit. **[139, 203-205]**

A költségek minimalizálása mellett az erőforrások hatékony felhasználása is cél a projekt megvalósítása során. Az erőforrások hatékony felhasználásának vizsgálatát végezhetjük az ütemezéssel együtt vagy az ütemezés után. Mindegyik módszernek megvan a maga előnye és hátránya. Fontos megjegyezni, hogy míg az ütemezés, illetve konvex költségfüggvények esetén a költségminimalizálás polinomiális időben megoldható (viszonylag kevés számítási időt igénylő) feladat **[51, 88, 94, 120]**, addig az erőforrások optimális allokálása (kombinatorikus) NP-nehéz feladat. **[7, 81, 144, 181, 242, 276, 306]**

Legújabb kutatások nyomán erőforrás-allokációra a mesterséges intelligenciában használt heurisztikus algoritmusokat, illetve genetikus algoritmusokat is alkalmaznak. Ezek a módszerek képesek a korábbi allokálásokból „tanulni”. Többszöri alkalmazás során hasonló erőforrás-allokációkra egyre jobb eredményt kapunk. **[10, 164, 173-177, 301]**

A legtöbb erőforrás-allokáció alkalmazása esetén először a következő lépéseket kell elvégeznünk.



1.5.-1 ábra: erőforrás-allokációs módszerek lépései

A logikai tervezés során olyan hálóttervet készítünk, amely a technológiai szempontból megengedhető maximális párhuzamosítási lehetőségeket tünteti fel. Így a logikai tervezés eredménye egy maximálisan tömörített háló lesz. Amennyiben meghatároztuk / megbecsültük a tevékenységek (várható) időtartamát, akkor valamely ütemezési algoritmussal kiszámítjuk a tevékenységek legkorábbi, illetve legkésőbbi kezdését, illetve befejezését. Ha további információink vannak, pl. meg tudjuk határozni vagy becsülni a tevékenységek (várható) erőforrásszükségleteit, illetve a rendelkezésre álló erőforrásainkat, akkor egy erőforrás-allokációs módszerrel lehetőségünk van olyan ütemtervet készíteni, melynél az erőforrásainkat megfelelően használjuk fel. [4, 73-75, 157, 158, 173-177, 288-289, 312] Léteznek olyan módszerek is, melyek az ütemezés során is figyelembe veszik a

tevékenységek erőforrásigényét, és a tevékenységek legkorábbi, illetve legkésőbbi kezdését, illetve befejezését ennek függvényében módosítják. [7, 28, 42, 118, 264, 340]

Az egyes tevékenységekhez felvett erőforrásigényt egy vonalas ütemterven szemléltethetjük, majd ezután felrajzolhatjuk a terhelési diagramot. Ennek tervezési szabálya (erőforráskorlátos erőforrás-allokáció esetén) a következő:

- Az erőforrásigényt négyszöggel jelöljük. Először mindig a kritikus út tevékenységeinek erőforrásigényét rajzoljuk be, mutatva ezzel azt, hogy ezen tevékenységeknek itt kell maradniuk ahhoz, hogy a kritikus időpont ne csússzon.
- Így kapunk egy olyan ún. terhelési diagramot, melyen feltüntetjük az egyes tevékenységeket a megfelelő erőforrásigényükkel.

Célunk kettős: egyrészt az egész programot továbbra is a legrövidebb idő alatt fejezzük be, de úgy, hogy egy meghatározott erőforráskorlátot ne lépjünk túl. [113, 341] A kiinduló állapotban minden tevékenységet a lehető legkorábbi időre ütemeztük be. Ezt az időpontot meg lehet határozni egyrészt az ütemezésnél (ebben az esetben viszont a legtöbb esetben túllépjük az erőforráskorlátot), [288-289] másrészt alsó korlátot lehet keresni, mely figyelembe veszi az erőforrásigényeket, illetve az erőforráskorlátokat is (ez a megoldás általában szintén nem lesz megengedett, de jobb becslést kapjuk a tevékenységek lehető legkorábbi kezdésének). [43, 169-171, 210, 280, 291-292, 294-295, 345-346, 396] Bármelyik módszert használjuk is, az erőforráskorlátot túlléphetjük. Ezért például egy úgynevezett simító eljárást alkalmazhatunk a kapott terhelési diagramon, mely segítségével egy *megengedett megoldást* kapunk, mely azt a feltételt teljesíti, hogy a munkafolyamatunk időtartama alatt nem fogjuk túllépni az adott erőforráskorlátot, valamint nem sértjük meg a rákövetkezési relációkat, melyek megadják, hogy egy tevékenységet milyen más munkafolyamat követ. [166, 282, 288-289]

Valamennyi eddig alkalmazott módszer nagy hiányossága, hogy a humán erőforrásoknál nem számol annak az adott munkában való jártasságával, illetve nem foglalkozik azzal a kérdéssel, hogy amennyiben az ajánlati ár kisebb, akkor mekkora az a maximális minőségi kompromisszum, amellyel még elvállalható a projekt megvalósítása.

Olyan projektek esetében, amelyek több mint 10 000 tevékenységet tartalmaznak, nincs mód az optimális eredmények előállítására, mivel a probléma naggyá és összetett kombinatorikussá válik. [202]

Vannak olyan algoritmusok is, melyek az ütemezést és az allokálást egyszerre végzik. Előnye, hogy időt takarítanak meg. Hátránya, hogy a közbenső lépések sokszor ütemezés szempontjából sem megengedettek. [7, 19, 21, 22, 92, 93, 347]

1.5.1 Allokálás

A hatvanas évek elején kifejlesztették az allokálási probléma két megközelítését, amelyek erőforráskorlátos és időkorlátos vagy simításos allokálási eljárásként ismertek. Azt remélték, hogy ezek esetleg megfelelő választ adnak az erőforrás- vagy időkorlátok alternatív problémájára. Mindegyikre több eljárást dolgoztak ki. Az eljárások két csoportba oszthatók: soros és párhuzamos feldolgozáson alapulóakra. [202, 213]

Az eltérés a tevékenységek „megszakításának” projektekbe tervezett mennyiségét érinti.

Egy **tevékenység megszakíthatóságát** a következőképpen definiálhatjuk: „Egy jelenleg folyamatban lévő tevékenység megállása azáltal, hogy erőforrásait elveszik azért, hogy egy magasabb prioritású tevékenységhez használják fel őket.” A tevékenység-megszakítást általában mindkét eljárásban megvalósítják, a procedúra kiválasztását a tervezett megszakítás várható mértékének kell irányítania. Ha a projektben a tevékenységek jelentős hányadát szándékozzák megszakítani, akkor párhuzamos allokálást kell alkalmazni; egyébként pedig soros allokálást. [202, 213]

A soros és párhuzamos eljárásokat valamilyen formában valamennyi jelenleg rendelkezésre álló projektmenedzsment-szoftverben alkalmazzák. Ez a csoportosítás ma is hasznos módja a programok osztályozásának. Különböző feltételek mellett mindkettő sikeresen célozza meg az erőforráskorlátos eset problémáját. [13]

1.5.1.1 A soros allokálási eljárás

Ez olyan eljárás, amelyben egy konstans prioritási szabály alkalmazásával még a tevékenységek ütemezése előtt rangsorolják a projekt(ek)ben szereplő tevékenységeket. E prioritási lista alapján szigorú rendben ütemezik be a tevékenységeket – a lehető legkorábbi időpontban, az erőforrások hozzáférhetőségének és a háló(k) követelményeinek megfelelően. Egy soros eljárásban megszokott a megelőzési korlátoknak (rákövetkezési relációnak) a háló időelemzésén alapuló, szabályozott, automatikus számításba vétele. A legelterjedtebben használt szabály az, hogy a tevékenység legkésőbbi kezdési időpontját használják a dátumok

emelkedő sorrendjében, és a teljes tartalékidő alkalmazása által felmerülő esetleges kötöttségeket feloldják – megintcsak növekvő sorrendben. [202, 213]

A figyelembe vett tevékenységek ütemezésekor – arra alapozva, hogy minden megelőző tevékenység befejeződött – a rendszer először azt a legkorábbi időpontot ellenőrzi, amelyhez a tevékenységet allokálni lehet. Azután meghatározza, hogy abban az időpontban elegendő erőforrás áll-e rendelkezésre a tevékenység megkezdéséhez. Ha nem, akkor előkeresi az erőforrás-hozzáférhetőségi táblázatokat, hogy meghatározható legyen a tevékenység-időszak ütemezése. A tevékenység ütemezése után a rangsorolt lista következő tevékenységét ugyanígy fontolóra veszik. [202, 213]

1.5.1.2 A párhuzamos allokálási eljárások

Ez olyan eljárás, amelyben prioritásuk szerint csak a kezdésre képes tevékenységek kerülnek rangsorolásra – az egyes ütemezési időszakoknál alkalmazott állandó szabállyal. Az ütemezéshez ebből a listából felmerülési sorrendben, az erőforrások hozzáférhetőségétől függően veszik ki a tevékenységeket. A be nem ütemezett tevékenységeket visszahelyezik a listába, hogy a következő ütemezési időszakban az új tevékenységekkel együtt rangsorolhassák őket.

Ez az ütemezés nagyon eltérő filozófiájú, mert egy listában gyűjti össze az összes olyan tevékenységet, amely egy meghatározott ütemezési időszakban figyelembe vehető. Tekintettel arra, hogy minden információ rendelkezésre áll, a tevékenység ütemezésére vagy késleltetésére irányuló döntést megalapozottan lehet meghozni. Ez különösen a megszakított tevékenységnél fontos, és ez az oka annak, hogyha a projektben sok tevékenységet szándékoznak megszakítani, ezt a megközelítést kell alkalmazni. [202, 213]

Ez az eljárás is jól megvalósítható ütemterveket hoz létre, de lényegesen nagyobb számítási igénnyel. Ennek oka a rangsorolandó tevékenységek élő listája. Ez a leglassabb művelet, amit a számítógép elvégez, és ráadásul a párhuzamos eljárásban minden időlépésnél, míg a soros eljárásban csak egyszer kerül végrehajtásra. [202, 213]

1.5.2 Erőforrás-allokáció (ERALL-módszer)

A logikai tervezés során (a **CPM-módszerrel**) olyan hálóttervet készítünk, amely technológiai szempontból megengedhető maximális párhuzamosítási lehetőségeket tünteti fel. A logikai tervezés eredménye maximálisan tömörített háló. Ennek megfelelően az időtervezésnél minimális időtartamokat kapjuk meg az egyes tevékenységekre nézve. Amennyiben

elkészítjük a hálóra vonatkozó erőforrásterhelési diagramot (lásd pl. 2.1.3-2 ábra), akkor láthatjuk, hogy egy adott erőforráskorlátot túlléphet. Az erőforrás-allokáció célja az, hogy (lehetőleg) az átfutási időt nem növelve, a kapacitáskorlátot nem túllépve a nem kritikus (alternatív) úton (utakon) lévő tevékenységeket a tartalékidejükön belül mozgassuk el úgy, hogy az erőforrásterhelési diagram a kapacitáskorlátot minden pontjában kielégítse. Ezt egy simítási eljárással valósíthatjuk meg, mely egy heurisztikus eljárás. Ez a módszer, ha létezik a feladatnak egy megengedett megoldása, akkor megtalálja. Az eljárás először megpróbálja úgy beütemezni a tevékenységeket, hogy a kritikus út ne növekedjen. Ha ez sikerül, akkor ezt a továbbiakban nevezzük **nem kritikus megoldásnak**. Ha ilyen megengedett megoldás nem létezik, akkor a módszer megpróbálja úgy beütemezni a tevékenységeket, hogy a kritikus út minél kevésbé növekedjen. Ha egy tevékenység erőforrásigénye nagyobb, mint az erőforráskorlát, akkor biztosan nincs megengedett megoldás. [288-289]

Miért csak megengedett megoldást talál ez a módszer? Azért, mert a nem kritikus úton lévő tevékenységeket nem egy adott célfüggvénynek megfelelően (pl. a lehető legkorábbi időpontra) ütemezi be. Az optimális megoldás az lenne, hogy ha a tevékenységeket úgy ütemeznénk be, hogy ezeket a célfüggvényeket figyelembe vesszük, de úgy, hogy az erőforráskorlátot egyetlen időpillanatban se lépjük túl.

1.6 Optimumkeresési eljárások

Dinamikus, Branch & Bound és lineáris programozási [251] technikákat alkalmaztak sikeresen 200 tevékenység alatti, néhány erőforrással rendelkező hálókra vonatkozó optimális megoldások kialakításához. Ennek ellenére az ilyen projektek csekélyek azokhoz viszonyítva, amelyek 5000 tevékenységgel rendelkeznek és 10-20 különböző erőforrást alkalmaznak. [26, 41, 61, 85, 118, 202, 319, 363, 394, 395]

A modern rendszerekben természetesen nem követelmény, hogy az erőforrásokat a tevékenység teljes időtartama alatt használják, vagy egy időben kizárólag egyet használjanak fel egy tevékenységre, vagy a tevékenységet ne osszák meg. Végül nem szabad elfelejteni, hogy a számítási eljárások által felhasznált adatok nem pontosak – mivel becsléseken alapulnak – és az esetleges hibák érvénytelenítik az optimumot. [7, 202]

1.6.1 Kiegyenlítés

Az erőforrás/idő korlátozottság problémájának megoldását az elsők között célozta meg a kiegyenlítési technika. Ez legegyszerűbb formájában azt jelenti, hogy valamilyen eljárással

előállított ütemtervnel megpróbálják kiegyenlíteni az erőforrásigények „csúcsait” és „völgyeit” – néhány tevékenység (alternatív utakon lévő tevékenységek) újraütemezésével. A kezdési ütemterv sok esetben a legkorábbi kezdési aggregáció, de lehet egy nagyon kifinomult allokációs eljárás által kialakított összetett ütemterv is. A kiegyenlítés egyszerűen hangzik, de a gyakorlatban nem az. Ha egynél több projekt létezik a szervezetben, akkor az eltérő erőforrások közötti kölcsönhatás rendkívül bonyolult helyzetet teremt. Továbbá – mivel az „egyensúly” jelentésének nincs általánosan elfogadott definíciója – rendkívül nehéz meghatározni a „kiegyenlítési” eljárás leállási pontját. Sok esetben az a gyakorlat, hogy a rendszer számára kijelölnek egy lefutási időt, és az idő végén megszerzett eredményt elfogadják. Sajnos a kereskedelmi szoftverek közül néhány helytelenül használja a kiegyenlítés kifejezést, ugyanis szerintük az allokációnak a rendelkezésre álló erőforrással szembeni egyensúlyát jelenti. Ugyanígy vannak olyan szoftverek is, amelyek szerint a kiegyenlítések tulajdonképpen allokálások. [202, 272]

Ezek az algoritmusok egy egészváltozós (IP) [148] vagy egy egészváltozóval kevert lineáris programozási feladatra (MILP) vezetik vissza a problémát. [242]

1.6.2 Heurisztikus és algoritmikus eljárások összehasonlítása

Erőforrás-tervezésre leggyakrabban a mai napig heurisztikus eljárásokat használnak. Ennek az oka, hogy bizonyos (10 000) tevékenységszám és néhány (5-10) erőforrástípus esetén a probléma kezelhetetlenné válik. [7, 59, 170, 202] A mai számítástechnikai teljesítmény ellenére is gyakran az optimális megoldás megtalálása reménytelen feladat. Az optimumkereső algoritmusok másik nagy hiányossága, hogy a közbenső megoldásai általában nem megengedettek. Ha tehát a feladat annyira bonyolult, hogy valószínűsíthető, hogy az adott erőforrás-tervezési problémát nem tudjuk meghatározott időn belül megoldani, akkor nem tudjuk megállítani a keresést, és az eddigi legjobb megoldást elfogadni, mert az sem garantált, hogy ez a megoldás egyáltalán megvalósítható (megengedett erőforrás-allokáció) lesz. [7]

Az erőforrás-tervezés során használt evolúciós eljárások kiinduló megoldásai ugyan lehetnek megengedettek, de itt sem garantált, hogy a közbenső lépések megengedettek lesznek. [7]

A következő táblázatban heurisztikus és algoritmikus eljárások összehasonlítását láthatjuk futási idő szempontjából.

Tevékenységek száma:=50

Algoritmus	Futási idő (sec)
Branch and Bound	1920,7
Dyn. Programming	0,0026

Tevékenységek száma:=1000

Algoritmus	Futási idő (msec)	Célfüggvény optimális megoldástól való eltérése
Dyn. Programming	9000	0,00 %
Peak Heuristic	< 1	123,81 %
Merge Heuristic	2	4,79 %
Split Heuristic	10	25,93 %
LP Heuristic	452	22,34 %
Merge and Split	5	3,85 %
LP + Merge and Split	452	1,77 %

1.6.2-1 táblázat: algoritmikus és heurisztikus módszerek átlagos futási ideje

A fenti táblázatból is láthatjuk, hogy a heurisztikus eljárások nagyon gyorsan viszonylag jó megoldást adnak. Ha a feladat során más a célfüggvény (pl. egyenletes erőforrás-kihasználás, lehető legkorábbi kezdés stb.), akkor ezekre más és más heurisztikus eljárásokat kell alkalmazni. [7, 98, 118, 165, 182, 315, 394-395]

Egy *erőforrás-allokációs probléma (adott célfüggvényre nézve) optimális megoldása* egy olyan megengedett ütemterv lenne, amely az adott célt legjobban kielégíti (vagy másképpen a célfüggvény értéke ebben az esetben minimális/maximális).

Az optimumkereső eljárások nagyságrendekkel lassabbak. Gyakorlatban kétfajta eljárás terjedt el ezek közül, az egyik az ún. Branch and Bound eljárás, a másik az ún. dinamikus programozás. [7, 314]

1.6.2.1 A Branch and Bound algoritmus

A kombinatorikus optimalizálásban illetve a mesterséges intelligenciában igen elterjedt módszer a korlátozás és szétválasztás módszere. A legtöbbször olyan NP-nehéz optimalizálási feladatok megoldásánál alkalmazzák, ahol a nem heurisztikus eljárásokkal történő megoldás meghaladná a mai számítási kapacitásokat. [65, 144, 179, 242, 306]

Az optimális erőforrás-tervezés is NP-nehéz feladat, ami azt jelenti, hogy nagyszámú erőforrás esetén lehetetlen belátható időn belül eredményhez jutni, pedig a feladat igazából nagyszámú erőforrás esetén válik izgalmassá és gyakorlati haszna is ekkor van. Mára már számos könyv és publikáció foglalkozik az eljárással, hol kiterjesztve azt a legáltalánosabb esetekre [242, 276], hol pedig egy-egy speciális problémát taglalva [85, 168, 181, 368]. A feladattípusok sokrétűsége ellenére azonban mindig jellemző, hogy valamiféle globális

minimum vagy maximum értéket keresünk, illetve az, hogy a feladat nagyságának növelésével jelentősen (sokszor exponenciálisan) nő a kiértékelendő részproblémák száma. Az, hogy az eljárás ilyen széles körben alkalmazható, az annak köszönhető, hogy ez csupán egy kereteljárás, melyet mindig az aktuális problémához kell igazítani. A kereteljárást úgy értelmezhetjük, hogy adott több részeljárás, vagy más néven szabályok, melyeknek csupán általános működését, valamint sorrendjét adja meg a B&B-algoritmus. Azt viszont, hogy az egyes részeljárásokat pontosan hogyan alkalmazzuk, azt már az adott feladat szabja meg. [242, 276]

A B&B-algoritmus formális leírása

Ahogy az előző részben láthattuk, a B&B-algoritmus leírása, különféle absztrakciós szinteken, már számos tudományos munkában megjelent. Mielőtt az eljárást áttekintenénk, figyelembe kell vennünk, hogy célunk csupán a megvalósítandó rendszer működésének megértése, illetve elméleti háttérének megalapozása, és nem az eljárás teljes részletekbe menő matematikai leírása, vagy például helyességének vizsgálata. Az elosztott rendszerünk egy általános keretet biztosít a már meglévő, speciális feladatokra optimalizált B&B-eljárások számára, így ennek az absztrakciós szintnek megfelelően elegendő a kereteljárással és annak komponenseivel megismerkedni. A jelen fejezet célja tehát az általános keretalgoritmus megismerése és formális leírása. Más, matematikailag részletesebb leírást találhatunk például Imreh Balázs „Kombinatorikus optimalizálás” című könyvében. Számunkra fontosabb az eljárás részeit képező szabályok kiemelése. [242, 276]

Az algoritmusban alkalmazott szabályok és szerepük

Az egyszerűség kedvéért a továbbiakban feltételezzük, hogy minden tekintett optimalizálási probléma minimalizálási feladat, azaz a lehetséges megoldások közül a minimális értékűt szeretnénk kiválasztani. A B&B algoritmus négy alapvető szabállyal jellemezhető, melyek a következők: [242, 276]

1. szétválasztási szabály (branching rule): ez a szabály határozza meg, hogy egy adott részproblémát mi alapján választunk szét kettő vagy több részproblémára.

2. korlátozási szabály (bounding rule): azt határozza meg, hogy miként számolható ki egy adott részprobléma optimális megoldásának alsó korlátja.

3. kiválasztási szabály (selection rule): ez a szabály definiálja, hogy miként válasszuk ki a következő szétválasztandó részproblémát a még feldolgozásra váró részproblémák közül.

4. kizárás szabály (elimination rule): ez a szabály adja meg, hogy mi alapján zárhatjuk ki azokat a részproblémákat, amelyek megoldásai nem vezethetnek a kiindulási probléma egy optimális megoldásához. Legyen P_0 a megoldandó minimalizálási probléma. A szétválasztási szabály ismételt alkalmazásával az eredeti P_0 problémának egyre finomodó felbontását hozzuk létre. Az így kapott új részproblémák egy $B=(P,A)$ -val jelölt fa struktúrába rendezhetők, ahol P a csúcsok, A pedig az élek halmaza. Ezt a fát *keresőfának* nevezzük. A keresőfa gyökere a P_0 kiindulási probléma. Ha egy P_j részprobléma a P_i részprobléma dekompozíciójából jött létre, akkor $(P_i, P_j) \in A$. Egy csúcs szintje a fában megegyezik a gyökértől az adott csúcsig vezető élek számával. A gyökér a 0. szinten helyezkedik el. Legyen továbbá $f(P_i)$ a P_i részprobléma optimális megoldása, valamint ha P_i részproblémát $P_1 \dots P_k$ részproblémákra bontjuk szét, akkor az f függvényre teljesül, hogy:

$$f(P_i) = \min_{j=1, \dots, k} \{f(P_j)\}. \quad (1.6.3.1-1)$$

Ezek után legyen $g(P_i)$ a P_i részproblémához tartozó alsó korlát, melyet a korlátozási szabály alapján számítunk ki, T pedig azoknak a részproblémáknak a halmaza, amelyek már nem bonthatók további részproblémákra, azaz a keresőfa leveleit képviselik. A g függvény a következő tulajdonságokkal rendelkezik:

$$g(P_i) \leq f(P_i), \text{ bármely } P_i \in P \text{ esetén}; \quad (1.6.3.1-2)$$

$$g(P_i) = f(P_i), \text{ bármely } P_i \in T \text{ esetén}; \quad (1.6.3.1-3)$$

$$g(P_i) \leq g(P_j), \text{ bármely } (P_i, P_j) \in A \text{ esetén}. \quad (1.6.3.1-4)$$

A tulajdonságok fentről lefelé a következőket jelentik: g egy alsó becslése f -nek, g pontos becslés, ha P_i dekompozíció nélkül megoldható, valamint az alsó korlátok monoton nem csökkenőek. A $B=(P,A)$ keresőfa adott P_i csúcsaihoz *címkeként* hozzárendeljük a $g(P_i)$ értéket. Ezeknek a címkéknek a továbbiakban fontos szerepe lesz abban, hogy eldöntsük, a keresőfa egy adott ágát lezárhatjuk-e, vagy tovább folytassuk a keresést. A keresés, azaz a

keresőfában található további felbontásra váró részproblémák feldolgozási sorrendje különféle stratégiák szerint történhet.

Ilyen stratégiák, vagy más néven *heurisztikus keresések* lehetnek például a mélységi, szélességi vagy legjobb értéken alapuló keresések. A keresés sorrendjét a kiválasztási szabály határozza meg, és jelentősen befolyásolhatja az algoritmus hatékonyságát. A kizárási szabályban háromféle tesztet alkalmazhatunk, hogy megállapítsuk, az adott részproblémát kizárhatjuk-e az optimális megoldás kereséséből. [242, 276]

Az első ilyen teszt azt vizsgálja, hogy az adott részproblémának lehetséges-e olyan megoldása, amely kielégíti a kezdeti feltételeket (*feasibility test*). Azt a megoldást, amely kielégíti a kezdeti feltételeket, lehetséges megoldásnak (*feasible solution*) nevezzük. [242, 276]

A második teszt azt vizsgálja, hogy az adott részproblémához címkeként hozzárendelt alsó korlát érték nagyobb, illetve egyenlő-e az eddig ismert legjobb lehetséges megoldás értékével (*bounding test*). Azaz P_i részprobléma esetén $g(P_i) \geq f^*(P_0)$ feltétel teljesül-e, ahol $f^*(P_0)$ az eddig ismert legjobb lehetséges megoldás, angol nevén: *incumbent*. Ugyanis, ha ez a feltétel teljesül, akkor a (1.6.3.1-4) és (1.6.3.1-2) egyenlőtlenségek alapján a P_i ágat a keresőfában lezárhatjuk, hiszen P_i bármely P_j gyermekére teljesülnek a $f(P_j) \geq g(P_j) \geq g(P_i) \geq f^*(P_0)$ egyenlőtlenségek. Más szóval ez azt jelenti, hogy P_i bármely P_j részproblémájának megoldása legfeljebb olyan jó lehet, mint az eddigi ismert legjobb megoldás, tehát felesleges a P_i részproblémát tovább bontani. [242, 276]

A harmadik teszt, a dominancia teszt (*dominance test*) ugyanazon a megfontoláson alapszik, mint az előző alsó korlát teszt, azzal a különbséggel, hogy itt a vizsgálandó P_i részproblémát nem az eddigi ismert legjobb lehetséges megoldáshoz hasonlítjuk, hanem minden korábban kiértékelt részproblémához. Ha a korábbi P_j részproblémák közül akár egy olyat is találunk, amelyre fennáll, hogy $f(P_j) \leq g(P_i)$, akkor azt mondjuk, hogy P_j domináns P_i -vel szemben, azaz P_i -nek nem lehet jobb megoldása, mint P_j -nek, tehát ezt az ágat szintén lezárhatjuk. [242, 276]

1.6.2.2 Dinamikus programozás

A dinamikus programozást rendszerint valamilyen numerikus paraméterektől függő érték optimumának meghatározására használjuk. A lényeg a következő: az optimális megoldást optimális részmegoldásokból állítjuk elő. Az optimális részmegoldásokat egy táblázatban

tároljuk, s a feladat ennek a táblázatnak szisztematikus feltöltésével oldható meg. A táblázat elemeit rendszerint egy rekurzív összefüggéssel határozhatjuk meg. [26, 101, 238, 242, 276]

A dinamikus programozás alapötlete, hogy a megoldandó problémát egy nagyobb problémacsomagba ágyazza be, amely problémacsomag alkalmas sorrendben könnyen „felgöngyölíthető”. Minden egyes probléma egyszerűvé válik a korábban megoldott más problémák eredményének ismeretében. [26, 101, 238, 242, 276]

A két módszert nagyon jól lehet ötvözni, előnyeiket kihasználni. További gyorsítási lehetőség, ha az egyes részproblémákat elosztjuk a különböző számítógépek között, és a részfeladatok kiértékelését külön számítógéppel végezzük, majd a kiértékelés eredményeit visszaküldjük a központi számítógéphez, amely újból kiosztja a feladatokat a többi munkaállomásra. [26, 101, 238, 242, 276]

A párhuzamos B&B algoritmus

A párhuzamos B&B algoritmussal foglalkozó művek [69, 80, 237, 258] száma szinte ugyanolyan bőséges, mint az általános algoritmussal foglalkozók száma.

Mielőtt azonban belekezdenék a párhuzamosítás lehetőségeinek tárgyalásába, fel kell tenni a kérdést, hogy valójában miért is kell egy hatékonyan működő eljárást párhuzamosítani? Ahogy azt az előző fejezet bevezetésében már említettem, a B&B-algoritmust nagyon gyakran NP-nehez problémák megoldásánál alkalmazzák. Ezeknél a problémáknál a feladat nagyságának növelésével drasztikusan nő a megoldási idő, és ezzel együtt a számítás erőforrásigénye is (pl. memória). A párhuzamosítás egyik előnye, hogy a feladat felosztásával és azok egyidejű végrehajtásával nagymértékben csökkenthető a feladat megoldásához szükséges idő, illetve ha a megközelítő B&B eljárásokat nézzük, akkor ugyanannyi idő alatt sokkal pontosabb eredményhez juthatunk. Másrészt gyorsan a szekvenciális számítógépek fizikai korlátjaiba ütközhetünk, például a processzor teljesítménye, vagy a bővíthető memóriahelyek száma erősen korlátozott. [26, 101, 242, 276]

Ezzel szemben, ha több számítógépet kapcsolunk össze, az erőforrások tetszőleges mértékben növelhetőek, a teljesítmény megsokszorozható.

Anomáliák

A párhuzamosítás előnyei ellenére a gyakorlatban igen nehezen valósítható meg, illetve csak nagyon speciális esetekben mondható el, hogy a processzorok (vagy a rendszerbe kapcsolt

számítógépek) számának lineáris növelésével a teljesítmény is lineárisan növekedjen. Előfordulhat, hogy a párhuzamos eljárásnak több időre van szüksége, mint a megfelelő szekvenciális (soros) algoritmusnak, ezt *lassulási anomáliának* (deceleration anomaly) nevezzük. Jó esetben azonban elérhető, hogy a párhuzamos algoritmussal szuper-lineáris vagy nagyobb gyorsulás érhető el a szekvenciálissal szemben, ezt *gyorsulási anomáliának* (speedup anomaly) nevezzük. A gyakorlati tapasztalatok azt mutatják [80], hogy leggyakrabban az úgynevezett *hátrányos anomália* (detrimental anomaly) érvényesül, azaz a végrehajtáshoz szükséges idő valahol a szekvenciális és a szuper-lineáris végrehajtási idő között helyezkedik el.

1.7 Az ütemezés, erőforrásallokáció alkalmazási területei

Ütemezést, erőforrástervezést nem csak a projektmenedzsment és a logisztika területén alkalmaznak, hanem pl. információs hálózatok optimális terhelésénél, forgalomirányítás, illetve kémiai folyamatok irányításában is. [5, 30, 52-54, 58, 138, 141, 145, 198, 330, 337, 339] Következőkben elsősorban a projektmenedzsment területén alkalmazható módszerek alkalmazási lehetőségeit mutatom be.

A legtöbb vállalat kevésbé használja ki a hálós, vagy más tervezési technikák adta lehetőségeket. Ennek oka számos problémára vezethető vissza.

1. Sokszor csak azzal van tisztában a szervezet, hogy mit szeretne létrehozni, megvalósítani (vagy sokszor ezzel sem). Ennek pontos mikéntjét, a tevékenységek, folyamatok meghatározását majd a megvalósítás során határozza meg részletesen. Ennek hátránya, hogy nagyon nehezen biztosítható, hogy az adott projekt a meghatározott időn belül befejeződjön.
2. Néhány vállalat meghatározza részletesen az elvégzendő tevékenységeket. Ezek között rákövetkezési relációt is felállítanak. Ennek ellenére úgy gondolják, hogy a megvalósítás közben a meghatározott tevékenységadatok annyit változnak, hogy ez teljesen megghiúsítja a terv alkalmazhatóságát. Ezért sok esetben csak Gantt-diagramokat alkalmaznak.

A csúszások a legtöbb esetben azért következnek be, mert

- a tevékenységek időtartamát a tervezés során fix időként és nem valószínűségi változóként kezelik, ezáltal az időtartamok bizonytalanságát a tervezés során nem veszik figyelembe;
- nem veszik figyelembe tervezéskor a tevékenységek erőforrás-szükségletét.

3. Ha egy vállalat erőforrás-allokációt is alkalmaz, akkor számára legtöbb esetben elegendő egy valamilyen megengedett megoldás megtalálása.

Látható, hogy az alkalmazott módszerek szervezési kérdéseket is felvetnek, hiszen a pontos ütemezéshez, illetve az erőforrástervezéshez meg kell becsülni a tevékenységek időtartamait, rákövetkezési relációit, valamint a tevékenységek erőforrás-szükségleteit is.

1.7.1 A módszerek alkalmazási feltételei, lehetőségei

Egy projektszervezet dönthet úgy, hogy nem alkalmazza az ütemezés és erőforrás-allokációs módszer lehetőségeit. Ebben az esetben azonban – főleg nagyobb projektek esetén – nehezen biztosítható, hogy a projekt meghatározott időn belül befejeződjön. Kisebb projektek esetén gyakran eltekintenek az ütemezés és erőforrás-allokációs módszer lehetőségeitől, mert azt egy tapasztalt projektvezető sok éves tapasztalatából adódóan átlátja, és az egyes váratlan eseményeket kezelni tudja.

Ha egy vállalat csak ütemezi a projektben elvégzendő tevékenységeit, akkor is meg kell becsülnie az egyes tevékenységek várható időtartamát. Két lehetősége van: vagy fix lefutási időként kezeli az egyes tevékenységek időtartamait, és a korrekciókat később a projekt futása közben végzi majd el a tervben, vagy eleve valószínűségi változóként kezeli a tevékenységek időtartamait, ezzel bizonyos határokon belül kezelni tudja a projekt átfutási idejének bizonytalanságát. Tervezni tudja, hogy adott valószínűségi szint mellett várhatóan mikor fog befejeződni a projekt.

Az ütemezés feltétele a logikai összerendelés, illetve a logikai háló megléte. Ennek ellenére a gyakorlatban használt projektmenedzsment szoftverek nem követelik meg a logikai háló elkészítését. Egy tevékenység kezdési, illetve befejezési idejét manuálisan is meg lehet adni. Ez a látszólagos kényelmi funkció gyakran nagyon csalóka lehet. Hiszen a tevékenységek csúszását már nagyon nehezen lehet ezzel a módszerrel kezelni, hiszen ha egy tevékenységhez nincsen hozzárendelve az őt követő tevékenység, akkor egy esetleges csúszás esetén a logikai sorrend akár teljesen fel is borulhat.

A csúszások egyik oka lehet, hogy nem megfelelően használja fel a vállalat a rendelkezésére álló erőforrásokat, így fontos szempont – főleg nagyobb beruházások esetén – az erőforrások helyes felhasználása, az egyes tevékenységekhez kapcsolódó erőforrás-igények összevetése a rendelkezésre álló erőforrás-kerettel. Ahol sok párhuzamos tevékenység folyik egy időben, és ezek elvégzéséhez pl. több munkaerőre van szükség, mint amennyi a cég

rendelkezésére áll, ott vagy más vállalat segítségét kell igénybe vennünk (pl. alvállalkozók bevonása), ha időben el akarjuk végezni a tevékenységeket (időkorlátos erőforrás-allokáció), vagy ha erre nincs mód, akkor időben későbbre kell beütemezni a tevékenységeket (erőforráskorlátos erőforrás-allokáció).

Ahogy az a korábbi fejezetekben láthattuk, sok esetben a beruházó cégek megelégszenek egy ún. megengedett megoldás megtalálásával, vagyis egy olyan ütemtervvel, ahol a megvalósítás során a rendelkezésre álló erőforrásainkat vagy időkeretünket nem lépjük túl. Egy ilyen megengedett megoldás néhány tevékenység esetén kézzel, több tevékenység esetén számítástechnikai szoftverekkel (pl. Microsoft Project, CA-SuperProject, Primavera stb.) viszonylag könnyen megtalálható. Egy ilyen ún. megengedett erőforrás-allokáció esetén azonban sokszor számos probléma merülhet fel. [255, 399-401]

Ha a tevékenységeket nem a lehető legkorábbi időpontra ütemezzük be, akkor előfordulhat, hogy a kritikus úton lévő tevékenységeken kívül alternatív úton lévő tevékenységek csúszása is a projekt átfutási idejének csúszását eredményezheti.

Néhány esetben fontos lehet, hogy az erőforrásainkat viszonylag egyenletesen használjuk fel. A megengedett megoldások a legtöbb esetben ezekkel az előnyös tulajdonságokkal nem rendelkeznek. Ezért a heurisztikus eljárások is igyekeznek egy adott célfüggvényre nézve optimális megoldást minél jobban megközelíteni. [182, 316-318]

1.8 Számítógépes projekttervező szoftverek

Nagy számítógépes programcsomagok (kezdeti alkalmazás)

IBM család:

Az International Business Machines Corporation nagy múlttal és tapasztalatokkal rendelkező számítástechnikai cég, mely a hardvereken kívül komoly programrendszerekkel és alkalmazási tapasztalatokkal is rendelkezik. [25, 182]

A különböző operációs rendszerek lehetőségeinek megfelelően több programcsomagot fejlesztettek ki, melyek közül hazánkban a PROJACS (Project Analysis and Control System) volt a legelterjedtebb.

A rendszer több, egymáshoz kapcsolódó modulból épül fel.

Ezek:

- főprogram (Main Processor)

- erőforrás-allokáló program (Resource Allocation Processor)
- költség-kiértékelő program (Cost Evaluation Processor);
- hálólőkészítő program (Network Preparation Processor).

Egyidejűleg 500 hálót tud kezelni; egy háló max. 100 alhálóra bontható, ezen belüli kapcsolatok száma max. 1000 lehet. Az alhálókat grafikusán CPM/PERT vagy PDM eljárás szerint ábrázolhatjuk. A tevékenységek időtervezése az MPM háló számításai szerint történik. **[25, 182]**

A PROJACS 6-féle kimutatást készít:

- tevékenységorientáltság, és
- aggregált tevékenység szerint,
- erőforrás-felhasználási,
- költség-,
- display-kimutatást,
- hálókirajzolást.

ICL család:

Az ICL cég két, hazánkban is üzemelt gépcsaládjára az 1900-as és a System 4 típusú gépekre egyaránt jól használható hálótervezési programcsomagot készített. A két programcsomag nagy méretű hálók feldolgozására volt alkalmas, segítségével idő-, erőforrás-, és költségtervezést végezhettek. **[25, 182]**

A két programcsomagban a hálókat modulszerűen, alhálónként állíthattuk össze, és típusháló könyvtárakat is kezelni tudtunk. Az időtervezést **CPM-módszerrel** lehetett végezni, de használható olyan tevékenység is, melyeknél hármassal időbecsléssel dolgoznak, s ezekre a tevékenységekre a programok elvégzik a szokásos PERT számításokat is. Az alháló struktúra teszi lehetővé, hogy a számítógép korlátozott belső memóriakapacitása ellenére szinte korlátlan méretű hálókat lehetett feldolgozni. **[25, 182]**

Egy hálóban max. 500 alháló lehet, ezek pedig max. 1500 tevékenységet tartalmazhatnak.

SINETIK programcsomagok

A programcsomagot a Siemens cég fejlesztette ki Siemens 4004 számítógépre. A program MPM és CPM típusú hálók feldolgozására volt alkalmas.

A hálón belüli tevékenységek száma max. 6000, és a kapcsolatok száma is max. 6000 lehet. A lehetséges alhálók száma tetszőleges. A háló kirajzolására alkalmas modullal a program nem rendelkezik. [25, 182]

A mindinkább előtérbe kerülő igények sürgetésére fejlesztették ki a mikroszámítógépekre alkalmazható hálótervezési programrendszereket, melyek közül elsőként a SZENZOR (korábban SYSTEM) Szervezési Vállalat jelentkezett a HSZR-mikro elnevezésű programjával. Ez a rendszer képes megoldani mindazon feladatokat, amelyeket korábban a hasonló, közép- és nagy-számítógépes hálótervezési programcsomagok. Jelenleg hazánkban – saját fejlesztésű programként – ez az egyetlen mikrogépre alkalmazható hálótervezési programcsomag. Ismertek, és gyakorlatban használatosak nyugati cégek által kifejlesztett olyan szoftverek, melyek többet tudnak hazai testvérüknél, a gyakorlati példa bemutatását azonban célszerűnek tartjuk a hazai változat lehetőségeit kihasználva végigkísérni a következő fejezetben. [25, 182]

Napjainkban a legtöbb kereskedelmi forgalomban kapható projektmenedzsment-szoftver az ún. tevékenység-csomópontú ábrázolástechnikát használja.

A mai korszerű számítógépes programok készítői igyekeznek a különböző módszerek elemeit ötvözni. Az elméleti kutatás és az eredményeknek a gyakorlatba történő átültetése a mai napig sem állt meg. Újabb és újabb modellezési megoldások és szoftverek jelennek meg, illetve az elméleti eredmények beépülnek a már elterjedt projektmenedzsment-szoftverek újabb verzióiba. [25, 182, 207, 208]

A gyakorlatban alkalmazott projekttervezési szoftverek**Microsoft Project (2000, XP, 2003)**

Az MS Project 2000/XP/2003 az MPM-modell egyszerűsített változatával dolgozik. Fő jellegzetessége a tevékenységek egymáshoz viszonyított helyzetének egyszerű kezelését lehetővé tevő négyféle kapcsolati típus és a kapcsolati idők alkalmazása. Az ütemezési

eredmények fő megjelenítési formája a vonalas ütemterv (Gantt Chart), az erőforrásterhelési és költségfelmerülési görbék, valamint ezek táblázatos formái. A Gantt Chart-on az ütemvonalak mellett a tevékenységek közötti kapcsolatokat is megjeleníthetjük, ezért ez az ábrázolás több, mint egy egyszerű vonalas ütemterv. A hálótér-modellel *Network Diagram* elnevezés alatt jeleníti meg a rendszer. (Az MS Project 98-ban még *PERT Chart* volt a hasonló funkciójú nézet neve.) A tevékenység-időtartamok felvitelénél lehetőségünk van hármass időbecslés megadására is. A program a PERT Analysis funkcióban kiszámítja a tevékenység-időtartamok és a projekt átfutási idejének pesszimista, optimista és várható értékét, de az értékek szórását nem. [399]

A Microsoft Project Standard minden eddiginél egyszerűbbé teszi az ütemezések és erőforrások kezelését, a projekt állapotának közlését és a projekt adatainak kimutatását. A jobb eszközök és a beépített támogatás révén a munka hamar megkezdhető, még a projektvezetéssel most ismerkedők is gyorsan megbarátkozhatnak a program kezelésével. A Microsoft Project projektvezetési szolgáltatásainak köszönhetően minden információ a rendelkezésére áll, így a projektek időben elkészülnek, és nem lépik túl a költségvetést. [399]

Az MS Project specifikációja

Taszkok (tevékenységek) száma projektenként	egy millió
Erőforrások száma projektenként	egy millió
Az erőforrás mennyisége hozzárendelésenként	60.000.000 egység
Erőforrás szint változások száma	100
Taszk-kapcsolatok száma projektenként	korlátlan
Kapcsolatok száma két taszk között	1
Megelőző kapcsolatok száma taszkonként	korlátlan
Követő kapcsolatok száma taszkonként	korlátlan
Konszolidált (összevont) projektek száma	998
Megnyitott projektfájlok száma konsz. projektenként	998
Közös erőforráshalmazt használó projektfájlok száma	999
Nyitott ablakok száma	50
Bázisnaptárak száma	korlátlan
Kivételek száma naptáranként	1.400
A havi naptárban nyomtatható taszkok száma	4.000

1.	Irodalmi áttekintés, alapfogalmak
Fajlagos költségábrák száma erőforrásonként	5
Változások száma költségábránként	25
A költségmező max. értéke	999.999.999.999
A munkamennyiség max. értéke órában	1.666.666.667
Munka hozzárendelésnél	999.999.999 perc
Változó anyag hozzárendelésnél	999.999.999 egység
Fix anyag hozzárendelésnél	60.000.000 egység
A nyomtatási skála beállítási tartománya	10-től 500 %-ig
Legkorábbi dátum	1984. január 1.
Legkésőbbi dátum	2049. december 31.

Előnyök:

- Hatékonyabb projektvezetés: a dinamikus ütemezés révén egy ütemezés vagy egy erőforrás változása azonnal tükröződik a projekt ütemezésében. A projektútmutató testreszabásával a már bevált projektvezetési eljárásokat is alkalmazhatjuk a programból.
- Több információ áll rendelkezésre: a Gantt-diagramokban és a használati nézetekben három időskálát jelenít meg, az oszlopok és sorok végösszegét pedig ki lehet nyomtatni.
- Egyszerűen használható eszközök: a projekttervek módosításakor alternatív ütemezési lehetőségeket ajánlanak fel. A Microsoft Excel vagy Outlook programokban már meglévő feladatlisták egyszerűen alakíthatók Microsoft Project-tervekké.

Hátrányok, hiányosságok:

- Nem követeli meg a logikai összerendeléseket. Ez a látszólagos könnyedség a projekt aktualizálásakor sok problémát okozhat a csúszások nyomonkövetésénél.
- Az erőforrástervező komponense általában nem optimális erőforrásallokációt határoz meg. (A Microsoft Project által alkalmazott erőforrás-allokációs módszer a párhuzamos erőforrásallokációhoz hasonlatos heurisztikus módszer)
- Programrövidítésre, költségoptimalásra, költségoptimalis erőforrás-allokáció meghatározására (pl. CPM/COST, MPM/COST, PERT/COST) nincs lehetőség.
- A PERT-módszer alkalmazásánál az átfutási idő bizonytalanságát, illetve egy adott valószínűségi szinthez tartozó átfutási időt nem tud számolni.

CA-SuperProject

A felhasználói programok ma már elérhető árúak, és többnyire jó minőségűek. A skála az alapfokú, ingyenes programoktól a mesterséges intelligenciával felruházott Primavera-ig terjed. A kettő közti széles skálán egy jó átlagos program pl. a Computer Associates SuperProject-je, amellyel közös erőforrás-csomagból dolgozó több projekt is kezelhető egyszerre. Felhasználóbarát, viszonylag könnyen kezelhető program. Jól használható, hasznos segítőtárs egy beruházási területen dolgozó mérnöknek. [401]

A CA SuperProject (4.0) specifikációja

Taszkok száma projektenként	16.000
Erőforrások száma projektenként	korlátlan
Az erőforrás mennyisége hozzárendelésenként	korlátlan
Erőforrás szint változások száma	254
Taszk-kapcsolatok száma projektenként	korlátlan
Kapcsolatok száma két taszk között	1
Megelőző kapcsolatok száma taszkonként	korlátlan
Követő kapcsolatok száma taszkonként	korlátlan
Prioritási szintek	1000
Outline szintek száma projektenként	65.535
Fajlagos költségábrák száma erőforrásonként	5
Változások száma költségábránként	37
A költségmező max. értéke	999.999.999.999
A munkamennyiség max. értéke órában	1.666.666.667

Előnyök:

- Gyors, egyszerű használat.
- Széles funkcionalitás (idő-, költség- és erőforrás-tervezés).
- Több projekt egyidejű kezelése (erőforrás-megosztási lehetőségek a projektek között).
- Együttműködés más szoftverekkel (importálás, exportálás pl. Excel).

Hátrányok, hiányosságok:

- Nem követeli meg a logikai összerendeléseket.
- Az erőforrástervező komponense általában nem optimális erőforrásallokációt határoz meg.
- Programrövidítésre, költségoptimalásra, költségoptimalis erőforrás-allokáció meghatározására (pl. CPM/COST, MPM/COST, PERT/COST) nincs lehetőség.

Primavera Project Planner (P3)

A Primavera Project Planner egy mesterséges intelligenciával felruházott projektmenedzsment-szoftver, mely képes egyszerre több projekt között hatékonyan elosztani az erőforrásokat. Többfajta, egymással hierarchikusan összerendelhető erőforrást is kezelni tud. [401]

A Primavera specifikációja

Projektek száma	korlátlan
Taszkok száma projektenként	100.000
Erőforrások száma projektenként	korlátlan
Az erőforrás mennyisége hozzárendelésenként	korlátlan
Erőforrás szint változások száma	korlátlan
Taszk-kapcsolatok száma projektenként	korlátlan
Kapcsolatok száma két taszk között	10
Megelőző kapcsolatok száma taszkonként	korlátlan
Követő kapcsolatok száma taszkonként	korlátlan
Prioritási szintek	1000
Outline szintek száma projektenként	korlátlan
A költségmező max. értéke	999.999.999.999
A munkamennyiség max. értéke órában	1.666.666.667

Előnyök:

- Széles funkcionalitás (idő, költség és erőforrás-tervezés).

- Több projekt egyidejű kezelése (erőforrás-megosztási lehetőségek a projektek között).
- Együttműködés más szoftverekkel (importálás, exportálás pl. Excel).
- Hierarchikus erőforrások kezelése.
- A mesterséges intelligencia alkalmazásának köszönhetően az optimális megoldáshoz közeli megengedett megoldás szolgáltatása.

Hátrányok, hiányosságok:

- Nem követeli meg a logikai összerendeléseket.
- Az erőforrástervező komponense általában nem optimális erőforrásallokációt határoz meg.

1.9 Módszerek, szoftveres alkalmazások értékelése

A korábban alkalmazott megoldások két részre bonthatók. Vannak algoritmikus megoldások, ilyen például a kiegyenlítéses módszer, és vannak heurisztikus megoldások, melyekre tipikus példa az allokáció. A heurisztikus megoldások általában gyorsabbak, de nem garantálnak optimális megoldásokat, és ezek a módszerek más eredményhez vezethetnek bizonyos esetekben. (Például, ha egy tevékenység megszakíthatóságát megengedjük, akkor más eredményt kapunk soros, illetve párhuzamos allokáció esetén).

A kiegyenlítéses algoritmus nem megengedett megoldásból indul ki, így nem igaz rá, hogy minden lépése megengedett megoldást adna. Így, ha olyan feladatot kell megoldanunk, ahol nagyszámú tevékenységet kell optimalizálnunk, akkor nem biztosított, hogy meghatározott időn belül legalább egy megengedett megoldást kapjunk.

A következő táblázat az irodalmi hivatkozások egyfajta kategorizálását tartalmazza.

		Időtervezés, ütemezés	Költség/időtervezés	Erőforrás-tervezés	
				Időkorlátos	Erőforráskorlátos
Determinisztikus	Heurisztikus	[17]; [82]; [184]; [214]; [301]; [313]; [357]	[83]; [338]	[28]; [39]; [110-111]; [182]; [297]	[7]; [10-12]; [24]; [28]; [31-34]; [37]; [39]; [41-43]; [53]; [57-59]; [75]; [108]; [111]; [133]; [135]; [137-138]; [164]; [166]; [171]; [173]; [176]; [182]; [204]; [210]; [212]; [213-216]; [243-244]; [246]; [250]; [273-274]; [280]; [282]; [288-292]; [294-296]; [297]; [299]; [302-305]; [317]; [338]; [340-341]; [343]; [347]; [363-364]; [366]; [372-374]; [377]; [379]; [393-394]
	Algoritmikus	[16]; [51]; [71]; [82]; [126]; [159-160]; [165]; [174-175]; [184]; [193]; [199-200]; [205]; [217-218]; [245]; [265]; [312]; [357]; [390]	[8-9]; [18]; [29]; [88-91]; [93-94]; [96]; [98]; [100]; [119-125]; [127-128]; [139]; [156]; [167]; [194]; [197]; [199-200]; [203]; [239]; [274]; [315]; [320]; [335]; [351-352]; [356]; [383]; [389]	[39]; [111]; [182]; [297]	[5]; [7]; [19]; [39]; [44]; [61]; [69]; [74-75]; [79]; [81]; [85-87]; [97-99]; [111]; [113]; [118]; [149]; [169]; [170]; [176]; [182]; [183]; [189]; [198]; [219]; [264]; [272]; [297]; [299]; [314]; [318-319]; [340]; [345-346]; [354-355]; [362]; [365]; [367];
Sztochasztikus	Heurisztikus	[49-50]; [143]; [147]; [152]; [154]; [191]; [233-236]; [254]; [281]; [293]; [331]; [369]; [381]		[92]	[361]
	Algoritmikus	[1-2]; [14]; [27]; [47-48]; [64]; [68]; [72]; [76-78]; [115]; [117]; [130-131]; [134]; [136]; [140]; [150-151]; [153]; [155]; [162-163]; [168]; [172]; [185-186]; [188]; [192]; [195]; [199-201]; [209]; [231]; [240-241]; [245]; [247-249]; [252-253]; [260-261]; [263]; [265-266]; [271]; [275]; [278-279]; [283-285]; [300]; [311]; [324-327]; [329]; [332-334]; [342]; [348-350]; [353]; [358-360]; [370]; [385-387]; [391]	[15]; [35]; [60]; [103-107]		[368]

1.9-1 táblázat: irodalmi hivatkozások csoportosítása

Mivel elsősorban erőforráskorlátos allokációval foglalkoztam, így a hivatkozások elsősorban erről a területről származnak. Ütemezés és költségminimalizálás területéről szinte kizárólag algoritmikus módszerekkel találkozhatunk, hiszen itt a feladat komplexitása nem túl nagy, így az ütemezéshez csak akkor célszerű valamilyen heurisztikus módszert alkalmazni, amikor nagyon nagy projektek valós idejű ütemezését kell megvalósítani. Az erőforráskorlátos erőforrás-allokáció problémája már sokkal nehezebben oldható meg még nagyteljesítményű számítógépek segítségével is. Éppen ezért itt már nagyon sok heurisztikus, illetve evolúciós módszerrel találkozhatunk.

Sztochasztikus költség-, illetve erőforrástervezés körébe soroltam azokat a módszereket, melyeknél már legalább egy paraméter (pl. időtartam, költség-, erőforrásigény stb.) sztochasztikus. Ebben a témakörbe még nagyon sok megválaszolatlan kérdés van. A projektmenedzsment számára fontos, hatékony eszköztárat nyújtó módszert, mely egyaránt alkalmazható olyan esetekben, amikor a tevékenység időtartama, költségigénye, illetve az erőforrás-szükséglet előre meghatározható (determinisztikus), illetve becsülhető (sztochasztikus), valamint kezeli a szakaszonként konstans erőforráskorlátokat, a projekt lefutása során bekövetkezett tervadatoktól való eltérését, mindezekig nem publikáltak.

A disszertációm készítése során a fenti problémákra kerestem a válaszokat. Módszerem az erőforrás-korlátos erőforrás-allokációs probléma megoldására alkalmazható megújuló erőforrások optimális erőforráselosztására. Az általam kifejlesztett módszer is

algoritmikus jellegű, vagyis garantáltan optimális megoldást kapunk egy adott célfüggvényre. Megengedett megoldásból indul, amit bármely heurisztikus módszer szolgáltat. Célfüggvény lehet a lehető legkorábbi/legkésőbbi kezdés.

Amennyiben a projekt működése közben az erőforráskorlát, vagy a tevékenységek erőforrásigénye, időtartamai megváltoznak, a javasolt módszer segítségével meghatározható egy új termelési program a még futó, illetve a még el nem kezdett tevékenységekre. Az eljárás segítségével olyan projektre is meghatározható várható átfutási idő, költség- illetve erőforrásigény, ahol a tevékenységek időtartamait, költség- illetve erőforrásigényeit csak becsülni tudjuk.

Ezzel a módszerrel több párhuzamosan működő projektet optimalizálhatunk, illetve többféle erőforrás egyidejű kezelését valósíthatjuk meg. Kezeln tudjuk továbbá a fel nem használt erőforrásokat is.

Munkám során olyan módszert fejlesztettem ki, mely hatékonyan használható a projektmenedzsment területén. Célom az volt, hogy a projektmenedzserek számára egy hatékony eszközt biztosítsak a projektek tervezésére. A javasolt módszer mind a kisebb (50-150 tevékenység, néhány erőforrás), mind pedig nagyobb (5000-50000 tevékenység 10-20 erőforrástípus) esetén is alkalmazható. Minél több információval rendelkezünk a tervezés során, annál pontosabb és megbízhatóbb tervet készíthetünk. A bemutatandó eljárás a szakirodalomban eddig publikált módszerekkel szemben olyan esetekben is használható, amikor költség-, erőforrás- és időoptimalás együttes cél. A módszer alkalmazható olyan esetekben is, amikor a tevékenység időtartama, közvetlen költsége illetve az erőforrás-szükségletek adatai determinisztikusak illetve valószínűségi változók.

1.10 Hipotézisek

- H1. A menedzserek számára a különböző tevékenységek szervezését biztosító gyakorlati módszerek, eljárások, szoftverek nem alkalmasak valamennyi erőforrás-tervezési/költségtervezési probléma megoldására.**
- H2. A gyakorlatban használt algoritmikus módszerek nagyméretű projekteknél (több mint 10 000 tevékenység) nem képesek „belátható” időn belül legalább megengedett megoldást találni.**
- H3. A gyakorlatban használt módszerek, eljárások, szoftverek nem kezelik a költségek és erőforrásigények bizonytalanságát.**

2. Az optimális erőforrás-allokáció keresésének módszertani bemutatása

2.1 Erőforrás-allokáció, megengedett megoldásból optimális megoldás keresése (ERALL-OPT, OPT-RALL)

Annak igazolására, amit a hipotéziseimben megfogalmaztam, első lépésként egy olyan módszert mutatok be, mellyel egy erőforráskorlátos erőforrás-allokációs probléma megengedett megoldásából adott célfüggvényre (pl. lehető legkorábbi kezdésre) egy optimális megoldást keres. Az általam kifejlesztett algoritmus az erőforráskorlátos erőforrás-allokációs probléma megengedett megoldásából indul ki. Egy olyan optimális megoldást keres, ahol az erőforráskorlátot nem túllépve a felhasznált tartalékidőket minimálisra csökkentjük. A továbbiakban megújuló erőforrásokkal foglalkozom (pl. munkaerő, berendezés stb.).

Az erőforrás-allokáció során egy megengedett megoldáshoz jutunk. Bemutatom, hogy ha egy ilyen problémának létezik megengedett megoldása, akkor létezik az erőforráskorlátot kielégítő optimális megoldása is, amely véges lépésben elérhető. [220-221, 223, 226]

A bemutatandó módszernek magyarul **ERALL-OPT** (erőforrás-allokáció optimális megoldása), angolul **OPT-RALL** (Optimized Resource Allocation) nevet adtam.

2.1.1 A feladat megfogalmazása

Definíció: Egy ütemezést *maximálisan párhuzamosított megoldás*nak nevezünk, ha valamennyi a projektben/termelési programban szereplő tevékenység a legkorábbi kezdési idejére van beütemezve.

Definíció: Egy tevékenység tényleges kezdése és a legkorábbi kezdés közötti időt *felhasznált tartalékidő*nek nevezzük.

Megjegyzés: A felhasznált tartalékidő mindig egy nemnegatív egész vagy valós szám, hiszen a tevékenységeket a legkorábbi kezdési idejüknél korábbra nem lehet beütemezni.

Definíció: Legkésőbbi befejezés és a tevékenység tényleges befejezése közötti időt *rendelkezésre álló tartalékidő*nek nevezzük.

Definíció: Egy erőforrás-allokációs probléma *megengedett megoldásának* nevezünk egy olyan ütemtervet, amelynél a projekt végrehajtása során minden időpillanatban az összes erőforrásigény nem haladja meg az erőforráskorlátot.

Definíció: Az erőforrás-allokáció (egy adott célfüggvényre) *optimális megoldásának* nevezünk egy olyan megengedett megoldást, ahol a célfüggvény értéke a lehető legkisebb (legnagyobb).

Megjegyzés: Ilyen célfüggvény lehet pl. a megengedett megoldásokban elmozgatott tevékenységek felhasznált tartalékidőinek minimuma, vagy a tevékenységek felhasznált tartalékidőinek összegének minimuma stb.

Legyen például a célfüggvény a megengedett megoldás érdekében elmozgatott tevékenységek lehető legkorábbi kezdése. Itt csak azokat a tevékenységeket kell optimalizálni, amelyeket elmozdítottunk annak érdekében, hogy egy optimális megoldást kapjunk (ezeket a tevékenységeket fehérrel jelöltem). Ugyanis a többi esetben a tevékenységeket nem mozgattuk el a megengedett megoldás keresésénél. Vagy azért, mert a kritikus úton helyezkednek el (zölddel jelöltem), vagy az erőforráskorlátot nem sértették meg (sárgával jelöltem).

Felírható az összefüggés:

Legyen: $x_{(i,j)}$ az (i,j) tevékenység által felhasznált tartalékidő (amelyeket nem mozgattuk el, azoké nulla). Legyen $z_{(i,j)}$ a tevékenység legkorábbi kezdési ideje ($EST_{(i,j)}$). Legyen $\phi(t_i)$ a t_i időpillanatban lévő erőforrásigény nagysága (pl. a 2.1.3-3 ábra látható 7. időpillanatban ez 15) stb. Legyen $w_{(i,j)}$ az (i,j) tevékenység kezdési ideje a megengedett megoldásban.

Mivel a feladat a lehető legkorábbi kezdések meghatározása, az egyik legnehezebb probléma alsó, illetve felső korlátot adni a lehetséges kezdéseknek. A legkorábbi kezdés általában elérhetetlen alsó korlátnak bizonyul. Az elmúlt két évben publikáltak olyan módszereket, melyekkel pontosabb becslést adhatjuk a lehető legkorábbi, illetve legkésőbbi kezdésnek. Ezek a módszerek már figyelembe veszik a tevékenységek erőforrás-szükségleteit is.

$w_{(i,j)}$ és $z_{(i,j)}$ meghatározása után $w_{(i,j)} - z_{(i,j)}$ konstans, és kezdetben $w_{(i,j)} - z_{(i,j)} = x_{(i,j)}$. Legyen továbbá P azoknak a tevékenységeknek a halmaza, amelyeket elmozdítottunk, valamint a

korlátozás legyen c . (Általános esetben P halmaz legyen az optimalizálandó tevékenységek halmaza.) Ekkor igaz a következő:

$$x_{(i,j)} \leq w_{(i,j)} - z_{(i,j)}, \text{ ahol } x_{(i,j)}, w_{(i,j)}, z_{(i,j)} \in \mathbf{R}_0^+ \quad (2.1.1-1)$$

$$\phi(z_{(i,j)} + x_{(i,j)}) \leq c,$$

$$\text{ahol } c \in \mathbf{R}_0^+, \phi \in \mathbf{R}_0^+ \rightarrow \{r_1, r_2, \dots, r_n\}, r_1, r_2, \dots, r_n \in \mathbf{R}_0^+, n \in \mathbf{Z}^+ \quad (2.1.1-2)$$

$$\forall (i,j) \in Q, \text{ ahol } Q \in \wp(P) \setminus \emptyset, \forall (k,l) \in A \setminus Q \text{ esetén. Ha } (i,j) \text{ tevékenység rákövetkezési relációban áll } (k,l)\text{-l, akkor } z_{(i,j)} + x_{(i,j)} \geq z_{(k,l)} + d_{(k,l)}. \quad (2.1.1-3)$$

Feltételek teljesülése esetén a feladat minimalizálni P halmaz tevékenységeinek felhasznált tartalékidejét:

$$q = \min_{(i,j) \in P} x_{(i,j)} \quad (2.1.1-4)$$

Ugyanígy feladatként lehet megfogalmazni, ha a felhasznált tartalékidők összegének minimumát szeretnénk meghatározni. Ekkor a 2.1-4 egyenlet a következőképpen módosul.

$$q = \min \sum_{(i,j) \in P} x_{(i,j)} \quad (2.1.1-5)$$

Hasonlóan lehet a felírni a problémát, ha a cél a lehető legkésőbbi kezdés; ekkor a feladat a felhasznált tartalékidők maximalizálása. Korlátot a (lehető) legkésőbbi kezdés adja meg. Ilyen felső korlátot adó algoritmusokat szintén az utóbbi két évben publikált módszerek segítségével lehet találni. Ezen módszerek sajátossága, hogy általában pontosabb becslését lehet adni a lehető legkésőbbi/legkorábbi kezdéseknek. Ezek a módszerek figyelembe veszik ugyan az erőforráskorlátokat, de a szolgáltatott megoldás nem biztos, hogy megengedett.

A modellből látszik, hogy a feltételek teljesülését egyrészt a töréspontokban kell vizsgálni, másrészt a rákövetkezési relációkból következik, hogy egy tevékenységet maximum meddig lehet visszatolni. E két idő határozza meg, hogy mikor kell döntenünk az egy lépésben optimalizálandó tevékenységekről (Q halmaz).

2.1.2 Az algoritmus leírása

Kihasználjuk a következőket a módszer során:

1. Csak a P halmaz elemeit optimalizáljuk. (Ez lehet az alternatív úton lévő tevékenységek halmaza, lehet továbbá a megengedett megoldásban későbbi időpontra ütemezett („elmozgatott”) tevékenységek halmaza).
2. Felhasználjuk, hogy ϕ -függvény minden olyan helyen, ahol nincs törés konstans, bármelyik tevékenységet is változtatva a módszer a megengedettségen nem változtat, ha figyelembe vesszük a rákövetkezési relációkat is.
3. A 2. pont szerint tehát egy „bizonyos ideig” a (2.1.1-2, 2.1.1-3) feltétel elhagyható. Ekkor viszont egy lineáris problémához (LP) jutunk. Tehát arra az intervallumra a választott kiválasztást alkalmazva a megengedettség nem sérül.

Definíció: Egy (i,j) tevékenységre vonatkozó töréspont értéke megmutatja, hogy az (i,j) tevékenységet elvéve az összes erőforrásra vonatkozó erőforrásigény függvény a tevékenység kezdése pillanatában hogyan változik. Ha az erőforrásigény csökken (nő) a tevékenység kezdetekor, akkor a töréspont ebben a pillanatban pozitív (negatív).

Példa: A 2.1.3-4 ábrán (6,9) illetve (7,8)-as tevékenységre vonatkozó töréspont nulla, míg ugyanez a 2.1.3-3 ábra esetében (6,9)-re 2 illetve (7,8)-ra 5.

Mennyi ez a „bizonyos idő”. Ennek kiszámolása nagyon egyszerű. Legyenek adottak azok a tevékenységek (Q), amelyek felhasznált tartalékidejét (együttesen) csökkenteni szeretnénk. Ekkor legyen t_i az az idő, amennyi ideig valamennyi csökkenthető úgy, hogy törésponthoz nem érnének, illetve ha eléri, akkor ez a töréspont negatív. Másrészt a rákövetkezési relációk meghatározzák, hogy az elmozgatható tevékenységek közül mennyivel mozgathatjuk el őket, hogy a rákövetkezési reláció ne sérüljön. Ezt az időt pedig úgy számíthatjuk ki, hogyha egy tevékenységnek van megelőző tevékenysége, akkor a megelőző tevékenység befejezéséből kivonjuk a követő tevékenység kezdési időpontját. Ez a szám legyen t_s (pl. Ez az idő a 2.1.3-4 ábrán a (7,8) – ra $33-(18+6)=9$). Továbbá legyen $Q \subseteq P$ azon tevékenységek halmaza, amelyeket az adott lépésben minimalizálni szeretnénk. Ekkor az az idő, ameddig a lineáris modellt használhatjuk (legyen t_l) az alábbi módon számítható:

$$t_l := \min(t_{s(i,j)}; t_{i(i,j)}), \text{ ahol } \forall (i,j) \in Q. \quad (2.1.2-1)$$

Ekkor $x_{(i,j)} := x_{(i,j)} - t_l$, ahol $(i,j) \in Q$.

Ebből következik, hogyha a kiválasztott felhasznált tartalékidőket az így kiszámolt idővel csökkentjük, a megengedettséget nem szegjük meg.

Az algoritmus leglényegesebb pontja most következik. Mégpedig amikor t_l -el csökkentettük $x_{(i,j)}$ -ket, akkor meg kell választanunk a következő Q halmazt. Nyilvánvaló, hogy $Q \subseteq P$, vagyis a kiválasztható tevékenységeknek (az elmozgatott tevékenységeknek) részhalmaza az a halmaz, amelyet ténylegesen kiválasztunk. A feladat megoldására egy úgynevezett Branch & Bound módszert alkalmaztam, amely a döntési pontokban a következőt teszi: P halmazban kezdetben az optimalizálandó tevékenységek szerepelnek. Q legyen részhalmaza P -nek. Ekkor rekurzívan ismételhetjük előlről, amíg nem lesz olyan tevékenység, amit „visszatolhatnánk”.

Az algoritmus informális leírásában találkozhatunk egy olyan esettel, amit eddig még nem említettem. A simítási eljárás során egyfajta speciális tulajdonságú megengedett megoldást kapunk, amelyre teljesül a következő:

Az alábbi eset nem fordulhat elő:



2.1.2-1 ábra: ütemezési csapdák

Ilyen a simításon alapuló megengedett megoldáskeresésnél azért nem fordulhat elő, mert csak addig kell simítani (eltolni) a tevékenységet, míg el nem érünk egy megengedett megoldást, tehát rögtön a második esetet kapnánk. Párhuzamos illetve soros allokációs eljárás alkalmazása esetében, valamint az algoritmus futása során köztes állapotként előfordulhat az első eset is. Éppen ezért, hogy a módszer ezt az esetet is kezelje, kibővítettem egy olyan vizsgálattal, amely megnézi, hogy van-e olyan hely, ahol a tevékenységet úgy be lehet ütemezni, hogy a korlátozó feltételek érvényessége ne boruljon fel. Nyilván itt is csak a töréspontokat kell figyelembe venni, valamint azt, hogy a tevékenység alatt van-e olyan töréspont, amely esetben a korlátozó feltétel sérülne. Ahhoz, hogy minden esetben helyes törésponti adatokkal számoljunk, a kiválasztott Q halmazbeli elemeket kivesszük a tevékenységek közül, majd az így kapott erőforrásterhelési diagram tetejére rendezzük.

Vezessük be a következő jelöléseket:

- Legyen P az optimalizálandó tevékenységek halmaza. Legyen továbbá A az összes tevékenységet tartalmazó halmaz (alaphalmaz).
- Legyen $Q \subseteq P$. Az optimalizálás során elmozgatandó tevékenységek halmaza.
- Legyen $z_{(i,j)}$ az (i,j) -tevékenység legkorábbi kezdésének időpontja.
- Legyen $x_{(i,j)}$ az (i,j) -tevékenység felhasznált tartalékideje, $z_{(i,j)}$ a (lehető) legkorábbi kezdése, $w_{(i,j)}$ (i,j) -tevékenység tényleges kezdési ideje a megengedett megoldásban, $d_{(i,j)}$ (i,j) tevékenység időtartama.
- Legyen T a töréspontok halmaza, ahol tároljuk „jobbról-balra” a töréspontok helyét és mértékét. Pl. a 2.3.-4 ábrára: $T := \{(40, 1), (33, -2), (27, -7), (24, 5), (20, 2), (18, -1), (15, 3)\}$. Korábbi töréspontokat nem érdemes tárolni, hiszen a korábbi időszakban kezdődő tevékenységeket nem mozgattuk el.
- A $T_{(i,j)}$ halmazban tároljuk a Q halmazra vonatkozó „legkorábbi kezdési időt” megelőző töréspont értékét. Tehát pl. a 2.3.-4 ábrán a $Q = \{(7,8)\}$ -as tevékenységre: a töréspontok $(27,-7)$, $(24, 5)$, $(20, 2)$, $(18, -1)$, $(15, 3)$, $(7, -6)$, $(4, 4)$. Vagyis 27. időpillanatban a töréspont értéke relatíve -7 , mivel ha a $(7,8)$ -as tevékenységet egy kis (dt) idővel a 27. időpont elé tolnánk, akkor az ott lévő erőforrás-kihasználás értéke 7-tel csökkenne az eredetihez képest. Így t_i értéke $Q = \{(7,8)\}$ -as tevékenységre $33-24=9$, mivel a $(7,8)$ -as tevékenységet maximum 9 időegységgel tolhatjuk el anélkül, hogy egy pozitív törésponthoz a 24. időpillanatban nem érnék. A töréspont értéke itt 5. Így $T_{(i,j)}$ -ben a $(7,8)$ -as tevékenységet a következőképpen tároljuk: $\{(7,8)\}, \{(6, -7), (9, 5)\}$. A többi töréspontot erre a tevékenységre nem érdemes tárolni, hiszen a 21. időpillanatnál előbbre nem tolhatjuk, mivel ekkor $x_{(7,8)}$ negatív lenne, ami ellentmondana azzal a feltevéssel, hogy a simítási eljárás előtti állapot (2.1.3-2 ábra) a felhasznált tartalékidőkre nézve optimális.
- Legyen T_s a rákövetkezési relációk idejének halmaza minden tevékenységre. Pl. a 2.1.3-4. ábrán $(7,8)$ -as tevékenységre $33-24=9$. Vagyis a $(7,8)$ tevékenység kezdete mínusz az $(5,8)$ -as tevékenység befejezése = 9. Ugyanígy itt is csak a nem nulla (negatív nem lehet) pozitív számokat tároljuk. Ezek kezdetben adottak, később mindig újra számíthatók. Tehát az előbbi példára egy elem a T_s -ben $\{(7,8); (5,7)\}, \{9\}$. Vagyis ez azt is jelenti, hogy a $(7,8)$ -as tevékenységet csak maximum 9 egységgel tolhatjuk el, hogy ne sértsük meg a rákövetkezési relációt.

- Legyen $\phi(t)$ a t . időpillanatban lévő összes erőforrásigény, $\phi(t)=c$ pedig az erőforráskorlát, ami jelen esetben konstans.
- Legyen továbbá a **ReCalculate** egy olyan függvény, amely kiszámítja a $T_{(i,j)}$ és T_s halmazokat a Q és A halmazok figyelembevételével, vagyis kiszámítja a töréspontok, és a rákövetkezési relációk idejét úgy, hogy a Q halmazbeli elemeket az erőforrásterhelési diagram „tetejére rendezi”, hogy valós töréspontokat számíthassunk. (Ha nem rendeznénk a minimalizálandó tevékenységeket a diagram tetejére, akkor a töréspont definíciója miatt nem vennénk észre valamennyi töréspontot.) Nyilvánvaló, hogy $T_{(i,j)}$ illetve T_s is véges számú elemet tartalmaz, hiszen véges számú tevékenység van.

Az algoritmus formális leírása:

Procedure ERALL-OPT($A, P, T_{(i,j)}, T_s, T$)

Begin

While $P \neq \emptyset$ **do** *{Addig ismételtetjük az eltolásokat, ameddig csak lehet.}*

Begin

Let be Q another subset of P where all $q_{(i,j)} \in Q$ satisfy that $\phi(z_{(i,j)} + x_{(i,j)} + t) \leq c$ and $x_{(i,j)} > 0$ where $t \in [0 - dt, d_{(i,j)}]$; *{csak olyan elemeket optimalizálhatunk, amelyeknél nem fog sérülni a megengedettség.}*

ReCalculate($T, T_{(i,j)}, T_s, Q, A$); *{Számítsuk ki újra $T_{(i,j)}, T_s$ halmazokat úgy, hogy a kiválasztott Q -beli elemeket az erőforrásterhelési diagram tetejére rendezzük.}*

$t_i := \max(x_{(i,j)})$, where $(i,j) \in Q$; *{Kezdetben a maximálisan csökkenthető eltolásra állítjuk, ugyanis ettől csak biztos, hogy kevesebbet tolhatjuk el az erőforrásokat.}*

if $(Q, \{w_1, w_2\}) \in T_{(i,j)}$ **and** $w_2 > 0$ **then** $t_i := w_1$ **else** $t_i := t_i$; *{Ha nincs ilyen korlátozó feltétel, akkor teljesen visszatolhatnánk, ha viszont van ilyen, akkor csak annyival toljuk vissza, amennyivel lehet.}*

For all $(i,j) \in Q$ **do**

Begin

if $(\exists(\{q_{(i,j)}, p_{(k,l)}, \{w\}\}) \in T_s$ where $q_{(i,j)} \in Q$ and $p_{(k,l)} \in A \setminus Q$ and w is minimal) **then** $t_{s(i,j)} := w$ **else** $t_{s(i,j)} := x_{(i,j)}$; *{Ha van olyan rákövetkezési reláció, amely korlátozná az elmozgatandó tevékenységet, akkor ez az érték legyen a korlátozó feltétel; egyébként legyen annyi, amennyivel a megengedett megoldás érdekében elmozdítottuk a tevékenységeket. Hiszen ha nincs ilyen korlátozás, akkor akár teljesen visszatolhatnánk az eredeti helyére.}*

$t_{(i,j)} := \min(t_{s(i,j)}, t_i)$

if $t_i > t_{(i,j)}$ **then** $t_i := t_{(i,j)}$; *{A minimálisan eltolható időértékkel kell valamennyi tevékenységet eltolni, hogy megőrizzük a megengedettséget.}*

End;

For all $(i,j) \in Q$ **do**

Begin

$T := T \setminus \{w_{(i,j)}, \phi(q_{(i,j)}) - \phi(q_{(i,j)} - dt)\}$; *{Kivesszük azt a töréspontot, ahonnan most elmozgatjuk az elemet}*

$x_{(i,j)} := x_{(i,j)} - t_i$; *{Minden kiválasztott elemet visszatolunk t_i -el}*

$w_{(i,j)} := z_{(i,j)} + x_{(i,j)}$

PRINT $x_{(i,j)}$;

$T := T \cup \{w_{(i,j)}, \phi(q_{(i,j)}) - \phi(q_{(i,j)} - dt)\}$; *{Betesszük azt a töréspontot, ahová most elmozgattuk.}*

PRINT Q ;

End;

$P := P \setminus \{p_{(i,j)} \mid p_{(i,j)} \in P \text{ and } \exists s_{(k,l)} \in A \setminus P \text{ where } \exists(p_{(i,j)}, s_{(k,l)}, w) \notin T_s\}$; *{P-ből kivonom azokat az elemeket, amelyeket a rákövetkezési relációk miatt nem lehetne már visszatolni.}*

If $Q = \emptyset$ **then For all** $(i,j) \in P$ **do** $t_i := \text{Hole}((i,j), A, T)$;

If $Q = \emptyset$ **and** $t_i = 0$ **then** $P := \emptyset$; *{Nem tudunk tovább optimalizálni, mivel sehogyan sem tudunk újabb részhalmazt kiválasztani, valamint semennyivel sem tudtuk visszatolni az elemeket.}*

ERALL-OPT($A, P, T_{(i,j)}, T_s, T$); *{Rekurzívan meghívjuk a függvényt.}*

End;

End.

Recalculate segédfüggvény, amely kiszámítja a $T_{(i,j)}$, T_s halmazokat.

Procedure ReCalculate(T , $T_{(i,j)}$, T_s , Q , A)

Begin

TempTi:={};

{Kezdetben a halmazok üresek}

$T_s := \{\{\}, \{\}\}$;

$q_{(i,j)} := (\exists q_{(k,l)} \text{ of } Q \text{ where } q_{(k,l)} \in Q \text{ and } w_{(i,j)} \text{ is minimal});$ *{egy Q-beli elemet kiválasztunk, mely a leghamarabb kezdődik.}*

$j_{(i,j)} := w_{(i,j)} - z_{(i,j)}$

While $j_{(i,j)} \geq 0$ **Do**

Begin

$TempT_i := TempT_i \cup \{(w_{(i,j)} - a, b) \text{ where } (a, b) \in T \text{ and } a - z_{(i,j)} < j_{(i,j)} \text{ and } a - z_{(i,j)} > 0\};$ *{Ebbe a halmazba gyűjtjük a Q halmazra vonatkozó legközelebbi töréspontokat és a töréspontok értékét.}*

$j_{(i,j)} := \{j, \exists (a, b) \in T j := a - z_{(i,j)} \text{ where } j \geq 0 \text{ and } j < j_{(i,j)}\}$ *{j_(i,j) értékét a következő töréspontig csökkentjük}*

End;

$T_i := \{Q, TempT_i\}$;

For all $(q_{(i,j)} \in Q \text{ and } p_{(k,l)} \in A \setminus Q)$ **Do**

Begin

If $(w_{(i,j)} - (w_{(k,l)} + d_{(k,l)}) > 0)$ **then** $T_s := T_s \cup \{(q_{(i,j)}, p_{(k,l)}), w_{(i,j)} - (w_{(k,l)} + d_{(k,l)})\};$ *{Feltöltjük a T_s halmazt.}*

End;

End;

ERALL-OPT algoritmus kiterjesztése Branch and Bound keretalgoritmus használatával

Kezdeti probléma alakja: $R_0 = \{A, P_0, T_0, T_{S_0}, c, Q_0, U_0\}$, ahol

A : tevékenységek alaphalmaza;

P_0 : optimalizálandó tevékenységek kiinduló halmaza;

T_0 : töréspontok kiinduló halmaza;

T_{S_0} : A P_0 halmaz tevékenységeinek rákövetkezési relációi ($T_{S_0} = \{\{i, j\}, \{t\}\}$, ahol $i, j \in P_0$,

valamint $t \in R_0^+$);

c : erőforráskorlát konstans;

Q_0 : az R_0 problémához tartozó optimalizálandó tevékenységek halmaza, $Q_0 = \emptyset$;

U_0 : $U_0 = \emptyset$;

Egy tetszőleges részprobléma alakja: $R_i = \{A, P_i, T_i, T_{S_i}, c, Q_i, U_i\}$, ahol

A : tevékenységek alaphalmaza;

P_i : az optimalizálandó tevékenységek aktuális halmaza; $P_i = \emptyset$, ha R_i a B&B fa egy levele, azaz tovább nem optimalizálható megoldás;

T_i : a töréspontok aktuális halmaza;

T_{S_i} : a P_i halmaz tevékenységeinek rákövetkezési relációi ($T_{S_i} = \{\{i, j\}, \{t\}\}$, ahol $(i, j) \in P_i$ valamint $t \in R_0^+$);

- c : erőforráskorlát konstans;
- Q_i : az R_i részproblémához tartozó optimalizálandó tevékenységek halmaza, $Q_i \in \wp(P_i)$. Megjegyzem, hogy Q_i vagy egy tevékenységet tartalmaz, vagy olyan tevékenységeket, melyek között átlapolás van;
- U_i : a P_0 halmaz azon elemei, amelyek már nem vonhatóak be az optimalizálási folyamatba, azaz $U_i = P_0 \setminus P_i$.

További jelölések:

- H : aktív halmaz, azon részproblémák halmaza, amelyek még nem kerültek kiértékelésre; kezdetben $H = \emptyset$;
- f_i : az R_i részproblémához tartozó megoldás értéke;
- g_i : az R_i részproblémához tartozó alsó korlát értéke, $g_i \leq f_i$;
- $x_{(k,l)}$: a (k,l) eseményhez tartozó tartalékidő, ahol $(k,l) \in A$;
- f^* : az aktuálisan legjobb megoldás;
- R : az aktuálisan legjobb megoldáshoz tartozó részprobléma;

Az algoritmus lépései:

1. R_0 kezdeti problémát betesszük a H aktív halmazba, $f_0 = \sum_{P_0} x_{(k,l)}$.
2. **A kiválasztási szabály alapján** kiválasztunk egy aktív R_i részproblémát H halmazból.
3. Ha $Q_i \neq \emptyset$, ahol $Q_i \in R_i$, akkor
 1. meghívjuk az *ERALL-OPT* algoritmust R_i részprobléma paramétereivel
 2. $f_i = \sum_{P_0} x_{(k,l)}$.

Megjegyzés: a feltétel csak akkor nem teljesül, ha $R_i = R_0$, ugyanis ekkor $Q_0 = \emptyset$, máskülönben $Q_i \neq \emptyset$.

4. Ha $P_i = \emptyset$, akkor az eddigi f^* legjobb megoldást illetve R részproblémát aktualizáljuk f_i alapján.
5. **A szétválasztási szabály alapján** P_i halmazból létrehozuk a Q_j halmazokat, azaz egy $Q \subset \wp(P_i)$ halmazt. Ha van olyan tevékenység, amelyet P_i -ből nem választhatunk, akkor azt U_i -hez adjuk.
6. Minden $Q_j \in Q$ halmazra:
 1. létrehozunk egy $R_j \{A, P_i, T_i, T_{Si}, c, Q_j, U_i\}$ részproblémát;

2. kiszámoljuk a következő képlet alapján g_j alsó korlátot:

$$g_j = \sum_{P_0 \setminus U_i} (x_{(k,l)} - t_{gj}) + \sum_{U_i} x_{(m,n)}, \text{ ahol}$$

$$t_{gj} = \min \left[\min_{P_0 \setminus U_i} (x_{(k,l)}), \min_{P_0 \setminus U_i} (t_{s(k,l)}) \right], \text{ valamint } t_s \text{ értékek az ERALL-OPT}$$

algoritmusból származnak;

3. ha $g_j < f^*$, akkor R_j -t betesszük H -ba (**korlátozási szabály**).
7. Ha $H \neq \emptyset$, akkor a 2. lépésre ugunk; egyébként az algoritmusnak vége és a legjobb megoldást az R részprobléma tartalmazza.

U halmazban elhelyezzük a P halmazból kieső elemeket.

Állítás: Az algoritmus nem ad rosszabb megoldást, mint amilyenből kiindultunk.

Bizonyítás: Minden lépésben $x_{(i,j)}$ -n, vagyis a felhasznált tartalékidőn csökkentünk, ha ez lehetséges. Így legrosszabb esetben a kiinduló megoldást kapjuk.

Állítás: Az algoritmus véges lépésben megáll.

Bizonyítás: Felhasználva, hogy egy lépés a töréspontoktól és a rákövetkezési relációktól függ, amelyek véges számúak, valamint az ilyen pontokban (legyenek továbbiakban döntési pontok) P -ből az elmozgatott tevékenységekből álló halmazból választjuk ki az optimalizálandó tevékenységeket, valamint az előző állítás értelmében mindig egy nem rosszabb állapothoz jutunk. Amint nem tudunk tovább optimalizálni, az algoritmus megáll. Véges helyen kell dönteni; amint egy döntési pontból nem tudunk tovább haladni (nem tudunk olyan Q -t kiválasztani, amelyre a feltételek igazak lennének), az algoritmus szintén megáll.

Állítás: Az algoritmus mindig megadja az optimális megoldást.

Bizonyítás: Ezt az biztosítja, hogy minden döntési pontban (rekurzió miatt) mindenféleképpen kiválasztjuk a P -beli elemeket, tehát az összes megoldást megkapjuk.

Állítás: Az algoritmus az összes optimális megoldást megadja.

Bizonyítás: Az előző állításból következik, hiszen minden lehetséges választást kipróbál.

Definíció: *Két megoldás ekvivalens*, ha valamennyi tevékenységre igaz, hogy a két megoldásban ugyanakkor kezdődnek, valamint minden időpillanatban ugyanannyi az erőforrásigényük.

Megjegyzés: Ekkor csak a felrajzolásban különbözhetnek (melyik helyezkedik el a másik felett az erőforrásterhelési diagramban).

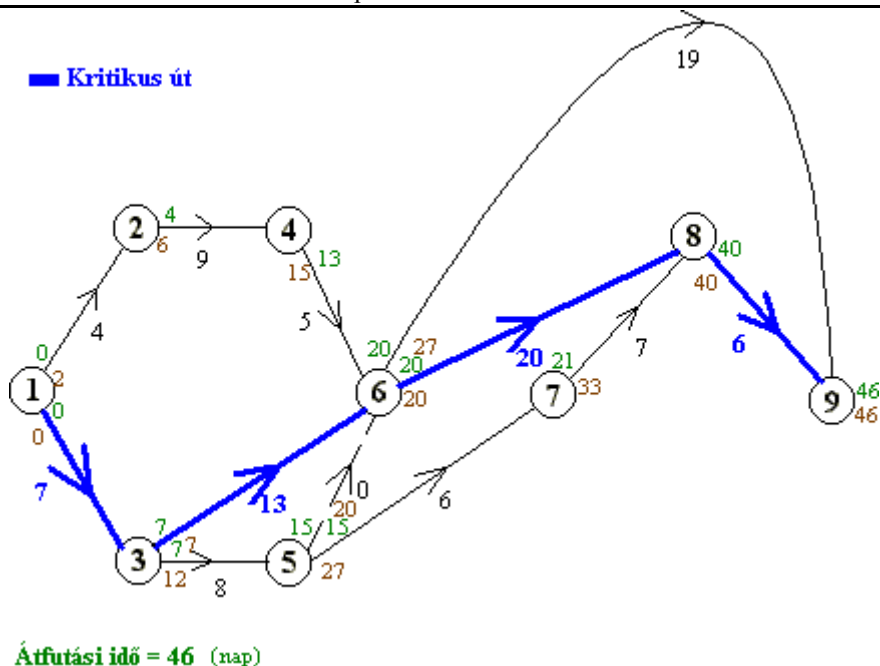
Definíció: Legyen egy (megengedett/optimális) *megoldáshalmaz fokszáma* a megoldáshalmazban lévő nem ekvivalens megengedett/optimális megoldások száma.

2.1.3 Példák

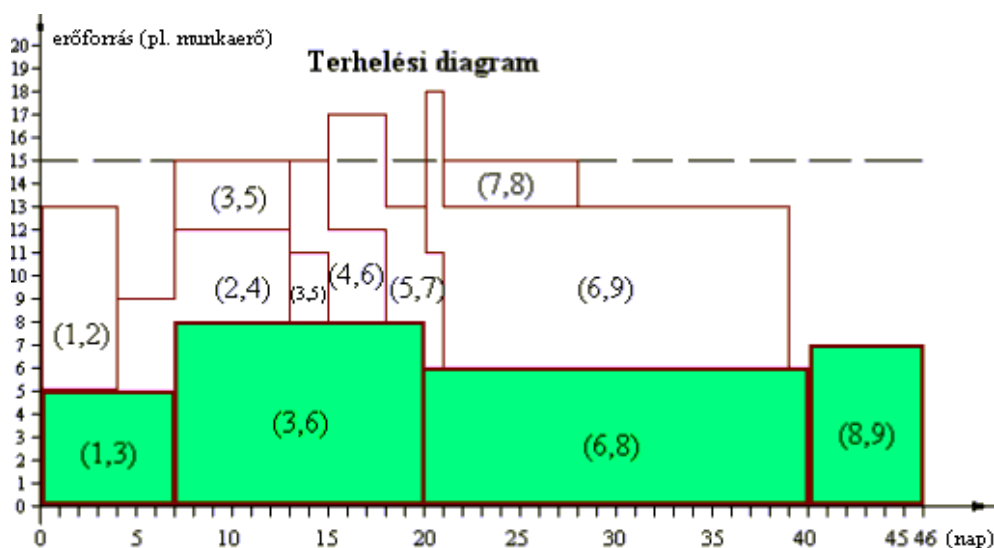
Adott egy tevékenységlista alapján felvett CPM-háló vagy MPM-háló (pl. 2.1.3-1 ábra), valamint a tevékenységekhez rendelt erőforrásigény (2.1.3-1 táblázat). Ekkor a tevékenységek legkorábbi beütemezésére felrajzolható egyértelműen az erőforrásterhelési diagram (pl. ehhez tartozó: 2.1.3-2 ábra). Ezen kívül adott egy megengedett megoldás, amelyre teljesül, hogy minden pontban az összes erőforrásigény nem nagyobb, mint az erőforráskorlát.

Tevékenység jele	Erőforrás-szükséglet	Időtartam
(1,2)	8	4
(1,3)	5	7
(2,4)	4	9
(3,5)	3	8
(3,6)	8	13
(4,6)	4	5
(5,6)	0	0
(5,7)	5	6
(6,8)	6	20
(6,9)	7	19
(7,8)	2	7
(8,9)	7	6

2.1.3-1 táblázat: CPM-tevékenységjegyzék



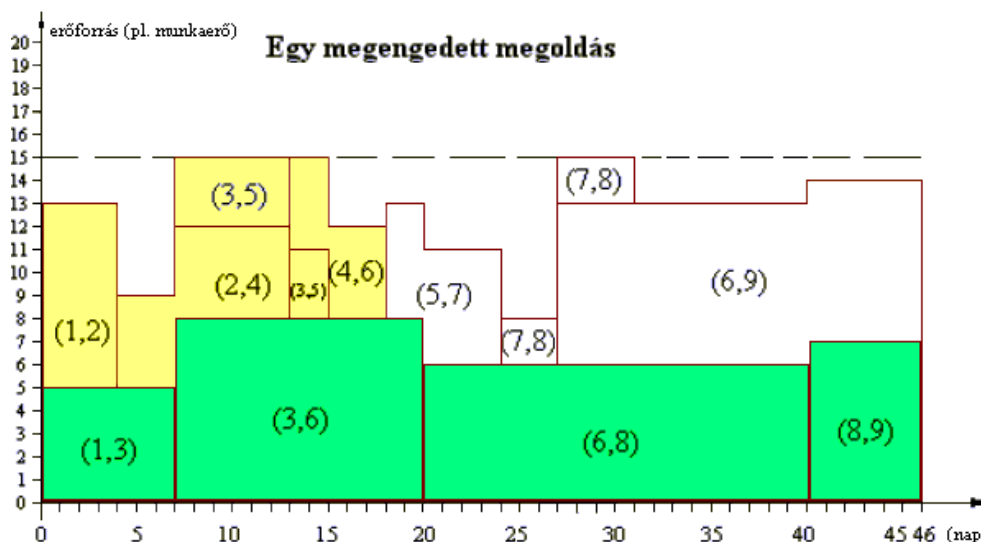
2.1.3-1 ábra: kritikus út egy CPM-hálóban



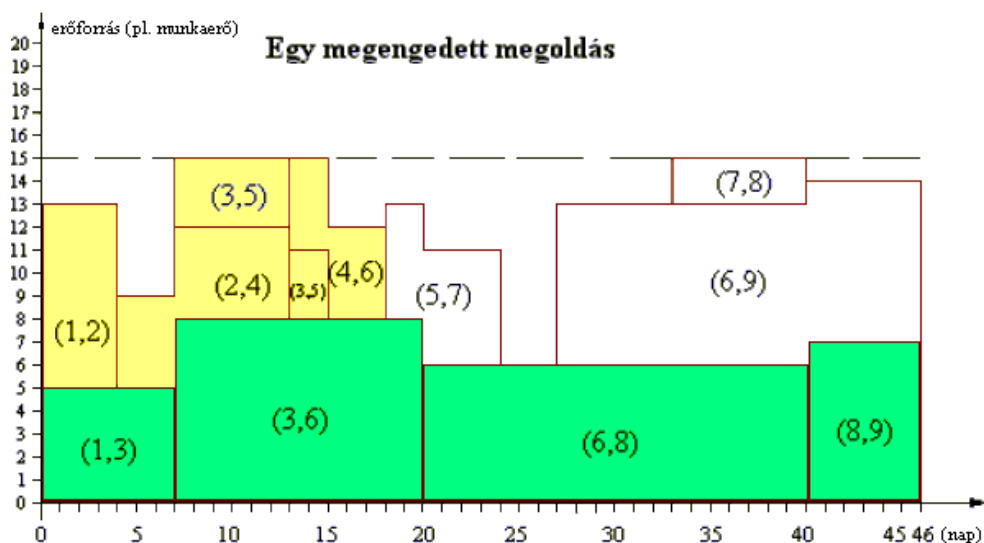
2.1.3-2 ábra: terhelési diagram

Ha csak azt vizsgálánk, hogy a rendelkezésünkre álló tartalékidőt mennyire használjuk ki, akkor ez az állapot lenne az optimális, mivel itt az egyes tevékenységeket olyan hamar kezdjük el, amilyen hamar csak tudjuk. Azonban van egy erőforráskorlátunk, amely korlátot úgy szeretnénk betartani, hogy a teljes átfutási idő (a kritikus út hossza) ne változzon. Erre a pl. a simító (allokációs) algoritmus ad egy heurisztikus megengedett megoldást. A bemutatandó módszer innen indul.

Az előző fejezetben említettem, hogy általában egy heurisztikus megoldás nem garantálja az optimumot, ellenben rendkívül gyors. A most bemutatandó algoritmus ezzel szemben garantálja az optimumot.



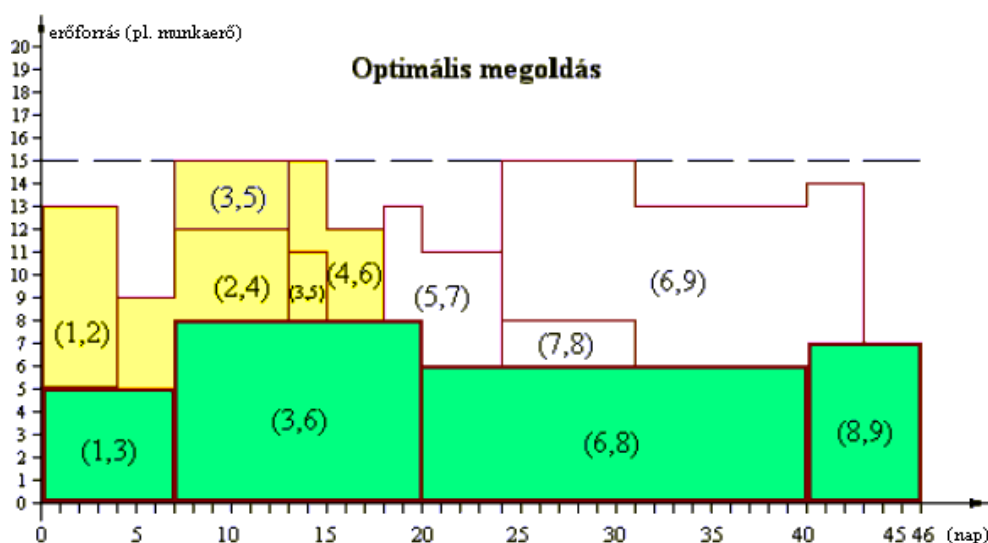
2.1.3-3 ábra: egy megengedett megoldás



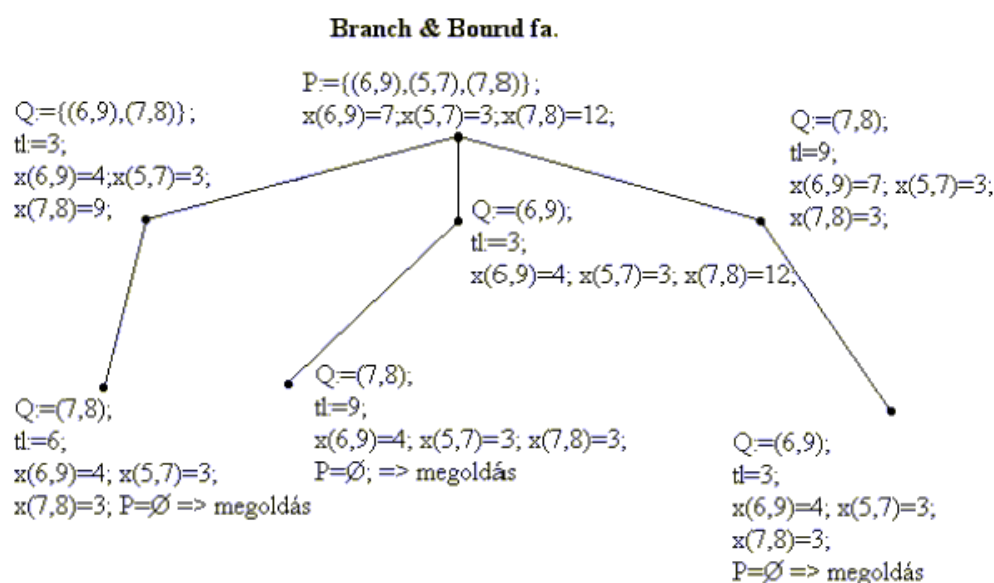
2.1.3-4 ábra: egy másik megengedett megoldás

Induljunk ki a 2.1.3-3 ábrán lévő megengedett megoldásból. Ekkor $P := \{(5,7), (6,9), (7,8)\}$. Q -nak csak olyan tevékenységet választhatok, amelyre nem lépünk túl a megengedési feltételt. Ez pedig csak a (6,9) tevékenység. Itt $t_i = 3$; vagyis maximum 3 egységgel kezdhetne korábban. Ezután már nincs olyan tevékenység, amit optimalizálni lehetne. Ekkor az optimális megoldás a 2.1.3-1 ábrán látható. Ha a 2.1.3-4 ábrán lévő megengedett megoldásból indulunk ki, akkor szintén $P := \{(6,9), (5,7), (7,8)\}$, Q lehet (6,9) vagy (7,8). Ennek megfelelően a Branch &

Bound fánk a 2.1.4-2 ábrán látható. Az algoritmus megadja a megoldásokat, amelyek ebben az esetben ugyanahhoz a megoldáshoz vezetnek.



2.1.3-5 ábra: az erőforrás-allokáció optimális megoldása



2.1.3-6 ábra: döntési fa

Általában egy ilyen algoritmus azért lassú, mert valamennyi megoldást megkeresi. Viszont egy megoldást viszonylag gyorsan elér. Ha úgy módosítjuk az algoritmust, hogy adunk egy olyan feltételt a befejezéshez, amelyet ha elérünk, az már elégséges, akkor az algoritmus rendkívül gyorsá tehető. A korlátozás és szétválasztás módszerét alkalmazva tovább gyorsíthatjuk az algoritmus lefutását. Ha egy megoldásról meg tudnánk mondani, hogy ez optimális, akkor szintén nem kell tovább keresnünk. Az algoritmus során

prioritásokat is kezelhetünk. Megmondhatjuk, hogy mely tevékenységeket optimálja előbb. Elmondható, hogy az algoritmusunk nem diszkretizál, tehát az egyes töréspontok, rákövetkezési relációk értékei felvehetik egy nemnegatív valós szám bármely részhalmazának tetszőleges értékeit a többi feltétel (rendelkezésre álló tartalékidő stb.) betartásával. (Ez különösen akkor lehet fontos, amikor a hálón alkalmazott időtartamok egysége nagyobb, mint egy óra, pl. nap, hét, hónap stb.)

2.2 Erőforrás-allokáció szakaszonként konstans erőforráskorlátozás esetén (ERALL-OPT/SZK, OPT-RALL/VRA)

A gyakorlatban sokszor előfordul, hogy a rendelkezésünkre álló erőforráskorlát függ az időtől. (Pl. egy szálloda építéseinél ünnepnapokon előfordulhat, hogy rendelkezésünkre jóval kevesebb munkaerő áll, mint más napokon.) Megmutatom, hogy ilyenkor bizonyos megszorításokkal vissza lehet vezetni a feladatot az eredeti problémára. Így egy megengedett megoldást kereső heurisztikus algoritmus (néhány apró ponton módosított változata) is működik, és egy megengedett megoldást talál. Látni fogjuk, hogy az általam kifejlesztett algoritmus viszont teljes egészében, lényegében változtatás nélkül visszavezethető az eredeti feladatra. [220-221, 223, 226]

A módszernek magyarul **ERALL-OPT/SZK** (optimális erőforrás-allokáció szakaszonként konstans erőforráskorlát esetén), angolul **OPT-RALL/VRA** (Optimized Resource Allocation with Variable Resource Availability) nevet adtam. (A /-jel után írodott rövidítések az alap algoritmushoz képest történt bővítést jelölik.)

2.2.1 A feladat és a megoldás leírása

Az előző fejezetben tárgyalt erőforrás-allokáció kizárólag konstans korlátozás esetén működött. Látni fogjuk, ha az erőforráskorlát szakaszonként konstans függvény, és a függvénynek csak véges sok helyen van szakadása, akkor egyszerűen visszavezethető az eredeti problémára.

Legyen adott egy φ függvény, mely az erőforráskorlátot adja meg minden pontban. Ennek a függvénynek véges sok helyen legyen csak szakadása, valamint e pontok kivételével legyen (szakaszonként) konstans függvény. Ilyen erőforráskorlátok mellett keressünk először egy megengedett megoldást. Mint azt látni fogjuk, első lépésként megpróbálunk egy olyan erőforráskorlátot keresni, amely konstans. Legyen ez a szám a φ függvény maximuma.

Azokon a szakaszokon, ahol φ függvény értéke kisebb ennél, ott vezessünk be olyan látszat-erőforrásigényt, amelyeket semmiképpen sem mozgathatunk el a megengedett megoldás-keresésben. Rendezzük ezeket a látszat-erőforrásigényeket az erőforrás-terhelési diagram alá. Ha létezik megengedett megoldás, akkor az algoritmusom megtalálja az optimális megoldást, hiszen a látszat-erőforrásigényeket nem mozgattuk el. Így ezeket a látszat-erőforrásigényeket optimálnunk sem kell.

Fontos megjegyezni, hogy a megengedett megoldás keresésébe a látszaterőforrás-igényeket olyan erőforrásigényekként/tevékenységekként kell kezelni, amelyeket semmilyen körülmények között sem szabad elmozdítani. Ha másképpen nem lehet megoldani a feladatot, akkor a feladatnak nincs megengedett megoldása, hiszen ez ellentmondana a kezdeti erőforráskorlát-/erőforrásigény-feltevéseinknek. Ha létezik megengedett megoldás, akkor az algoritmusunk ezeket a tevékenységeket semmiképpen sem mozgatja el (nem optimalizálja), ugyanis a megengedett megoldás keresésekor ezeket a tevékenységeket TILOS volt elmozdítani!

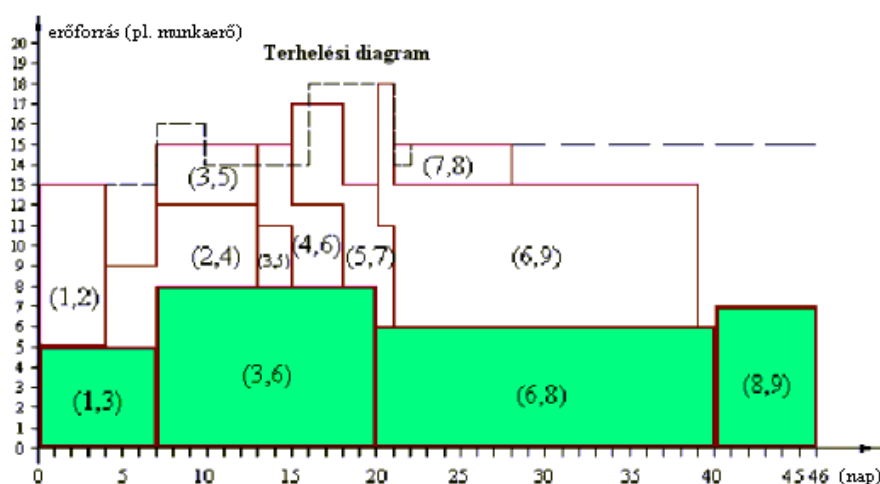
Definíció: Egy szakaszonként konstans φ erőforráskorlát-függvény maximuma, és a φ függvény értelmezési tartományában lévő pont közötti erőforrásigényt *látszat-erőforrásigénynek* nevezzük.

Állítás: A módosított **ERALL-OPT/SZK-algoritmus** is véges lépésben megadja az optimális megoldást.

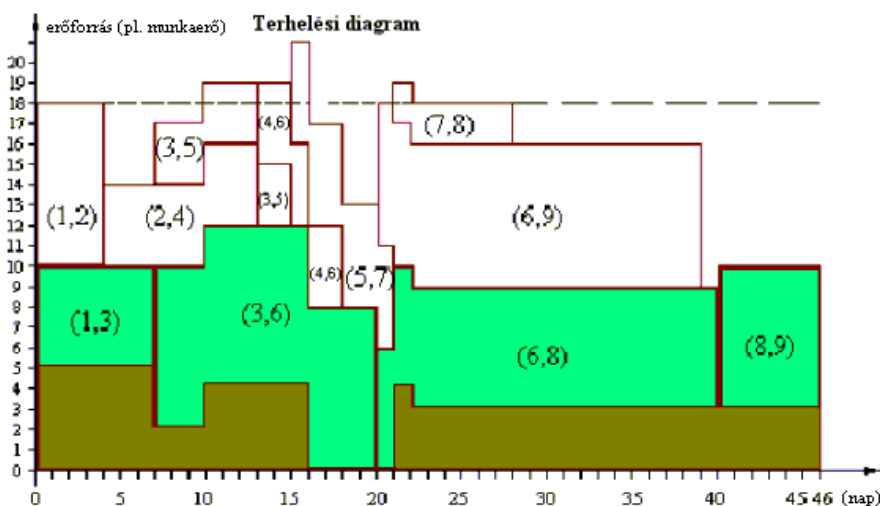
Bizonyítás: Visszavezetjük a feladatot az előző problémára. Tegyük fel, hogy a feladatnak létezik megengedett megoldása. (Amennyiben nem létezik, úgy optimális megoldása sem lehet, hiszen az optimális megoldás egyben megengedett megoldás is.) Használjuk ki, hogy a megengedett megoldás keresésében a „látszat-erőforrásigényeket” nem mozgattuk el. (Ebben az esetben ugyanis ez szintén nem lenne megengedett a megoldás). Ezért az **ERALL-OPT-algoritmus** ezeket a tevékenységeket nem is fogja optimalizálni. Rendezzük a látszat-erőforrásigényeket a terhelési diagram alá. Az optimalizálandó tevékenységek halmaza legyen P , ezután alkalmazzuk az **ERALL-OPT-algoritmust**.

2.2.2 Példa

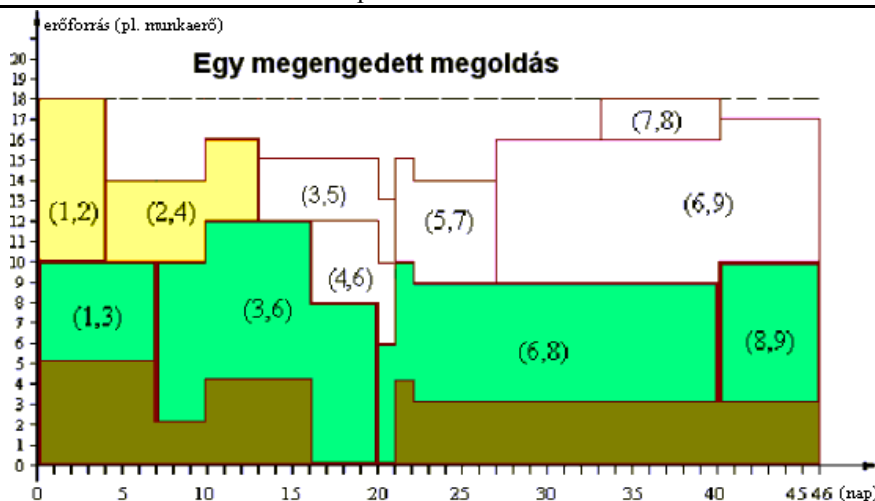
A 2.2.2-1 ábrán láthatunk egy olyan esetet, melynél az erőforráskorlátnak véges sok helyen szakadása van. Ezen pontok kivételével viszont az erőforráskorlát egy konstans függvény. A 2.2.2-2 ábrán az eredeti feladatot úgy rajzoljuk át (az erőforráskorlát és az erőforrás-terhelés függvényeinek különbségét olyan látszat-erőforrásigényeknek tekintve, melyet nem mozgathatunk el), hogy az előző feladattal ekvivalens feladatot kapjunk. Egy megengedett megoldást (2.2.2-7 ábra) optimalizálunk az eredeti algoritmussal.



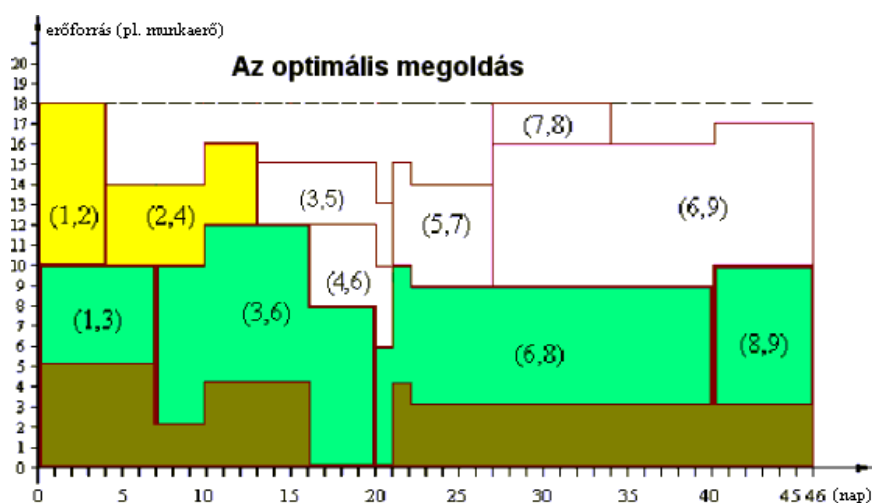
2.2.2-5 ábra: terhelési diagram



2.2.2-6 ábra: a terhelési diagram látszat-erőforrásigény elrendezése után



2.2.2-7 ábra: egy megengedett megoldás



2.2.2-8 ábra: az optimális megoldás

Megjegyzés: Olyan eseteket vizsgáltunk, ahol a maximális erőforráskorlátozó függvény független volt a tevékenységektől, csak az időtől függött. Legyen az erőforráskorlát, tehát $\varphi: [0, m] \rightarrow \{y_1, \dots, y_n\}$, ahol $m \in R^+, n \in Z^+$. Az erőforráskorlát-függvényt 0-tól m -ig értelmeztük. $m = TPT$ legyen a projekt átfutási ideje; a függvénynek véges sok (n db) helyen van szakadása. Itt a függvény $\{y_1, \dots, y_n\}$ értékeket vehet fel.

Amint láthatjuk, ez az algoritmus sem diszkretizál. A szakadások a projekt átfutási idején belül bármely nem negatív valós számot felvehetnek.

2.3 További alkalmazások (tevékenységek, erőforrásigények követése, megszakítható tevékenységek kezelése)

Nemcsak a projektek tervezése, de a projektek nyomonkövetése is elengedhetetlen egy sikeres projekt végrehajtásához. A tevéhez képest gyakran előfordulhatnak a megvalósítás során kisebb-nagyobb változások (pl. tevékenység időtartama, költségigénye, erőforrás-szükséglete stb.). Az ilyen eltérések kezelésére két fajta módszert is választhatunk. Vagy beépítjük a projekttervbe az egyes változók (tevékenység időtartama, költség- és erőforrásigénye stb.) bizonytalanságát (lásd. 2.8. fejezet), vagy a megváltozott paraméterekkel „újra ütemezzük” a tevékenységeket. Ebben a fejezetben először a második módszer lehetőségeit mutatom be.

Legyen $\nu(t)$ egy adott időpont a koordinátarendszeren. Tegyük fel, hogy eddig az időpontig a tevékenységek egy adott módon már végbementek. (pl. valamilyen optimalizációs eljárás által meghatározott módon. Lehet pl. $\nu(t)$ a mai dátum, amikor tudomásunkra jut a változás).

Mint látni fogjuk, lényegében a $\nu(t)$ időpont előtt befejezett tevékenységekkel nem kell foglalkoznunk, hiszen az már végrehajtott. A végrehajtás alatt lévő tevékenységeket két csoportra bonthatjuk. Az első csoportot azok a tevékenységek alkotják, melyeket semmilyen körülmények között nem szabad megszakítani (ilyen lehet pl. egy kémiai kísérletsorozat, építőiparban bizonyos betonozási munkák stb.). A másik csoportba azok a tevékenységek tartoznak, amelyeket megszakíthatunk. A továbbiakban először a nem megszakítható tevékenységekkel foglalkozunk, majd a modellelmet kiterjesztmem megszakítható tevékenységek kezelésére is. A futás alatt lévő be nem fejezett, nem megszakítható tevékenységek még be nem fejezett részét nem szabad elmozgatnunk. Így az előző módszerhez hasonlóan úgy kezeljük őket, mintha látszat-erőforrásigények lennének. Ily módon biztosítjuk azt, hogy ne szakad hassanak meg. [220-221, 223, 226]

2.3.1 Erőforráskorlát változása (ERALL-OPT/ON-LINE/VK, OPT-RALL/ON-LINE/VRA)

Az erőforráskorlátok, erőforrásigények, tevékenységek időtartamainak nyomonkövetése fontos feladat, mert így e paraméterek esetleges változásaira a szervezet jobban fel tud készülni.

Az előbb felsorolt paraméterek lehetséges változásai közül most tekintsük a legegyszerűbb problémát, ha a projekt megvalósítása közben kiderül, hogy az erőforráskorlát

megváltozik. Az éppen futó tevékenységeket két csoportba kell bontani aszerint, hogy a tevékenységek megszakíthatók vagy sem. Először tehát azzal az esettel foglalkozom, amikor a tevékenységeket nem lehet megszakítani. (Azzal az esettel, amikor definiáljuk egy tevékenység megszakíthatóságát, később foglalkozom). Tehát ügyelnünk kell arra, hogy a már elkezdett tevékenység ($v(t)$ időpillanat előtt elkezdett tevékenység) nem szakadhat meg. Ezt úgy érhetjük el, hogy a folyamatban lévő tevékenységet az előző fejezetben tárgyalt látszat-erőforrásigények módjára kezeljük. Ezzel elérhetjük, hogy egy megengedett megoldást kereső algoritmus (pl. párhuzamos allokáció, soros allokáció stb.) nem fogja a tevékenységet elmozdítani. Az **ERALL-OPT** módszer (mivel elmozdítás nem történt) így nem fogja a látszat-erőforrásigényeket optimalizálni. Ezzel a megszorítással elegendő a $v(t)$ időpont után optimalizálni, hiszen a már végrehajtott tevékenységeket szükségtelen is lenne optimalizálni. [220-221, 223, 226]

A módszernek magyarul **ERALL-OPT/ON-LINE/VK** (on-line optimális erőforrás-allokáció erőforráskorlát változása esetén), angolul **OPT-RALL/ON-LINE/VRA** (On-line Optimized Resource Allocation with Variable Resource Availability) nevet adtam.

Állítás: A módosított **ERALL-OPT/ON-LINE/VK** megadja az optimális megoldást véges lépésben.

Bizonyítás: Visszavezetjük a feladatot az előző problémára. Legyen adott egy $v(t)$ időpont, ahol $0 \leq v(t) \leq TPT$, vagyis ez az időpont a kezdés és a projekt átfutási ideje között van. Az eddig az időpontig befejezett tevékenységeket nem kell ütemeznünk. Az ebben a pillanatban folyamatban lévő tevékenységek közül a nem megszakíthatókat a következőképpen kezeljük: Legyen $z_{(i,j)}$ a folyamatban lévő tevékenység kezdési ideje, $d_{(i,j)}$ a tevékenység időtartama valamint A_n a folyamatban lévő nem megszakítható tevékenységek halmaza, ahol

$$z_{(i,j)} + d_{(i,j)} - v(t) > 0 \quad \forall (i,j) \in A_n \quad (2.3.1-1)$$

A folyamatban lévő tevékenységeket a terhelési diagram aljára rendezzük. A továbbiakban csak $v(t)$ és a TPT időpont között lévő időintervallumban optimalizálunk. A folyamatban lévő tevékenységeknek a látszaterőforrással azonos prioritást adunk, ezzel biztosítva azt, hogy a megengedett megoldáskeresésben ne mozgassuk el. Jelölje A_m a $v(t)$ időpontban megszakítható tevékenységek halmazát. Ezeket a tevékenységeket a terhelési diagram tetejére rendezzük, az alternatív úton lévő tevékenységekkel azonos

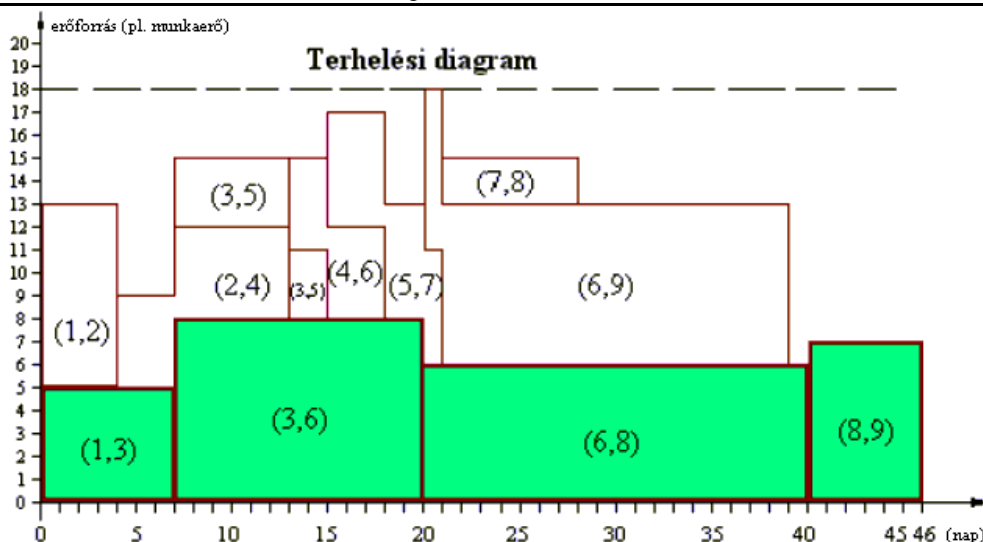
prioritást adunk ezeknek a tevékenységeknek. Amennyiben létezik megengedett megoldás és $A_m = \emptyset$, úgy az **ERALL-OPT/SZK-algoritmussal** optimális megoldást kaphatunk; amennyiben $A_m \neq \emptyset$, úgy erre az intervallumra ($v(t)$ és **TPT** között) használjuk az **ERALL-OPT/MSZT** (2.3.4. fejezet) módszert optimális megoldás keresésére.

2.3.1.1 Példa

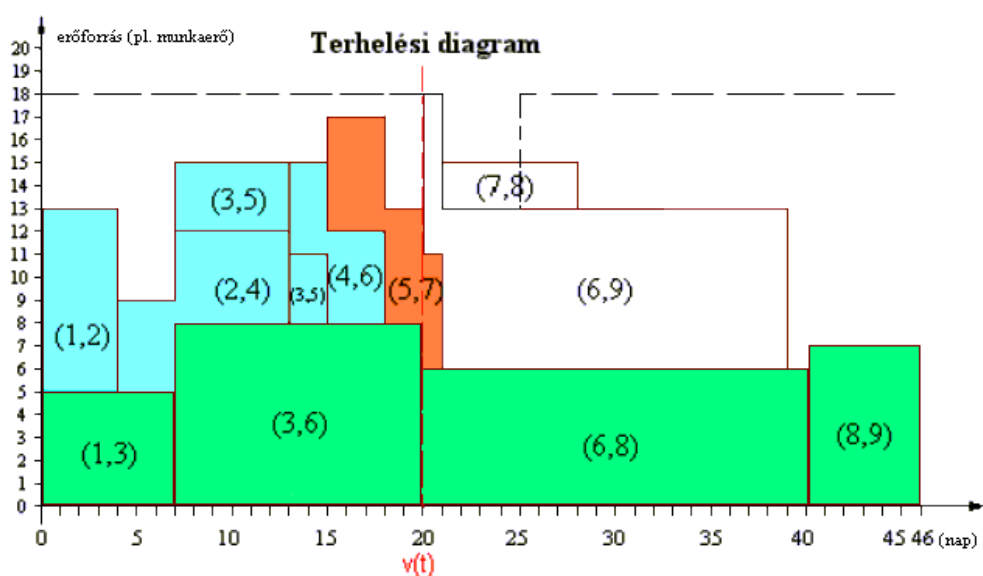
A gyakorlati életben sokszor előfordul, hogy a már működő projektbe menet közben kell beavatkoznunk. Ennek több oka is lehet. Előfordulhat, hogy a projekt/termelés végrehajtása közben bizonyos nehézségek lépnek fel. Például egy adott tevékenység a tervezettnél több erőforrást igényel, vagy éppen időtartama tovább tart, mint arra számítottunk. Az is lehetséges, hogy az erőforráskorlát másképpen alakul, mint azt eredetileg elterveztük. Megmutatom, hogy ebben az esetben - az előző két algoritmust kombinálva - hogyan oldhatjuk meg ezeket a feladatokat.

A következő példában az egyszerűség kedvéért olyan kapacitáskorlátból indultunk ki, amelynél nem volt szükség sem simításra, sem optimalizálásra. A 20. napon azonban kiderül, hogy a 21. naptól a kapacitáskorlát (pl. munkások száma) lecsökken átmenetileg 13 -ra (pl. váratlan influenza-járvány következtében 5 ember nem tud munkába jönni), a 25. napon visszaáll 18-ra (meggyógyultak).

Formálisan: $v(t) := 20$, $\varphi_1(t_1) = 18$, ha $0 \leq t_1 \leq 21$, vagy $25 < t_1$, és $\varphi_2(t_2) = 13$, ha $21 < t_2 \leq 25$. Láthatjuk, hogy ezekkel a megszorításokkal visszavezethetjük az előző fejezetekben tárgyalt megoldási módszerekre.

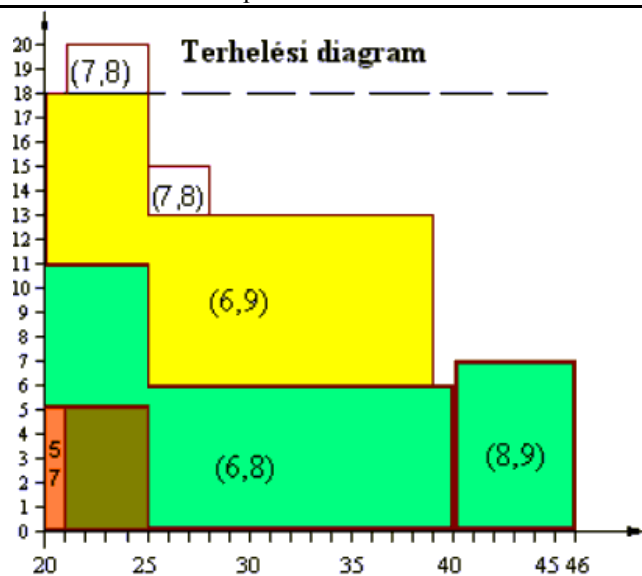


2.3.1.1-9 ábra: terhelési diagram kiinduló állapota

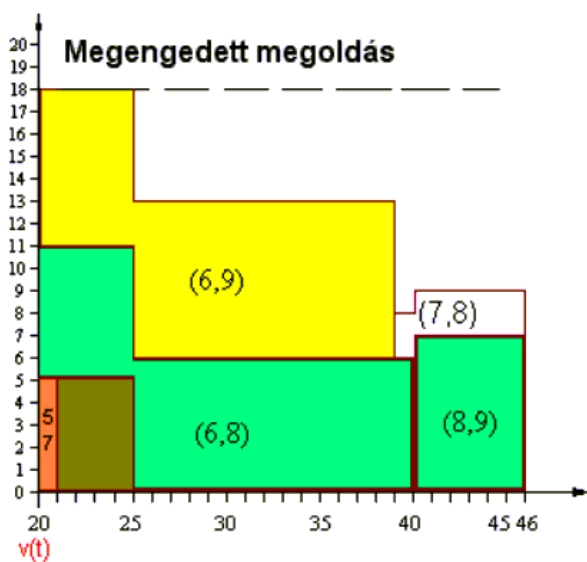


2.3.1.1-10 ábra: terhelési diagram az erőforráskorlát megváltozása után

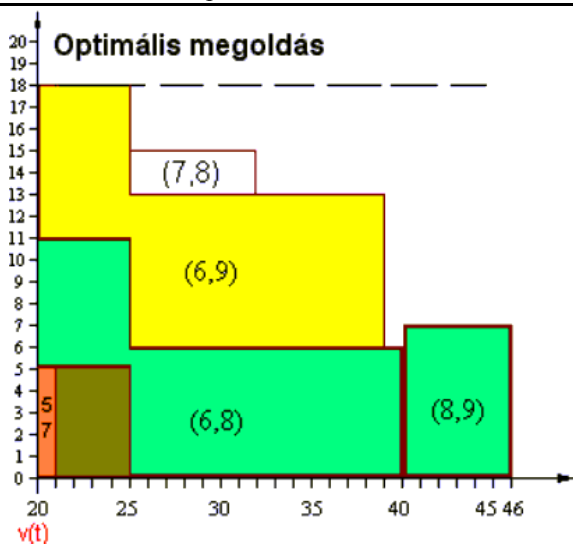
Ez a gyakorlatban azt jelentheti, hogy a 21. naptól 4 napon keresztül 4 emberrel kevesebbre számíthatunk. Ilyen változások például egy-egy influenzajárvány esetén fordulhatnak elő. Ha az erőforrás egy berendezés, akkor ez a változás azt jelenti, hogy a 21. napon (például meghibásodás miatt) 4 napig (a javítási munkák befejezéséig) 4-gyel kevesebb berendezéssel kell megvalósítani a projektet. A kérdés az, hogy a projekt átfutási ideje változik-e, vagy tud-e alkalmazkodni a vállalat a megváltozott körülményekhez.



2.3.1.1-11 ábra: az ütemezendő tevékenységek és erőforrásaik



2.3.1.1-12 ábra: a megengedett megoldás



2.3.1.1-13 ábra: az optimális megoldás

Megjegyzések:

- Láthatjuk, hogy a feladatot visszavezettük az eredeti problémára. Hasonlóan az előző fejezetekben tárgyalt példákhoz: a kritikus úton lévő tevékenységeket zölddel, az optimalizálandó tevékenységeket fehérrel, a megengedett megoldás érdekében elmozgatott tevékenységeket pedig sárgával jelöltem. A már végrehajtott tevékenységeket kékkel, a folyamatban lévő tevékenységeket pedig narancssárgával jelöltem.
- Az (5,7)-es tevékenység még be nem fejezett részét, hasonlóan a látszat-erőforrásigényhez (okkersárgával jelöltem), nem optimalizáltuk, mivel a modellünkben egy tevékenység megszakítását nem engedélyeztük.
- Ha ezzel a megszorítással nem találunk megengedett megoldást, akkor a feladatnak (hasonlóan az előző fejezetben tárgyaltakhoz) nincs megengedett megoldása.

2.3.2 Tevékenységek erőforrás-szükségletének változása (ERALL-OPT/ON-LINE/EV, OPT-RALL/ON-LINE/VR)

Ebben az esetben a projekt végrehajtása közben változik az egyes tevékenységek erőforrás-szükséglete. Elképzelhető, hogy egy már megkezdett tevékenység erőforrásigénye is változik. Ezt az esetet is megvizsgáltam, mivel a legtöbb heurisztikus módszer, köztük az **ERALL-módszer** sem alkalmazható abban az esetben, ha egy tevékenység erőforrás-szükséglete időben változik [288-289], kivéve ha ez a változás egybeesik $v(t)$ -vel. Ha viszont létezik erre az esetre egy megengedett megoldás, akkor azt az algoritmusunk megtalálja azzal a

módosítással, hogy a töréspontokhoz az erőforrás-szükséglet változását is hozzávesszük.

[220-221, 223, 226]

A módszernek magyarul **ERALL-OPT/ON-LINE/EV** (on-line optimális erőforrás-allokáció erőforrásigény változása esetén), angolul **OPT-RALL/ON-LINE/VR** (On-line Optimized Resource Allocation with Variable Resources) nevet adtam.

A már megkezdett tevékenységeket szintén az előző fejezetben megtárgyalt módon látszaterőforrások módjára kezeljük. Nem mozgathatjuk el őket.

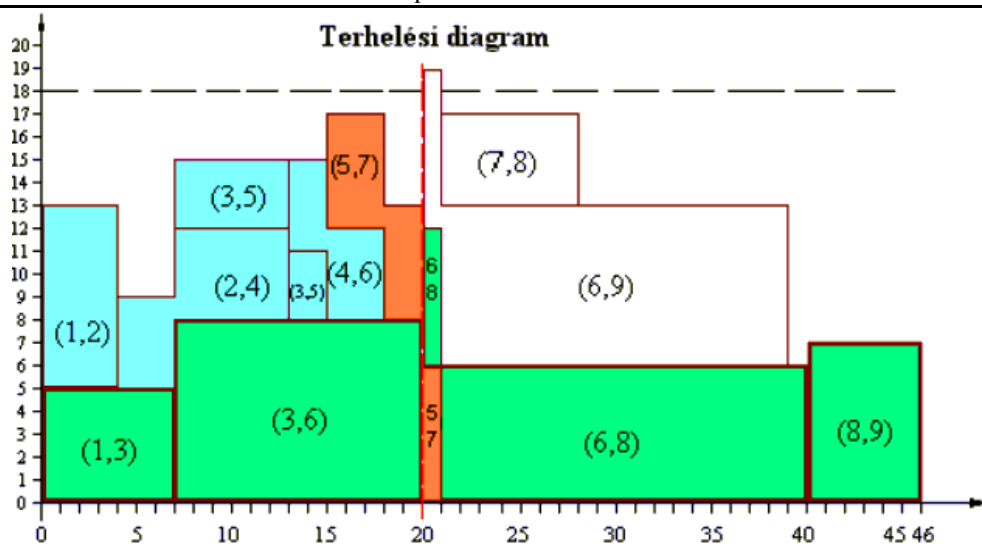
Állítás: A módosított **ERALL-OPT/ON-LINE/EV** megadja az optimális megoldást véges lépésben.

Bizonyítás: Lásd. 2.3.1. bizonyítást.

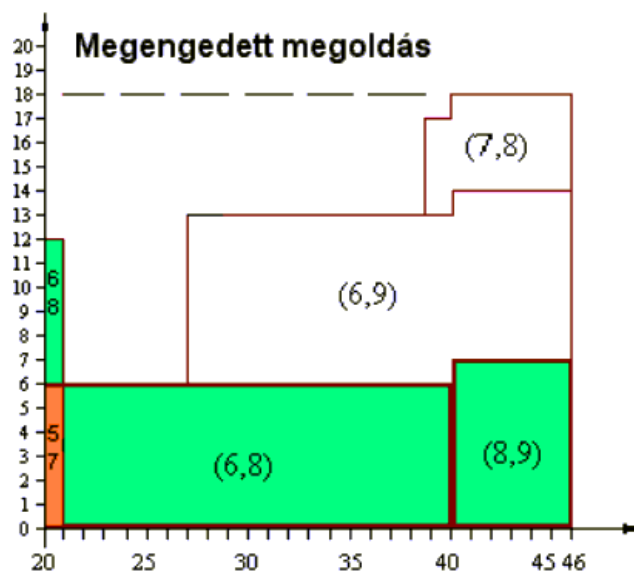
2.3.2.1 Példa

Induljunk ki szintén a 2.3.2.1-1 ábrán látható feladatból. $v(t)$ legyen ismét 20. Növekedjen meg (5,7) tevékenység erőforrás-szükséglete 1-el. (7,8) tevékenység erőforrás-szükséglete pedig növekedjen 2-vel. Erőforrás-szükséglet megváltozása legtöbbször akkor fordul elő, ha helytelenül mértük fel a tevékenységek erőforrásigényét, illetve valamilyen időjárási ok miatt az előzetesen kikalkulált erőforrásigény előreláthatólag kevésnek bizonyul.

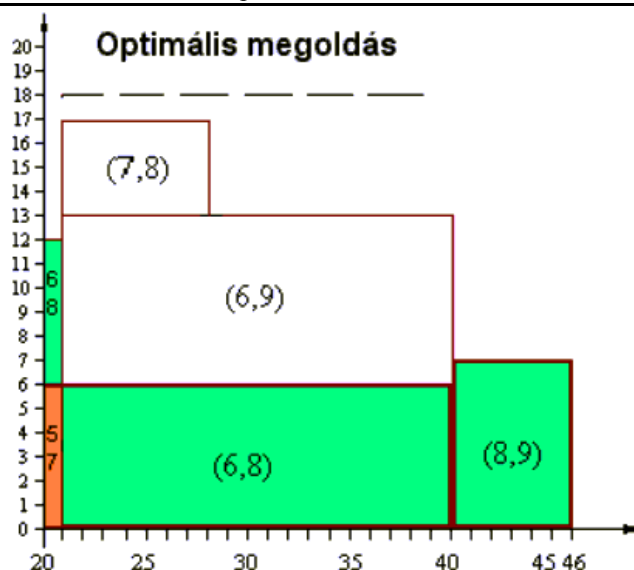
Különösen az építési projektek során a vállalatok sokszor úgynevezett tevékenységnormákat alkalmaznak, melyek standard tevékenységre (pl. betonozás, alap ásása stb.) tartalmazzák a szükséges tevékenység időtartamát és erőforrásigényét. Ettől az értéktől a vállalat az alkalmazott technológia, valamint a különböző külső körülmények (pl. időjárás) miatt eltérhet. Ha a vállalat folyamatosan figyeli a tevékenységek időtartamainak, erőforrásigényeinek betartását, akkor a tapasztalatokból nyert adatok alapján módosíthatja a tervét, ezáltal pontosabban megbecsülheti a még végrehajtásra váró tevékenységek erőforrás-szükségletét. Valamint arra a számára legfontosabb kérdésre is pontosabb választ tud adni, hogy vajon a megváltozott adatok figyelembevételével is be tudja-e határidőre fejezni a projektet.



2.3.2.1-1 ábra: terhelési diagram az erőforrásigény megváltozása után



2.3.2.1-2 ábra: megengedett megoldás



2.3.2.1-3 ábra: az optimális megoldás

Megjegyzések:

- Láthatjuk, hogy ebben az esetben is a módszer ugyanúgy használható. Ha olyan problémával találkozunk, ahol a két probléma kombináltan fordul elő, azaz mind az erőforráskorlát, mind pedig a rendelkezésre álló tevékenységek erőforrás-szükséglete is megváltozik, akkor a problémát ugyanígy kezelhetjük.
- Szintén elmondható, ha nem található megengedett megoldás, akkor a problémának nincs optimális megoldása sem.

2.3.3 Tevékenységekhez szükséges idő változása (on-line módon) (ERALL-OPT/ON-LINE/TLV, OPT-RALL/ON-LINE/VRT)

Az előző fejezetben már említettem, hogy a tevékenységek időtartamainak és erőforrás-szükségleteinek változása elsősorban abból adódik, hogy helytelenül mértük fel ezen paramétereket. A problémát az is nehezíti, hogy a változás során a kritikus út is megváltozhat. Ennek következtében egyrészt az átfutási idő is változhat, másrészt megváltoznak a tevékenységek tartalékidői. Ebből adódóan olyan tevékenységek időbeni megvalósulására is fokozott figyelmet kell fordítani, amelyek eddig az alternatív utakon helyezkedtek el, és esetleg jelentős tartalékidővel rendelkeztek.

Ezért tehát joggal tekinthetjük ezt a problémát az on-line ütemezés legnehezebb problémájának. Ez azért nehezebb feladat, mint az előzőek, mert ekkor egy újabb hálót kell létrehoznunk, amennyiben a változás nagyobb, mint a rendelkezésre álló tartalékidő. Elképzelhető az is, hogy az eddig kritikus úton lévő tevékenységek esetleg alternatív útra

kerülhetnek, s fordítva. Ezért első lépésként fel kell rajzolnunk egy új hálót. Ebből a hálóból az erőforrásigényt tartalmazó táblázat segítségével egy új terhelési diagram rajzolható. Erre egy heurisztikus módszert (pl. **ERALL-módszert**) alkalmazva ismét egy megengedett megoldáshoz jutunk. Ezt optimalizálva juthatunk ismét optimális megoldáshoz. [220-221, 223, 226]

A módszernek magyarul **ERALL-OPT/ON-LINE/TLV** (on-line optimális erőforrás-allokáció tevékenység időtartamainak változása esetén), angolul **OPT-RALL/ON-LINE/VDT** (On-line Optimized Resource Allocation with Variable Duration Time) nevet adtam.

Definíció: Az olyan eseményt, melyet az on-line ütemezés során, a háló tulajdonságainak fenntartása érdekében vezettünk be, *látszateseménynek* nevezzük.

Megjegyzés: Az ilyen eseménynek a szerepe csupán annyi, hogy megőrizzük a háló tulajdonságait. Amennyiben az ütemezés során tevékenység-csomópontú hálóval dolgoztunk, úgy látszatesemény helyett látszattevékenységet kell bevezetnünk a háló definíciójának megtartása érdekében. Ha nem vezetnénk be látszateseményt (vagy látszattevékenységet), akkor nem biztos, hogy teljesülne a hálónak az a tulajdonsága, hogy csak egy kezdő eseménye (tevékenység-csomópontú hálónál egy kezdő tevékenysége), kezdőpontja lehet. Ezt a (látszat)kezdőpontot a folyamatban lévő tevékenységekkel, illetve az éppen induló tevékenységekkel össze kell kötni látszattevékenységekkel, szintén a háló tulajdonságainak fenntartása érdekében. Az itt alkalmazott látszattevékenységeknek azonban van átfutási idejük: $\nu(t)$.

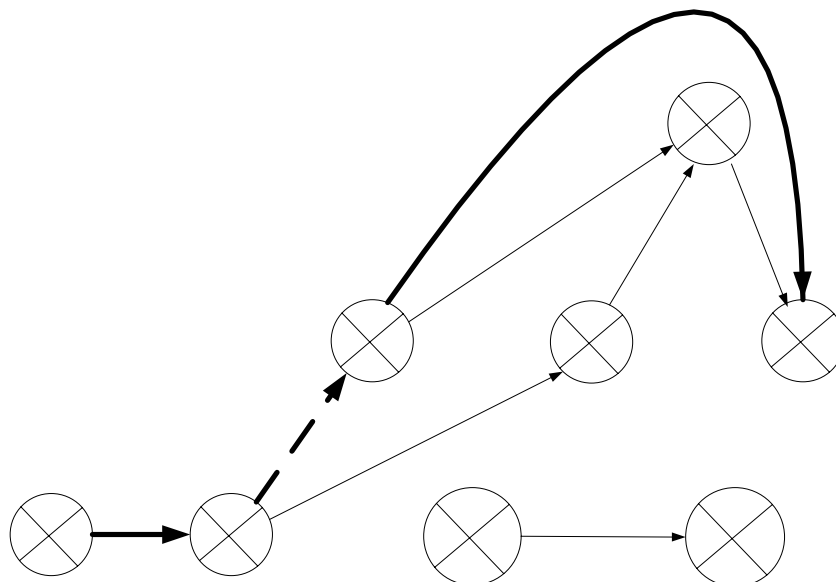
Állítás: A módosított **ERALL-OPT/ON-LINE/TLV** megadja az optimális megoldást véges lépésben.

Bizonyítás: Amennyiben bármely tevékenység rendelkezésére álló tartalékideje kisebb, mint a változás mértéke, úgy nem kell új gráfot felrajzolni. Ellenkező esetben fel kell rajzolni a végre nem hajtott, illetve folyamatban lévő tevékenységekre egy új gráfot úgy, hogy egy látszat-kezdőeseményt (tevékenység-csomópontú hálónál látszat-kezdőtevékenység) vezetünk be. Ezután a megengedett megoldás megkeresése után használjuk az **ERALL-OPT/VK** módszert.

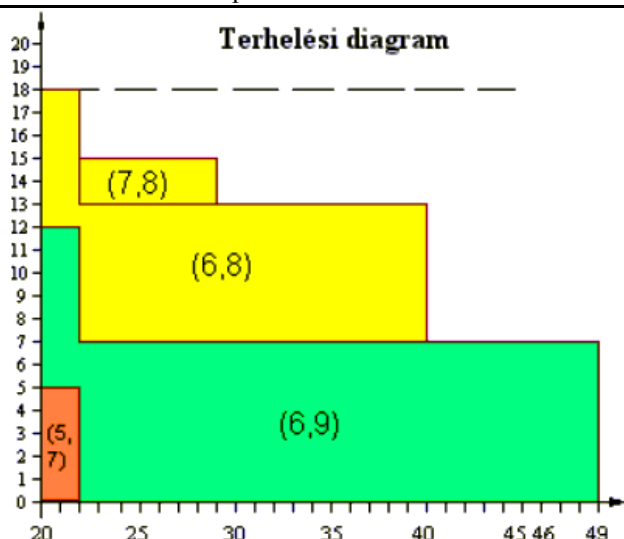
Megjegyzés: Az on-line algoritmusokat kombinálhatjuk, vagyis alkalmazhatók olyan feladatokra is a módszerek kombinációja, melynél valamennyi paraméter megváltozik.

2.3.3.1 Példa

Szintén induljunk ki a 2.3.1.1-9 ábra alapján felvázolt problémából az egyszerűség kedvéért. Legyen $\nu(t)$ ismét 20. Növekedjen (5,7) tevékenység ideje 1 egységgel, (6,9) 10 egységgel. Vegyünk fel egy 0. látszateseményt, hogy a háló definícióját biztosítani tudjuk. Ennek erőforrásigénye legyen 0 (bármennyi lehetne, hiszen csak $\nu(t)$ -től optimalizálunk), hossza pedig $\nu(t)$. Nyilván a $\nu(t)$ időpont előtt elkezdődött, még folyamatban lévő tevékenységeket az előzőekhez hasonlóan kezeljük.



2.3.3.1-1 ábra: a CPM-háló tevékenységek időtartamainak megváltozása után



2.3.3.1-2 ábra: a terhelési diagram

Megjegyzések:

- Ebben a példában a gráf utáni reprezentáció egyben az optimális megoldást is szolgáltatta. Ez általában persze nem igaz. Mégis minden esetben ilyen ütemezés esetén, ahol a tevékenységekhez szükséges idők változnak, először egy új „segédgráfot” kell felrajzolnunk, amennyiben a változás nagyobb, mint a tevékenység rendelkezésre álló tartalékidő. Ezután (ha szükséges) először megengedett megoldást kell keresni, majd pedig ezt optimalizálni.
- Az olyan on-line ütemezéseknél, ahol egyszerre több paraméter változik (erőforráskorlát, tevékenység(ek) erőforrásigénye, tevékenység(ek)hez szükséges idő), azt, hogy fel kell-e rajzolni egy újabb gráfot, az dönti el, hogy van-e olyan tevékenység, amelyhez a szükséges idő megváltozott. Ha igen, akkor valószínűleg fel kell rajzolni.
- Nem kell felrajzolni a gráfot, ha egy tevékenységhez szükséges idő úgy változott meg, hogy erre a változásra a rendelkezésre álló tartalékidők fedezetet nyújtanak.
- Ezek az optimális megoldások $v(t)$ függőek, hiszen $v(t)$ időpont előtt lezajlott tevékenységekkel már nem foglalkozunk, nem optimalizáljuk, az algoritmusunk során nem vesszük őket figyelembe.

2.3.4 Tevékenységek megszakíthatósága (ERALL-OPT/MSZT, OPT-RALL/IA)

A gyakorlatban vannak olyan tevékenységek, melyeket megszakíthatunk, és vannak olyan tevékenységek, amelyeket nem. A megszakítható tevékenységekre is igaz, hogy egy bizonyos

ideig (pl. a technológia miatt) nem szakíthatók meg. Továbbá értelmezhető egy olyan (maximális) idő, amely elteltével az adott tevékenységet mindenképpen folytatni kell. Ha egy tevékenység esetében megengedjük a megszakíthatóságot, akkor az pl. **ERALL-algoritmus** erre az esetre – egy megengedett megoldás keresésére – nem használható.

Az irodalmi áttekintésben említettem a soros illetve párhuzamos allokációkat. Ezek mindegyike használható ebben az esetben egy megengedett megoldás keresésére. Bár ezek az optimumkereső algoritmusok csak heurisztikusak. [202, 213]

Ha ebből a megengedett megoldásból indulunk ki úgy, hogy a megszakított tevékenységeket külön tevékenységekként kezeljük azzal a megszorítással, hogy azt a maximális időt, amely e két megszakított tevékenység között van, ne lépje túl az előírtat, akkor egy optimális megoldást kaphatunk. Vagyis be kell vezetnünk egy $t_{f(k,l,l,m)}$ időt, mely megadja, hogy két megszakított résztevékenység $((k,l), (l,m))$ mennyi idő után követi egymást. Ez egy optimalizálási lépésben nem lépheti túl az előírt értéket. [220-221, 223, 226]

A módszernek magyarul **ERALL-OPT/MSZT** (optimális erőforrás-allokáció tevékenységek megszakíthatósága esetén), angolul **OPT-RALL/IA** (Optimized Resource Allocation with Interruptable Activities) nevet adtam.

Definíció: Egy megszakított tevékenység egyik résztevékenységének befejezése, illetve követő résztevékenységének kezdése között eltelhető maximális időt nevezzük *maximális megszakítási időnek*.

Állítás: A módosított **ERALL-OPT/MSZT** megadja az optimális megoldást véges lépésben.

Bizonyítás: Tegyük fel, hogy létezik megengedett megoldás. Legyen M a megszakítható tevékenységek halmaza. Legyen $M_{(i,j)}$ egy megszakítható tevékenység résztevékenységeinek halmaza, ahol $(i,j) \in M$. Legyen $z_{(k,l)}$ egy adott résztevékenység kezdési ideje, $d_{(k,l)}$ a időtartama, ahol $(k,l) \in M_{(i,j)}$, $(i,j) \in M$. Két résztevékenység közötti maximális megszakítási idő legyen $t_{fmax(k,l,l,m)}$, ahol $(k,l), (l,m) \in M_{(i,j)}$, $(i,j) \in M$, és legyen $t_{f(k,l,l,m)}$, ahol $(k,l), (l,m) \in M_{(i,j)}$, $(i,j) \in M$ két résztevékenység között eltelt idő. Bármely előző fejezetekben tárgyalt optimumkeresési eljárást használhatjuk, az egyes megoldásoknál azonban figyelembe kell venni, hogy $t_{f(k,l,l,m)} \leq t_{fmax(k,l,l,m)}$. A megadott megoldáshalmazból

kiválasztjuk az olyan megoldás(oka)t, amelyek a fenti egyenlőséget kielégítik $\forall (k,l), (l,m) \in M_{(i,j)}, \forall (i,j) \in M$ esetén.

2.4 Költség-, idő-, erőforrás-optimalás egyidejű megvalósítása (ERALL-OPT/KLTG, OPT-RALL/COST)

Az eddigi fejezetekben ütemezéssel és erőforrás-optimalással foglalkoztam. A menedzsment számára azonban a költségek figyelembevétele, esetleges csökkentése is fontos szempont lehet. Egy projekt megvalósítása során – az irodalmi áttekintésben az ISO 8402-es szabvány projektre vonatkozó definíciójának megfelelően – az időtartamok és erőforrás-szükségletek megfelelő kezelése mellett a költségek minimális szinten tartása is fontos célunk. Amennyiben a költségeket is figyelembe vesszük az optimalizálás során, számos cél(függvény)t fogalmazhatunk meg. Ilyen feladat lehet a költségek felmerülésének kiegyenlítése, a projekt végrehajtása lehető legrövidebb idő alatt a legkisebb (változó) költségnövekménnyel, optimális erőforrás-kihasználás mellett, illetve lehető legkisebb összköltség meghatározása. Az első esetben a költségeket mint (speciális) erőforrást kell kezelnünk. A második és harmadik esetet vizsgálom meg részletesen ebben a fejezetben. Láthatjuk majd, hogy ebben az esetben valamennyi előzőekben tárgyalt módszerre szükség lehet. Ezeket integráltan kell használni a cél elérése érdekében. [220-221, 223, 226]

Először meghatározzuk a **CPM-módszer** (vagy **MPM-módszer**) segítségével a tevékenységek legkorábbi kezdési idejét. Ezután egy **CPM/COST-** (vagy **MPM/COST-**) módszerrel meghatározzuk, hogy minimális költségnövekménnyel mennyivel lehet gyorsítani a tevékenységek időtartamait. Ennek a módszernek számos (számunkra) előnyös tulajdonságát kihasználjuk. Kihasználjuk például, hogy adott költségkorlát esetén is (a költségkorlát figyelembevételével) költségoptimalis megoldást szolgáltat a változó költségekre nézve. Meghatározzuk a (változó) költség – időcsökkenés, erőforrásigény – időcsökkenés függését.

Az időrövidítés során ugyanis a menedzsment a következő problémával kerül szembe: Ha csökkenti a program átfutási idejét, akkor növekszik a tevékenységek változó költség-igénye. Elmondható azonban az is, hogy a program fixköltsége viszont a program rövidítésével szintén csökken. Ebből viszont következik, hogy konvex költségfüggvények esetén az összköltség minimalizálható, vagyis található egy olyan program, melynek

összköltségigénye minimális. Ezáltal a vezetés mind a projekt átfutási idejét, mind pedig az összköltségét csökkentheti.

Egy **CPM/COST-módszerrel** szolgáltatott hálót tekintünk kiindulási állapotnak. Ezután egy heurisztikus módszerrel megengedett megoldást keresünk (pl. alkalmazhatjuk az **ERALL**-módszert, soros vagy párhuzamos allokációt). Ha van olyan megengedett megoldás, melynél a módszer alkalmazása után nem változott a teljes projektidő, akkor alkalmazzuk az **ERALL-OPT** optimalizálást (2. fejezet). Megvizsgálunk olyan eseteket is, amikor e három kritérium (minimális átfutási idő, minimális összköltség, optimális erőforrásfelhasználás) csak kompromisszumokkal elégíthető ki egyszerre. [220-221, 223, 226]

A módszernek magyarul **ERALL-OPT/KLTG** (optimális erőforrás-allokáció költségek figyelembevételével), angolul **OPT-RALL/COST** (Optimized Cost and Resource Allocation) nevet adtam.

2.4.1 A feladat megfogalmazása

Ahogy az az előző fejezetben már említettem, a három kritérium:

1. minimális összköltség elérése,
2. minimális átfutási idő elérése, valamint
3. az optimális erőforrás-felhasználás biztosítása

legtöbb esetben kompromisszumok nélkül nem elégíthető ki egyszerre.

Tegyük fel, hogy egy projektet, melynél a tevékenységeket, a tevékenységekhez szükséges időt, költséget, erőforrásigényt már meghatározták. Adott minden tevékenységre vonatkozóan, hogy ha az adott tevékenységet hamarabb szeretnénk befejezni, akkor az milyen költség- és erőforrás-növekménnyel jár.

Ebben a helyzetben két feladatot is meghatározhatunk: az első szerint egy olyan programot kell meghatároznunk, mely a minimális költségnövekménnyel, a legrövidebb idő alatt, maximális párhuzamosítás mellett beütemezi a tevékenységeket a legkorábbi időpontra úgy, hogy egy adott erőforráskorlátot ne lépjen túl. Ez a program tulajdonképpen egy minimális átfutási idővel rendelkező projekt. Ennél rövidebb idő alatt nem lehet végrehajtani a programot. Ezenkívül az is elmondható, hogy nem található olyan projekt, mely kisebb összköltséggel ugyanennyi idő alatt végrehajtható.

A másik feladat a minimális összköltség meghatározása. Ebben az esetben olyan programot kell meghatározni, melyben az összköltség minimális.

A módszer alkalmazásához a következő feltételezésekkel élünk:

1. A költség-idő függvények konvexek.
2. A fixköltségek (az idő előrehaladtával) monoton nőnek.
3. A változóköltség-idő függvény minimuma a normál átfutási idejű projekt esetén van.

Ezek a megszorítások a legtöbb projekt esetén teljesülnek. Az első megszorítás szerint olyan költségfüggvényekkel dolgozunk, melyek konvexek. Nem konvex költségfüggvények esetén a költségminimumok meghatározása igen nehéz feladat. Ilyen esetekben legtöbbször genetikus algoritmusokat alkalmaznak a függvényminimum megtalálásának céljából. Ez a módszer bármilyen gyors is, nem szolgáltat minimumot. Speciális esetekben mód van arra, hogy algoritmikus módszerekkel is belátható időn belül eredményt szolgáltatathassunk. Ilyen speciális eset, amikor a feladat szeparálható, jelen esetben ez azt jelenti, hogy a tevékenységek változóköltség-igénye adja meg az összes változó költséget, valamint a változó költségek és fixköltségek összege adja meg az összköltséget. Ezek a költségfüggvények egymástól függetlenül vizsgálhatók. Az összköltség ezen költségek összege lesz. Ha ez teljesül, akkor lehetőség van a költségfüggvények szeparálására. A konvex függvények minimumát hagyományos módon, a nem konvex függvények minimumát egyre kisebb intervallumon vett konvex burkuk segítségével lehet meghatározni.

A módszer szempontjából nem lényeges, hogy egy megengedett erőforrásallokációt milyen módszerrel találunk meg. Ezért a továbbiakban csak azt feltételezzük, hogy egy ilyen megengedett megoldást meg lehet találni. Ha nincs megengedett megoldás, akkor ebben az esetben sincs optimális megoldás.

Az egyszerűség kedvéért most költségoptimalis erőforrásallokáció esetén a módszer végigkövetése céljából **CPM/COST-módszert** alkalmazunk az ütemezési fázis végrehajtására. Már most meg kell azonban jegyezni, hogy bármely más minimális összköltségű, illetve minimális átfutási idejű programot meghatározó költségoptimalizáló módszert alkalmazhattunk volna.

A **CPM/COST-módszernél** fel kell tennünk, hogy az előzőekben tárgyalt három megszorítás (költségfüggvények konvexitása, fixköltségfüggvény monoton növekedése, valamint az, hogy a változóköltség-idő függvény minimum helye a normál átfutási idejű projekt esetén van) teljesül.

Ha ezek a feltételek teljesülnek, akkor számos további következtetést is levonhatunk.

1. A minimális összköltségű program a normál és a minimális átfutási idejű projekt között van.

2. Ha a költségfüggvények szigorúan konvexek, akkor csak egy összköltség-minimumhely van.

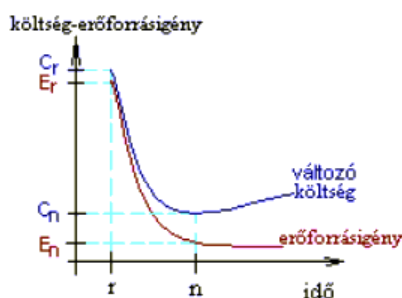
A három kritérium (lehető legrövidebb átfutási idő, lehető legkisebb összköltség és optimális erőforrásfelhasználás mellett történő megvalósítás) nem mindig elégíthető ki egyszerre kompromisszumok nélkül. A továbbiakban vizsgáljuk meg a (változó)költség-növekmény – időtartam csökkentése, valamint az erőforrásigény és az időtartam csökkentése közötti kapcsolatot. Az alábbiakban definiáljuk néhány fogalmat, melyet a költségtervezés kapcsán a **CPM/COST**-, **MPM/COST**-**módszerek**nél (is) alkalmaznak.

Normál (időtartam) idő: Az az időmennyiség, amely a tevékenység normál/tervszerű végrehajtásához szükséges.

Megjegyzés: A továbbiakban feltesszük, hogy a tevékenység normál időtartama esetén legkisebb a (változó)költség-igénye.

Minimális vagy rohamidő: Az a legkisebb időmennyiség, amely alatt a tevékenységet végre lehet hajtani.

Elmondható, hogy a (változó) költségek és az időtartamok között általában fordított arányosság figyelhető meg. [372]



2.4.1-14 ábra: változó költség-idő, erőforrásigény-idő kapcsolata

Az ábrán látható, hogy a rohamidőnél (r) rövidebb idő alatt nincs értelme a költséget és az erőforrásigényt vizsgálni, hiszen ennél kevesebb idő alatt a tevékenységet nem lehet végrehajtani, így ezek a görbék ilyen időpontokra nincsenek is értelmezve. A tevékenység erőforrásigénye (attól függ persze, hogy mit tekintünk erőforrásigénynek) fordítottan arányos

az időtartamokkal (például, ha az erőforrásigény a munkaerő, akkor több idő alatt kevesebb, míg kevesebb idő alatt több munkás tudja elvégezni ugyanazt a munkát).

Később látni fogjuk, hogy az algoritmus szempontjából teljesen mindegy, hogy az idő és az erőforrásigény között milyen függvénykapcsolat van. Egy fontos követelmény van csupán, mégpedig az, hogy bármely (diszkrét) időpontban (de legalább a normál idő (n) és a rohamidő (r) közötti időintervallumban) meg tudjuk mondani az adott időhöz tartozó erőforrásigényt.

Az előző pontban a költségfüggvényekre szorosabb kritériumokat tettünk. A (változó) költségek általában a normál idő (n) és a rohamidő (r) időintervallum között fordítottan arányosak a tevékenység időtartamával. Viszont, ha egy tevékenység időtartama a tervezettnél tovább tart, akkor költségnövekménnyel (is) járhat. Mégis látni fogjuk, hogy mi csak a normál és a rohamidő közötti intervallumot tekintjük, hiszen ezen az időintervallumon belül optimalizálunk (ugyanis az előző pontban leírt költségfüggvényekre vonatkozó kritériumokból következik, hogy csak ezen intervallumon belül lehet összköltségminimális program). Ezen az időintervallumon belül megkívánjuk a függvénytől, hogy szigorúan monoton csökkenő legyen. (Ez a **CPM/COST-módszer** alkalmazhatósága miatt szükséges.) Továbbá mindkét függvénytől elvárjuk, hogy a vizsgált pontokban értelmezve legyenek.

2.4.2 Az algoritmus leírása

A továbbiakban a fenti feltételeknek megfelelő erőforrásigény és (változó)költség függvényekkel foglalkozunk. Először a legkorábbi (normál) időre vonatkozó ütemezést végezzük el. Ez a módszer megadja a tevékenységek maximális párhuzamosítása melletti legkorábbi kezdési időpontokat. Ezután költségoptimalizálást végzünk. Ilyen a példában szereplő **CPM/COST-módszer** is. A módszer lépéseit egy táblázatba összefoglaljuk. Ha költségoptimalizálásra a **CPM/COST-módszert** alkalmazzuk, akkor ezen módszer néhány speciális tulajdonságát is kihasználhatjuk.

Egyik fontos, számunkra nagyon kedvező tulajdonsága ennek a módszernek, hogy ez egy mohó algoritmus. Minden lépésben a legkisebb költségnövekedéssel járó kritikus úton lévő tevékenység(ek) időtartamát csökkenti. Tehát, ha az összes (változó) költségre megszabnánk egy korlátot, akkor ez a módszer az ennek megfelelő legjobb megoldást szolgáltatja. Ekkor a módszer lépéseit csak addig kell végrehajtani, ameddig a változó költségek összege kisebb vagy egyenlő, mint ez a korlát. A módszer csak a kritikus úton lévő tevékenységeket csökkenti úgy, hogy figyel arra, hogy egy nem kritikus út a kritikus úton

lévő tevékenységek időtartamainak csökkentésével már kritikus úttá válhat. Továbbá egy lépésben nem lehet többet csökkenteni az időtartam(ok)on, mint a nem kritikus úton lévő tevékenységek tartalékideje.

Ha egy **CPM/COST-módszerrel** meghatároztuk a lépéseket, akkor meghatározzuk a terhelési diagramot. Ha van olyan időpillanat, ahol az összes erőforrásigény nagyobb, mint az erőforráskorlát, akkor a megfelelő erőforrásadatokkal megengedett megoldást keresünk. Ha valamely megengedett megoldást kereső algoritmussal (mint pl. az **ERALL-módszerrel**) találtunk olyan termelési programot, amelyben a kritikus utat nem kellett megváltoztatni, vagyis a kritikus út hossza nem nőtt a megengedett megoldást kereső módszer alkalmazása után, akkor az általunk kifejlesztett **ERALL-OPT-algoritmust** futtatjuk le. Ekkor a feladat költség-, idő- és erőforrás-optimális lesz egyben. [220-221, 223, 226]

Nevezzük a továbbiakban az olyan megengedett megoldásokat — melyeknél a megengedett megoldás keresése során a módszer a kritikus út hosszát megváltoztatta — kritikus megoldásnak, valamint az olyan megengedett megoldásokat, melyeknél a kritikus út hossza nem változik, *nemkritikus (megengedett) megoldásnak*.

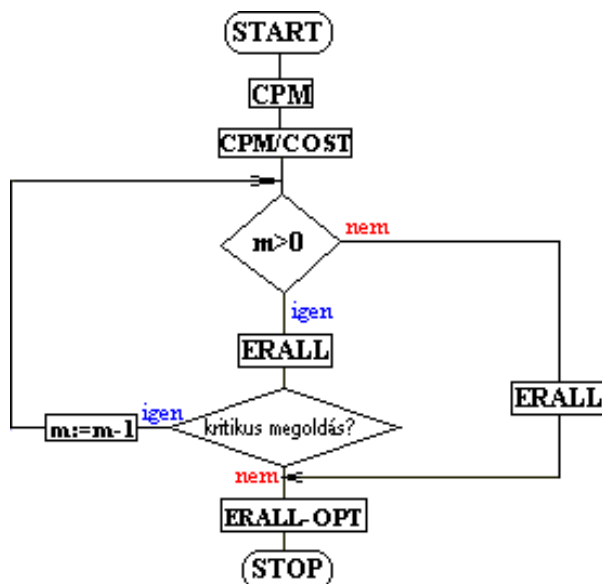
Ha nincs olyan nemkritikus megengedett megoldás, akkor a **CPM/COST-módszer** által szolgáltatott lépések közül az utolsó előtti végezzük el a fenti vizsgálatokat. Ha egy lépésben több időegységnyit javítottunk a projekt átfutási idején, akkor ezekre a közbenső állapotokra is el kell végezni a fenti vizsgálatokat, mivel elképzelhető, hogy ebben az esetben találunk egy nemkritikus megoldást. Ha a **CPM/COST-módszer** egyetlen lépése sem szolgáltat nemkritikus megoldást (valamint a közbenső lépéseknél sem található ilyen megoldás), akkor azt mondjuk, hogy a teljes projekt időtartamát az adott erőforráskorlát mellett nem lehet lerövidíteni. Ekkor visszaérkeztünk a második fejezetben tárgyalt esethez, vagyis először egy megengedett megoldást kereső heurisztikus algoritmust, majd egy **ERALL-OPT-módszert** kell végrehajtanunk a feladaton.

Meg kell továbbá jegyezni, hogy az előző fejezetekben tárgyalt esetekre alkalmazott algoritmusokat az **ERALL-OPT-algoritmus** helyett itt is alkalmazhatjuk. Tehát, ha az erőforráskorlát nem konstans függvény, akkor használható az **ERALL-OPT/VK**, on-line ütemezésnél az **ERALL-OPT/ONLINE/VK** és így tovább.

Az algoritmus menetét az alábbi folyamatra is reprezentálja (ahol *m* a **CPM/COST** lépésinek száma).

Definíció: Az olyan megengedett megoldást, ahol a projekt átfutási ideje (*TPT*) megegyezik a maximálisan párhuzamosított megoldásban lévő kritikus út hosszával, *nemkritikus megoldásnak* nevezzük.

Definíció: Az olyan megengedett megoldást, ahol a projekt átfutási ideje (*TPT*) nagyobb, mint a maximálisan párhuzamosított megoldásban lévő kritikus út hossza, *kritikus megoldásnak* nevezzük.



2.4.2-1 ábra: az ERALL-OPT/COST algoritmus folyamatábrája

Állítás: A módosított **ERALL-OPT/KLTG** megadja az optimális megoldást véges lépésben.

Bizonyítás: Amennyiben a költségoptimalizálás (pl. a **CPM/COST-módszer**) lefutása után létezik nemkritikus megoldás, akkor valamennyi (általam készített) optimáló eljárással optimális megoldás adható. Amennyiben ez a megoldás kritikus, akkor ezen idő alatt, ilyen költség szint mellett nem lehet optimális erőforrás-allokációt meghatározni, mivel nincs olyan megengedett megoldás, mely ütemtervét ezen idő alatt végre lehetne hajtani. Kihasználjuk a következőt: csak ott lehet költség-növekmény-optimális megoldást találni, amely rajta van a **CPM/COST** által megadott költség-növekmény-idő függvényen. Ezért vizsgáljuk meg valamennyi diszkrét időpontra, hogy található-e nemkritikus megoldás. Amennyiben található, ezek közül a legrövidebb átfutási időt meghatározó ütemtervre végezzünk optimálást valamely optimáló algoritmusunkkal. Amennyiben a költség-növekmény-idő függvény egyetlen

diszkrét pontjára sem határozható meg nemkritikus megoldás, akkor azt mondjuk, hogy a projekt átfutási idejét nem rövidíthetjük úgy, hogy az egyszerre legyen erőforrásigény-, költségnövekmény- és időoptimális. Ekkor az eredeti feladatra végezzük el a megengedett megoldás keresését, majd ha létezik megengedett megoldás, akkor az optimalizálást. Minimális összköltség meghatározása esetén is hasonlóan járunk el. A költségoptimalizálás során kiválasztjuk azt a megoldást, melyre az összköltségek minimálisak. Ha van nemkritikus megengedett megoldás, akkor van optimális megoldás is, melyeket el lehet véges lépésben érni (2. fejezet szerint). Ha egyetlen pontban sem található összköltségminimális megoldás, akkor az előzőekhez hasonlóan visszajutottunk a 2.1 fejezetben tárgyalt problémához: normál átfutási idejű projektre kell megengedett, majd pedig optimális megoldást találni. Ha nincs megengedett megoldás, akkor optimális megoldás sem található.

2.4.3 Példa

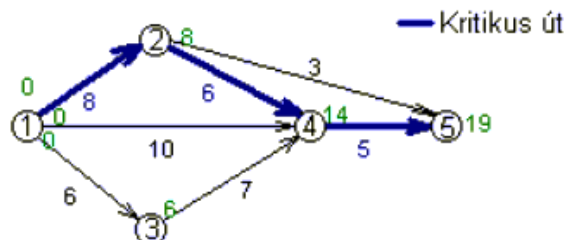
A következőkben az egyszerűsítés kedvéért a tevékenységek esetében az erőforrásigény- és költségnövekmény legyen azonos egységnyi időtartam-csökkenésre. Az adatokat az alábbi táblázat foglalja össze.:

Tevékenység	Normál idő ($t_{ni,j}$)	Rohamidő ($t_{ri,j}$)	Normál költség ($C_{ni,j}$)	Egységnyi költség-növekedés ($\Delta C_{i,j}$)	Normál erőforrásigény ($R_{ni,j}$)	Egységnyi erőforrás-igény növekedése ($\Delta R_{i,j}$)
(1,2)	8	6	4500	100	3	0,5
(1,3)	6	5	12000	200	2	1
(1,4)	10	7	10500	150	2	1
(2,4)	6	4	7500	300	4	1
(2,5)	3	3	3500	-	2	-
(3,4)	7	6	1500	500	3	1
(4,5)	5	4	2500	650	1	1
Összesítés			42000	-	-	-

2.4.3-1 táblázat

Az erőforrás legyen a munkaerő, az erőforráskorlát pedig 10 fő. A költségek eFt-ban értendők.

Az adatokból megszerkeszthető a **CPM-háló**, valamint felírhatók a **CPM/COST-módszer** lépései.



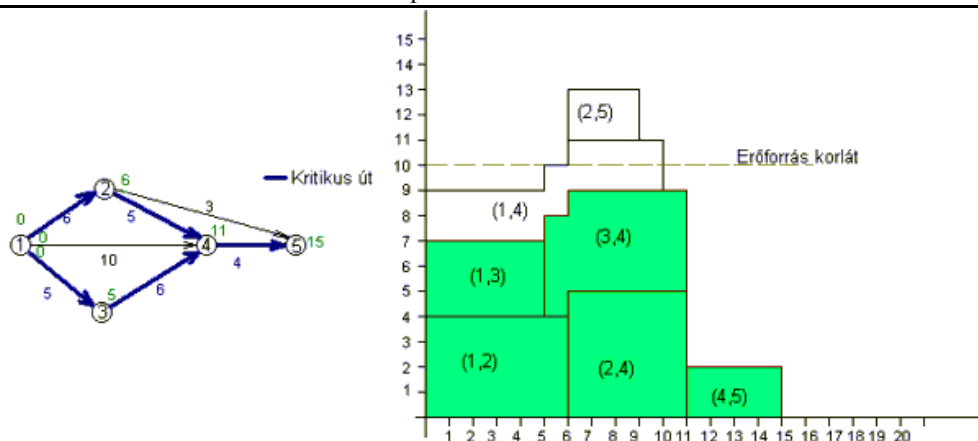
2.4.3-2 ábra: CPM-háló

A **CPM/COST-módszer** lépései

Lépé- sek száma	Csökken- tett tevé- kenységek	Csök- kentés	Összes költség	Egységnyi költség- növekedés	Összes költség- növekedés	Teljes pro- jektidő
0	-	-	42000	-	-	19
1	6	1	42100	100	100	18
2	10	1	42400	100+200=300	300	17
3	6	1	43050	650	650	16
4	3	1	43850	500+300=800	800	15

2.4.3-2 táblázat

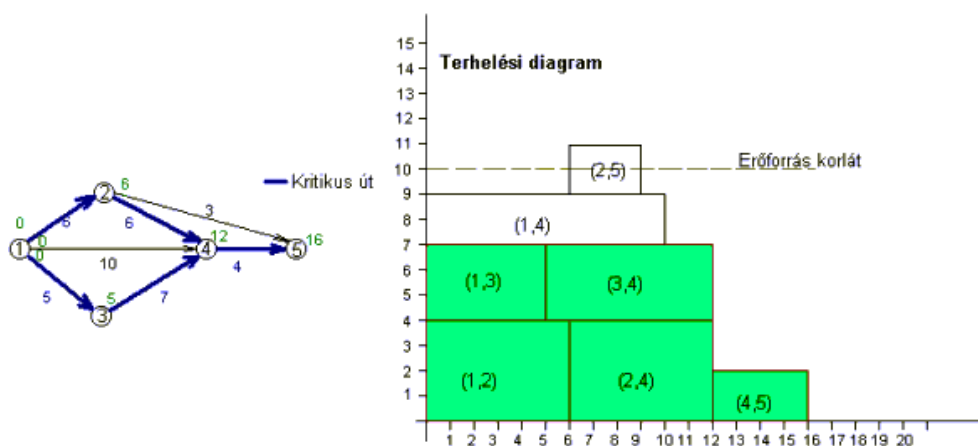
Az utolsó (negyedik) lépés alapján meghatározhatjuk a hálót, és a hozzá tartozó terhelési diagrammot.



2.4.3-3 ábra: a CPM-háló és a hozzá tartozó terhelési diagram a CPM/COST-módszer utolsó lépésére

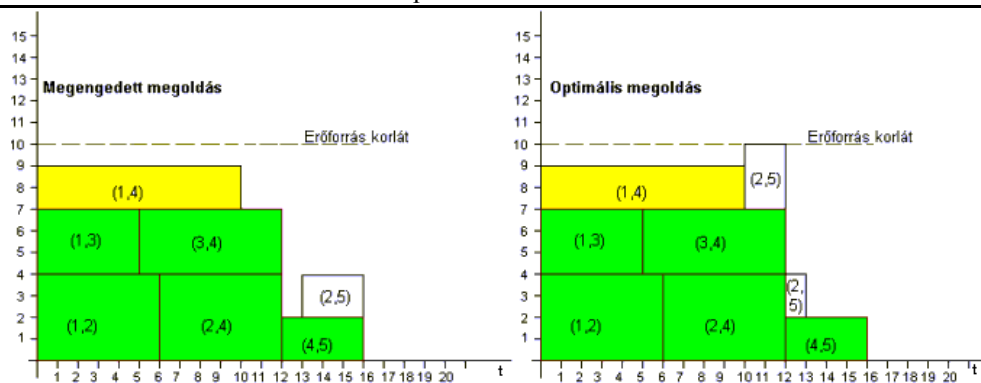
Látható az erőforrásterhelési diagram alapján, hogy nemkritikus megoldás nem található, ezért egy lépéssel vissza kell lépni a **CPM/COST** táblázatban. Erre a lépésre is meghatározható egy háló, és a hozzá tartozó terhelési diagram.

Tekintettel arra, hogy a harmadik lépés után csak egy időegységgel javult az átfutási idő, így nem kell közbenső időegységekre megvizsgálni a terhelési diagramot.



2.4.3-4 ábra: a CPM-háló és a hozzá tartozó terhelési diagram a CPM/COST-módszer utolsó előtti lépésére

Ebben az esetben már található nemkritikus megengedett megoldás. Pl. az **ERALL-algoritmus** alkalmazása után az **ERALL-OPT-algoritmussal** optimális megoldást kapunk az adott feltételrendszerre.



2.4.3-5 ábra: megengedett illetve optimális megoldás

Az ábrákból leolvasható, hogy a projekt összes (változó) költsége 43050 eFt, 10 munkással a program 16 nap alatt végrehajtható. Az egyes tevékenységek kezdési ideje leolvasható az optimális erőforrás-allokáció terhelési diagramjából.

Amennyiben a költségekre is meghatározunk egy korlátot, akkor a **CPM/COST-módszert** csak addig kell folytatni, ameddig el nem érjük ezt a korlátot, utána az algoritmust ugyanígy folytatjuk tovább.

2.5 Költségcsökkentés alternatív megoldások segítségével

A projektek tervezése és megvalósítása során nagyon sokszor előfordul, hogy az előzetes kalkuláció után a minimális összköltséggel megvalósítandó optimális erőforrás-allokációjú projektet sem lehet azon az áron megvalósítani, amelyen a tendert kiíró megvalósítani szeretné. [229]

Ezzel a problémával a gyakorlatban sokszor találkozhatunk, különösen építési, illetve beruházási projektek esetén. Erre a problémára egy más megközelítésű megoldást láthatunk Dr. Papp Ottó: Projektmenedzsment a gyakorlatban című könyvének 15. fejezetében: „A projektköltségek csökkentése értékelemzéssel – Építési vállalkozáson illetve szervezetfejlesztési példán bemutatva” című esettanulmányban. [310]

A módszer segítségével meghatározott maximális költségvetés esetén meg lehet határozni a lehető legjobb minőséggel végrehajtható projekt ütemtervét, költségigényét és erőforrás-szükségletét.

2.5.1 Alternatív megoldások

Ha az előzetes kalkulációk során kiszámított minimális összköltségre vonatkozó optimális erőforrás-allokációs megoldás esetén a költségek magasabbak, mint a program megvalósítására szánt költségvetés, akkor három lehetőségünk van: vagy lemondunk a projekt megvalósításáról, vagy veszteséggel, de megvalósítjuk a projektet, vagy pedig az egyes megvalósítandó tevékenységeket – költségcsökkentés céljából – más tevékenységekkel váltjuk ki. Az első választási lehetőséggel az elemzés során most a továbbiakban nem foglalkozunk, hiszen ebben az esetben elesünk az üzlettől, és itt nincs értelme tovább optimális erőforrás-tervezést készíteni. A második választási lehetőséget néha akkor is „bevállalják”, ha az átmeneti veszteség ellenére arra számítanak, hogy későbbi projektek megvalósításakor ez a veszteség megtérül. Ekkor meg kell határozni egy minimális összköltséggel járó adott célfüggvényre optimális erőforrás-allokációt. A továbbiakban a harmadik esettel foglalkozom, vagyis egyes megvalósítandó tevékenységeket – költségcsökkentés céljából – más tevékenységekkel váltjuk ki.

A harmadik esetben először egy listát kell készítenünk az egyes tevékenységek alternatív megvalósításairól.

Itt a célunk megfogalmazásakor három szempontot kell figyelembe venni (itt a pontok a választás során prioritásokat is jelölnek):

- 1. először is a lehető legjobb minőségben végezzük el az adott tevékenységeket**
- 2. lehető legtöbb költségcsökkenéssel járjon az alternatív megoldás bevezetése**
- 3. lehető legkisebb mértékben változzon (és ha lehet, inkább csökkenjen) a tevékenység időtartama, erőforrásigénye.**

Ezen kritériumok mellett minden tevékenység esetén meghatározható egy minimális minőségi követelmény, melyet mindenképpen a szerződés értelmében teljesítenünk kell.

Célszerű az adott tevékenységhez tartozó alternatív megoldásokat ezen szempontok szerint rendezni. Azokat a tevékenységeket, melyek a minőségi követelményeknek nem felelnek meg, ki kell zárni a lehetséges megvalósítandó alternatív tevékenységek listájából. A kiválasztás során a célnak leginkább megfelelő alternatív megoldással újraütemezzük a tevékenységeket, illetve újból megkeressük az optimális megoldást. Az alternatív megoldások

keresése egy több célfüggvényre meghatározott kiválasztási probléma. Optimális megoldást Branch & Bound módszerrel vagy dinamikus programozás segítségével kereshetünk.

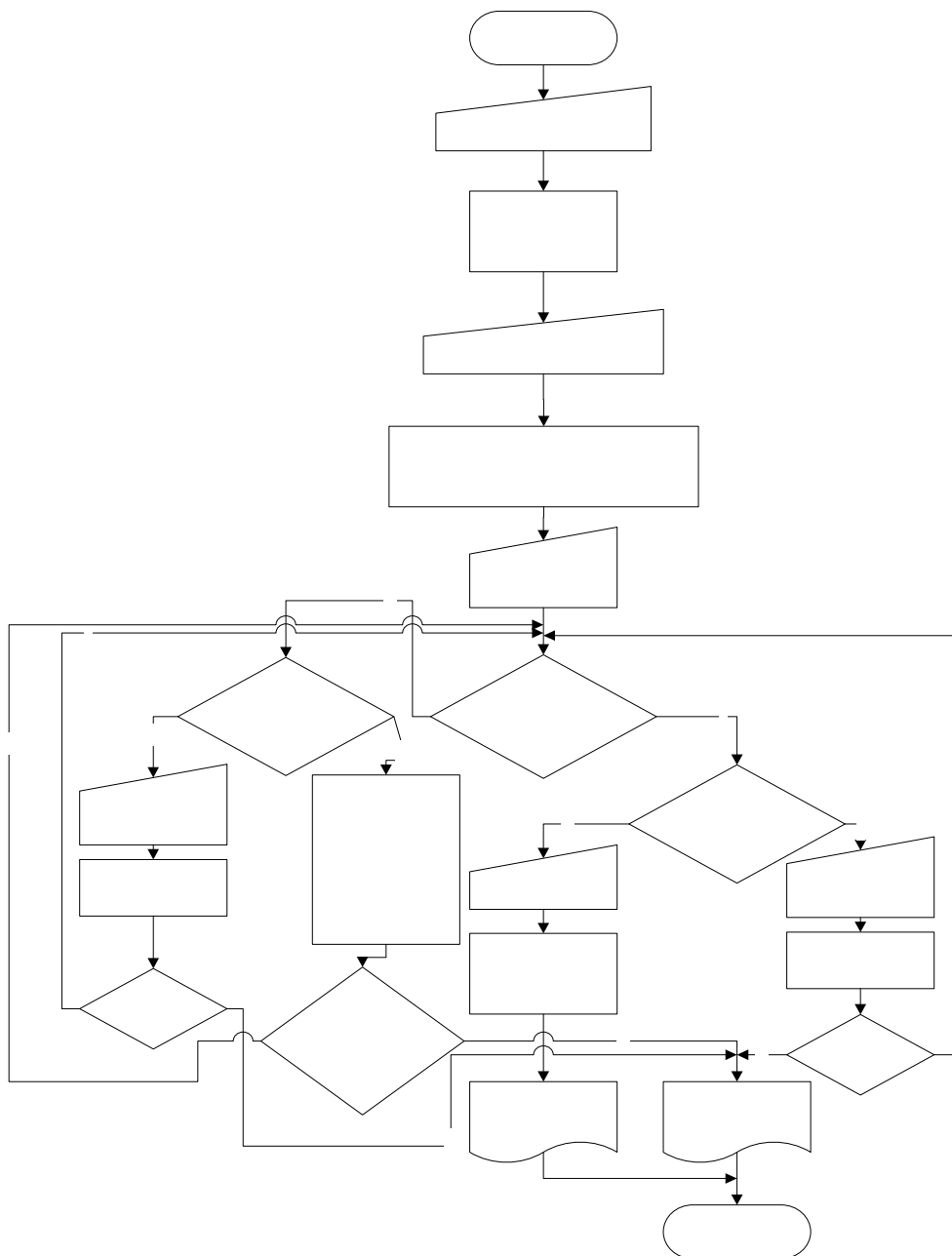
2.5.1.1 Alternatív megoldások keresése

A módszer kiindulópontja egy költség-optimalis megengedett, vagy adott célfüggvényre nézve optimalis erőforrás-allokáció. Ilyen megoldást úgy kereshetünk, hogy először meghatározunk egy összköltség-minimális ütemtervet. Ha található erre az ütemtervre nézve nemkritikus megengedett megoldás, vagyis található olyan megengedett megoldás, amely az erőforráskorlátot nem túllépve a kritikus út hosszát sem váltatja meg, akkor a 2. fejezetben meghatározott célfüggvénynek megfelelően kell alternatív tevékenységeket keresnünk úgy, hogy az alternatív tevékenységek alkalmazása után létezzen nemkritikus megoldás. Ha nemkritikus megoldás létezik, akkor létezik optimalis erőforrás-allokáció is, és ez véges lépésben megtalálható.

A módszer menetét az alábbi (2.5.1.1-1) folyamatábra szemlélteti. A feladat megoldásához tehát először ütemezniünk kell. Ehhez szükségesek a tevékenységek időtartamainak és rákövetkezési relációinak ismerete. Az ütemezést bármely gyakorlatban használt módszerrel (**CPM**, **MPM**, **PERT** stb.) elvégezhetjük. Ezután meg kell határozni egy összköltség-minimális megoldást. Ezt bármely, gyakorlatban használt költségoptimalizáló módszerrel (**CPM/COST**, **MPM/COST**, **PERT/COST** stb.) elvégezhetjük. A módszer használatához szükség van a költségigények, valamint a költségkorlátok és időtartam-költségigény függvények ismeretére. Az időtartam-költségigény függvényeket vagy megbecsüljük, vagy – például kissorozatgyártás esetén – statisztikai módszerekkel adott valószínűséggel meghatározzuk. (Kissorozatgyártás esetén a költség-idő függvények statisztikai becslését részletesen a 2.8.3-as fejezetben tárgyalom). A költségfüggvények meghatározása/beclsése után meg kell vizsgálni, hogy van-e nemkritikus megengedett megoldás. Az előző bekezdésben tárgyaltak szerint ha létezik ilyen, akkor az adott célfüggvényre nézve optimalis erőforrás-allokáció véges lépésben meghatározható. Ha viszont nincs ilyen, akkor itt is alternatív megoldást kell keresni, de ekkor az előző fejezetben vázolt célfüggvény 2. és 3. prioritási pontjai felcserélődnek.

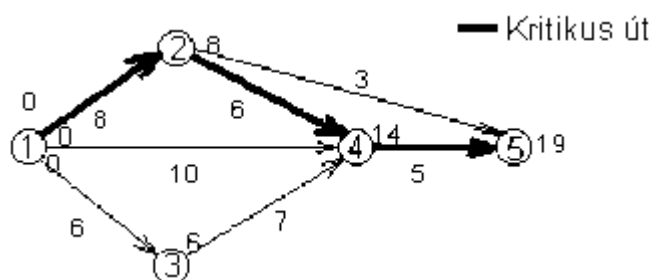
Továbbra is az a cél, hogy az adott tevékenységeket a lehető legjobb minőségben végezzük el, de ahhoz, hogy találhassunk nemkritikus megoldást, olyan tevékenységet kell alternatív módon megvalósítanunk, amelynek erőforrásigénye és/vagy időtartama kisebb.

Ha nem található több alternatív megoldás, akkor a **CPM/COST**-, **MPM/COST**-, **PERT/COST-módszer** megoldási listájából válasszuk ki a minimális összköltségigényű programhoz legközelebb eső megoldást, és a vizsgálatot végezzük el újból.



Tevékenység	Normál idő ($t_{ni,j}$)	Roham-idő ($t_{ri,j}$)	Normál költség ($C_{ni,j}$) (eFt)	Egységnyi költség-növekedés ($\Delta C_{i,j}$) (eFt)	Normál erőforrás-igény ($R_{ni,j}$) (fő)	Egységnyi erőforrás-igény növekedése ($\Delta R_{i,j}$) (fő)
(1,2)	8	6	4500	100	3	0,5
(1,3)	6	5	12000	200	2	1
(1,4)	10	7	10500	150	2	1
(2,4)	6	4	7500	300	4	1
(2,5)	3	3	3500	-	2	-
(3,4)	7	6	1500	500	3	1
(4,5)	5	4	2500	650	1	1
Összesítés			42000	-	-	-

2.5.2-1 táblázat: tevékenységek időtartamait, költség- és erőforrásadatait tartalmazó táblázat



2.5.2-1 ábra: normál időtartamokra vonatkozó CPM-háló

Tegyük fel, hogy a fixköltség 20000 eFt 19 hétre. Ha az átfutási idő csökken, az hetenként 1000 eFt megtakarítást eredményez. Ekkor a minimális összköltségű program a minimális átfutási idejű program lesz, a program teljes költsége pedig 43850 eFt (összes változó költség + 20000 eFt (összes fixköltség) - 4 x 1000 eFt megtakarítás, amiért hamarabb tudtuk a projektet befejezni. Tehát az összes költség = $43850 + 20000 - 4000 = \underline{\underline{59850 \text{ eFt}}}$. A lehetséges megoldásokat szemlélteti az alábbi táblázat.

Lépések száma	Csökkenett tevékenységek	Csökkentés	Összes változó költség (eFt)	Egységnyi költség-növekedés (eFt)	Összes változó-költség-növekedés	Teljes projektidő (hét)
0	-	-	42000	-	-	19
1	(1,2)	1	42100	100	100	18
2	(1,2)+(1,3)	1	42400	100+200=300	300	17
3	(4,5)	1	43050	650	650	16
4	(2,4)+(3,4)	1	43850	500+300=800	800	15

2.5.2-2 táblázat: költségcsökkentés lépései CPM/COST-módszerrel

Vizsgáljuk meg azt az esetet, amikor az erőforráskorlát 10 fő. A rendelkezésre álló költségkeretünk 61000 eFt. A második esetben az erőforráskorlát 11 fő. A rendelkezésre álló költségkeret 59000 eFt.

Az alábbi táblázat azokat a tevékenységeket tartalmazza, amelyeknek van alternatív megoldásuk.

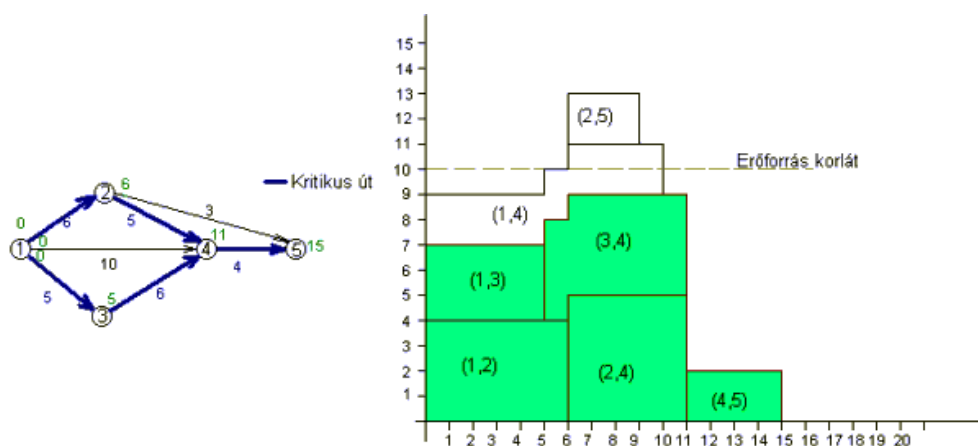
(1,2)			(3,4)			(4,5)		
q	vc/Δvc	r/Δr	q	vc/Δvc	r/Δr	q	Vc/Δvc	r/Δr
100	4500/100	3/0,5	100	1500/500	3/1	100	2500/650	1/1
90	4000/90	2/0,5	90	2000/90	2/1	96	2000/600	1/1,5

2.5.2-3 táblázat: minőség- és költségadatok

q jelöli a minőségi tényezőt. A módszer alkalmazásához nem kell feltétlenül ezeket a minőségi tényezőket számokkal ellátni. Elegendő az alternatív megoldásokat minőségi szempontból sorrendbe rakni. Ha azonban az alternatív megoldásokhoz valamilyen elv szerint számszerű minőségi tényezőt rendelünk, akkor a célfüggvényt könnyebben meghatározhatjuk.

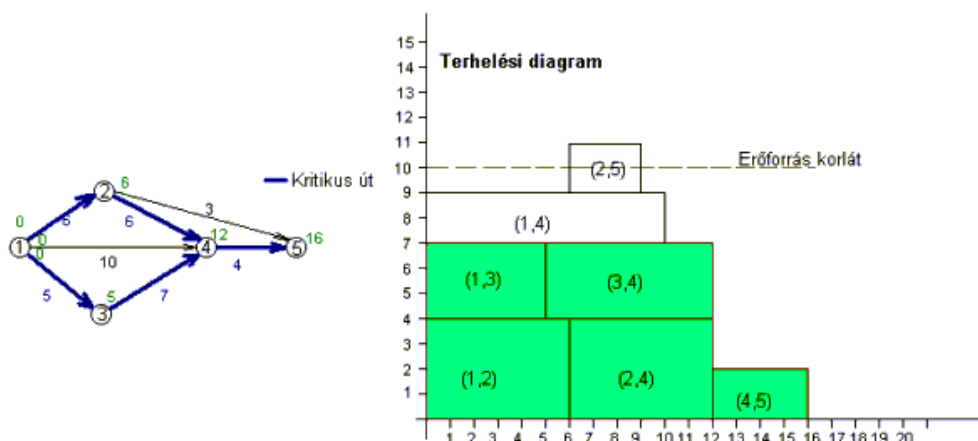
Ha bizonyos tevékenységek megvalósításának minőségére különösen oda kell figyelnünk, akkor célszerű a kiválasztáshoz használt célfüggvényünket úgy megválasztani, hogy e fontos tevékenységek minőségi faktorait súlyozzuk.

Tekintsük a továbbiakban azt az esetet, amikor az erőforráskorlát 10 fő. A rendelkezésre álló költségkeretünk 61000 eFt. Az erőforrás-terhelési diagram 15 hétre a következőképpen néz ki:

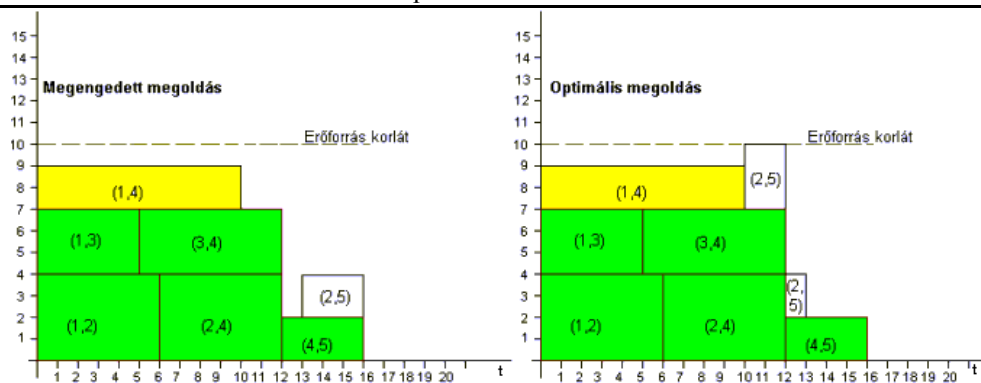


2.5.2-2 ábra: projekt legrövidebb átfutási idejéhez tartozó terhelési diagram (itt nincs nemkritikus megoldás)

Látható a terhelési diagramon, hogy nincs nemkritikus megoldás. Ahhoz, hogy az erőforráskorlátot ne lépjük túl, a CPM/COST megoldásaiból válasszuk ki a 3. termelési programot.



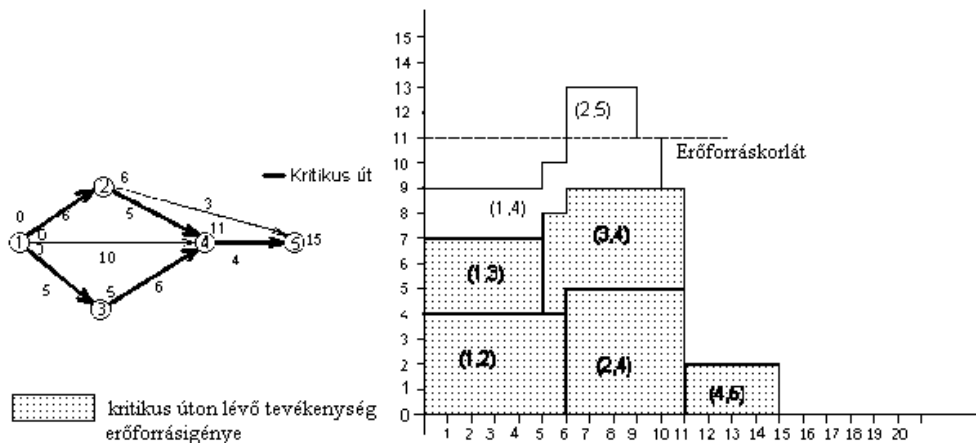
2.5.2-3 ábra: hosszabb átfutási időhöz tartozó terhelési diagram



2.5.2-4 ábra: nemkritikus megengedett és optimális erőforrás-allokáció

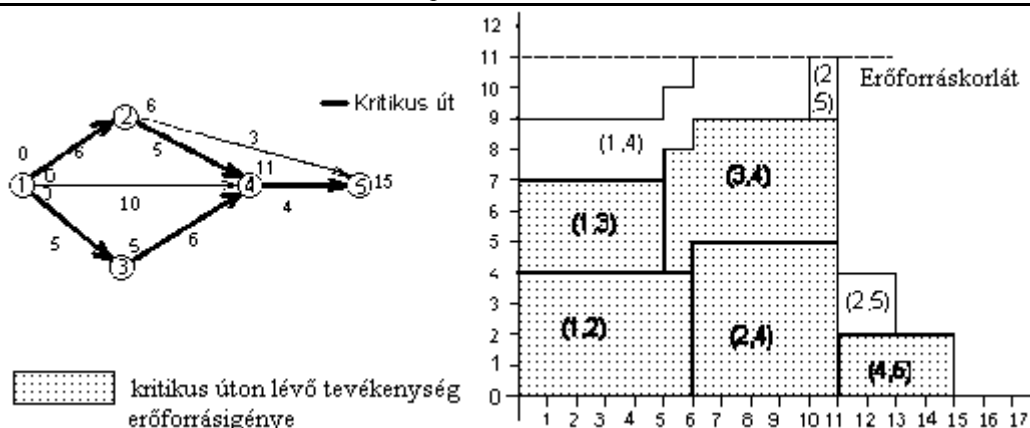
Az optimális erőforrás-allokáció meghatározásánál a cél a lehető legkorábbi kezdés volt. Ekkor nem kellett minőségi kompromisszumokat tennünk. Az összes költség 43050 eFt (összes változó költség) + 20000 eFt (összes fixköltség) – 3000 eFt (a rövidebb átfutási időből adódó fixköltség-megtakarítás) = 60050 eFt < 61000 eFt (költségkeret).

A második esetben sajnos már kénytelenek vagyunk a tevékenység(ek)et minőségileg kevésbé magas színvonalon végrehajtani ahhoz, hogy a tervezett költségkeretet ne lépjük túl.



2.5.2-5 ábra: Projekt legrövidebb átfutási idejéhez tartozó terhelési diagram

Itt található nemkritikus megengedett megoldás és ebből optimális megoldás is.



2.5.2-6 ábra: optimális erőforrás-allokáció

Az összes költség az előzőekben kiszámolt 59850 eFt, ami nagyobb, mint az 59000 eFt-os költségkeret. Ha a (3,4)-es tevékenységet egy alternatív tevékenységgel helyettesítjük, akkor az erőforrásigény és az időtartam nem változik, de a költség csökken 900 eFt-tal. Ekkor az összes költség 58850 eFt lesz, ami már kisebb, mint a rendelkezésre álló költségkeret.

Minél nagyobb a verseny egy projekt megvalósításáért, annál nagyobb az esélye, hogy olyan alacsony áron kell megvalósítani a projektet, hogy az már a költségeinket sem fedezi. Sokszor a cégek nincsenek tisztában, hogy az adott projektet mennyi pénzből lehet megvalósítani, és így sokszor kevesebb pénzért elvállalják a beruházás megvalósítását, és csak később jönnek rá, hogy a költségeik sokkal magasabbak lesznek, mint arra előzőleg számítottak.

Ha a szerződés szerinti juttatások nem fedezik a költségeket, akkor vagy elállunk az üzlettől, vagy elvállaljuk még akkor is, ha tudjuk, hogy csak veszteséggel tudjuk megvalósítani a projektet, vagy megpróbáljuk a tevékenységeket alternatív megoldásokkal kiváltani. Ilyenkor viszont figyelembe kell venni, hogy a lehető legmagasabb minőségi színvonalon valósítsuk meg az egyes tevékenységeket.

2.6 Többféle erőforrás egyidejű kezelése, párhuzamos projektek közötti erőforrás-elosztás

Nagyobb projektek esetén nem csak egy erőforrást kell figyelembe vennünk, hanem általában többet (munkaerő, anyagfelhasználás, berendezések stb.) Ezeket együttesen kell kezelni. Ráadásul általában párhuzamosan több projekt is működhet, amelyek között optimálisan el kell osztani az erőforrásokat. [220-221, 223, 226] Módszereket ilyen típusú problémákra korábban is publikáltak, azonban ezek mindegyike az irodalmi áttekintésben bemutatottak

szerint nem megengedett megoldásokból indultak. És rendelkeztek azzal az összes hátránnyal (számítási idő, speciális alkalmazási terület stb.), amit már az első fejezetben említettem. [74, 97, 314, 319, 367]

Amennyiben a tevékenységek többfajta erőforrásigényét optimálni kívánjuk, úgy ezekre az erőforrásokra is fel kell rajzolni a terhelési diagramot. Megengedett megoldást a soros vagy a párhuzamos allokációval kereshetünk. Az optimálásnál figyelniünk kell arra, hogy a terhelési diagramokon az egyes tevékenységek szinkronban kezdődjenek, illetve fejeződjenek be. Hiába lehetne az egyik terhelési diagramon az adott tevékenységet korábbi időpontra beütemezni, ha a másik terhelési diagramon erre nincs lehetőség.

Párhuzamos projektek esetén, amennyiben az azonos típusú erőforrások (pl. munkaerő) átcsoportosíthatók a projektek között, akkor ezeket a projekteket összevonhatjuk egy hálóba, itt optimálva biztosan globálisan is optimális megoldást kapunk. Az optimálás és a szétbontás után leolvashatók a diagramról a megváltozott erőforráskorlátok is.

Ha a projektek között nem valósítható meg az erőforrás átcsoportosítása (pl. nem azonos típusúak az erőforrások), akkor két független projektként kezelve külön-külön kell optimálnunk. [220-221, 223, 226]

2.6.1 Párhuzamos projektek közötti erőforrás-elosztás (ERALL-OPT/PP, OPT-RALL/PP)

Egy vállalatnál párhuzamosan több beruházási projekt is folyhat. Amennyiben azonos erőforrásokat igényelnek, akkor elképzelhető, hogy érdemes bizonyos erőforrásokat átcsoportosítani az egyik projektből a másikba a hatékonyabb erőforrás-kihasználás érdekében. Erőforrásokat a két projekt között csak akkor lehet átcsoportosítani, ha:

1. azonos erőforrásokat használnak,
2. az erőforrás átcsoportosítása lehetséges (nincs térbeli, földrajzi, technológiai, egyéb akadálya) és
3. az erőforrás-átcsoportosítást a vállalat engedélyezi.

Amennyiben a fenti feltételek teljesülnek, akkor a hálók, és a terhelési diagramok összevonhatók egy hálóba, illetve egy diagramba. Az összevont hálón, illetve diagramon sem kaphatunk a felhasznált tartalékidőkre jobb megoldást, mint a legkorábbi időre beütemezett, maximálisan párhuzamosított megoldás, hiszen a rákövetkezési relációkat itt is be kell tartani.

Mégis általában jobb, hatékonyabb megoldást kapunk, mintha két független párhuzamosan működő projektet külön-külön optimálnánk.

Amennyiben a fenti feltételek bármelyike nem teljesül, akkor csak külön-külön optimálhatjuk a projekteket.

A módszernek magyarul **ERALL-OPT/PP** (optimális erőforrás-allokáció párhuzamosan működő projektek esetén), angolul **OPT-RALL/PP** (Optimized Resource Allocation with Parallel Projects) nevet adtam.

Állítás: A módosított **ERALL-OPT/PP** megadja az optimális megoldást véges lépésben.

Bizonyítás: Amennyiben két projekt között nem valósítható meg erőforrás-megosztás, akkor független projektként kell kezelnünk. A két projektre külön-külön kell elvégezni az optimalizálást, valamely fent említett optimáló eljárásunkkal. Ha a projektek között az erőforrások megoszthatók, akkor egy új háló rajzolható, ahol az erőforrások összeadhatók. Az összetett projektben kell először egy megengedett megoldást keresni. Amennyiben létezik megengedett megoldás, akkor valamely optimáló megoldással határozzuk meg az optimális megoldást. A szétválasztás után fennmaradó kihasználatlan erőforrások elosztása az optimumot nem befolyásolja.

A fennmaradó erőforrást többféleképpen is eloszthatjuk. Amennyiben például az erőforrás a munkaerő, úgy csak annyi munkást kell foglalkoztatni az adott időben a projekten, amennyit a projekt erőforrásigénye megkíván. Ha az erőforrás nem a munkaerő, hanem pl. valamilyen anyag, berendezés, akkor a ki nem használt erőforrást az alábbiak figyelembe vételével oszthatjuk el:

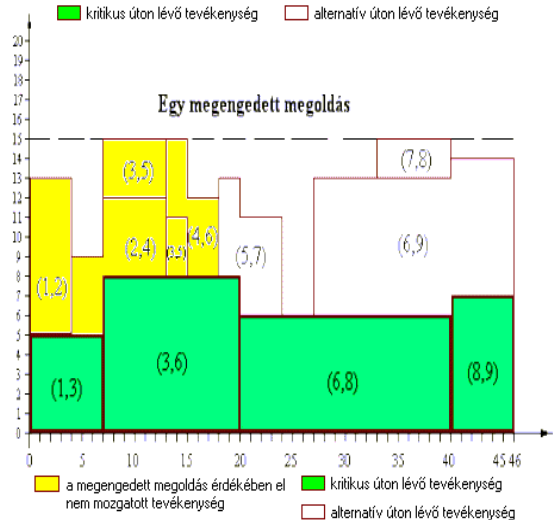
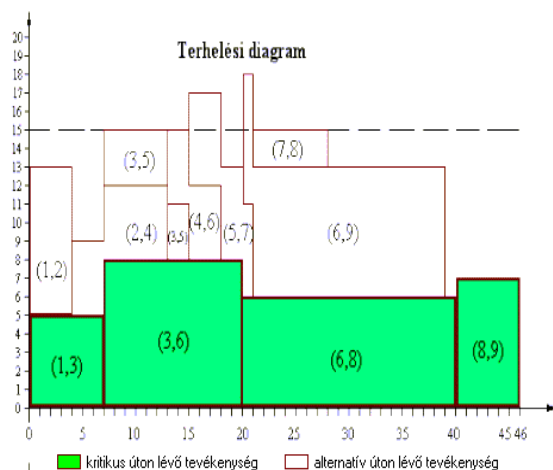
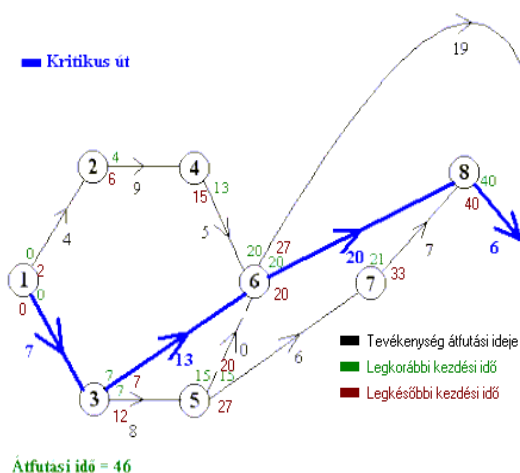
1. Ha a legfontosabb szempont, hogy a termelési program akkor se változzon, ha valamilyen többlet-erőforrásigény lép fel, akkor célszerű a fennmaradó erőforrásokat vagy egyenlően, vagy valamilyen súlytényező figyelembevételével ahhoz a termelési programhoz rendelni, ahol az erőforrásokra inkább szükség lehet.
2. Az előző esetben előfordulhat, hogy sokszor kell egyik projektből a másikba szállítani olyan anyagokat, amelyeket lehet, hogy fel sem használunk. Ha az a szempont, hogy a projektek közötti anyagszállítás minimális legyen, akkor a következőképpen járunk el. Kihasználjuk, hogy a projektekhez szükséges

összes erőforrás elegendő az összes projekt számára, így azt kell meghatároznunk, hogy hogyan lehet kielégíteni az összes erőforrásigényt minimális anyagmozgással.

2.6.1.1 Példa

Az első projekt:

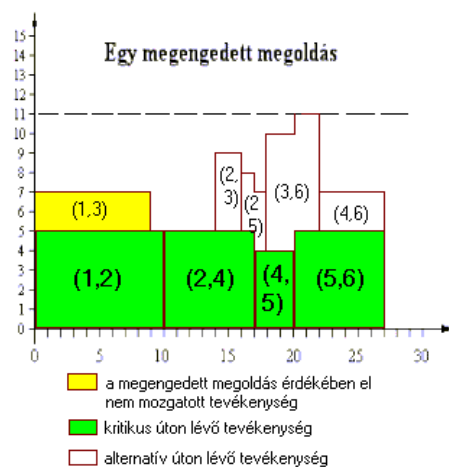
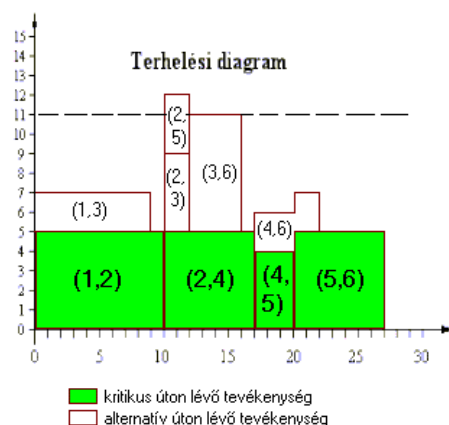
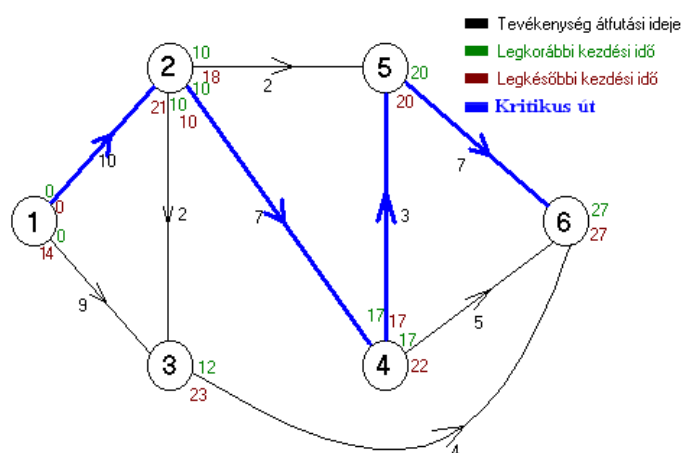
Tevékenység jele	Erőforrás-szükséglet	Lefutási idő
(1,2)	8	4
(1,3)	5	7
(2,4)	4	9
(3,5)	3	8
(3,6)	8	13
(4,6)	4	5
(5,6)	0	0
(5,7)	5	6
(6,8)	6	20
(6,9)	7	19
(7,8)	2	7
(8,9)	7	6



2.6.1.1-15 ábra: első projekt

A második projekt:

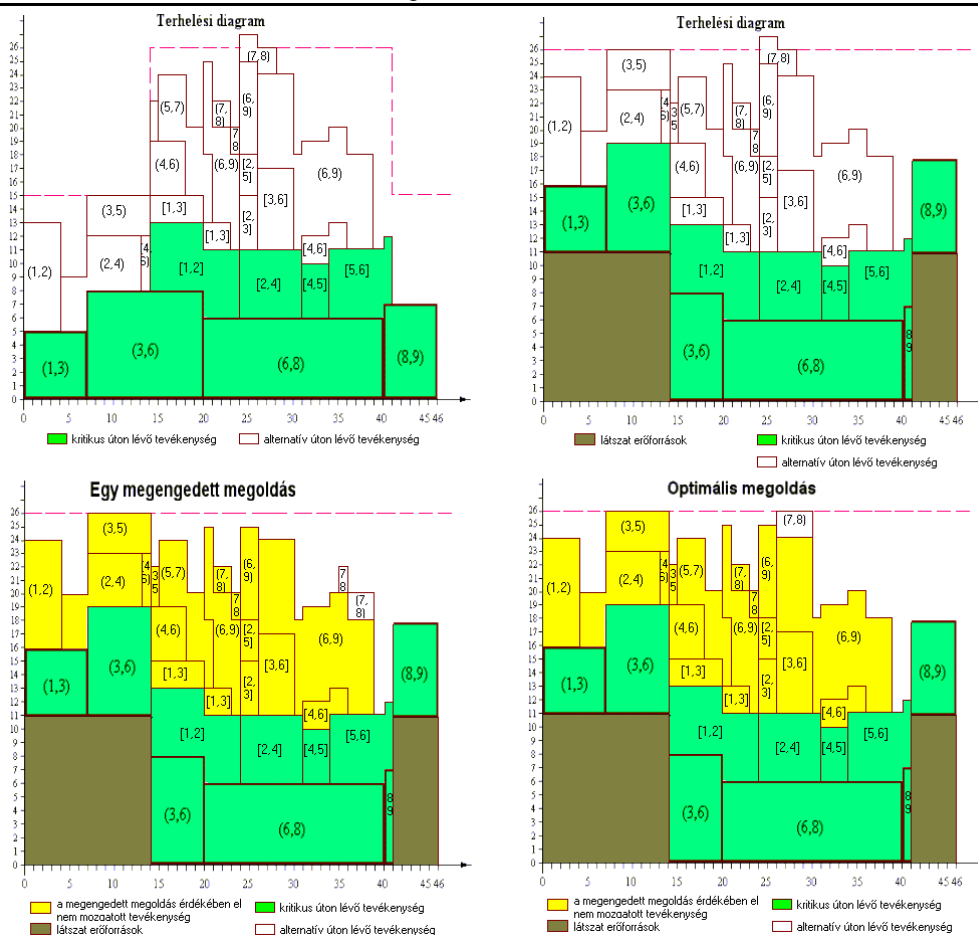
Tevékenység jele	Erőforrás-szükséglet	Lefutási idő
(1,2)	5	10
(1,3)	2	9
(2,3)	4	2
(2,4)	5	7
(2,5)	3	2
(3,6)	6	4
(4,5)	4	3
(4,6)	2	5
(5,6)	5	7



2.6.1.1-16 ábra: második projekt

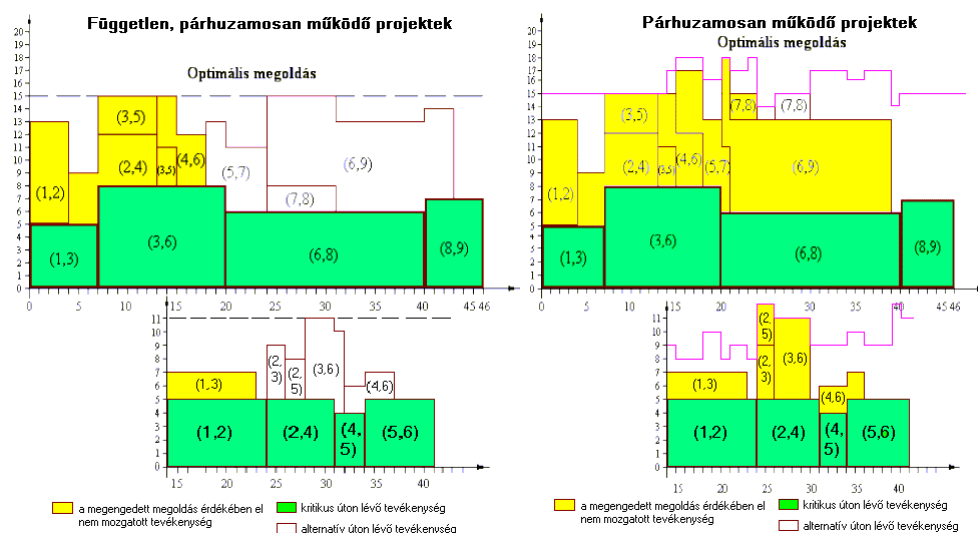
Az első esetben a projektek között az erőforrások nem csoportosíthatók át. Ekkor csak külön-külön optimalizálhatjuk a projekteket.

Az optimális megoldások ebben az esetben könnyen számíthatók. Az optimális megoldás bemutatása előtt azonban nézzük meg azt az esetet, amikor a projektek párhuzamosan is végezhetők. A második projektet az első elkezdése után 14 nappal kell elkezdni. Az első projektben lévő (7,8)-as tevékenységet elkezdése után a harmadik napon meg lehet szakítani, de 9 nap elteltével folytatni kell a tevékenységet. *(Az első projektben lévő tevékenységeket kerek zárójellel, míg a második projektben lévő tevékenységeket szögletes zárójelben jelölöm.)*



2.6.1.1-17 ábra: projektek közötti erőforrás-megosztás

Ezek után hasonlítsuk össze az optimális megoldásokat:



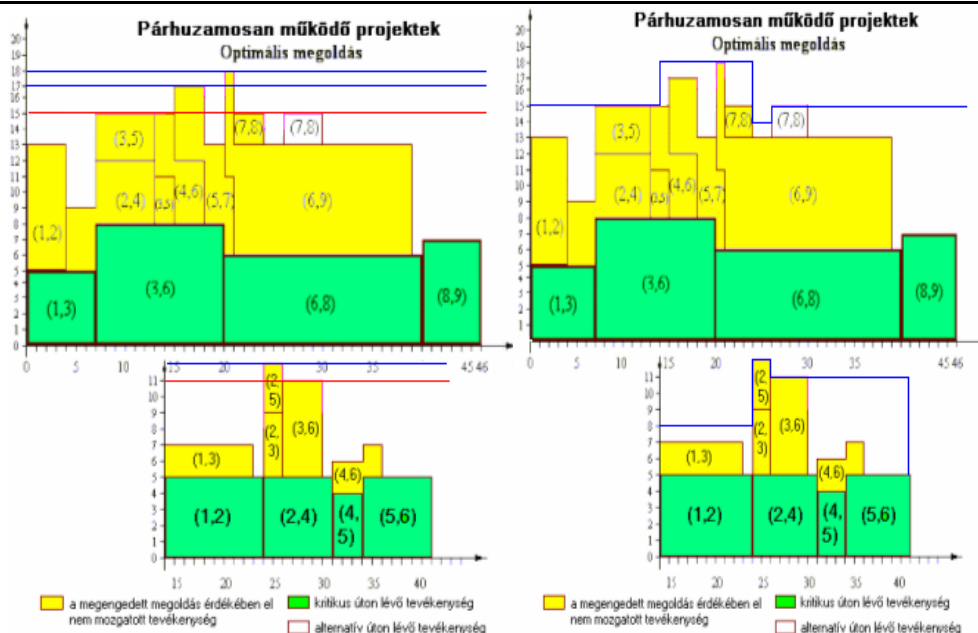
2.6.1.1-18 ábra: optimális megoldások: megoszthatóság, illetve meg nem oszthatóság esetén

Az első oszlopban látható az az eset, amikor a projektek közötti erőforrás-megosztás nem lehetséges. A második oszlopban a párhuzamosan működő projektek között az erőforrások megoszthatók. Látható, hogy a legkorábbi beütemezéshez képest alig változott a tevékenységek beütemezése. Amennyiben lehetőségünk van a projektek közötti erőforrás-megosztásra, akkor elmondható, hogy általában igaz az, hogy a tevékenységeket hamarabb be tudjuk fejezni, mint ha az erőforrásokat nem oszthatnánk meg a projektek között.

A másik fontos szempont, hogy míg az első esetben, független projekteknél a feladat (optimálás) bonyolultsága a két részfeladat bonyolultságának összege, addig jól látható, hogy amennyiben megengedjük a projektek közötti erőforrások megoszthatóságát, a feladat bonyolultsága általában kisebb, de legfeljebb egyenlő a két részfeladat bonyolultságának összegével. Előfordulhat az az eset is (pl. a fenti példa), hogy a feladat egyszerűsödik, tehát jóval kevesebb lépés alatt juthatunk el az optimális megoldásig.

A fennmaradó erőforrásokat tetszés szerint eloszthatjuk a párhuzamosan működő projektek között. Választhatjuk pl. azt a megoldást, hogy viszonylag egyenlően osszuk el a projektek között a fel nem használt erőforrásokat, így azonnal megkapjuk az erőforráskorlátot is. Ez azért célszerű, mert így kisebb változás esetén még részleges (on-line) újraütemezésre nincs szükség.

A fennmaradó erőforrások kezelésénél először meghatározzuk az összevont erőforrásterhelési diagram lokális maximumait. Ezután meghatározzuk az egyedi projektek terhelési diagramjainak maximumait. Ezekhez a lokális maximumokhoz szintvonalakat húzunk. Olyan szintvonalat kell meghatároznunk, amely a lehető legkevesebb lokális maximumot metsz ki a terhelési diagramon. Mivel az összes erőforráskorlát rögzített, úgy kell a szintvonalakat meghatározni a terhelési diagramokon, hogy a lehető legkevesebb helyen lépje túl az erőforrásigény a szintvonalak által meghatározott erőforráskorlátot. Ahol túllépi, ott szintvonalat kell lépni, de úgy kell ezeket a szintvonalakat összekötni, hogy a lehető legkevesebb törés legyen az erőforráskorlátban. Az algoritmus egy megoldását láthatjuk a 7.1.1-5 ábrán.



2.6.1.1-5 ábra: maradék erőforrások optimális erőforrás-elosztása

2.6.2 Többfajta erőforrás egyidejű kezelése (ERALL-OPT/TE, OPT-RALL/MR)

A gyakorlati életben a legtöbb projekt megvalósítása komplex feladat, így természetes igényként jelentkezhet az az eset, hogy egy tevékenységet több erőforrás felhasználásával valósítunk meg.

Egy létesítményprojekt esetén például a legtöbb esetben egy tevékenységhez több erőforrást is figyelembe kell venni. (Pl. munkaerő, berendezések együttes kezelése stb.). Meghatározható valamennyi erőforrás-felhasználásra egy-egy korlát. Ekkor fel kell rajzolnunk valamennyi erőforrásfajtára egy-egy terhelési diagramot. Ezeket csak együttesen optimalizálhatjuk, hiszen az itt megkapott ütemterveknek meg kell egyezniük. A tevékenységeket mindegyik diagramról leolvasva ugyanakkor kell kezdeni, befejezni, megszakítani illetve tovább folytatni.

Míg az előző eset nem feltétlenül járt a probléma bonyolódásával, az optimaláshoz szükséges lépések növekedésével, ebben az esetben egy bonyolultabb feladatot kell megoldanunk. [220-221, 223, 226]

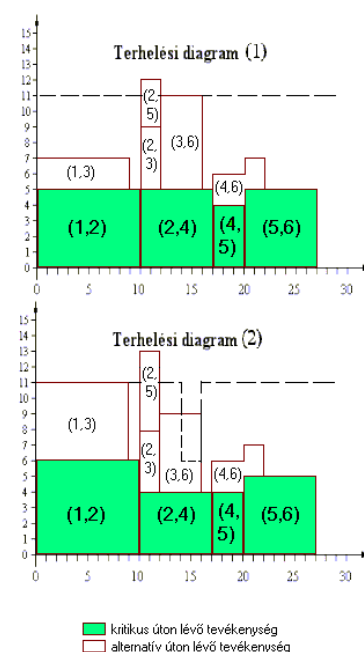
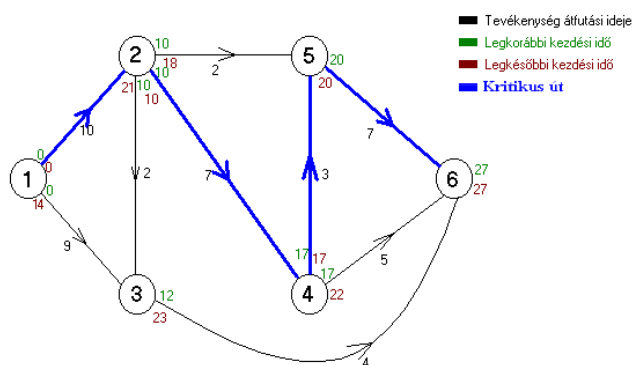
A módszernek magyarul **ERALL-OPT/TE** (optimális erőforrás-allokáció több erőforrás egyidejű kezelése esetén), angolul **OPT-RALL/MR** (Optimized Resource Allocation with Multiple Resources) nevet adtam.

Állítás: A módosított **ERALL-OPT/TE** megadja az optimális megoldást véges lépésben.

Bizonyítás: Tegyük fel, hogy létezik megengedett megoldás. Az optimális megoldás keresésénél figyelembe kell venni, hogy csak olyan ütemtervet fogadhatunk el, amely mindkét korlátozási feltételt kielégíti. Valamennyi erőforrásfajtára meg kell határozni a terhelési diagramot. A rákövetkezési relációk itt is ugyanúgy érvényesek. A töréspontok halmazainak kiszámításánál valamennyi diagramot figyelembe kell venni, így a töréspontok halmaza ezek uniója lesz. Ekkor használhatjuk valamely optimáló eljárást, ügyelve arra, hogy valamennyi korlátozást betartsuk.

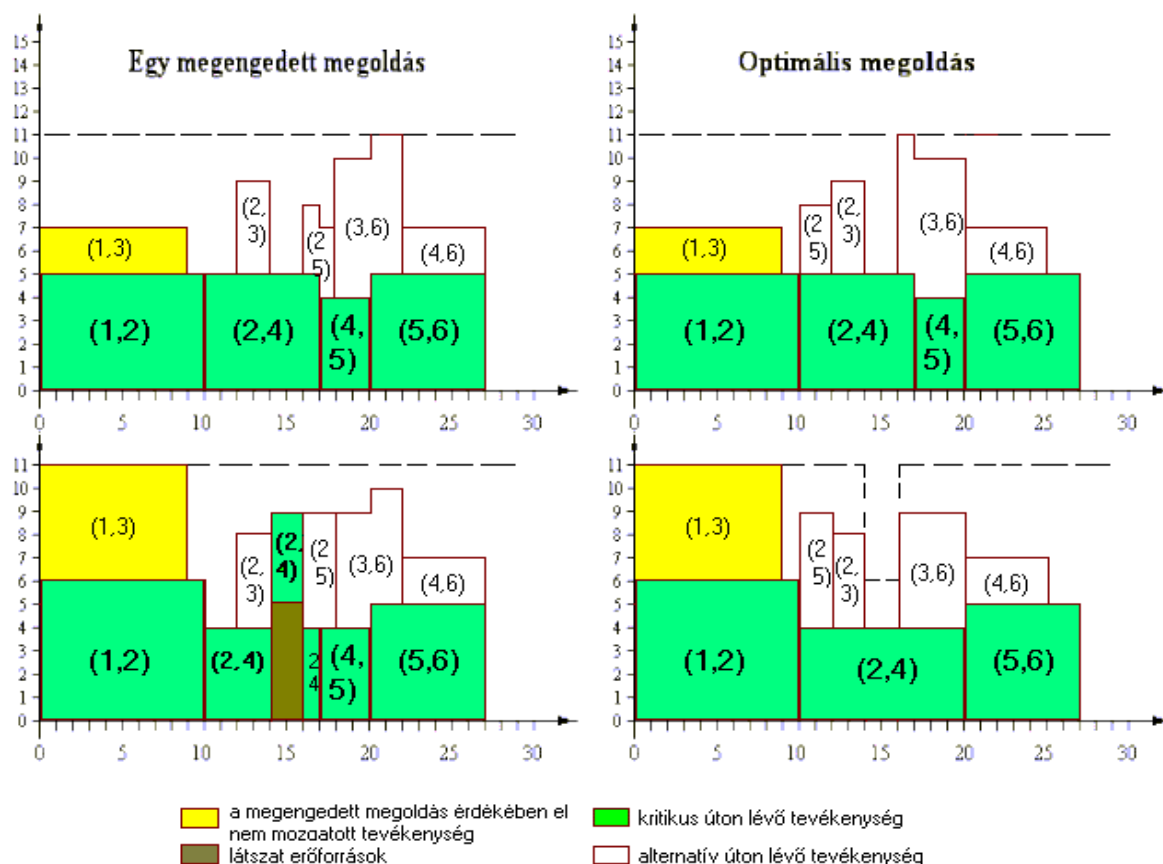
2.6.2.1 Példa

Tevékenység jele	Erőforrás-szükséglet (1)	Erőforrás-szükséglet (2)	Lefutási idő
(1,2)	5	6	10
(1,3)	2	5	9
(2,3)	4	4	2
(2,4)	5	4	7
(2,5)	3	5	2
(3,6)	6	5	4
(4,5)	4	4	3
(4,6)	2	4	5
(5,6)	5	5	7



2.6.2.1-1 ábra: többfajta erőforrás egyidejű kezelése

A megengedett, illetve az optimális megoldás keresésénél a tevékenységeket együtt kell mozgatni. Csak olyan ütemterv fogadható el, ami mind a kettő feltételrendszernek megfelel.



2.6.2.1-2 ábra: optimális megoldás többfajta erőforrás egyidejű kezelése esetén

2.7 Erőforráscsoport – csoportos erőforrás-tervezés

A vállalati vezetés egyfajta újszerű szervezeti törekvése a csoportos munkavégzés. Óriási jelentősége van a komplex problémamegoldásban, a döntéselőkészítésben és a projektek kivitelezésében is. Egy projekt kivitelezésénél bevett gyakorlat például, hogy bizonyos munkákat egy-egy csoport végez. Sokszor előfordul az is, hogy bizonyos tevékenységeket alvállalkozóknak adnak ki. A fővállalkozó sok esetben csak e munkák koordinálásával foglalkozik.

A projektmenedzsmentben használt erőforrás-tervezési módszerek legnagyobb hiányossága, hogy ezeket az eseteket nagyon nehezen kezelik. A leírt módszer segítséget nyújt, hogy a fenti – gyakorlatban előforduló – problémákat kezelni tudjuk. Arra a kérdésre keressük a választ, hogy ha egy-egy munkát egy alvállalkozó vagy egy csoport végez, akkor hogyan kell az adott munkát elosztani a csoporton belül ahhoz, hogy azt a lehető legrövidebb idő alatt az erőforráskeretét nem túllépve költség-hatékonyan végezze el.

2.7.1 Erőforráscsoportok

A mai projektmenedzsment-szoftverek (MS-Project, Primavera, CA-SuperProject stb.) hatékonyan kezelnek ún. erőforráscsoportokat (resource group). Ennek lényege, hogy az azonos típusú erőforrásokat (pl. munkaerő, ezen belül: segédmunka, szakmunka stb.) egy csoportba szervezhetjük. Ezen erőforrások bizonyos tulajdonságai megegyeznek (pl. munkaidő, költségfelmerülés stb.). Fontos kérdés lehet azonban még, hogy az egyes erőforrástípusok mennyire helyettesíthetők egymással. Ez a probléma azonban a mai projektmenedzsment szoftvereken – mint ahogy később látni fogjuk – túlmutat.

Felmerülhet a kérdés, hogy erre a plusz információra egyáltalán miért lehet szükség? Ha feltesszük, hogy egy-egy tevékenységet egy-egy csoport végez, akkor feltehetjük azt is, hogy a csoport tagjainak közel azonos típusú munkát kell elvégezniük, tehát az általuk elvégzett munkák (többnyire) azonos típusú erőforrásokat igényelnek. Továbbá feltesszük, hogy a csoport érdeke, hogy a rendelkezésére álló erőforrásaival gazdálkodva a lehető legrövidebb idő alatt és a lehető legkisebb költséggel végezze el a rábízott tevékenységeket. Ha megengedünk az azonos jellegű erőforrások között bizonyos „konverziót”, akkor akár egy hatékonyabb erőforrás-terheléshez, kisebb időtartamokhoz, vagy kevesebb költséggel járó ütemtervhez juthatunk.

Mit értünk az azonos jellegű erőforrások „konverzióján”? Egy erőforrásigény – bizonyos megszorításokkal – kiváltható egy másik (hasonló jellegű) erőforrással. Gyakorlatban is előfordul, hogy egy segédmunkát egy szakmunkás végez. (Sajnos ennek fordítottja is előfordul.) Ennek következménye sokszor a munka nem megfelelő minősége, költséghatékonysága stb.

Először a modell szempontjából két szélsőséges esetet tekintünk. Az egyik eset, amikor két erőforrás egyáltalán nem helyettesíthető egymással. Ebben az esetben a több erőforrás egyidejű kezelésére használható technikákat (Multi Resource Management) alkalmazhatjuk. Ebben az esetben erőforrás-típusonként fel kell rajzolni egy-egy terhelési diagramot. Az optimalizálás során a tevékenységek kezdési és befejezési idejét szinkronizálnunk kell az erőforrás-terhelési diagramoknak megfelelően.

A másik – modell szempontjából – szélsőséges eset, amikor két erőforrás tökéletesen helyettesíthető egymással (pl. azonos típusú munkák különböző épületrészen). Ekkor a feladat a párhuzamosan működő projektek közötti erőforrás-megosztásra vezethető vissza. Ekkor az erőforrásokat egy közös terhelési diagramba rajzolhatjuk fel. A gyakorlatban előfordulhat

azonban olyan eset is, amikor két erőforrásigény egymással – bizonyos megszorításokkal – helyettesíthető. Ennek kezelésével foglalkozunk a következő fejezetekben.

2.7.2 Csoportos erőforrás-tervezés

Egy szervezetben vagy egy csoportban egy-egy ember általában többféle munkához is érthet különböző mértékben. Kérdés lehet, hogy kit mire alkalmazzunk, ha a fenti célt tartjuk szem előtt. Egyénnel, vagy csoportokkal dolgozzunk? Ha csoportokat foglalkoztatunk, akkor hogyan válogassuk össze a csoportot? Erre milyen informatikai támogatottságot vehetünk igénybe?

Először arra a kérdésre válaszoljunk, hogy egyénnel vagy csoportokkal dolgozzunk? Melyik hatékonyabb? Amint azt később látni fogjuk, ezt általában a feladat jellege – ezen belül is az erőforrásigények – határozzák meg.

Ha az erőforrások egymást egyáltalán nem helyettesíthetik, akkor erőforrás-kihasználás szempontjából az egyéni, vagy csoportos hozzárendelés bármelyike jó megoldásnak bizonyulhat. Ebben az esetben egy tevékenység erőforrásigénye (pl. 8 CNC-hegesztőmunkás) más erőforrással nem helyettesíthető. A modell szempontjából lényegtelen, hogy 8 CNC-hegesztő végzi-e el a munkát vagy egy 8 tagú CNC-hegesztő csapat. (A csoportban lévő esetleges szinergiahatásokkal most nem foglalkozunk).

Amennyiben az erőforrások tökéletesen helyettesíthetők egymással, abban az esetben a csoport foglalkoztatása már általában célravezetőbb, mint egyes egyének alkalmazása. A feladat ebben az esetben a párhuzamosan futó projektek közötti erőforrás-megosztásra vezethető vissza. Ha egyénnel dolgozunk, és az erőforrások nem megoszthatók, akkor párhuzamosan futó független projektekhez analóg feladatot kapunk. Amennyiben az azonos típusú tevékenységeket egy csoport végez, úgy az erőforrásigények összevonhatók. A csoport tagjai között eloszthatja a különböző tevékenységeket, illetve az ehhez szükséges erőforrásigényeket.

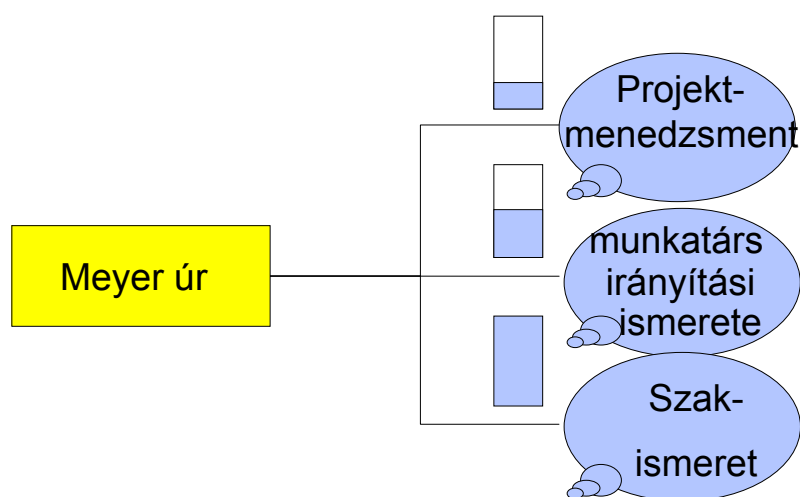
A fenti megfontolások alapján csoportot célszerű alkalmazni abban az esetben is, ha az erőforrások bizonyos mértékig helyettesíthetők egymással. A modell részletezését a következő fejezet tárgyalja.

A következőkben azt szeretnénk megvizsgálni, hogy – a feladat szempontjából – hogyan érdemes összeválogatni a csoport tagjait. A csoport tagjainak megválasztásánál a következő szempontokat kell figyelembe venni.

1. Lehetőleg hasonló kompetenciákkal rendelkező csoporttagokat célszerű egy csoportba összeválogatni. Erre a csoportra olyan tevékenységeket célszerű rábízni, amelyek azonos típusúak, részben vagy teljesen helyettesíthetők egymással. Ha a csoportban hasonló kompetenciájú egyének dolgoznak, akkor közöttük a munkamegosztás, illetve ezáltal hatékony erőforrás-kihasználás valósítható meg.
2. Mivel a modellből is kitűnik, hogy a csoport mérete csak addig bővíthető, ameddig található olyan erőforrás, ami egymással tökéletesen vagy részben helyettesíthető, a csoport méretére fokozottan figyelniük kell. A csoport méretének még egyéb szempontok is gátat szabnak. A túl nagy csoport nehezen kezelhetővé válhat. A csoportra hatékony erőforrás-megosztására felírandó probléma túlságosan bonyolulttá válhat.

2.7.3 Kompetencia alapú kiválasztás informatikai támogatása

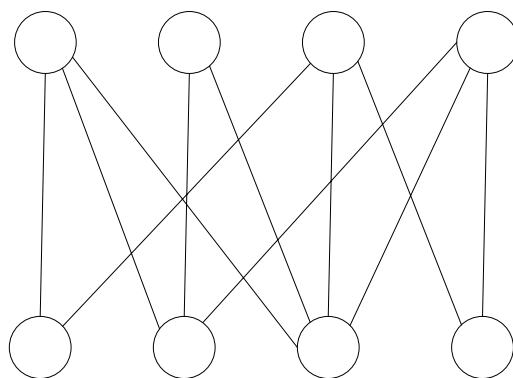
A kompetencia alapú kiválasztást néhány információs rendszer már ma is támogathatja. Ezek a rendszerek ún. tudástérképet használnak, melyben az alkalmazottak tudását, kompetenciáját térképezik fel. Az információs rendszerek tervezése során olyan metodikák is léteznek, melyek az ilyen tudástérképeket támogatják. Ilyenre egy példa az ARIS-rendszer. A metodika a kompetenciákhoz mértéket is rendel. Ezáltal nyilvántarthatjuk, hogy ki milyen területen a leghatékonyabb. Ezt az ARIS-metodikában lefedettségi foknak nevezzük.



2.7.3-1 ábra: tudástérkép az ARIS-metodikában

A modell során feltesszük, hogy akinek az adott területen nagyobb a lefedettségi foka, nagyobb hatékonysággal is végzi el a kompetenciájába tartozó munkákat. Ebből egyenesen következik, hogy az adott tevékenységekre a legnagyobb lefedettségi fokkal rendelkező egyéneket válasszuk ki.

Természetesen előfordulhat az az eset is, hogy az adott tevékenységekre nem tudjuk kiválasztani a legmegfelelőbb embert, mert ők már párhuzamosan más munkán dolgoznak. Az optimális hozzárendelést egy párosítási feladatként lehet meghatározni, melyben az egyik csoport az elvégzendő tevékenységek, a másik csoport a rendelkezésünkre álló munkaerő. A páros gráf élein pedig a lefedettségi fok szerepel. Két csúcs pontosan akkor van összekötve, ha az illető az adott munkát el tudja végezni, vagyis a lefedettségi fok nem nulla.



2.7.3-2 ábra: „hozzárendelési” gráf

Ha kiválasztottuk a megfelelő embereket a megfelelő munkára, akkor az előző fejezetben tárgyalt módszer szerint meghatározhatjuk a csoportokat is, akik majd elvégzik a rájuk bízott tevékenységet. Ha előző feltételezésünkkel élünk, vagyis ha feltesszük, hogy akinek a lefedettségi foka az adott munkát illetően magasabb, az hatékonyabban/gyorsabban végzi el a munkát, akkor ezáltal időtervezésünket, erőforrás-tervezésünket pontosíthatjuk is. Ha az időtervezésünket úgy készítettük el, hogy a lehető legnagyobb lefedettségi fokkal rendelkező egyéneket választottuk ki a megfelelő munkára, akkor feltételezhetjük, hogy kisebb lefedettségi fokkal rendelkező egyén lassabban végezné el a munkát, illetve azonos idő alatt több embert kellene foglalkoztatnunk. Feltesszük tehát, hogy az időtartam, illetve az erőforrás-szükséglet függvénye a tevékenységet végzők lefedettségi fokától fordított arányban függ. Ha a lefedettségi fok alacsony, vagyis ha a tevékenységet végzők kevésbé értenek az elvégzendő munkákhoz, akkor a tevékenységek időtartamai is megnövekedhetnek, ha viszont

a rendelkezésre álló idő alatt el szeretnék végezni a feladatot, akkor pótlólagos erőforrásokra lehet szükségünk.

A módszer menete tehát a következő: először meg kell határoznunk a megvalósításhoz szükséges tevékenységeket, meg kell állapítani a rákövetkezési relációkat. Ezután felrajzolhatunk egy ún. **logikai hálót**. Ez a háló megmutatja a tevékenységek közti logikai kapcsolatokat (rákövetkezési relációkat).

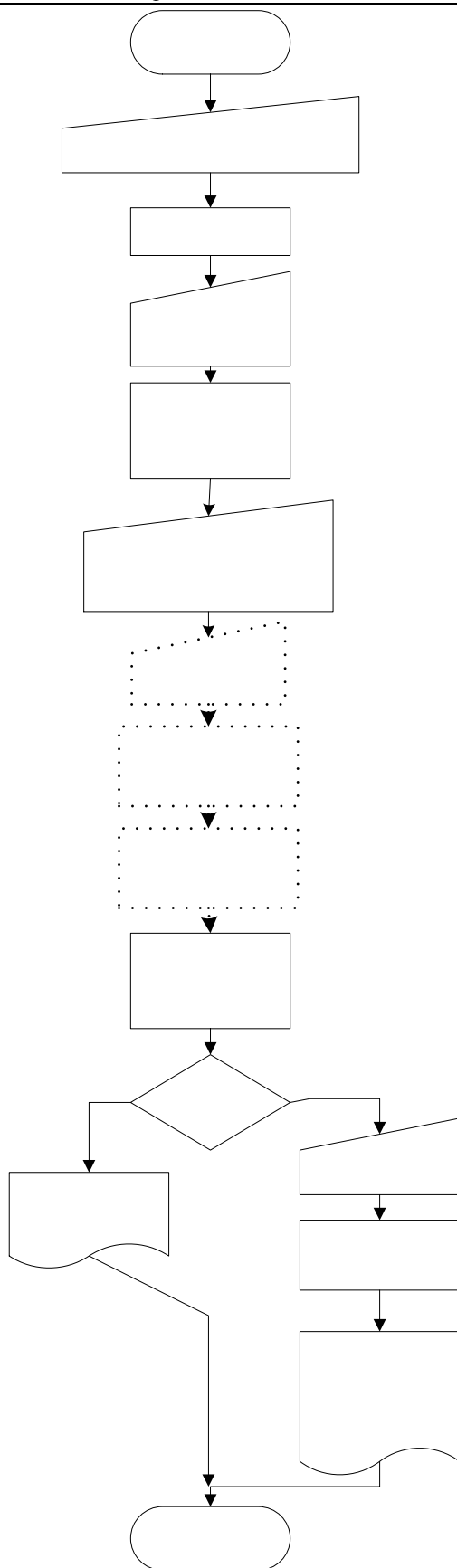
Ezután meg kell határoznunk, vagy meg kell becsülnünk a tevékenységek (várható) **időtartamát**, majd egy ütemezési módszerrel (pl. **CPM**, **PERT**, **MPM** stb.) meghatározhatjuk a **teljes projekt átfutási idejét**.

Az ütemezési feladat megoldása után meg kell határozni, vagy meg kell becsülni a tevékenységek végrehajtásához szükséges erőforrásokat, illetve ezek mennyiségét. (Pl. egy adott munka végrehajtásához hány fő szükséges, milyen anyagokat kell felhasználnunk, illetve milyen berendezésekre van szükségünk.)

Ha a szükséges kompetenciákról tudomásunk van, akkor ezáltal kiválaszthatjuk a megfelelő embereket a megfelelő munkákra, illetve összeállíthatjuk a megfelelő csoportokat. A hozzárendelések után az időtartam- és/vagy erőforrásigény-adatokat pontosíthatjuk.

Ha adott(ak) az erőforrás-korlát(ok), akkor keresnünk kell egy megengedett megoldást. Ebben az esetben egy olyan ütemtervet kapunk, amely során a teljes projekt átfutási ideje lehetőleg nem növekszik és a rendelkezésre álló erőforráskorlátot a program megvalósítása során egyszer sem lépjük túl.

Ezután meg kell határoznunk, hogy milyen erőforrás-felhasználást szeretnénk megvalósítani, vagyis meg kell határozni a célfüggvényt. Ha található megengedett megoldás, akkor található véges lépésben az adott célfüggvényre nézve legjobb megoldás, amit a továbbiakban **az erőforrás-allokációs probléma adott célfüggvényre nézve optimális megoldásának** nevezünk. A módszer leegyszerűsített folyamatábráját láthatjuk az alábbi ábrán.



2.7.3-3 ábra: a módszer lépései

2.7.4 Tökéletes helyettesítéstől a tökéletes kiegészítésig

Az előző fejezetben azokkal az esetekkel foglalkoztam, amikor az erőforrások tökéletesen helyettesítik egymást (ekkor egy terhelési diagramba összevonhatók a tevékenységek), illetve tökéletesen kiegészítik egymást (ekkor két terhelési diagramot kell felrajzolni és a tevékenységeket pedig szinkronizálni kell). A kérdés az, hogy e két szélsőséges eset között van-e más lehetőség is.

Mint azt az előző fejezetben már láthattuk, hogy ki milyen munkát végez, már a tervezés során eldőlt. Kérdés az, hogy a terhelési diagramot külön, vagy egybe kezeljük. Ha külön-külön rajzolunk fel erőforrás-terhelési diagramot minden egyes erőforrástípusra, akkor az esetleges erőforrás-átcsoportosítások nem jelennek meg. Ha pedig egy diagramban kezeljük az erőforrásokat, akkor hamis képet kapunk, hiszen az erőforrásigények nem minden esetben helyettesíthetők egymással.

Csak azokat az erőforrásokat célszerű egy diagramba összevonni, amik viszonylag jól helyettesíthetők egymással. Mint azt a korábbi fejezetekben említettem, a helyettesíthetőséget a dolgozók kompetenciájának lefedettségi foka határozza meg. Ebbe az esetben a – lefedettségi fok reciprokával súlyozott (a fordított arányosság miatt) – idő, vagy erőforrásadatokat kell figyelembe venni. Láthatjuk, hogy tökéletes kiegészítés esetén a súly végtelen lenne, tehát nem lenne tartható az egy diagramba való összevonás. Látható továbbá, hogy minél kisebb a lefedettségi fok, vagyis minél inkább helyettesíthetik egymást olyan emberek, akik az adott tevékenység elvégzésére kevésbé kompetensek, annál több lesz a becsült átfutási idő és/vagy az erőforrás-szükséglet. Ebből is következik, hogy a megfelelő csoport kiválasztása, illetve a tevékenységekhez rendelő kompetens személyek kiválogatása időt és/vagy erőforrást takarít meg a vállalat számára. [286]

Ezek az eljárások bármely – a projektmenedzsmentben használt (pl. Microsoft Project, CA-SuperProject, Primavera stb.) – szoftver által szolgáltatott megengedett megoldásból indulhatnak. Figyelembe veszik a tevékenységeket végző kompetenciáját, ezáltal pontosabb becslést kaphatunk a szükséges erőforrásigény és időtartam nagyságáról. Ezeket a megoldásokat egy adott cél (vagy célfüggvény) szerint a módszerünk minden lépésben javítja. Több projektre, több erőforrást kezelve minimális összköltségű, optimális erőforrás-terhelésű programokat határozhatunk meg. [399-401]

2.8 Bizonytalan átfutási idejű projektek optimális erőforrás-elosztása

A gyakorlati életben mind a fejlesztési, mind a beruházási, de különösképpen a kutatási projektek esetén nem tudjuk biztosan meghatározni egy adott projekt vagy termelési program tevékenységeinek lefutási idejét. Különösen igaz ez kutatási és fejlesztési programokra, ahol a tevékenységek időtartamai kevésbé ismertek. A tervezett és a tényleges tartam sokszor különbözik egymástól. [222, 224-225, 230]

A módszer segítségével tetszőleges projekt (költség-, erőforrásigény-, idő-)optimális erőforrás-allokációját lehet meghatározni úgy, hogy figyelembe véve a tevékenységek időtartamainak bizonytalanságát is.

A módszer egy megengedett erőforrás-allokáció alternatív úton lévő tevékenységeit adott célfüggvénynek megfelelően ütemezi be (ez a célfüggvény lehet pl. lehető legkorábbi kezdés, lehető legkésőbbi kezdés stb. lásd 2. fejezet), figyelembe véve, hogy az adott tevékenységek várható időtartamai valószínűségi változók, melyeknek van várható értéke illetve szórása. A szakirodalom szerint általában 10-12% költséget takaríthatunk meg azáltal, hogy a tevékenységek időtartamait nem determinisztikus változóként, hanem egy valószínűségi változóként kezeljük, hiszen így az időtartamok bizonytalanságát már előre figyelembe tudjuk venni. Hozzávetőlegesen meg tudjuk határozni, hogy egy adott biztonsági szint mellett mennyi lesz a program átfutási ideje. Mégis nagyon sokszor nem várt események is befolyásolhatják a projekt vagy a termelési program átfutását. Amennyiben a projekt működése közben az erőforráskorlát, vagy a tevékenységek erőforrásigénye, időtartama megváltozik, a módszer segítségével meghatározható egy új termelési program a még futó, illetve a még el nem kezdett tevékenységekre. [152, 222, 224-225, 230]

Az algoritmus segítségével olyan projekteket tervezhetünk, ahol a lehető legrövidebb idő alatt a legkisebb (változó) költségnövekménnyel, optimális erőforrás-kihasználással, maximális párhuzamosítás mellett hajthatjuk végre a projektünket a lehető legkisebb bizonytalanság mellett.

Ezzel a módszerrel több párhuzamosan működő projektet optimalizálhatunk, illetve többféle erőforrás egyidejű kezelését valósíthatjuk meg. Kezelnünk tudjuk továbbá a fel nem használt erőforrásokat is.

2.8.1 Sztochasztikus időtervezés

A sztochasztikus időtervezés során az egyes tevékenységek időtartamát egy-egy valószínűségi változóként kezeljük. Két eset lehetséges:

- a szóban forgó tevékenységek vagy nem teljesen ismeretlenek, és mindegyikükre közelítőleg ismerjük tartamuk valószínűség-eloszlását;
- vagy pedig teljesen ismeretlenek (vagy legalábbis nagyon kevésbé ismertek), és nem ismerjük minden tartam valószínűség-eloszlását.

Ha ismerjük az időtartamok eloszlását, akkor minden nehézség nélkül meghatározhatjuk az egyes tevékenységek átlagos időtartamát (az időtartam várható értékét) és szórásnégyzetét (varianciáját) valamint a szórását (standard bizonytalanságát). [1, 2, 195]

Ezeket így jelöljük: $E(t_{i,j}) = \bar{t}_{i,j}$, $D^2(t_{i,j}) = \sigma_{i,j}^2 = u^2(t_{i,j})$.

Ha nem ismerjük az időtartamok eloszlását, akkor a projekt tervezésekor a számítások megkönnyítése érdekében a – projektmenedzsmentben használt – **PERT-módszert** alkalmazzuk és feltesszük, hogy az időtartamok β -eloszlásúak. [195]

Egyedi- és kissorozatgyártás termelésirányításában egy-egy termék elkészítésének menete szintén sok szempontból felfogható úgy, mint egy projekt megvalósítása. Itt azonban a tevékenységek tartamának eloszlásáról sokszor semmilyen információnk sincs, és gyakran nem is feltételezhetjük, hogy az adott tevékenység időtartama β -eloszlást követ, így a várható értékeket, illetve az időtartam várható értékének szórását (standard bizonytalanságát) is nehéz meghatározni. [267] Itt segítségünkre az lehet, hogy azt az adott terméket minden bizonnyal többször is legyártjuk. Ha az egyes gyártásoknál az egyes tevékenységek időtartamait feljegyezzük, akkor ebből a várható érték a következőképpen becsülhető: N gyártás, illetve M tevékenység esetén az egyes tevékenységek várható értéke:

$$E(X_i) = \bar{X}_i = x_i = \frac{1}{N} \sum_{j=1}^N X_{i,j}, \text{ ahol } i=1,2,\dots,M. \quad (2.8.1-1)$$

A standard bizonytalanság négyzete:

$$u^2(x_i) = s^2(x_i) = s^2(\bar{X}_i) = \frac{s^2(X_i)}{N} = \frac{1}{N(N-1)} \sum_{j=1}^N (X_{i,j} - \bar{X}_i)^2, \text{ ahol } i=1,2,\dots,M. \quad (2.8.1-2)$$

Megjegyzés: Hasonló becsléseket lehetne adni az egyes tevékenységek változó költség-, illetve erőforrásigény vonzatára is.

A továbbiakban mindvégig (erősen) stacioner, ergodikus folyamatokkal foglalkozunk. (Lásd: 1.3.7. fejezet)

Ha a három módszer valamelyikével (ismert eloszlás esetén közvetlenül, ismeretlen eloszlásnál β -eloszlást feltételezve, illetve több gyártás során szerzett tapasztalat útján) meghatároztuk a várható értéket, illetve az időtartamok várható értékeinek szórásnégyzetét (standard bizonytalanságának négyzetét), akkor kiszámítható a projekt átfutási idejének szórása (összetett standard bizonytalansága) az alábbi megfontolásokat felhasználva: [397]

- A központi határeloszlás tételét felhasználva: a projekt, illetve a termelési program átfutási idejének várható értéke az időtartamok várható értékeinek összege.

$$E(y) = \sum_{i=1}^M E(X_i) \quad M \text{ tevékenységet feltételezve a kritikus úton.} \quad (2.8.1-3)$$

- A projekt, illetve termelési program átfutási idejének szórásnégyzete (az összetett standard bizonytalanság) általános esetben a következőképpen számítható:

$$u_c^2(y) = \sum_{i=1}^M \sum_{j=1}^M \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) = \sum_{i=1}^M \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + 2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j), \quad (2.8.1-4)$$

ahol u_c az un. összetett standard bizonytalanság, $u(x_i, x_j)$ x_i, x_j tevékenységek közötti kovariancia, $y = \bar{Y} = f(x_1, x_2, \dots, x_M)$ pedig a modellfüggvényünk. A projektek, illetve a termelés során legtöbbször feltehetjük, hogy $y = \bar{Y} = f(x_1, x_2, \dots, x_M) = x_1 + x_2 + \dots + x_M = \bar{X}_1 + \bar{X}_2 + \dots + \bar{X}_M$. (Ezt a feltételezést tettük tulajdonképpen az első pontban is. Vannak olyan modellek, melyek az időt, illetve a költséget is egységes függvényként kezelik. Ebben az esetben az összetett standard bizonytalanság kiszámításánál más lesz a modellfüggvényünk.) Ha a fenti feltételezést elfogadjuk, akkor deriváltak 1-ek, és a képlet a következőképpen egyszerűsödik:

$$u_c^2(y) = \sum_{i=1}^M \sum_{j=1}^M u(x_i, x_j) = \sum_{i=1}^M u^2(x_i) + 2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M u(x_i, x_j) = \sum_{i=1}^M u^2(x_i) + 2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M u(x_i) u(x_j) r(x_i, x_j), \quad (2.8.1-5)$$

ahol $r(x_i, x_j)$ x_i, x_j tevékenységek közötti korreláció. Ha a korreláció értéke egy, vagyis az egyes tevékenységek várható időtartama függ a másik tevékenység várható időtartamától, akkor az összetett standard bizonytalanság (a projekt

$$(\text{várható}) \text{ átfutásának szórása): } \sigma = u_c(y) = \sum_{i=1}^M u(x_i). \quad (2.8.1-6)$$

Ha a tevékenységek időtartamai egymástól (lineárisan) független (vagyis a korreláció nulla valamennyi esetben), akkor a projekt várható átfutásának

szórásnégyzete (az összetett standard bizonytalanság négyzete) a következőképpen

$$\sigma^2 = u_c^2(y) = \sum_{i=1}^M u^2(x_i). \quad (2.8.1-7)$$

- Minden tevékenységről meg lehet állapítani, hogy az (adott valószínűségi szint mellett) a kritikus úton van-e vagy sem. Tevékenységek legkorábbi/legkésőbbi kezdésének/befejezésének standard bizonytalanságának meghatározásához számos módszer létezik. Ezek a módszerek elsősorban abban különböznek egymástól, hogy mennyire veszik figyelembe, hogy egy alternatív úton lévő tevékenység mikor válhat kritikus úton lévő tevékenységgé, ezáltal szórása hogyan befolyásolja az átfutási idő szórását. A módszer szempontjából adottnak tekintem egy tevékenység időtartama standard bizonytalanságát, és a továbbiakban elsősorban az erőforrásallokációra koncentrálok.

2.8.1.1 Példa

Egy termék gyártása három egymást követő folyamatból áll, melyeknek - 10 gyártást követően – az adatai a következőképpen alakultak.

Gyártás	1. tevékenység		2. tevékenység		3. tevékenység		Összesen	
	idő (nap)	költség (eFt)	idő (nap)	költség (eFt)	idő (nap)	költség (eFt)	idő (nap)	költség (eFt)
1	4,0	100	8,0	150	4,0	100	16,0	350
2	4,5	100	7,8	151	4,2	100	16,5	351
3	4,0	100	8,0	150	3,8	100	15,8	350
4	4,2	102	8,2	150	4,0	102	16,4	354
5	4,3	100	8,0	151	4,0	100	16,3	351
6	4,2	100	8,0	150	4,2	100	16,4	350
7	4,2	101	8,0	150	4,1	101	16,3	352
8	4,1	100	8,2	150	4,0	100	16,3	350
9	4,2	100	8,1	150	4,0	100	16,3	350
10	3,8	99	8,0	151	4,1	99	15,9	349
Átlag	4,15	100,20	8,03	150,30	4,04	100,20	16,22	350,70
Bizonytalanság	0,0601	0,2494	0,0367	0,1528	0,0371	0,2494	0,0742	0,4485
Korreláció (idő 1,2)	-0,3278							
Korreláció (idő 1,3)	0,3985							
Korreláció (idő 2,3)	-0,4245							
Korreláció (kltg. 1,2)	-0,4666							
Korreláció (kltg. 1,3)	1,0000							
Korreláció (kltg. 2,3)	-0,4666							
Összetett bizonytalanság (idő):	0,0742							
Összetett bizonytalanság (kltg.):	0,4485							

2.8.1.1-1 táblázat: tevékenységek időtartamai és változó költségek közötti korreláció.

Látható, hogy az egyes tevékenységek időtartamai illetve változó költségei között van korrelációs kapcsolat, így az összetett standard bizonytalanság kiszámításánál ezeket a korrelációs együtthatókat is figyelembe kell venni. Ez azonban még mindig csak egy statikus modell, mely a különböző költségnövekedéseket (pl. inflációs hatásokat), illetve a begyakorlásokat nem veszi figyelembe. Ha olyan modellt szeretnénk alkalmazni, ami ezeket a változásokat is figyelembe veszi, akkor a standard bizonytalanságok, illetve átlagok helyett

alkalmazhatunk súlyozott átlagokat, illetve súlyozott bizonytalanságokat, melyek a régebbi megvalósításokat kevésbé veszik figyelembe.

2.8.2 Költségtervezés bizonytalan átfutási idő esetén

A projekt, vagy a termelési program során előfordul, hogy a változó költség növekedése árán kell átfutási időt csökkentenünk.

Először bemutatok néhány gyakorlatban használt eljárást, melyek nem veszik figyelembe a költségek esetleges bizonytalanságát, majd bemutatok néhány lehetséges módszert a költségek bizonytalanságainak meghatározására.

2.8.2.1 Determinisztikus költségtervezés sztochasztikus időtervezés esetén

A tevékenységek időtartamainak becslésénél (az egyszerűség kedvéért) feltételezzük, hogy az időtartamok β -eloszlást követnek.

A gyakorlatban költségoptimalásra számos módszert alkalmaznak; ezekre egy példa Bowman, Chrétienne vagy Cox összköltségminimáló módszere [60, 64, 68]. Mivel a módszerem az erőforrások optimalására koncentrál, így tulajdonképpen a módszer használhatóságának szempontjából érdektelen, hogy az ütemezés illetve a költségminimalizálás fázisában milyen módszert alkalmazunk. A könnyebb áttekinthetőség szempontjából egy egyszerű költségoptimaló módszert alkalmazunk, melyet a továbbiakban egyszerűen **PERT/COST-módszernek** nevezek. A **PERT/COST-módszer** alkalmas mind minimális összköltség-, mind pedig minimális átfutási idejű program meghatározására. A sztochasztikus időtervezés során azonban felléphetnek a tervezésből adódó problémák. Nézzünk ezek közül néhányat, illetve vizsgáljuk meg, hogyan lehet ezeket a problémákat kiküszöbölni.

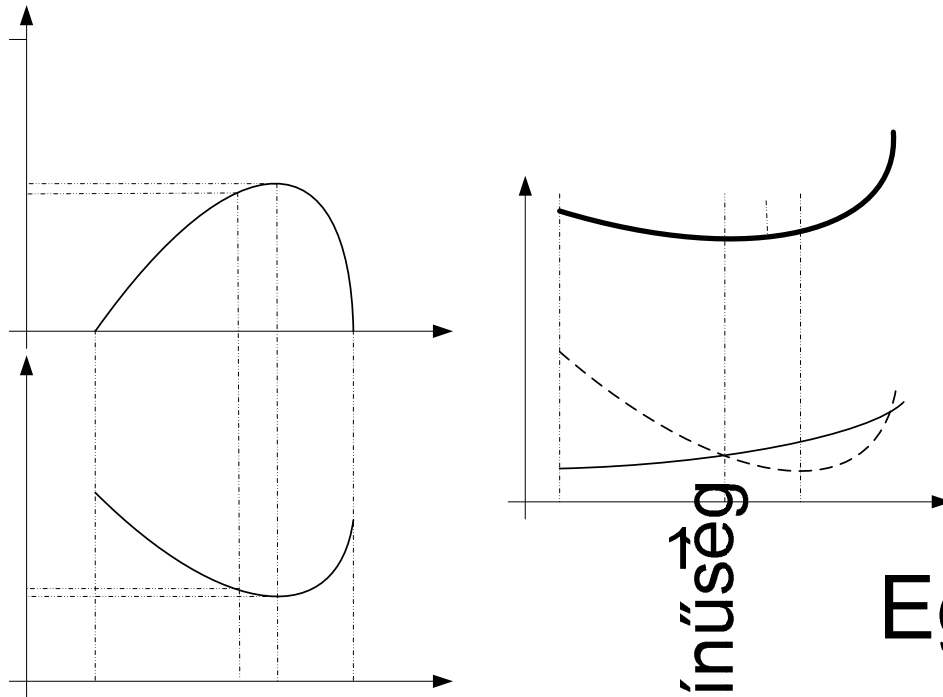
1. Elsőként megemlíthetjük, hogy a normál megvalósítású program összköltsége általában nincs rajta az (össz)költségoptimalis görbén. (Megjegyzem, hogy ez valamennyi sztochasztikus költségoptimaló módszer esetében megfigyelhető. Ennek megoldása minden esetben az alábbiakban bemutatott módszer segítségével történhet.)

Ez a következők miatt van így:

- a. A determinisztikus időtervezés esetén egy tevékenység normál időtartamának azt az időt választjuk, ahol a megvalósításához szükséges (változó) költség a legkisebb. Ha minden tevékenységnél így járunk el, akkor egy olyan (ún.

normál átfutási idejű) programot kapunk, ahol a tevékenységek megvalósításához tartozó összes változó költség minimális.

- b. A **PERT-módszernél** legtöbb esetben három adatból (legvalószínűbb időtartam, optimista- és pesszimista becslés) becsüljük meg a tevékenység várható időtartamát. Ez viszont egyáltalán nem biztos, hogy minimális változó költséggel jár. Ugyanez igaz sajnos más költségminimalizáló eljárásokra, ahol első lépés a tevékenységek várható időtartamának megbecslése. Ez viszont egyáltalán nem szükségszerűen a minimális változó költséggel járó normál időtartam.
 - c. Ha a **PERT-** és **CPM-**, vagy **MPM-módszer** tevékenységidőtartam-becslését összevetjük, akkor láthatjuk, hogy a **CPM-**, **MPM-módszernél** gyakran úgy becsülünk, hogy a tevékenység normál időtartamának a legvalószínűbb időtartamot választjuk. (Ez persze nem minden esetben teljesül, ellenben a legtöbbször a **CPM**, **MPM** által becsült tevékenység időtartamok nem fognak megegyezni a **PERT-módszer** által becsült várható tevékenység időtartamokkal.)
 - d. A legtöbb esetben a **PERT-módszer** által meghatározott eloszlás nem szimmetrikus, és gyakran fennáll a következő egyenlőtlenség: $a_{i,j} < t_{i,j} < m_{i,j} < b_{i,j}$.
2. Ha feltesszük az 1-es pontban leírtakat, akkor az ábrákból könnyen látható, hogy a **PERT-módszernél** alkalmazott becslések miatt a program összköltsége (általában) nem lesz rajta az összköltségoptimális görbén (ha például a **PERT-módszer** által kiszámolt várható időtartamok nem egyeznek meg a **CPM-**, **MPM-módszerek** által meghatározott normál időtartamokkal. Ugyanez figyelhető meg más költségoptimális megoldások esetében is.)



2.8.2.1-1 ábra: költségek és várható tartamok kapcsolata determinisztikus költségtervezés esetén

(Béta-e

3. Abban az esetben sem illeszkedik a **PERT-módszer** által becsült időtartamokból számolt projekt átfutási idejének költségvonzata az összköltségoptimalis görbére, ha az egyes tevékenységekre a **CPM/MPM-módszer** által becsült legvalószínűbb normál időtartam kisebb, mint a **PERT-módszer** által meghatározott várható (normál) időtartam.
4. Ha feltételezzük az egyes, illetve a kettes pontban leírtakat az egyes tevékenységekre, akkor egy – a gyakorlatban használt – módszer segítségével csökkenthetjük a program (össz)költségét, anélkül hogy a program várható átfutási idejét megnövelnénk. Ennek a módszernek a lényege, hogy az alternatív úton lévő tevékenységek szabad tartalékidejét csökkentjük (vagyis a tevékenységek várható időtartamát növeljük). Ez (feltételezve 1., 2. pontokban leírtakat) (össz)költségcsökkenéssel jár. (Ekkor kiszámítható a szórás (standard bizonytalanság), és a módusz változása.) Ezzel a módszerrel a kritikus utak száma megnőhet, ilyenkor a belépő kritikus utak miatt a program várható átfutási ideje is bizonytalanabbá válik. Ezért nem növeljük a tevékenységek tartalékidejét a szabad tartalékidőn felül, hiszen akkor szélsőséges esetben a teljes program összes tevékenysége kritikus úton lévő tevékenység lehet.

Ezzel a módszerrel tehát költséget takaríthatunk meg, anélkül hogy a várható átfutási idő növekedne.

5. Ennél egy jobb módszer, ami azt az esetet is figyelembe veszi, ahol az egyes tevékenységek változó költsége, illetve várható időtartama magasabb, mint a (legvalószínűbb) normál időtartam, illetve (az ehhez az értékhez tartozó) normál költség. Itt az elsőrendű cél a normál átfutási idő elérése. Ekkor az egyes tevékenységek várható időtartamait úgy módosítjuk, hogy az minimális változó költséggel járjon. A várható értéket tehát erre az értékre kell módosítani. Ha a minimális változó költséghez tartozó időtartam éppen a legvalószínűbb időtartam, akkor a tevékenységek várható értékét erre az értékre kell módosítani. (A 2.8.2.1-2 – 2.8.2.1-5 egyenletek megadják a szórások illetve az időtartamok változásának számításához használható képleteket).

Ezzel a módszerrel tehát szintén változó költséget takaríthatunk meg. Az így meghatározott átfutási idővel megvalósított program minimális változó költséggel fog rendelkezni, és meg fog egyezni a **CPM**-, **MPM**-módszerekkel kiszámított normál átfutási idővel.

A továbbiakban feltesszük, hogy az időtartamok és a változó költségük között függvénykapcsolat áll fenn. Az előzőekben vizsgált mennyiségek $a_{i,j}$, $m_{i,j}$, $b_{i,j}$, $t_{i,j}$ és $\sigma_{i,j}$ most $vc_{i,j}$ változó költség függvényei.

A **PERT/COST-módszerben** feltesszük, hogy az alábbi $(r_{1i,j}, r_{2i,j})$ arányok nem változnak a költségnövekedés hatására:

$$r_{1i,j} = \frac{a_{i,j}}{t_{i,j}}, \quad r_{2i,j} = \frac{b_{i,j}}{t_{i,j}}. \quad (2.8.2.1-1)$$

Az alábbi mennyiségek tehát a következőképpen változnak (a változó mennyiségeket $\sim$ - jellel jelöltem):

Szórásnégyzet (standard bizonytalanságnégyzet) változása:

$$\tilde{u}^2(\tilde{t}_{i,j}) = \tilde{\sigma}_{i,j}^2 = \left[\frac{1}{6} (\tilde{b}_{i,j} - \tilde{a}_{i,j}) \right]^2 = \frac{1}{36} (r_{2i,j} - r_{1i,j})^2 \tilde{t}_{i,j}^2. \quad (2.8.2.1-2)$$

Tehát, ha változó költség növekedése árán csökkentjük a $t_{i,j}$ átlagos időtartamot, akkor ezzel egyenes arányban csökken a standard bizonytalanság (tevékenység szórása) is. Az arányossági tényező pedig: $\frac{1}{6}(r_{2,i,j} - r_{1,i,j})$.

Úgyszintén arányos a $t_{i,j}$ átlagos időtartammal a módusz változása:

$$\tilde{m}_{i,j} = \frac{1}{4} \left[6 - (r_{1,i,j} - r_{2,i,j}) \right] \cdot \tilde{t}_{i,j}. \quad (2.8.2.1-3)$$

Az optimista, illetve pesszimista becslések változása az alábbi módon írható fel:

$$\tilde{a}_{i,j} = r_{1,i,j} \tilde{t}_{i,j} \quad (2.8.2.1-4)$$

$$\tilde{b}_{i,j} = r_{2,i,j} \tilde{t}_{i,j} \quad (2.8.2.1-5)$$

Az eloszlás paramétereinek becslése:

$$\alpha \approx 6 \frac{1 - r_{1,i,j}}{r_{2,i,j} - r_{1,i,j}} \quad (2.8.2.1-6)$$

$$\beta \approx 6 \frac{r_{2,i,j} - 1}{r_{2,i,j} - r_{1,i,j}}$$

Látható, hogy az eloszlás paraméterei a várható időtartamok csökkenésétől függetlenek, így a tartamok csökkenése esetén is (feltételezve, hogy r_1, r_2 arányok állandóak) azonos paraméterű (azonos „alakú”) eloszlással becsülhetők az időtartamok (egy adott biztonsági szintet feltételezve).

Más költségoptimalizációs módszer alkalmazása esetén is a végeredményként egy összköltségminimális, vagy egy minimális átfutási idejű programot kapunk. Megkapjuk továbbá a tevékenységek adott költségshinthez tartozó (várható) legkorábbi/legkésőbbi kezdését/befejezését, illetve ezek standard bizonytalanságát.

2.8.2.2 Sztochasztikus költségtervezés sztochasztikus időtervezés esetén

A tervezés során általában nem csak az egyes tevékenységek várható időtartamáról, de a várható költség-, illetve erőforrásigényeiről sincs biztos információnk. Általában ezeket is csak becsülni tudjuk. Hogy a költségek, illetve erőforrásigények becslésénél miért nem terjedt el az időbecsléshez hasonló sztochasztikus modell, annak számos oka van.

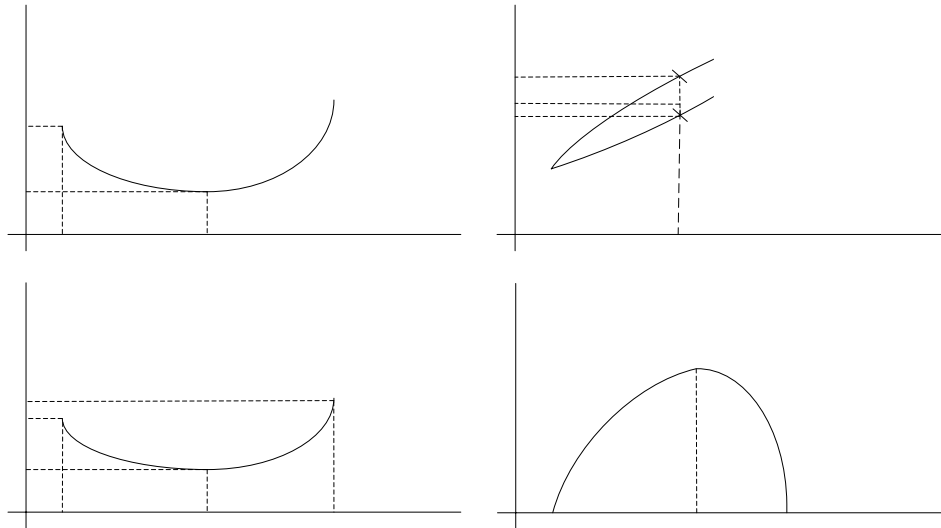
1. A költségek, illetve erőforrásigények becslésére nincs általánosan elfogadott eloszlásfüggvény (mint például projektek esetén időtartamok becslésére a β -eloszlás).
2. Változó költségek, illetve a fixköltségek valószínűleg más és más eloszlást követnek.
3. Erőforrásigény-típusok (megújuló, nem megújuló) erőforrások szintén valószínűleg más és más eloszlást követnek.

Látható a fenti nehézségek miatt, projektek esetén a költségek, illetve az erőforrásigények bizonytalanságát nagyon nehéz megbecsülni.

Termelési programok esetén azonban használhatjuk a 2.8.2.1 példában leírtakat. A gyártás során feljegyezzük a tevékenységek időtartamait, költség- és erőforrásigényeit. Ebből becsülhető a tevékenységek időtartamainak várható értéke, és az átlag szórása (standard bizonytalansága), illetve a program várható átfutási ideje és a várható átfutási idő szórása (összetett standard bizonytalansága). Becsülni kell továbbá azt, hogy a kritikus úton lévő tevékenységek közül melyik tevékenységet mennyi idővel lehet csökkenteni, illetve hogy ez az időtartam csökkenés mennyi változó költség növekedéssel jár. Ha ezek közül ugyanúgy, ahogy a **CPM/COST**-, **MPM/COST**- vagy a **PERT/COST-módszernél** a legkisebb (változó)költség-növekedéssel járó tevékenység időtartamát csökkentjük, és így módosítjuk a gyártást, akkor több gyártás esetén szintén becsülhetjük az egyes tevékenységek időtartamait, a várható időtartam szórását, a várható átfutási időt, illetve annak szórását, valamint a becslés (standard) bizonytalanságát. Ezt a folyamatot addig ismételhetjük, ameddig a kívánt átfutási időt (valamely biztonsági szint mellett) el nem érjük.

Ha más információnk nincs az eloszlásokat illetően, akkor projektek esetén várható érték, illetve szórás becslésére az előző megvalósított projektek szolgáltathatnak információt. Hasonló tevékenységek esetén becsülhető a tevékenységek időtartamának várható értéke illetve a szórása. Vagyis a gyakorlatban ez azt jelenti, hogy amennyiben adott tevékenységeket elvégeztünk korábban más projektek keretében, akkor az ott szerzett tapasztalatokat (időtartam, költségigény, erőforrás-szükséglet) feljegyezve, a 2.8.1.1-es példa alapján megbecsülhetjük a várható időtartamot, a költségigényt és az erőforrás-szükségletet, illetve meghatározhatjuk ezen adatok standard bizonytalanságát. Minél több hasonló tevékenységet végeztünk el korábbi projektek megvalósítása során, annál pontosabban tudjuk ezeket az adatokat (időtartamot, költség- és erőforrásigényt) meghatározni.

Általában a következő összefüggések állapíthatók meg egy projekt vagy egy kissorozatgyártás termelésirányításában.



2.8.2.2-1 ábra: egy tevékenység időtartamának és költségigényének kapcsolata

A tevékenységek időtartamának adott költség szint melletti valószínűségének sűrűségfüggvényét a 2.8.2.2-1 (d) ábra mutatja (β -eloszlás feltételezve). Valamennyi valószínűségi szint mellett elmondható, hogy a normál időtartam esetén minimális változó költség-igény (2.8.2.2-1 (a) ábra). Ha ennél az időnél rövidebb idő alatt akarjuk végrehajtani a tevékenységet, akkor az általában csak a költség növekedésével érhető el. Ebben az esetben viszont az optimista és pesszimista időtartamok is változnak (2.8.2.2-1 (b) ábra). Egy adott költség szint mellett a legvalószínűbb időtartamtól való extrém eltérések általában szintén költségvonzzattal járhatnak.

Valamennyi tevékenységre meghatározva a költség optimalis időtartamot meg lehet mondani, hogy mely tevékenységek időtartamait kell csökkenteni, illetve növelni. Meg lehet továbbá határozni a minimális változó költséggel járó átfutási időt, hiszen a minimális változó költséggel járó időtartamok esetén az átfutási idő is minimális változó költséggel fog járni.

Ezenkívül közelítéssel kapunk minden tevékenységre egy olyan költség-idő függvényt, mely segítségével a programot esetlegesen rövidíthetjük. Ha a fixköltségeket is mérjük adott időszakban, akkor olyan programot határozhatunk meg, amellyel minimális összköltség érhető el.

változó költség

$vc(n)$

$vc(n)$

r

változó költség

(a)

Tevékenység
idejének
költségigénye
kapcsolata

(c)

Változó költség
különböző le
eset

Erőforrás-tervezést ezután lehet elvégezni. Ekkor meghatározható egy adott valószínűségi szintre (pl. legvalószínűbb átfutási idő) a lehető legkisebb összköltséggel járó optimális erőforrás-allokáció.

2.8.3 Bizonytalanság kezelése kissorozatgyártás termelésirányításában

Sok esetben a projektekhez hasonlóan lehet a kissorozatgyártás termelésirányítását is kezelni. Amíg azonban a projektek esetén a tevékenységidők, költség- és erőforrásigények jórészt csak becsülhetők, addig itt a tevékenységek időtartamának, költségigényének és erőforrás-szükségleteinek becslésére a korábbi termelés adatait is felhasználhatjuk. A következőkben bemutatjuk, hogyan becsülhetők meg statisztikai módszerekkel e paraméterek várható értékei, szórásai, valamint ebből hogyan becsülhetjük meg az adott üzemben a várható termelést. [228]

2.8.3.1 Paraméterek becslése

A tevékenységek időtartamait, költségigényeit és erőforrás-szükségleteit az ISO Guide to the Expression of Uncertainty in Measurement 1993-ban kiadott segédlete alapján becsüljük. Ez a segédlet többfajta bizonytalanságot különböztet meg. Becsléseink során ezeket a bizonytalansági fogalmat fogjuk használni.

2.8.3.1.1 Példa az időtartamok becslésére

Egy termelő vállalat nyílászárókat készít. A tevékenységeinek (percekben mért) időtartamát feljegyzi, és a következő megállapításokat teszi (tapasztalatai alapján). Naponta 75-80 nyílászárót tud elkészíteni. 25 munkása van a különböző munkák elvégzéséhez. Kérdés az, hogy hogyan tudja fokozni a termelését, ha tudjuk, hogy a tevékenységidők a legvalószínűbb időtartamtól maximum $-10 - +30\%$ között tér(het)nek el. A költségigényekkel most még nem foglalkozunk. Az erőforrás-szükségletet pedig adott tevékenység esetén konstansnak vesszük. Az egyszerűség kedvéért tételezzük fel, hogy a tevékenység időtartama, mint valószínűségi változó β -eloszlást követ. Feltételezzük továbbá, hogy a tevékenységeket egymás után sorban hajtjuk végre.

Srsz.	Tevékenység megnevezése	a	b	m	t	σ^2	r_1	r_2	α	β
1	Darabolás	9,0	13,0	10,0	10,3	0,44	0,87	1,26	2	4
2	Víz kifolyó és kilincshely kimaratása	9,0	13,0	10,0	10,3	0,44	0,87	1,26	2	4
3	Merevítő vas vágása	4,5	6,5	5,0	5,2	0,11	0,87	1,26	2	4
4	Csavarozás	4,5	6,5	5,0	5,2	0,11	0,87	1,26	2	4
5	Fő szerkezeti egységek összeállítása és hegesztése	6,3	9,1	7,0	7,2	0,22	0,87	1,26	2	4
6	Hegesztés utáni sorjázás	5,4	7,8	6,0	6,2	0,16	0,87	1,26	2	4
7	Vasalatok és pántok szerelése (szárny)	13,5	19,5	15,0	15,5	1,00	0,87	1,26	2	4
8	Vasalatok és pántok szerelése (tok)	13,5	19,5	15,0	15,5	1,00	0,87	1,26	2	4
9	Üvegezés (panel), üveglécezés	6,8	9,8	7,5	7,8	0,25	0,87	1,26	2	4
10	Funkcionális elemek beállítása és ellenőrzése	3,6	5,2	4,0	4,1	0,07	0,87	1,26	2	4
Σ	Összesen	76,1	109,9	84,5	87,3	3,8				

2.8.3.1.1-1 táblázat: tevékenységek várható időtartamának, standard bizonytalanságának illetve a β -eloszlás paramétereinek becslése

Ha az egyes tevékenységek várható időtartama, mint valószínűségi változó β -eloszlást követ, akkor használjuk a **PERT-módszert** az egyes paraméterek becslésére. Ekkor a legyen az optimista becslés (itt a tevékenység legvalószínűbb időtartamának m 90%-a). Legyen b a pesszimista becslés (itt a tevékenység legvalószínűbb időtartamának m 130%-a).

Az előző fejezet egyenleteit felhasználva kiszámíthatók a fenti paraméterek (várható érték, szórás, eloszlás paraméterei stb.)

Ha a fenti paraméterek ismertek, meghatározható az átfutási idő (TPT) várható értéke $E(TPT)$, illetve szórása σ_{TPT}^2 (összetett standard bizonytalansága $u_c^2(TPT)$).

$$TPT := E(TPT) = \sum_{i=1}^M E(t_i) := \sum_{i=1}^M t_i, \quad (2.8.3.1.1-1)$$

$$\sigma_{TPT}^2 = u_c^2(TPT) = \sum_{i=1}^M \sum_{j=1}^M \frac{\partial f}{\partial t_i} \frac{\partial f}{\partial t_j} u(t_i, t_j) = \sum_{i=1}^M \left(\frac{\partial f}{\partial t_i} \right)^2 u^2(t_i) + 2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M \frac{\partial f}{\partial t_i} \frac{\partial f}{\partial t_j} u(t_i, t_j). \quad (2.8.3.1.1-2)$$

A számítás megkönnyítése érdekében tegyük fel a következőket:

1. f egy egyszerű összegző függvény, tehát az átfutási idő nem lesz más, mint a kritikus úton lévő tevékenységek (itt az összes tevékenység) időtartamának összege. Ekkor a deriváltak 1-ek lesznek.
2. A tevékenységek között nincs korreláció. Ekkor az összeg második fele 0.

Így az összefüggés a következőképpen egyszerűsödik:

$$\sigma_{TPT}^2 = u_c^2(TPT) = \sum_{i=1}^M u^2(t_i) = \sum_{i=1}^M \sigma_i^2. \quad (2.8.3.1.1-3)$$

Tehát az átfutási idő szórásnégyzete nem lesz más, mint az időtartamok szórásnégyzeteinek összege. Látható azonban, hogy a modell akkor is alkalmazható, ha a tevékenységek között létezik korreláció.

Meghatározható továbbá adott valószínűségi szintek (90%, 95%, 99%) mellett, mennyi lesz legrosszabb esetben a tevékenységek időtartama.

Srsz.	Tevékenység megnevezése	<i>m</i>	<i>t</i>	<i>t_{p=0,9}</i>	<i>t_{p=0,95}</i>	<i>t_{p=0,99}</i>	<i>t_{p=0,9}</i>	<i>t_{p=0,95}</i>	<i>t_{p=0,99}</i>	<i>p_{p=0,9}</i>	<i>p_{p=0,95}</i>	<i>p_{p=0,99}</i>
1	Darabolás	10,0	10,3	11,3	11,6	12,1	10,6	10,7	10,9	0,68	0,71	0,77
2	Vízkefolyó és kilincshely kimaratása	10,0	10,3	11,3	11,6	12,1	10,6	10,7	10,9	0,68	0,71	0,77
3	Merevítő vas vágása	5,0	5,2	5,7	5,8	6,1	5,3	5,4	5,4	0,68	0,71	0,77
4	Csavarozás	5,0	5,2	5,7	5,8	6,1	5,3	5,4	5,4	0,68	0,71	0,77
5	Fő szerkezeti egységek összeállítása és hegesztése	7,0	7,2	7,9	8,1	8,5	7,4	7,5	7,6	0,68	0,71	0,77
6	Hegesztés utáni sorjázás	6,0	6,2	6,8	7,0	7,3	6,4	6,4	6,5	0,68	0,71	0,77
7	Vasalatok és pántok szerelése (szárny)	15,0	15,5	17,0	17,4	18,2	15,9	16,1	16,3	0,68	0,71	0,77
8	Vasalatok és pántok szerelése (tok)	15,0	15,5	17,0	17,4	18,2	15,9	16,1	16,3	0,68	0,71	0,77
9	Üvegezés (panel), üveglécezés	7,5	7,8	8,5	8,7	9,1	8,0	8,0	8,2	0,68	0,71	0,77
10	Funkcionális elemek beállítása és ellenőrzése	4,0	4,1	4,5	4,7	4,8	4,3	4,3	4,3	0,68	0,71	0,77
Σ	Összesen	84,5	87,3	95,8	98,3	102,3	89,82	90,53	91,86			

2.8.3.1.1-2 táblázat: tevékenységek időtartamai különböző biztonsági szintek esetén

Ebben az esetben *p* a tevékenységekhez tartozó biztonsági szintet jelöli. Tehát pl. *t_{p=0,9}* azt jelenti, hogy 90%-os biztonsági szint mellett az esetek 90%-ában a tevékenység legalább ennyi idő alatt végre fog hajtódni. Minél magasabb ez a biztonsági szint, annál nagyobb a (becsült) időtartam. Ha egy tevékenység csúszni is fog, de a közöttük lévő korreláció 0, annak az esélye, hogy minden tevékenység csússzon, elég kicsi. Ezért itt a tevékenységek 90%-os biztonsági szint melletti időtartamainak egyszerű összegzése helyett célszerű – az átfutási idő összetett standard bizonytalanságát felhasználva – valamely biztonsági szintre meghatározni az átfutási időt. *P* ezt a valószínűséget jelöli. *t_{p=0,9}* tehát azt jelenti, hogy ha az átfutási idő az esetek 90%-ában legfeljebb *TPT_{p=0,9}*=89,82 óra, akkor átlagosan *t_{p=0,9}* lesz a tevékenység időtartama. Az esetek *p_{p=0,9}*=68%-ában egy adott tevékenység időtartama nem lesz nagyobb ettől a *t_{p=0,9}* értéktől.

Srsz.	a	b	m	t	σ^2	r	r_{max}	n	r_n	r_m	t_n	$t_{np=0,9}$	$t_{np=0,95}$	$t_{np=0,99}$	$t_{nP=0,9}$	$t_{nP=0,95}$	$t_{nP=0,99}$
1	9,0	13,0	10,0	10,3	0,44	2	4	2	4	0	5,17	5,67	5,81	6,06	5,31	5,36	5,44
2	9,0	13,0	10,0	10,3	0,44	2	4	2	4	0	5,17	5,67	5,81	6,06	5,31	5,36	5,44
3	4,5	6,5	5,0	5,2	0,11	1	3	1	1	2	5,17	5,67	5,81	6,06	5,31	5,36	5,44
4	4,5	6,5	5,0	5,2	0,11	1	3	1	1	2	5,17	5,67	5,81	6,06	5,31	5,36	5,44
5	6,3	9,1	7,0	7,2	0,22	1	3	2	2	1	3,62	3,97	4,07	4,24	3,72	3,75	3,80
6	5,4	7,8	6,0	6,2	0,16	1	4	2	2	2	3,10	3,40	3,49	3,63	3,19	3,21	3,26
7	13,5	19,5	15,0	15,5	1,00	1	5	4	4	1	3,88	4,25	4,36	4,54	3,99	4,02	4,08
8	13,5	19,5	15,0	15,5	1,00	1	8	4	4	4	3,88	4,25	4,36	4,54	3,99	4,02	4,08
9	6,8	9,8	7,5	7,8	0,25	2	6	1	2	4	7,75	8,50	8,72	9,08	7,97	8,03	8,15
10	3,6	5,2	4,0	4,1	0,07	1	23	1	1	22	4,13	4,53	4,65	4,84	4,25	4,29	4,35
Σ	76,1	109,9	84,5	87,3	3,8			25									

2.8.3.1.1-3 táblázat: tevékenységek várható időtartamai, erőforrás-szükségletei

A feladatban ismert minden tevékenység erőforrásigénye r . Legyen ez az erőforrás a tevékenységek végrehajtásához szükséges munkások száma. n legyen a gépek száma tevékenységenként. Ekkor egy adott tevékenység összes erőforrásigénye: $r_n = r \cdot n$. Ez az érték nem lehet magasabb, mint az adott tevékenységet elvégezni tudó dolgozók száma: r_{max} .

r_m legyen a maradék erőforrás, tehát: $r_m := r_{max} - r_n$. t_n az egységnyi időtartam, amely megmutatja, hogy n gép esetén egy gépen átlagosan mennyi lesz egy adott tevékenység végrehajtási ideje. Ekkor $t_n := t / n$. Ennek maximuma ($t_{nmax} := \max(t_n)$) lesz a szűk keresztmetszet (ebben a példában a dőlt betűkkel szedett 9. sorszámú tevékenység). Ha adott a munkaidő, illetve a műszakok száma, akkor kiszámítható, hogy átlagosan mennyit lehet termelni egy munkanapon.

$TPT_{P=0,5}$	87,32	$Q_{P=0,5}$	77,4	
$TPT_{P=0,9}$	89,82	$Q_{P=0,9}$	75,3	$Q_{p=0,9}$ 74,3
$TPT_{P=0,95}$	90,53	$Q_{P=0,95}$	74,7	$Q_{p=0,95}$ 73,5
$TPT_{P=0,99}$	91,86	$Q_{P=0,99}$	73,6	$Q_{p=0,99}$ 72
σ_{TPT}	1,95			
munkaórák száma	10			
műszakok száma	1			

2.8.3.1.1-4 táblázat: különböző biztonsági szinthez tartozó átfutási idő, illetve megtermelhető termékek mennyisége

50%-hoz tartozó átfutási idő a várható átfutási idő. Feltételezhetjük a központi határeloszlás tételéből fakadóan, hogy elegendően sok tevékenység esetén az átfutási idő,

mint valószínűségi változó normális eloszlást követ. Ekkor a várható megtermelhető termékek száma: $Q_{P=0,5} = \text{műszakok száma (itt 1)} * \text{munkaórák száma (itt 10)} * 60 \text{ (perc)} / TPT_{P=0,5}$.

Hasonlóan kiszámítható Q értéke a többi biztonsági szinten is. Így pl. az esetek 90%-ában ($Q_{P=0,9} = 1 * 10 * 60 (=600) / TPT_{P=0,9} = 75.3$ nyílászáró/nap) ettől kevesebb terméket nem fogunk termelni.

Menedzseri, vezetői döntést befolyásoló döntés lehet, hogy a kritikus tevékenység esetleges csúszása hogyan befolyásolja a megtermelhető tevékenységek számát. Ezt mutatja meg a Q_p érték. Kiszámításakor m db termelés esetén szintén feltételezhetjük, hogy az összes megtermelt mennyiség t_{nmax} várható értékű $\sigma_{t_{nmax}}$ szórású normális eloszlást követ. Ennek megfelelően pl. $Q_{p=0,9}$ kiszámítása a következőképpen történik.

$$Q_{p=0,9} = \frac{600}{N_{p=0,9}^{-1}(t_{nmax}, \sigma_{t_{nmax}})} = 74,3 \text{ nyílászáró/nap, ahol a } N_p^{-1}(\cdot) \text{ függvény a } p=0,9$$

valószínűségi értékhez tartozó t_{nmax} középvértékű, $\sigma_{t_{nmax}}$ szórású inverz normális eloszlás.

Elemzés:

Az elemzés során fontos megállapításokat tehetünk:

1. Várhatóan 77,4 darab nyílászárót fogunk tudni egy nap alatt megtermelni. Ez az előzetes becslésekkel teljesen szinkronban van.
2. Elmondható, hogy 72-nél kevesebbet 99%-os valószínűséggel egyetlen nap sem termelünk.
3. Csak úgy tudunk egy nap többet termelni, ha az időtartamokat csökkentjük és/vagy több munkagép segítségével a termelési folyamatot párhuzamosítjuk. Ez utóbbi viszont az erőforrásigény növekedésével, egyúttal költségnövekedéssel járhat.
4. Ahhoz, hogy eldöntsük, van-e értelme párhuzamosítani, költségelemzést kell végezni.
5. Csak olyan tevékenységeket célszerű párhuzamosítani, amelynek egységnyi (várható) időtartama magas. Jelen esetben ez a 9., illetve esetlegesen az 1-4. tevékenységek lennének.

2.8.3.1.2 Változó költség és az időtartam kapcsolatának meghatározása

Ha maradunk az előzőekben vázolt termelési struktúránál, akkor is célszerű a termelést a lehető legkisebb költséggel végrehajtani. Ehhez azonban fontos, hogy felderítsük az

időtartamok, illetve a költségigények kapcsolatát. Ez a kapcsolat lehet determinisztikus vagy sztochasztikus. Ha kitöltünk az alábbi ábrához hasonló táblázatot, meghatározott (pl. heti, kétheti, havi) rendszerességgel, akkor kellően sok táblázat kitöltése esetén (legalább 10) statisztikai következtetéseket vonhatunk le időtartamok illetve költségek, esetleg az időtartamok és erőforrásigények között.

Feldolgozott időszak:		Lefutási idők			Változó költségek						Erőforrásigény					
Srsz.	Tevékenység megnevezése	a	b	m	vc(a)	vc(b)	vc(m)	vc _{min}	vc _{max}	vc _{átlag}	r(a)	r(b)	r(m)	r _{min}	r _{max}	r _{átlag}
1	Darabolás															
2	Vízki folyó és kilincshely kimaratása															
3	Merevítő vas vágása															
4	Csavározás															
5	Fő szerkezeti egységek összeállítása és hegesztése															
6	Hegesztés utáni sorjázás															
7	Vasalatok és pántok szerelése (szárm)															
8	Vasalatok és pántok szerelése (tok)															
9	Üvegezés (panel), üveglécezés															
10	Funkcionális elemek beállítása és ellenőrzése															
Σ Összesen											Összes fix költség					

2.8.3.1.2-1 táblázat: kitöltendő táblázat

A tevékenységek időtartamai közül **a**-oszlopba azt az időtartamot írjuk, ami az adott időszakban a legkisebb volt. Ennek változó költsége **vc(a)**, erőforrás-szükséglete pedig **r(a)**. Ezeket szintén írjuk a megfelelő oszlopba. Hasonlóan az előzőkhöz, a tevékenység leghosszabb időtartamát **b**, leggyakrabban előforduló időtartamát **m**, ezek költségét **vc(b)**, **vc(m)** és erőforrás-szükségletét **r(b)**, **r(m)** is írjuk a megfelelő oszlopba. Fontos lehet, hogy átlagosan, minimálisan és maximálisan mennyi volt a költség- és erőforrásigény.

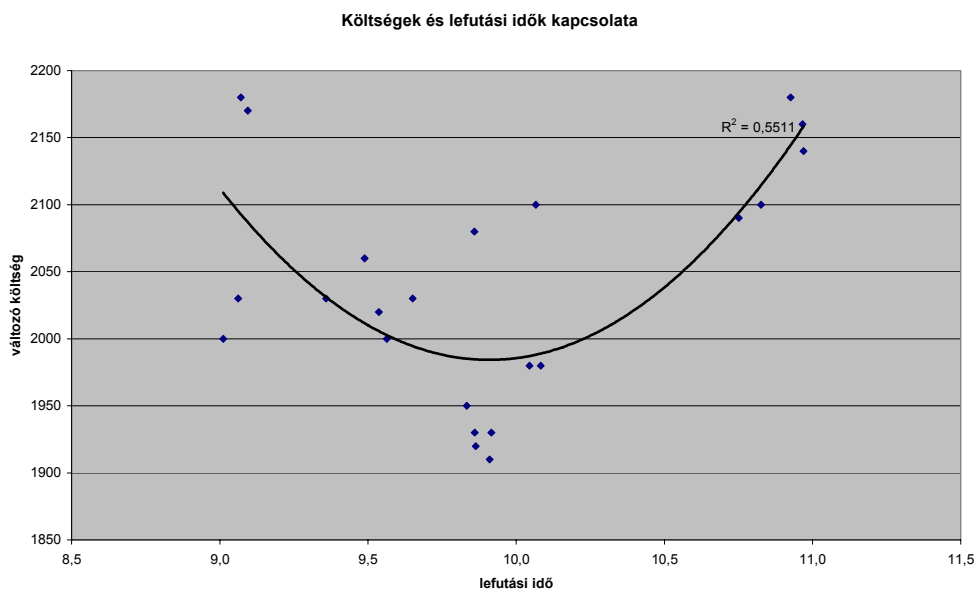
Ha a tevékenységek száma kicsi, akkor egy egyszerű táblázatkezelő segítségével magunk is elemezhetjük az adatokat. Ha a tevékenységek száma jelentősebb, akkor a problémára egyszerű szoftver készíthető. Az adatokat rendezzük a legvalószínűbb időtartam szerint.

A következő tevékenységet elemzésünkhöz egy hosszabb átfutási idejű kissorozatgyártásból vettük. (Az időadatok órákban, a költségadatok euróban értendők.) Egy tevékenység elemzésénél a következő táblázatot kapjuk. (Az adatokat a legvalószínűbb időtartam szerint sorba rendeztük.)

Srsz.	a	b	m	vc(a)	vc(b)	vc(m)	vc _{min}	vc _{max}	vc _{átlag}	r(a)	r(b)	r(m)	r _{min}	r _{max}	r _{átlag}
7	8,9	9,1	9,0	2100	2060	2000	1980	2120	2010	4	4	4	4	4	4
18	8,9	11,8	9,1	2160	2100	2030	2020	2160	2040	4	4	4	4	4	4
16	8,7	11,5	9,1	2250	2210	2180	2180	2260	2180	4	4	4	4	4	4
5	8,8	11,3	9,1	2250	2210	2170	2150	2260	2190	4	4	4	4	4	4
22	8,8	10,0	9,4	2150	2060	2030	2010	2160	2040	4	4	4	4	4	4
6	8,7	10,1	9,5	2130	2090	2060	2060	2140	2070	4	4	4	4	4	4
23	8,9	9,9	9,5	2120	2100	2020	2010	2130	2040	4	4	4	4	4	4
2	9,0	10,8	9,6	2150	2080	2000	1980	2170	2000	4	4	4	4	4	4
8	9,1	10,2	9,7	2180	2130	2030	2020	2200	2030	4	4	4	4	4	4
11	9,1	10,1	9,8	2070	2040	1950	1940	2080	1950	4	4	4	4	4	4
4	9,8	10,2	9,9	2140	2150	2080	2070	2160	2100	4	4	4	4	4	4
10	9,7	10,7	9,9	2060	2020	1930	1910	2070	1950	4	4	4	4	4	4
9	9,7	11,9	9,9	1990	1950	1920	1910	2000	1920	4	4	4	4	4	4
12	9,4	11,0	9,9	2030	2000	1910	1910	2040	1930	4	4	4	4	4	4
15	9,0	12,8	9,9	2020	1970	1930	1920	2020	1950	4	4	4	4	4	4
1	9,3	11,1	10,0	2050	2010	1980	1960	2070	1980	4	4	4	4	4	4
3	9,8	11,1	10,1	2160	2160	2100	2090	2160	2110	4	4	4	4	4	4
19	10,0	12,0	10,1	2120	2030	1980	1980	2130	1990	4	4	4	4	4	4
13	10,7	11,7	10,8	2150	2120	2090	2080	2150	2090	4	4	4	4	4	4
21	10,1	13,7	10,8	2160	2180	2100	2090	2180	2120	4	4	4	4	4	4
17	10,0	13,6	10,9	2300	2200	2180	2160	2300	2200	4	4	4	4	4	4
14	10,1	13,3	11,0	2220	2250	2160	2150	2260	2170	4	4	4	4	4	4
20	10,2	12,1	11,0	2290	2170	2140	2140	2310	2140	4	4	4	4	4	4
Átlag	9,43	11,30	9,90	2141,30	2099,57	2042,17	2031,30	2153,48	2052,17	4	4	4	4	4	4

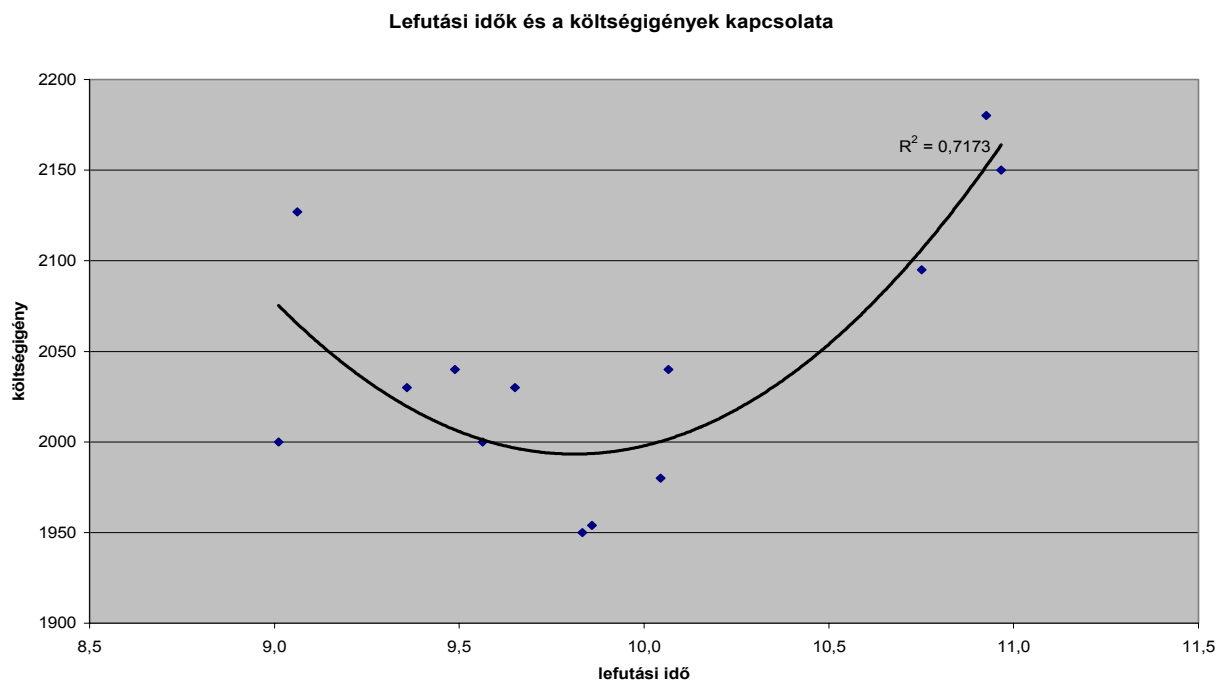
2.8.3.1-2 táblázat: egy tevékenység időtartamának és költség-/erőforrásigénye kapcsolatának meghatározására szolgáló táblázat

A táblázatból azonnal kitűnik, hogy a tevékenységek időtartamai és erőforrásigényük között nincs függvénykapcsolat. A költségek és az időtartamok között viszont megfigyelhető sztochasztikus kapcsolat.



2.8.3.1.2-1 ábra: egy tevékenység időtartama és költség-/erőforrásigényének kapcsolata

Az összefüggés jobban szemléltethető, ha az azonos időtartamokhoz tartozó költségigényeket átlagoljuk.



2.8.3.1.2-2 ábra: egy tevékenység időtartamának és költség-/erőforrásigényének kapcsolata

Mindkét görbére illesztett másodfokú polinom minimuma 9,8 óránál van. Itt most a legvalószínűbb időtartamokra néztük meg az időtartamok és a költségigények kapcsolatát. Ez természetesen elvégezhető a többi, valószínűsíthető időtartamra, illetve az optimista és pesszimista becslésekre is.

Ha azt mondjuk, hogy a normál időtartam legyen a lehető legkisebb változó költséggel járó időtartam, akkor a normál időtartam meghatározható valamennyi biztonsági szintre is.

A kissorozatgyártás sok esetben felfogható úgy is, mint egy kis projekt (azzal a különbséggel, hogy a projektek mindig egyediek, de kissorozatgyártás során is időtartamokkal, költség- és erőforrásigényekkel számolunk, így ebből a szempontból az alkalmazott módszerek is használhatók ebben az esetben is). Hasonló módon lehet itt is meghatározni a költség-optimalis programokat, valamint erre a programra egy optimalis erőforrás-allokációt. Itt azonban a becsléseinket pontosíthatjuk mérésekkel. Elemezhetjük, hogy a termelés vajon a legtöbb esetben minimális költséggel zajlik-e vagy sem. Meghatározhatjuk, hogyan lehet rövidíteni a programot, ez mennyi költségbe kerül. Itt

azonban nem feltételezünk egy adott költség-idő, vagy erőforrás-idő függvényt, hanem a mért adatokból ezt meghatározhatjuk.

A 2.8.2-1 ábra segítségével megoldható az a feladat, hogy adott valószínűségi szintre, pl. a legvalószínűbb esetre a legkisebb költséggel járó időtartamot határozzuk meg. Ez az elemzések szerint ebben a feladatban 9,8 volt. Az átlagban mért 9,9 közel van ehhez az értékhez, ennek ellenére a jelentős szórás miatt az átlagos költségigény messze felette van a minimális költségigénynél. Ezért a termelés során célszerű a minél pontosabb időtartam betartása.

2.8.4 Példa a bizonytalan átfutási idejű projektek erőforrás-tervezésére

Adott vállalat műszaki fejlesztési tervében szerepel a gyártás célgépesítése. Ennek keretében a soron következő feladat egy olyan célgép kialakítása, amelynek kiszolgálását pneumatikus beemelő berendezés könnyíti meg. A vállalatnál néhány folyamat célgépesítését már elvégezték, így megfelelő tájékoztató adatok rendelkezésre állnak az egyes tevékenységek végrehajtási időigényének meghatározására. A végrehajtó létszám ismert, tehát az egyes tevékenységek átfutási ideje becsülhető. A projektet 12 hét (=84 nap) alatt mindenképp be kell fejezni. Ekkor a várható fixköltség 1225 eFt. Az átfutási idő rövidítésével a fixköltségek várhatóan napi 25 eFt-tal csökkenthetők. A számítások megkönnyítése érdekében feltételezzük, hogy az erőforrások közül csak a tevékenységek elvégzéséhez szükséges munkaerőt kell optimalizálni. A projekt kezdetén maximum 24 embert tudunk egyidejűleg foglalkoztatni az egyes tevékenységek elvégzésére. A 8. héttől (56. naptól) már csak maximum 12 embert, míg a 10. héttől (70. naptól) már csak maximum 6 embert tudunk ezen a projekten foglalkoztatni.

A feladat tehát az, hogy legkisebb (várható) (össz)költséggel (legalább 90%-os valószínűséggel) a lehető leghamarabb végezzük el az egyes tevékenységeket úgy, hogy a megadott erőforráskorlátot sehol ne lépjük túl.

Az alábbi táblázatban leírtam a tevékenységek időtartamait (legvalószínűbb, optimista és pesszimista becslés alapján). Feltételeztük, hogy a tevékenységek változó költségei a legvalószínűbb megvalósítási idő mellett minimálisak (normál megvalósítás) $vc(m)$. Feltételezzük továbbá, hogy az optimista becslésnél kevesebb, illetve a pesszimista becslésnél több ideig nem tart a tevékenység megvalósítása. Az optimista becsléshez rendelünk költséget ($vc(a)$) (ez lesz a rohamköltség, vagy másképpen a minimális időtartam mellett

megvalósítható tevékenység költsége). Ugyanígy rendelhetünk a pesszimista becsléshez is költséget, mely szintén magasabb lesz, mint a normál átfutási időhöz tartozó normál költség $vc(m)$. Az egyszerűség kedvéért most feltételezzük, hogy a három paraméter között a változóköltség-igény lineáris. Legyen adott továbbá az optimista, illetve a pesszimista megvalósításhoz tartozó (emberi) erőforrásigény. Az egyszerűsítés miatt most itt is feltételezzük, hogy ezen az intervallumon az erőforrásigény lineáris *(A módszer alkalmazhatósága szempontjából csak azt kell kikötni, hogy a költség-idő függvény konvex legyen, valamint hogy minden pontjában értelmezve legyen. Az erőforrásigény-idő függvényről csak azt kell feltennünk, hogy minden pontjában értelmezve van).* [222, 225, 227]

Jel	Sisz.	Megnevezés	a	b	m	t	σ^2	r_1	r_2	vc(m)	vc(a)	vc(b)	vc(f)	r(a)	r(b)	α	β
A	1	A megoldás műszaki adatainak meghatározása	3	5	4	4	0,111	0,750	1,250	200	250	240	200,00	4	3	3	3
B	2	A célgép megtervezése	23	33	31	30	2,778	0,767	1,100	150	200	180	156,25	5	3	4	1,8
C	3	A célgép legyártása	37	43	40	40	1,000	0,925	1,075	200	290	280	200,00	6	4	3	3
E	4	A célgép helyszíre szállítása	1	1	1	1	0,000	1,000	1,000	120	120	120	120,00	2	2	--	--
D	5	Helykielölés és területfeszabadítás	4	6	5	5	0,111	0,800	1,200	25	30	28	25,00	3	1	3	3
F	6	Alapkészítés	1	3	2	2	0,111	0,500	1,500	50	60	55	50,00	6	4	3	3
M	7	Beton kötése	1	3	2	2	0,111	0,500	1,500	250	300	280	250,00	5	3	3	3
G	8	Gépfelállítás	1	1	1	1	0,000	1,000	1,000	150	150	150	150,00	4	4	--	--
N	9	Csatlakozások bekötése	2	2	2	2	0,000	1,000	1,000	250	250	250	250,00	2	2	--	--
P	10	Próbaüzem	2	4	3	3	0,111	0,667	1,333	75	80	78	75,00	3	3	3	3
V	11	Villamos szerelési tervek készítése	3	5	4	4	0,111	0,750	1,250	75	50	77	75,00	4	4	3	3
H	12	Villamos vezeték- és szerelvényyszerelés	3	5	4	4	0,111	0,750	1,250	25	55	26	25,00	6	4	3	3
S	13	Sűrítettlevegő-vezeték tervezése	3	5	4	4	0,111	0,750	1,250	50	60	55	50,00	1	1	3	3
I	14	Sűrítettlevegő-vezeték készítés és szerelés	4	8	6	6	0,444	0,667	1,333	50	70	60	50,00	6	4	3	3
J	15	Pneumatikus anyagmozgató berendezés tervezése	9	11	10	10	0,111	0,900	1,100	50	70	60	50,00	1	1	3	3
K	16	Pneumatikus anyagmozgató berendezés gyártása	9	17	16	15	1,778	0,600	1,133	55	70	60	57,14	6	4	4,5	1,5
L	17	Pneumatikus anyagmozgató berendezés szerelése	9	11	10	10	0,111	0,900	1,100	50	70	60	50,00	4	3	3	3

2.8.4-1 táblázat: tevékenységjegyzék

Első lépés: A tevékenységek várható értékét úgy kell módosítani, hogy az a minimális változó költséggel járó időtartamra (itt most a móduszra, vagyis a legvalószínűbb időtartamra) essen. Ezzel tehát (várható) változó költséget takarítunk meg. Ekkor a változásokat az alábbi táblázatba lehet összefoglalni. *(A változó mennyiségeket nagybetűvel jelöltük.)*

Jel	Srsz.	A	B	M	T	S2	T(90%)	vc(T)	vc(T(90%))	r(T(90%))
A	1	3,00	5,00	4,00	4	0,111	4,507	200,000	220,269	3,753
B	2	23,77	34,10	32,03	31	2,966	33,171	150,000	182,565	5,034
C	3	37,00	43,00	40,00	40	1,000	41,520	200,000	240,538	5,507
E	4	1,00	1,00	1,00	1	0,000	1,000	120,000	120,000	2,000
D	5	4,00	6,00	5,00	5	0,111	5,507	25,000	26,520	2,507
F	6	1,00	3,00	2,00	2	0,111	2,507	50,000	52,534	5,507
M	7	1,00	3,00	2,00	2	0,111	2,507	250,000	265,202	4,507
G	8	1,00	1,00	1,00	1	0,000	1,000	150,000	150,000	4,000
N	9	2,00	2,00	2,00	2	0,000	2,000	250,000	250,000	2,000
P	10	2,00	4,00	3,00	3	0,111	3,507	75,000	76,520	3,000
V	11	3,00	5,00	4,00	4	0,111	4,507	75,000	76,013	4,000
H	12	3,00	5,00	4,00	4	0,111	4,507	25,000	25,507	5,507
S	13	3,00	5,00	4,00	4	0,111	4,507	50,000	52,534	1,000
I	14	4,00	8,00	6,00	6	0,444	7,013	50,000	55,067	5,507
J	15	9,00	11,00	10,00	10	0,111	10,507	50,000	55,067	1,000
K	16	9,60	18,13	17,07	16	2,023	17,623	55,000	63,115	6,156
L	17	9,00	11,00	10,00	10	0,111	10,507	50,000	55,067	3,753

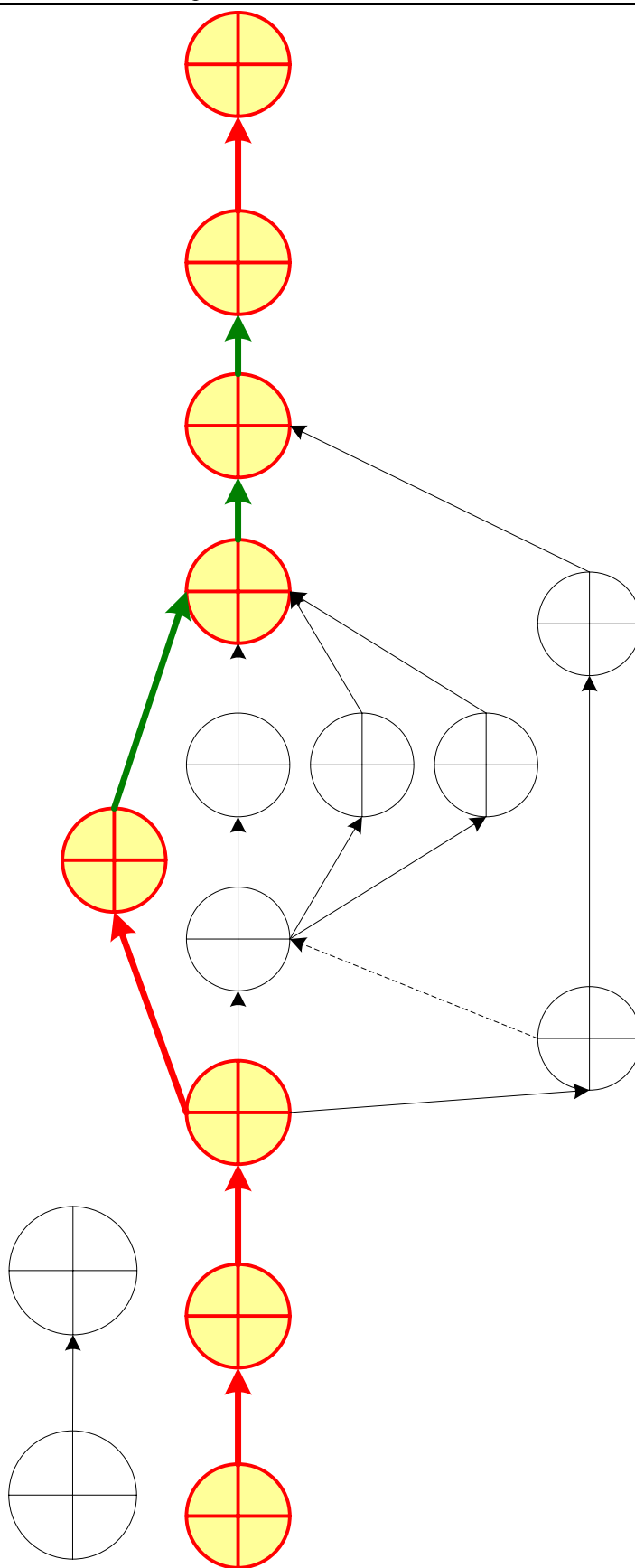
2.8.4-2 táblázat: időtartamok, költség- és erőforrásigények változása

A tevékenységidők változásával mind a költségszint, mind pedig az erőforrás-szükséglet változik (lásd részletesen a harmadik lépésnél). A várható időtartamok változásának hatására változnak az intervallumok (optimista, illetve pesszimista becslések) is (2.8.3-as egyenlet szerint).

Második lépés: Felrajzoljuk a PERT-hálót, valamint megállapítjuk a tevékenységek átfutási idejét, illetve az átfutási idő szórásnégyzetét. Ezután meg kell határozni a kritikus utat.

A háló felrajzolásához először meg kell határozni a logikai kapcsolatokat, a követő és a megelőző tevékenységeket. A logikai háló felrajzolása után először egy odafelé történő (progresszív) elemzés segítségével a **PERT-módszernek** megfelelően meghatározzuk a projekt várható átfutási idejét, majd egy visszafelé történő elemzéssel meghatározzuk a kritikus ut(ak)at. Ezután szintén odafelé történő elemzés segítségével meghatározzuk a projekt várható átfutási idejének szórásnégyzetét (összetett standard

bizonytalanságának négyzetét), ami közelítőleg a kritikus ut(ak)on lévő tevékenységek várható időtartamai szórásnégyzeteinek (standard bizonytalanságuk négyzetének) összege lesz. (Ezt a közelítést csak akkor lehet alkalmazni, ha az alternatív utak tartalékideje elegendően nagy ahhoz, hogy alternatív úton lévő tevékenység adott valószínűségi szint mellett (pl. 99%-os valószínűséggel) ne válhasson kritikus úttá. Ellenkező esetben az alternatív utak szórását nem hagyhatjuk figyelmen kívül. Ebben az esetben ez a feltétel teljesül, így ez a közelítés elvégezhető. Ha adott valószínűséggel alternatív úton lévő tevékenység is válhat kritikussá, akkor Fatemi Ghomi – Teimouri módszerét kell alkalmazni az átfutási idők meghatározására). Ha több kritikus út is van, akkor óvatosságból a legnagyobb összetett standard bizonytalanságúval számolunk a továbbiakban. Az alábbi ábrán látható az induló PERT-háló. A táblázatból látható, hogy E, G, N tevékenység időtartama nem csökkenthető.



2.8.4-1 ábra: a PERT-háló. A kritikus úton lévő tevékenységek: A-B-C-E-G-N-P. Tovább már nem csökkenthető az időtartama az E,G,N tevékenységeknek.

Harmadik lépés: Csökkentsük a kritikus úton lévő tevékenységek várható időtartamát.

Ezután pedig határozzuk meg a minimális átfutási idejű és a minimális összköltséggel járó programokat. A programok meghatározásához készíthetünk táblázatot.

Az időtartamok csökkentésénél három szempontot kell figyelembe venni:

- 1. Azon (kritikus úton lévő) tevékenység(ek) várható tartamát csökkentjük, amelyek (várható) egységnyi (változó)költségnövekedése a legkisebb.**

Megjegyzés: Az egységnyi költségnövekedési tényezőt közvetlenül is megadhatjuk, vagy a következő képlettel számíthatjuk (lineáris költségnövekedés esetén): $\Delta vc = \frac{vc(r) - vc(n)}{n - r}$, ahol r a roham- vagy

minimális megvalósítási idő, amely mellett a tevékenység még elvégezhető. n pedig a normál megvalósítási idő, amely mellett az adott tevékenységet normál körülmények között elvégezhetjük. Feltettük a példában, hogy a minimális (várható) megvalósítási idő az eredeti időtartam becslésénél a tevékenység időtartamának optimista becslése, vagyis $r=a$, valamint hogy a normál megvalósítási idő a legvalószínűbb megvalósítási idővel egyezik meg, képlettel: $n=m$. Feltettük továbbá, hogy a normál időtartamtól hosszabb időtartam szintén költségnövekedéssel jár (lineáris esetben hasonlóan az előzőhöz itt is számítható költségnövekedési tényező, amely azt mutatja, hogy egységnyi késés mennyi többletköltséggel jár). Ha a tevékenységek időtartama és változó költsége közötti függvénykapcsolat nem lineáris, akkor folytonos esetben a költségnövekedési tényező a költségfüggvény idő szerinti deriváltja, diszkrét esetben pedig a differenciáltja lesz.

- 2. Ha a költségnövekedési tényezők megegyeznek, akkor azon tevékenység időtartamát célszerű rövidíteni, amelynek rövidítésével nem alakul ki újabb kritikus út.** Ugyanis a kritikus úton lévő tevékenységet nem szabad eltolni (hiszen ez a program átfutási idejének növekedéséhez vezetne), ezáltal erőforrásigényeit nem lehet későbbi időpontra ütemezni. Minél több kritikus tevékenységünk

van, annál jobban „meg van kötve a kezünk”. Szélső esetben még megengedett megoldást sem tudunk találni az erőforrás-allokációs problémára. Ez pedig azt jelenti, hogy ebben az esetben az adott a költségszinten, adott időtartamokat, illetve korlátokat figyelembe véve nem lehet a programot határidőre teljesíteni. Sokszor előfordul, hogy a költségkeretünk szűkössége miatt nem tudjuk, vagy nem éri meg a minimális átfutási idejű termelési programot megvalósítani, hanem valamilyen közbenső termelési programot választunk és valósítunk meg.

- 3. Ha több tevékenység is teljesíti az előző két pont követelményeit, vagyis a költségnövekedési tényezőjük minimális, és nem lépnek be újabb kritikus utak az időtartamok csökkentése hatására (vagy bármely választás hatására ugyanúgy lesznek alternatív utak, amelyek kritikus utakká fognak válni), akkor azt a tevékenységet (azokat a tevékenységeket) választjuk, amely(ek)nek hatására az átfutási idő szórása (összetett standard bizonytalansága) a legnagyobb mértékben csökken.**

Lsz.	Tev.jel	Δt	Δvc	$\Delta vc * \Delta t$	Σvc	ΔFC	$\Delta FC * \Delta t$	ΣFC	ΣTC	TPT	σ	TPT(90)	TPT(95)	TPT(99)
0	--	--	--	--	1825	--	--	1175	3000	82	2,05	84,62	85,37	86,76
1	P	1	5	5	1830	25	25	1150	2980	81	2,03	83,60	84,34	85,73
2	B	8	6,25	50	1880	25	200	950	2830	73	1,67	75,14	75,75	76,89
3	C	3	30	90	1970	25	75	875	2845	70	1,63	72,09	72,68	73,79
4	A	1	50	50	2020	25	25	850	2870	69	1,61	71,07	71,65	72,75

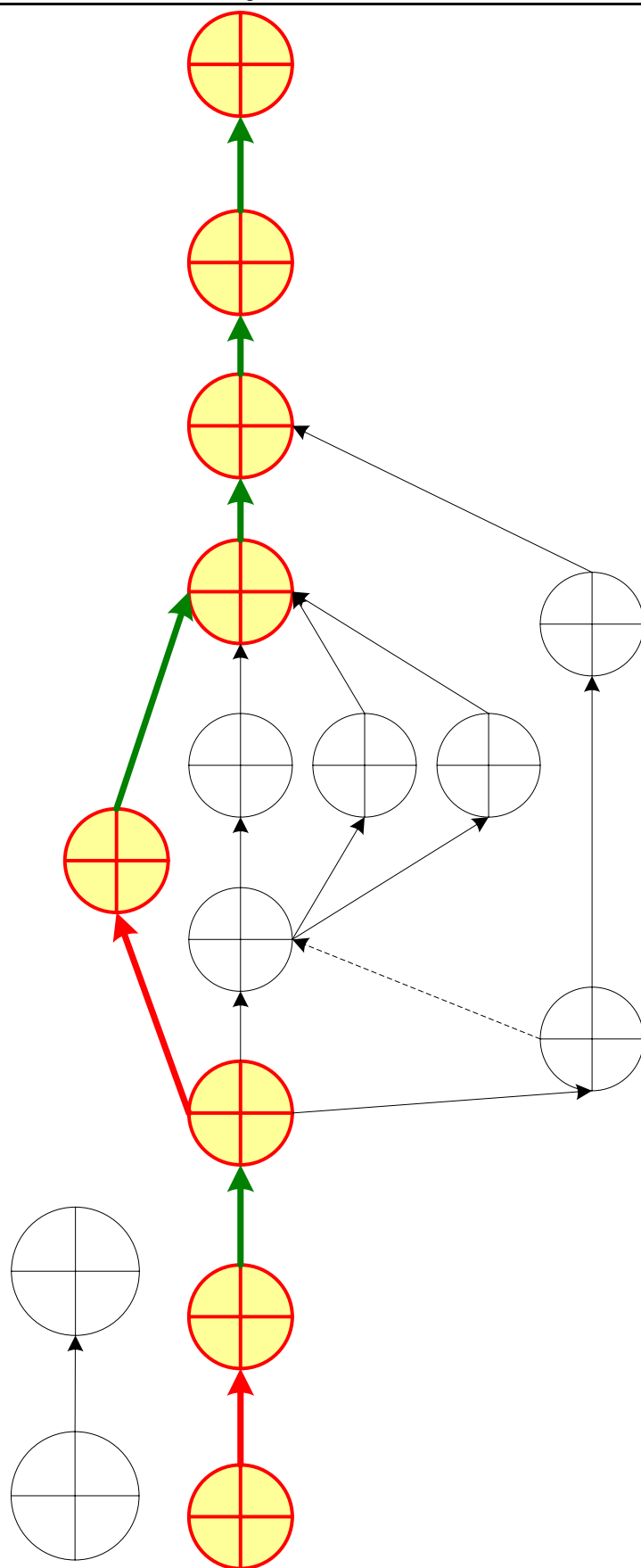
2.8.4-3 táblázat: a programok várható átfutása, illetve várható költségei.

A táblázatban feltüntettük a lépésszámon és az azonosító adatokon kívül az időtartam csökkentését (Δt), a változó költség egységnyi (Δvc), és összes várható költség növekedését ($\Delta vc * \Delta t$), az összes várható változó költséget (Σvc), a fixköltségcsökkenést (ΔFC), a várható összköltség értékeit (ΣTC), illetve a várható átfutási időt (TPT), az összetett standard bizonytalanságot (σ) és az adott biztonsági szintekhez tartozó átfutási időket (TPT(90), TPT(95), TPT(99)).

A táblázatból látható, hogy az összes várható költség tekintetében a második programtervezet a minimális, és a negyedik tervezet rendelkezik a legkisebb minimális átfutási idővel.

A programtervezeteknél a várható időtartamokkal számoltunk. Ha az egyes tevékenységek adott biztonsági szint melletti időtartamait szeretnénk megtudni, akkor β -eloszlást feltételezve meghatározhatók a tevékenységek adott biztonsági szint melletti időtartamai, az ehhez tartozó változóköltség-igény illetve erőforrás-szükséglet. Az időtartamok nem lesznek feltétlenül egész számok. Ha csak egész számokat fogadunk el, akkor ezeket az értékeket kerekítenünk kell. Ha egy tevékenység időtartamának valószínűsége egy meghatározott biztonsági szintet mindenképpen el kell, hogy érjen, akkor csak felfelé kerekíthetjük a kapott időtartam értékét. A várható erőforrásigény kiszámításánál feltételeztük, hogy az optimista időtartam (a) és a pesszimista időtartam alatt az erőforrásigény (a számítások megkönnyítése érdekében) egy lineáris függvény, így egy adott biztonsági szinthez tartozó időtartamhoz rendelhető (várható) erőforrás-szükséglet. Az erőforrás fajtájától függően esetlegesen szintén kerekítenünk kell az erőforrás-szükséglet értékén.

A példában mi minden tevékenység esetében meghatároztuk azt az időtartamot, amely 90%-os valószínűséggel teljesül. A kapott értékeket kerekítettük. A várható költségeket ezen időtartamok figyelembevételével számítottuk ki. A tevékenységek időtartamaihoz tartozó erőforrás-szükségleteket is kerekítettük, mivel itt az erőforrásigény a tevékenység elvégzéséhez szükséges munkaerő volt. A továbbiakban a minimális várható összköltséggel járó programot választjuk, és ezzel számolunk tovább. Itt 90%-os valószínűséggel a program átfutási ideje: 75,14 nap.



2.8.4-2 ábra: minimális várható összköltséggel rendelkező program PERT-hálój

Jel	Sorsz.	Megnevezés	A	B	M	T	S2	T(90%)	T(90%)	vc(T)	vc(T(90%))	Ir(T(90%))
A	1	A megoldás műszaki adatainak meghatározása	3,00	5,00	4,00	4	0,111	4,507	5	200,000	240,000	4
B	2	A célgép megtervezése	17,63	25,30	23,77	23	1,633	24,611	25	200,000	187,500	3
C	3	A célgép legyártása	37,00	43,00	40,00	40	1,000	41,520	42	200,000	253,333	6
E	4	A célgép helyszínre szállítása	1,00	1,00	1,00	1	0,000	1,000	1	120,000	120,000	2
D	5	Helykijelölés és területfelszabadítás	4,00	6,00	5,00	5	0,111	5,507	6	25,000	28,000	3
F	6	Alapkészítés	1,00	3,00	2,00	2	0,111	2,507	3	50,000	55,000	6
M	7	Beton kötése	1,00	3,00	2,00	2	0,111	2,507	3	250,000	280,000	5
G	8	Gépfelállítás	1,00	1,00	1,00	1	0,000	1,000	1	150,000	150,000	4
N	9	Csatlakozások bekötése	2,00	2,00	2,00	2	0,000	2,000	2	250,000	250,000	2
P	10	Próbaüzem	1,33	2,67	2,00	2	0,049	2,338	3	80,000	75,000	3
V	11	Villamos szerelési tevek készítése	3,00	5,00	4,00	4	0,111	4,507	5	75,000	77,000	4
H	12	Villamos vezetékek- és szerelvényyszerelés	3,00	5,00	4,00	4	0,111	4,507	5	25,000	26,000	6
S	13	Sűrített levegő-vezeték tervezése	3,00	5,00	4,00	4	0,111	4,507	5	50,000	55,000	1
I	14	Sűrített levegő-vezeték készítés és szerelés	4,00	8,00	6,00	6	0,444	7,013	8	50,000	60,000	6
J	15	Pneumatikus anyagmozgató berendezés tervezése	9,00	11,00	10,00	10	0,111	10,507	11	50,000	60,000	1
K	16	Pneumatikus anyagmozgató berendezés gyártása	9,60	18,13	17,07	16	2,023	17,623	18	55,000	65,000	6
L	17	Pneumatikus anyagmozgató berendezés szerelése	9,00	11,00	10,00	10	0,111	10,507	11	50,000	60,000	4
Összesen										1880,000	2041,833	

2.8.4-4 táblázat: minimális várható összköltséggel rendelkező program tevékenységjegyzéke

Ha a táblázat adatai alapján határozzuk meg az átfutási időt, akkor ez A-B-C-E-G-N-P tevékenységre kerekítés

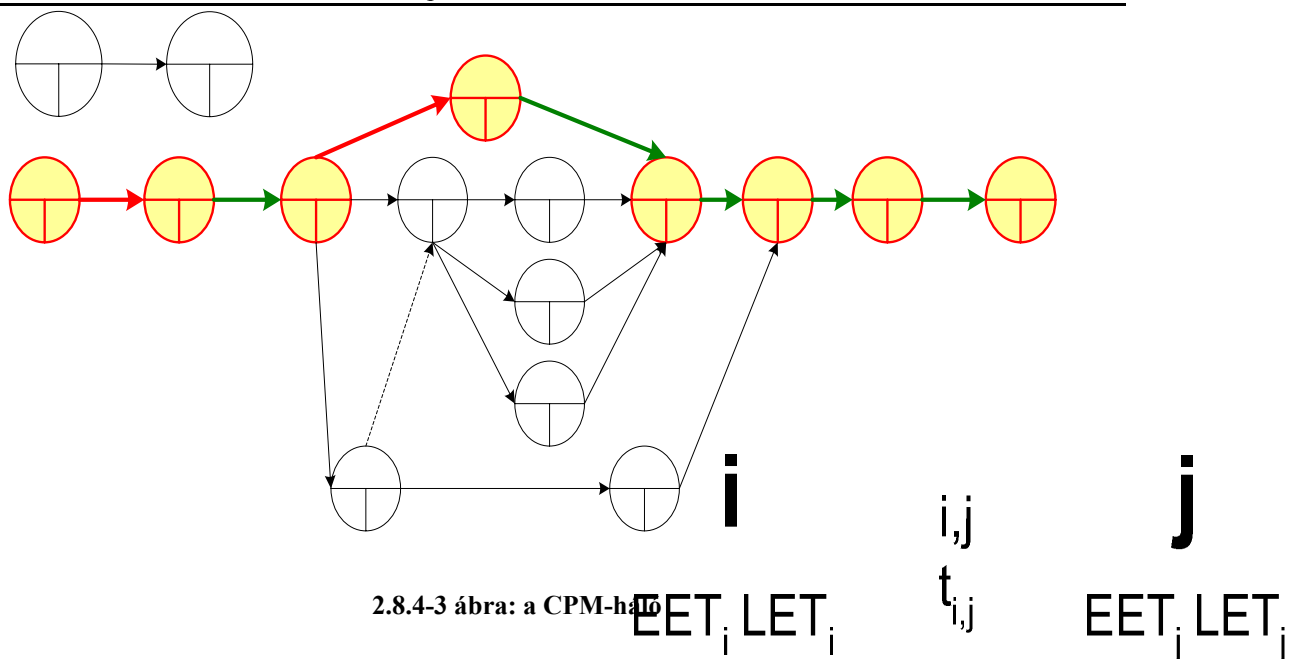
nélkül: $=4,5+24,6+1+41,5+1+2+2,3 \approx 77$ nap. A tevékenységek

időtartamainak kerekítéseit figyelembe véve: $=5+25+42+1+1+2+2=78$ nap.

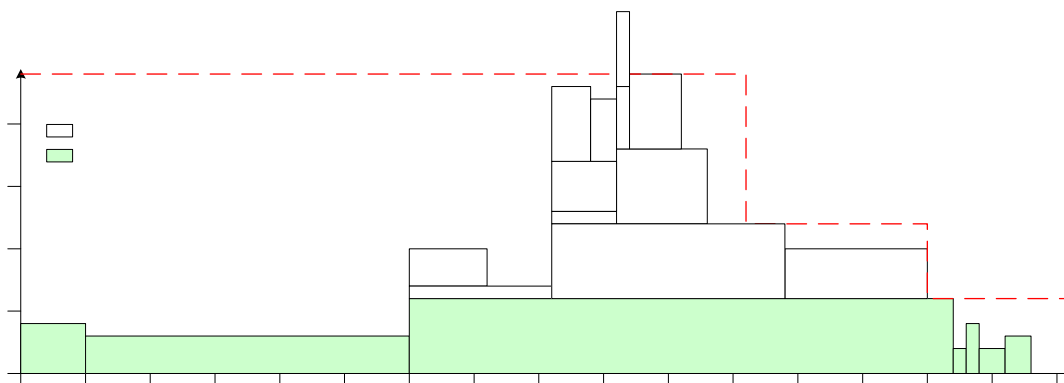
Ha minden tevékenységre előírjuk a 90%-os valószínűségi szintet, akkor a teljes átfutási időre nézve ez a szint már több mint 99%. Ha a 90%-os szintet szeretnénk tartani a tevékenységekre, akkor elegendő a tevékenységre 62-63% körüli valószínűségi szintű időtartamot biztosítani. Ennek kiszámítása a következőképpen történhet: a 90%-os valószínűségi szint melletti átfutási időből indulunk ki, ami 75,14 nap. Az átfutási idő várható értéke 73 nap. Ezt a 2,14 napot kell szétosztani a tevékenységek között az időtartam arányában. Egy napra tehát $2,14/73$ pótlólagos nap jut. Ezek után kiszámítható a tevékenységek időtartama, feltételezve, hogy az átfutási idő 90%-os valószínűséggel 75,14 nap lesz. Például A tevékenység időtartama így: 4 nap (A tevékenység várható időtartama) + $4 \cdot 2,14/73$ nap = 4,117 nap lesz. A tevékenység valószínűségi szintje ezek után (β -eloszlást feltételezve): 62,17% lesz.

A továbbiakban a számítások megkönnyítése érdekében a kerekített időtartamokat vesszük figyelembe. Tehát pl. 90%-os biztonsági szinthez meghatároztuk a tevékenységek időtartamát (β -eloszlást feltételezve), költségigényét és az erőforrás-szükségletét. Ezeket az adatokat most már úgy kezeljük, mintha determinisztikus adatok lennének.

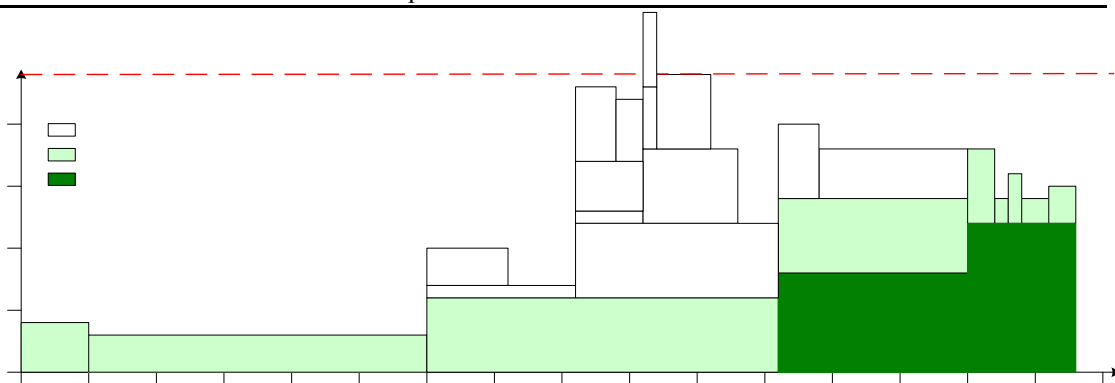
Ekkor felrajzolható egy CPM-háló a megfelelő tartamok figyelembevételével.



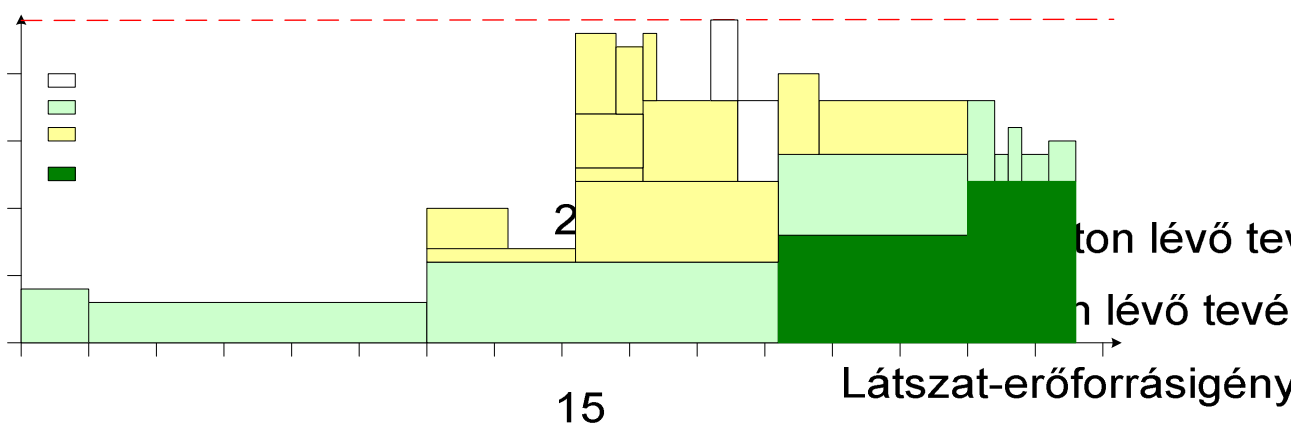
Negyedik lépés: Felrajzoljuk az erőforrásterhelési diagramot. Ezután keresünk megengedett, majd optimális erőforrás-allokációt. Ha eddig a pontig eljutottunk, akkor a problémának megfelelő erőforrás-allokációs algoritmust választhatunk. Számos problémát megoldhatunk (párhuzamos projektek közötti erőforrás-megosztás, többfajta erőforrás egyidejű kezelése stb.) az előző fejezetekben leírtak alapján. Itt egy megengedett megoldás keresése után (amelyet pl. a Microsoft Projekt programjával kereshetünk) egy célfüggvényt kell meghatározni. Legyen most ez a célfüggvény a lehető legkorábbi kezdés.



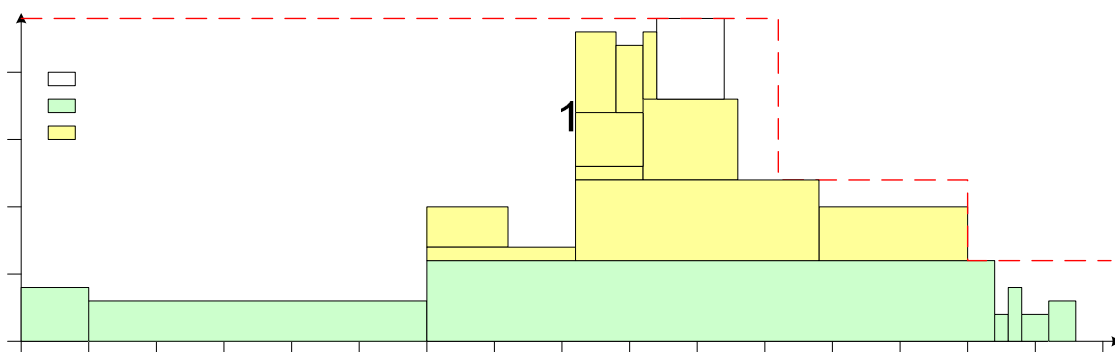
2.8.4-4 ábra: a terhelési diagram



2.8.4-5 ábra: a terhelési diagram látszat-erőforrásigények bevezetése után



2.8.4-6 ábra: egy lehetséges megengedett megoldás



2.8.4-7 ábra: az optimális megoldás

Az optimális megoldás meghatározásához választhatunk más célfüggvényt is. Ilyen célfüggvény lehet pl. a lehető legkevesebb átlagos erőforráskihasználás (erőforrásigények kiegyenlítése), a maximális erőforráskihasználások minimalizálása stb.

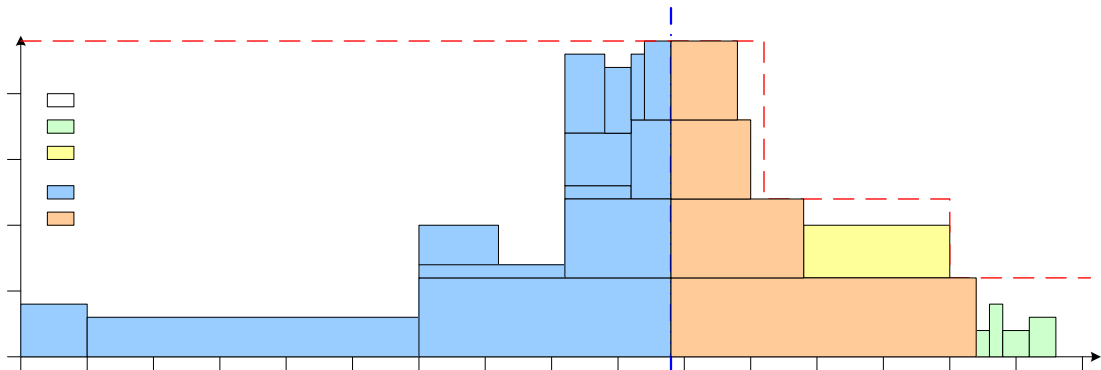
Az optimális erőforrás-allokáció meghatározása után kaptunk egy tervet a projekt erőforrásigény-, költség- és időszükségleteire.

(Ötödik lépés:) A hálós-, illetve az erőforrás-tervezési technikák nem csak a projekt tervezési szakaszában, hanem a megvalósítás nyomonkövetésére, ellenőrzésére is használhatók.

Amennyiben megváltozik egy tevékenység erőforrás-szükséglete, időtartama, vagy az erőforráskorlát, akkor on-line ütemezést kell alkalmaznunk. A projekt követésénél a hálóba a már lefutott tevékenységek tényleges időtartamát kell beírni, a tevékenység szórását (standard bizonytalanságát) pedig nullának kell tekinteni. A tevékenységek időtartamának követése során így csökkenni fog az átfutási idő szórása is. A program végére a sztochasztikus hálóból egy determinisztikus háló lesz, mely megadja, hogy a tevékenységek ténylegesen mikor hajtódtak végre, mikor kezdődtek, és mikor fejeződtek be.

Tegyük fel, hogy az előző példában szereplő projektben a 7. hétig (49. napig) szereplő valamennyi tevékenység a 90%-os biztonsági szint melletti időtartam alatt megvalósult. A *villamosvezeték- és szerelvényyszerelés* (H tevékenység) és a *sűrítettlevegő-vezeték készítése és szerelése* (I tevékenység) viszont a 90%-os valószínűséggel teljesülő várható időtartamokhoz képest is várhatóan még két napot fog csúszni. Először rajzoljuk fel az aktualizált hálót, határozzuk meg a várható átfutási időt! A háló felrajzolása után a várható átfutási idő szintén az A-B-C-E-G-N-P tevékenységek várható időtartamainak összege: $= 5+25+40+1+1+2+2=76$ nap lesz. Mivel A és B tevékenység már lefutott, ezért ezen tevékenységek szórását nullának, várható időtartamát pedig a tényleges időtartamával azonosnak tekintjük. Az átfutási idő szórásnégyzete: $=0+0+1+0+0+0+0,049=1,049$. Ebből az átfutási idő szórása: $= 1,024$ nap. Ekkor a 90%-os biztonsági szint mellett az átfutási idő becslése: $= 77,3$ nap. Ha β -eloszlás szerint, 90%-os biztonsági szint mellett összegezzük a tevékenységek várható időtartamát, akkor kerekítések nélkül az átfutási idő: $=5+25+41,52+1+1+2+2,338=77,858$ nap. Kerekítésekkel pedig továbbra is 78 nap. Látható, hogy a különböző becslések egyre jobban közelítenek egymáshoz. Ez elsősorban az átfutási idő szórásának (az összetett standard bizonytalanság) csökkenésével

magyarázható. Rajzoljuk fel a terhelési diagramot, keressünk megengedett, majd egy adott célfüggvényre optimális megoldást!



2.8.4-8 ábra: az optimális megoldás on-line ütemezés után

2.9 Összefoglalás

A projektmenedzsment előtérbe kerülése tette a hálótervezést ismét fontossá. Egy projektben már nem csak az a fontos, hogy a feladatot minél hamarabb oldjuk meg, hanem az is lényeges, hogy esetenként szűkös erőforrásokkal is gazdálkodni tudjunk. Továbbá fel tudunk készülni olyan nem várt eseményekre is, amelyek egy adott tevékenység időtartamát, erőforrás-szükségletét megváltoztatják.

Munkám során olyan algoritmust fejlesztettem ki, amely a fent említett céloknak megfelel.

1. Garantáltan véges lépésben megadja az optimális megoldást.
2. Használható változó (nem konstans) erőforráskorlát esetén is.
3. Használható a már elkezdett projektekben lévő, megváltozott időtartamú, erőforrás-szükségletű tevékenységek újraütemezésére.
4. Olyan problémákat is kezelni tudunk, ahol az idő-, költség-, erőforrás-igények együttes optimalizálása a cél.
5. Olyan algoritmust készítettem, mely többfajta erőforrás egyidejű kezelésére, illetve párhuzamosan működő projektek közötti erőforrás-megosztásra is alkalmazhatók.
6. Használható bizonytalan átfutási idejű projektek, illetve termelési programok tervezésére.

A

Összefoglalás

A módszer menetét ábrákkal illusztrálva mutattam be, melyekkel olyan menedzseri/vezetői problémák megoldására is mód nyílik, amit a szakirodalomban eddig még nem publikáltak.

A következő táblázat azt mutatja, hogy a bemutatott módszereket milyen problémákra lehet alkalmazni. A táblázatban megtalálható, hogy egy adott probléma esetén mely fejezetek segítségével lehet megoldást találni.

Optimális erőforrás-tervezés		Egyfajta erőforrás				Többfajta erőforrás					
		Tervütemezés		On-line ütemezés		Tervütemezés			On-line ütemezés		
		Egy vagy független projekt	Több projekt közötti erőforrás-megosztás	Egy vagy független projekt	Több projekt közötti erőforrás-megosztás	Egy vagy független projekt	Több projekt közötti erőforrás-megosztás	Egy vagy független projekt	Több projekt közötti erőforrás-megosztás	Egy vagy független projekt	Több projekt közötti erőforrás-megosztás
		Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása	Erőforrás-helyettesítések figyelmen kívül hagyása
Determinisztikus	Normál átfutás	ERALL-OPT (2.1)	ERALL-OPT/PP (2.6.1)	ERALL-OPT/ON-LINE (2.3)	2.3 + 2.6.1	ERALL-OPT/TE (2.6.2)	2.6.1 + 2.6.2	Csoportos erőforrás-tervezés (2.7) + 2.6.1 + 2.6.2	2.3 + 2.6.2	2.3 + 2.6.1 + 2.6.2	2.3 + 2.6.1 + 2.6.2 + 2.7
	Minimális vagy költség-optimalis átfutás	ERALL-OPT/KLTG (2.4)	2.4 + 2.6.1	2.3 + 2.4	2.3 + 2.4 + 2.6.1	2.4 + 2.6.2	2.4 + 2.6.1 + 2.6.2	2.4 + 2.7	2.3 + 2.4 + 2.6.2	2.3 + 2.4 + 2.6.1 + 2.6.2	2.3 + 2.4 + 2.6.1 + 2.6.2 + 2.7
	Minimális vagy költség-optimalis átfutás	Alternatív megoldások keresése (2.5) + 2.4	2.4 + 2.5 + 2.6.1	2.3 + 2.4 + 2.5	2.3 + 2.4 + 2.5 + 2.6.1	2.4 + 2.5 + 2.6.2	2.4 + 2.5 + 2.6.1 + 2.6.2	2.4 + 2.5 + 2.7	2.3 + 2.4 + 2.5 + 2.6.2	2.3 + 2.4 + 2.5 + 2.6.1 + 2.6.2	2.3 + 2.4 + 2.5 + 2.6.1 + 2.6.2 + 2.7
Sztochasztikus	Normál átfutás	Bizonytalan átfutási idejű projektek optimalis erőforrás-elosztása (2.8)	2.6.1 + 2.8	2.3 + 2.4 + 2.8	2.3 + 2.4 + 2.6.1 + 2.8	2.4 + 2.6.2 + 2.8	2.4 + 2.6.1 + 2.6.2 + 2.8	2.4 + 2.7 + 2.8	2.3 + 2.4 + 2.6.2 + 2.8	2.3 + 2.4 + 2.6.1 + 2.6.2 + 2.8	2.3 + 2.4 + 2.6.1 + 2.6.2 + 2.7 + 2.8
	Minimális vagy költség-optimalis átfutás	Költségtervezés bizonytalan átfutási idejű projektek esetén (2.8.2) + 2.8	2.6.1 + 2.8 + 2.8.2	2.3 + 2.4 + 2.8 + 2.8.2	2.3 + 2.4 + 2.6.1 + 2.8 + 2.8.2	2.4 + 2.6.2 + 2.8 + 2.8.2	2.4 + 2.6.1 + 2.6.2 + 2.8 + 2.8.2	2.4 + 2.7 + 2.8 + 2.8.2	2.3 + 2.4 + 2.6.2 + 2.8 + 2.8.2	2.3 + 2.4 + 2.6.1 + 2.6.2 + 2.8 + 2.8.2	2.3 + 2.4 + 2.6.1 + 2.6.2 + 2.7 + 2.8 + 2.8.2
	Minimális vagy költség-optimalis átfutás	Minimális vagy költség-optimalis átfutás	2.5 + 2.8 + 2.8.2	2.3 + 2.4 + 2.5 + 2.8 + 2.8.2	2.3 + 2.4 + 2.5 + 2.6.1 + 2.8 + 2.8.2	2.4 + 2.5 + 2.6.2 + 2.8 + 2.8.2	2.4 + 2.5 + 2.6.1 + 2.6.2 + 2.8 + 2.8.2	2.4 + 2.5 + 2.7 + 2.8 + 2.8.2	2.3 + 2.4 + 2.5 + 2.6.2 + 2.8 + 2.8.2	2.3 + 2.4 + 2.5 + 2.6.1 + 2.6.2 + 2.8 + 2.8.2	2.3 + 2.4 + 2.5 + 2.6.1 + 2.6.2 + 2.7 + 2.8 + 2.8.2

2.9-1 táblázat: optimális erőforrás-tervezés alkalmazása különböző problémák esetén

Valamennyi algoritmus a 2.1-es fejezetben tárgyalt **ERALL-OPT**-módszeren alapul. Az egyes alfejezetekben azt mutattam be, hogy egy adott feladatot milyen kiterjesztésekkel, módosításokkal illetve megszorításokkal lehet visszavezetni az alapproblémára. Mivel az egyes algoritmusok modulárisan építhetők fel, így egy komplex problémára több fejezetben tárgyalt módszert egyidejűleg is alkalmaznunk kell (egy lehetséges moduláris felépítést mutat az 5.3.1-es ábra).

3. A módszer egy gyakorlati alkalmazása az INTEGRAL-HEXA Rt-nél

Az esettanulmányban egy zöldmezős beruházásra került sor, mely csaknem 350 tevékenységet tartalmaz. A feladat és cél az volt, hogy a tevékenységeket áttekintve megvizsgáljuk, hogy az előre elkészített ütemtervhez képest lehet-e időt, költséget illetve erőforrást megtakarítani.

A vállalat elsősorban zöldmezős beruházásokat végez. Több áruház, bevásárlóközpont építése fűződik a cég nevéhez. A vállalat a beruházásokra pályázik, mely pályázatokhoz részletes ütemtervet, illetve költségtervet mellékel. Az ütemtervet Microsoft Project segítségével, míg a költségtervet Microsoft Excel segítségével készítették el. A kérdés tehát az volt, hogy idő-, erőforrás-optimaló módszereket alkalmazva jelentősen lehet-e csökkenteni a projekt átfutási idejét, illetve költségigényét.

Az optimális erőforrás-tervezést az **ERALL-OPT**-módszer segítségével végeztem. Ez a módszer hatékonyan alkalmazható projektek optimális erőforrástervének meghatározására. A módszer részletes ismertetését a 2. fejezet tartalmazza.

Optimális megoldás keresésére párhuzamos Branch and Bound algoritmust alkalmazó elosztott optimalizáló rendszert használtunk. Ez a rendszer egy platformfüggetlen JINI-alkalmazás [114, 398], mely az adatokat .XML-formátumban olvassa be, illetve az eredményeket is ilyen formátumban szolgáltatja vissza. [230, 257]

3.1 Meglévő adatok feldolgozása

Microsoft Projectben megadott ütemterv, illetve egy Excel táblázat állt rendelkezésre. Az ütemterv tartalmazta az egyes tevékenységek nevét, időtartamát napokban megadva, tényleges kezdésének és befejezésének időpontját, valamint az egyes tevékenységek megelőző tevékenységeit. Az Excel táblázat a felmerülő anyag- és bérköltségeket összesítette az egyes tevékenységcsoportoknál. Mivel az előzőekben említett módszer használatához szükség van az egyes tevékenységek erőforrásigényére, illetve követő tevékenységeire, ezért az elsődleges feladat ezen adatok kiszámítása volt a meglévő adatokból, és a módszerhez szükséges adatok felvitele .XML állományba.

A projektben szereplő tevékenységeket – az elvégzendő munka jellege szerint - 6 csoportba sorolta a vállalat: **A**, **B**, **C**, **D**, **E** és **F** tevékenységcsoportba. Az **A**

tevékenységcsoportba kerültek az építőmesteri munkák, a **B** tevékenységcsoportba a gépész szerelési munkák, a **C** tevékenységcsoportba az elektromos szerelési munkák, a **D** tevékenységcsoportba a külső közművel kapcsolatos munkák, az **E** tevékenységcsoportba a külső létesítmények építésével kapcsolatos munkák, az **F** tevékenységcsoportba pedig a társvállalkozók tevékenységei. Az egyes tevékenységeket az alábbi táblázatok tartalmazzák.

A - ÉPÍTŐMESTERI MUNKÁK		
Előkészítő munkák	Dokkoló, aknák, padló alatti vb munka	Padló lapburkolás
Építési szerződés aláírás	Földszint padló acélhaj beton	Üvegalkészítés
Telek, épület kitűzés	Emelet aljzatok	Álmennyezetek, pult hátfal belsőép.
Munkaterület átadás	Nyílászárók elhelyezése	Ütközésvédelem, szerelvények
Építési konténer telepítés	Hidegburkoló munkák	Takarítás
Ideiglenes energia kiépítés	Festő előkészítés	A-D 0-7a Kassza felülete
Ideiglenes vízellátás kiépítés I.	Gipszkarton munkák	Tömbalapok földmunkái
Ideiglenes vízellátás kiépítés II.	Fém és lakatos munkák	Tömbalapok betonozása
Ideiglenes közl. létesítmények I.	Kapuk, rámpakiegyenlítő	Előregyártott kehelynyak
Ideiglenes közl. létesítmények II.	Hőszigetelő és díszítő munkák	Monolit vb kehelytálp, tömbalapok
Építési tábla kihelyezés	Álmennyezetek	Szerelő, fagykizáró betonok
Meglévő trafó bontás	Festő és mázoló munkák	Előregyártott talpgerendák
Telefonvonal biztosítás	Asztalos munkák	Monolit talpgerendák
Bontási maradványok eltávolítása	Árnyékolási munkák	Visszatöltések
Földmunkák	Melegburkolás, műgyanta	Előregyártott vasbeton oszlop
Humuszleszedés	Szerelvényezés	Függőleges monolit vb szerkezetek
Szennyezett föld-eltávolítás	Takarítás	Acél szerkezetek
Alsófeltöltés	D-G 0-7 Eladótér	Monolit vasbeton földem
Padló alatti feltöltés	Tömbalapok földkiemelés	Falazatok
G-H 7 Raktárak, irodák	Tömbalapok betonozása	Mall acélszerkezet
Tömbalapok földkiemelés	Előregyártott kehelynyak elhelyezése	Tető trapézlemez elhelyezés
Beton a tömbalapba	Monolit kehelytálp betonozása	Tető hőszigetelés
Előregyártott kehelytalp	Szerelő, fagykizáró betonok	Tető vízszigetelés
Monolit kehelytálp	Előregyártott talpgerendák	Homlokzat szerelés
Pontalapok	Visszatöltések	Főbejárat szerkezetépítés
Szerelő, fagykizáró betonok	Előregyártott vasbeton oszlop	Reklámtartó acélszerkezet
Előregyártott talpgerendák	Előregyártott vb gerenda, szelemen	Bádogos munkák
Monolit talpgerendák, alsó falak	Homlokzati acélszerkezetek	Acélhajás betonpadló
Visszatöltések	Tető trapézlemez elhelyezés	Válaszfalak
Előregyártott vasbeton oszlopok	Tető acélszerkezetek	Vakolatok
Előregyártott vb gerenda, szelemen	Tető hőszigetelés	Alumínium-szerkezetek a tetőn
Földszint falazási munkák	Hő- és füstelvezetők	Üvegezés a tetőn
Földszint mon.vasbeton munkák	Tető vízszigetelés	Alumínium-szerkezetek
Tető trapézlemez elhelyezés	Homlokzatszerelés	Üvegezés
Tető hőszigetelés	Bádogos munkák	Nyílászárók elhelyezése
Tető, vízszigetelés	Padlócsatornák betonból	Lakatos munkák
Földem vasbeton munka	Földelő hálózat	Hidegburkolatok
Emelet falazási munka	Nyílászárók elhelyezése	Festő előkészítés
Előtető monolit vb munka	Acélhajás padló	Gipszkarton munkák
Előtető vízszigetelés	Padlócsatorna acélból	Álmennyezetek
Homlokzati acélszerkezet	Füstkötény építés	Rácsok, automata ajtók
Homlokzati vasbetonpanel-szerelés	Fém lakatos munkák	Festő, mázó munkák
Homlokzati szendvicspanel-szerelés	Csemegesziget kialakítás	Asztalos munkák
Bádogos munkák	Festő, mázó munkák	Szerelvényezés
Vakolási munkák	Festő, mázó munkák	Takarítás

3.1-1 táblázat: az A tevékenységcsoport tevékenységei

B - GÉPÉSZ SZERELÉSI MUNKÁK	C - ELEKTROMOS SZERELÉSI MUNKÁK	
Földbe kerülő gép.vezetékek kassza	Villámhárító földelő szondák	Kasszafelügyelet
Földbe kerülő gép.vezetékek eladótér	Villámhárító földelő háló	Védőcsövezés
Földbe kerülő gép.vezetékek Raktár	Villámhárító felfogó rudak	Kábeltálca szerelés
Geberit Pluvia földbe kerülő része	Villámhárító tetőn ép.gép. szerelv.	Kábelezés
Geberit Pluvia csőszerelés	Gazdasági rész	Kasszák szerelése
sprinkler csőszerelés az eladótérben	<i>Földszint</i>	Világítás szerelés
sprinkler csőszerelés az eladótérben	20kV-os kapcsoló és transzformátor	Elosztók
sprinkler csőszerelés szoc. és ek.	Diesel aggregátor	Reklámok, kapuk
sprinkler csőszerelés szoc. és ek.	0,4 kV-os elosztó	Gépészeti villanszerelés
Sprinkler csőszerelés Kassza	Süllyesztett védőcsőszerelés	Tűzjelző szerelés
sprinkler földalatti vezeték	Aljzatok alatti védőcsőelhelyezések	Sprinkler gépház
sprinkler földalatti vezeték	Pékség, raktár elosztók	Kábeltálca
sprinkler gépház szerelése	Kábelszerelés	Kábelezés , beüzemelés
víz- és tűzivízszerelés eladótér	Világítás és szerelvényezés	Légkezelők
víz- és tűzivízszerelés szoc. és ek.	Gépészeti villanszerelés	Védőcsőszerelés
víz- és tűzivízszerelés Kassza	Kábeltálca szerelés	Kábelezés
fűtésszerelés az eladótérben	Tűzjelző szerelés	Beüzemelés
kapulégfűgönyök felszerelése	<i>Emelet</i>	Épületfelügyelet
fűtésszerelés szoc. és ek.	Süllyesztett védőcsőszerelés	Kábelezés
Fűtésszerelés Kassza	Kábeltálca szerelés	Perifériák szerelése
légkezelők elhelyezése	Kábelszerelés	Elosztók szerelése
radiátorok felrakása szoc. és ek.	Szerelvényezés, világítás szerelés	Bekötések, tesztelés
kazánház szerelése	UPS beüzemelés	Beüzemelés, próba
gázszerelés	Gépészeti terek	Mérések, jegyzőkönyvek
szellőzésszerelés az eladótérben	Elosztószekrények	
szellőzésszerelés szoc. és ek.	Tűzjelző szerelés	
szellőzésszerelés Kassza	Eladótér	
nagy szellőzőgép bekötése	Kábeltálca szerelés	
splitek elhelyezése	Erősáramú kábelezés	
hűtésszerelés eladótérben	Sensormatic kábelezés	
hűtésszerelés szoc. és ek.	Hangosítás szerelés	
hűtésszerelés Kassza	Antenna szerelés	
folyadékűtő felrakása	Tűzjelző szerelés	
szigetelés	Felülvilágítók kábelezése	
beüzemelés, besabályozás	Gépészeti Villanszerelés	

3.1-2 táblázat: a B és C tevékenységcsoport tevékenységei

D - KÜLSŐ KÖZMŰ	E - KÜLSŐ LÉTESÍTMÉNYEK	F - TÁRSVALLALKOZÓK TEV.
Vízellátás	Külső felszerelési tárgyak	energia biztosítás
Földmunka	Reklámtorony alapozás	padlócsatorna
Csővezeték szerelvények	Reklámtorony villamos megtáplálás	padlócsatorna csőszerelés
Próbák, üzembehelyezés	Reklámtorony szerelés	padlócsatorna lefedés
Csapadékvíz elvezetés	Reklámtorony üzembehelyezés	hűtőpultok szerelés
Földmunka	Kerítés- kapu	gépház zárható
vezeték építés	Kerítés- kapu	gépház szerelés
Aknák, iszapfogó beépítés	Tűzivíz és sprinkler tározó	reklámtorony alapkosár szállítás
Folyókák, víznyelők		reklámtorony alapozás
Próbák, üzembehelyezés		reklámtorony szerelés
Szennyvízelvezetés		reklámtorony elektromos betáplálás
Földmunka		homlokzati reklámtartó
Vezeték építés		szerelésre munkaterület
Aknák, homokfogó, zsírfogó		hűtőkamrák szerelése
Fedlapok		lámpaszállítás
Próbák, üzembehelyezés		Kassza felügyelethez munkaterület
Külső gázellátás		kocsitároló szerelés
Földmunka		kemencéhez munkaterület
Csővezeték, nyomásszabályozó		kémény és egyéb bekötések
Próbák, üzembehelyezés		hangosítás vezetékezés
Út-járda parkoló építés		szerelés, beüzemelés
Földmunka		Sensormatik vezetékezés
Feltöltések		zártver megadás
Szivárgó építés		zárbetét csere
szegélyképzés		
Útalapok		
Térköburkolat készítés		
Bitumenes burkolatok		
Jelzőtáblák, felfestések		
Kertépítési munka		
Földmunka		
Növénytelepítés		

3.1-3 táblázat: a D, E és F tevékenységcsoport tevékenységei

Az egyes tevékenységeket a tevékenységcsoportokon belül feladatcsoportokba soroltam. A feladatcsoportok összesítették a bennük szereplő tevékenységek anyagköltségét, bérköltségét. A 3.1-4 táblázatban példaként a D és E tevékenységcsoportok feladatcsoportjait láthatjuk azok költségigényével együtt.

Főösszesítő - Hauptsumme				
Ssz.	Munkanem	Anyag Material (nettó HUF)	Díj Lohn (nettó HUF)	Összes Summe (nettó HUF)
D	Külső munkák (telekhatáron belül)			
061.	Külső vízellátás	5 851 820	3 569 735	9 421 555
062.	Külső csapadékvízvezetés	18 894 060	9 264 421	28 158 481
063.	Külső szennyvízvezetés	2 483 506	2 964 694	5 448 200
064.	Külső gázellátás	1 505 742	808 036	2 313 778
066.	Út-, járda- és parkolóépítés	80 216 543	33 228 985	113 445 528
067.	Kertépítési munkák	9 480 016	2 305 935	11 785 951
D	Külső munkák összesen	118 431 687	52 141 806	170 573 493
E	Külső létesítmények (telekhatáron belül)			
071.	Külső felszerelési tárgyak	2 472 466	403 615	2 876 081
072.	Reklámtorony	408 672	218 935	627 607
073.	Kerítés, kapu	2 210 162	1 021 001	3 231 163
074.	Tűzivíz és sprinkler tározó	6 557 572	2 774 481	9 332 053
E	Külső építmények összesen	11 648 872	4 418 032	16 066 904

3.1-4 táblázat: a D és E tevékenységcsoport feladatcsoportjai, illetve azok anyag- és bérköltsége

Minden tevékenység adott feladatcsoportba történő besorolását a rendelkezésünkre álló – költségvetést is tartalmazó – Excel-file segítségével végeztem. Ez az állomány minden egyes tevékenységcsoportban lévő feladatcsoportról részletes leírást adott. A 3.1-5 táblázatban példaként a D tevékenységcsoport 67-es feladatcsoportjának tevékenységeit láthatjuk azok részletes leírásával együtt.

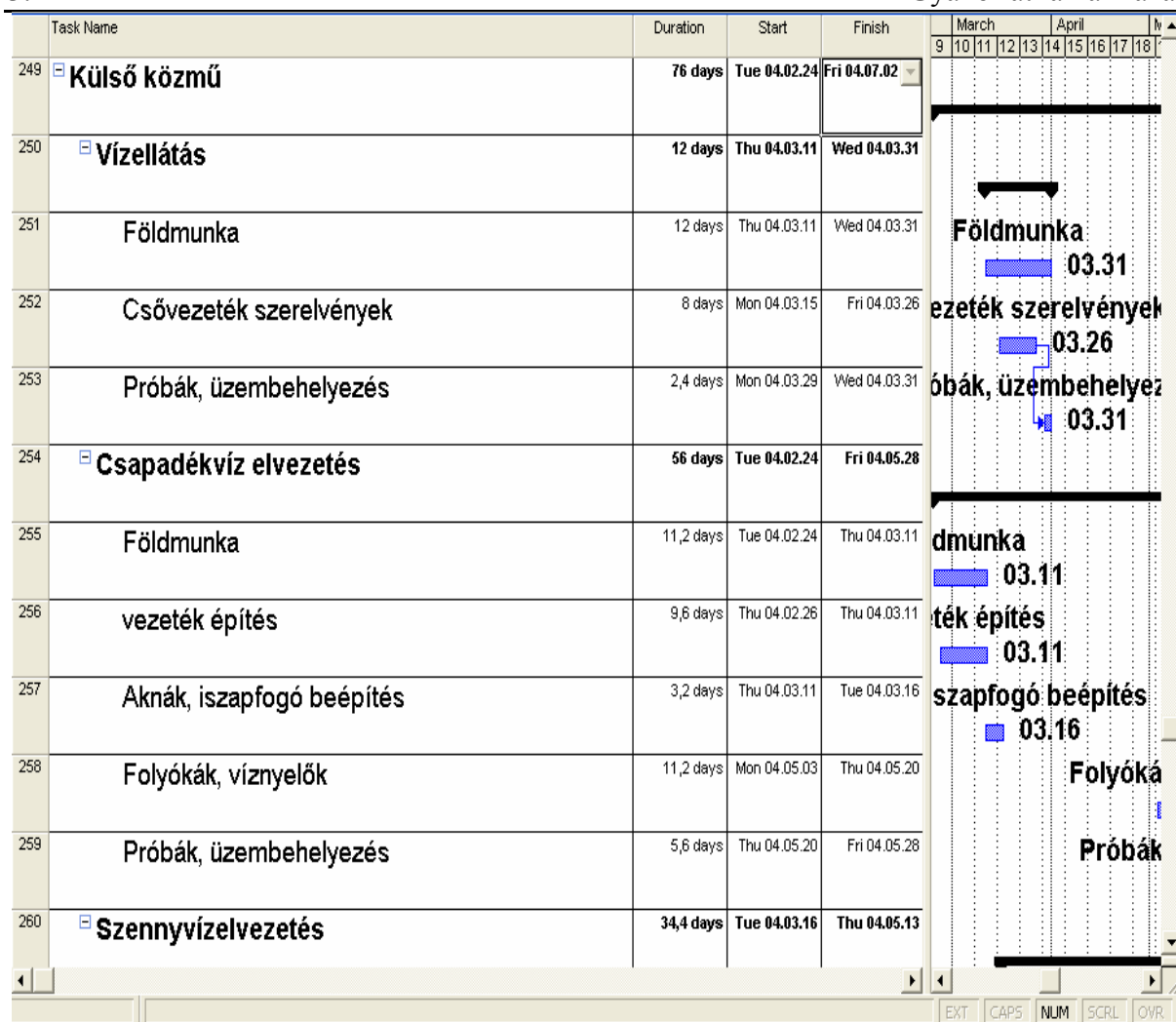
067. Parkosítási munkák	
Talajjavítás, füvesítés	
Talajjavítás szerves trágyával	
Füvesítés 20%-nál kisebb rézsűn, talajelőkészítéssel, 50-60 dkg/10 m ²	
Füvesítés 20%-nál nagyobb rézsűn, talajelőkészítéssel, 50-60 dkg/10 m ²	
Talajjavítás, füvesítés	
Növényültetés	
Földlabdás facsemete ültetése,gödörásással, víztányér készítéssel, öntözéssel ACER PLATANOIDES	
Földlabdás facsemete ültetése,gödörásással, víztányér készítéssel, öntözéssel ACER PLATANOIDES	
Földlabdás facsemete ültetése,gödörásással, víztányér készítéssel, öntözéssel TILIA TOMENTOSA, min 20 cm	
Földlabdás facsemete ültetése, gödörásással, víztányér készítéssel, öntözéssel MALUS PROFUSION, min 20	
Földlabdás facsemete ültetése,gödörásással, víztányér készítéssel, öntözéssel KOELREUTERIA	
Földlabdás facsemete ültetése, gödörásással, víztányér készítéssel, öntözéssel ROBINIA PSEUDOACACIA VAR. UMBRACULIFERA, min 20 cm	
Földlabdás facsemete ültetése, gödörásással, víztányér készítéssel, öntözéssel SALIX ALBA cv. TRISTIS, min	
Földlabdás facsemete ültetése, gödörásással, víztányér készítéssel, öntözéssel FRAXINUS ORNUS, min 20	
Földlabdás facsemete ültetése, gödörásással, víztányér készítéssel, öntözéssel SORBUS AUCUPARIA, min	
Konténeres facsemete ültetése,gödörásással, víztányér készítéssel, öntözéssel CHAMAECYPARIS	
Konténeres facsemete ültetése, gödörásással, víztányér készítéssel, öntözéssel PICEA PUNGENS, 125/150	
Kiültetett fa karózása 2db karóval	
Cserje ültetése szoliterként,gödörásással, víztányér készítéssel, öntözéssel COTONEASTER DAMMERI,	
Cserje ültetése szoliterként, gödörásással, víztányér készítéssel, öntözéssel JUNIPERUS MEDIA "OLD GOLD",	
Cserje ültetése szoliterként, gödörásással, víztányér készítéssel, öntözéssel COTONEASTER	
Szabadgyökerű cserje ültetése, gödörásással, víztányér készítéssel, öntözéssel BERBERIS JULIANAE, 40/60	
Szabadgyökerű cserje ültetése, gödörásással, víztányér készítéssel, öntözéssel FORSITHIA x INTERMEDIA	
Szabadgyökerű cserje ültetése sövényként, gödörásással, víztányér készítéssel, öntözéssel LIGUSTRUM	
Kéregtakarás a parkolósávok közti zöld sávban 5 cm vtg-ban	
Humusz visszatöltés az ültető gödrökbe	
Szellőző és öntöző drén elhelyezése facsemetéknél	
A telepített növényzet gondozása az áruház átadását követő két év időtartamban	
Növényültetés	

3.1-5 táblázat: a D tevékenységcsoport 67-es feladatcsoportjának egyes tevékenységei

Miután minden egyes tevékenységről megállapítottam, mely feladatcsoportba tartozik, meghatározhattam a tevékenységek erőforrásigényét. Az erőforrásigényeket a feladatcsoportok bérköltségeiből számítottam ki. A feladatcsoportok bérköltsége három tényező szorzatából tevődik össze: erőforrások összegének, a munkaórák számának valamint a munkabérnek a szorzatából. A teendő tehát a következő volt: a feladatcsoportok bérköltségét leosztani a munkabérrel, majd a munkaórák számával, azaz a tevékenységek időtartamainak összegével (órában mérve), majd az így kapott erőforrásigényt szétosztani az

egy-tes tevékenységek között. Mindezek előtt azonban ki kellett számítani az egyes tevékenységek időtartamát órában mérve, mivel a vállalat az időtartamokat napokban adta meg.

Példaként tekintsük a 61-es feladatcsopórtot, melybe a külső vízellátás tevékenységei tartoznak (3.1-6 táblázat). Ezen belül három tevékenység van: földmunka, csővezeték-szerelvények lefektetése valamint az üzembe helyezés, kipróbálás. A feladatcsopórt bérköltsége 3569735 Ft. Ezt az összeget osztottuk a munkabérrel, valamint a tevékenységek – órákban mért – időtartamával, így megkaptuk az erőforrásigények összegét, 62-t. Tehát a földmunkán, a csővezeték-szerelvények lefektetésén valamint az üzembe helyezésén összesen 62 ember dolgozott. Ezt a 62 embert el kellett osztani az egyes tevékenységek között. Az időtartamok rendre 120, 80 és 24 óra. Matematikailag megfogalmazva a problémát, 120 óráig dolgozott x ember, 80 órán át y , 24 órán át pedig z . Ezek alapján felírhatjuk az alábbi összefüggéseket: $120 \cdot x + 80 \cdot y + 24 \cdot z = 3569735 / \text{munkabér}$, valamint $x + y + z = 62$. E két egyenletből a földmunkára és a csővezeték-szerelvények lefektetésére 21-21 ember jutott, a próbákra, üzembe helyezésre pedig 20. Elmondható, hogy egy adott feladatcsopórtot általában azonos típusú munkaerővel meg lehetett valósítani. A feladatcsopórtok tevékenységei nem tartalmaztak átlapolást, így elsősorban az összes erőforrás-szükségletre voltunk kíváncsiak, illetve arra, hogy hány emberrel lehet ezt a projektet megvalósítani.



3.1-6 táblázat: a „Vízellátás” feladatcsoportjának egyes tevékenységei

Ezzel a módszerrel az erőforrásigények egész jó közelítését kapjuk. A 3.1-6 táblázatból látható, hogy a vállalat az egyes tevékenységek kezdési és befejezési időpontját pontos dátumokkal adta meg. A jelenlegi programunk azonban – amely még tesztelés alatt áll –, csak egész számokkal tud dolgozni. Ezért minden egyes dátum helyett egy egész számot adtam meg, amely azt mondja meg, hogy az adott tevékenység a projekt kezdési időpontjához képest hány munkaóra múlva fog elkezdődni, illetve befejeződni. A kezdő tevékenység a 0. órában kezdődik el, ehhez képest adtuk meg a többi tevékenység kezdési idejét. Ennek számítását egyszerűen az Excel-cellákban végeztük oly módon, hogy az adott tevékenység tényleges kezdési időpontjából kivontam a projekt kezdési idejét. Ekkor megkaptam, hogy az adott tevékenység hány nappal kezdődött később a projekt kezdetéhez képest. Ha ezt beszorozzuk a munkaórák számával (figyelembe véve, hogy a cég átlag napi 10 munkaórával számolt szombat, vasárnapot is beleértve) megkapjuk, hogy a kezdési időponthoz képest mennyi munkaórával később kezdődött el az adott tevékenység.

A módszer használatához szükséges, hogy megadjuk minden egyes tevékenység követő tevékenységét vagy tevékenységeit, hogy megkapjuk a tevékenységek hálóját. A háló egy körmentes, irányított gráf, amelynek egy kezdő és egy végpontja van. A gráf csomópontjai maguk a tevékenységek. A háló definícióját teljesítő gráf egyben mindig topológikusan rendezhető. Ha a hálót topológikusan rendezzük (vagyis visszafelé mutató nyilakat nem engedünk meg), akkor a hálóból pontosan látni lehet, hogy az egyes tevékenységek elvégzése után mely tevékenységek következhetnek. Minden olyan gráf, mely irányított és körmentes, topológikusan rendezhető. A program topológikusan rendezte is a gráfot a könnyebb áttekinthetőség érdekében.

A rendelkezésre álló megelőző tevékenységek alapján állapítottam meg a tevékenységek követő tevékenységeit. Több esetben a tevékenységek közötti kapcsolat nem szigorú vég-kezdet jellegű volt, hanem megfigyelhetők bizonyos helyeken átlapolások is. Ez azt jelentette, hogy nem kell megvárni, míg a megelőző tevékenység teljesen befejeződik, el lehet előbb is kezdeni az aktuális tevékenységet. Ilyen esetekben szükségessé vált a megelőző tevékenység szétbontása kettő (esetleg több) résztevékenységre. Például:

	Task Name	Duration	Start	Finish	Predecessors
114	Tömbalapok betonozása	8,8 days	Mon 04.01.12	Mon 04.01.26	113SS
115	Előregyártott kehelynyak	4 days	Mon 04.01.26	Fri 04.01.30	114FS-4 days
116	Monolit vasbeton kehelytalp, tömbalapok	7,2 days	Mon 04.01.26	Thu 04.02.05	115FS-4 days
117	Szerelő, fagykizáró betonok	3,2 days	Thu 04.02.05	Tue 04.02.10	116SS
118	Előregyártott talpgerendák	7,2 days	Tue 04.02.10	Fri 04.02.20	117SS
119	Monolit talpgerendák	9,6 days	Tue 04.02.10	Wed 04.02.25	117SS
120	Visszatöltések	4,8 days	Mon 04.03.01	Mon 04.03.08	118FS-7 days
121	Előregyártott vasbeton oszlop	4 days	Mon 04.03.01	Fri 04.03.05	118FS-7 days

3.1-7 táblázat: több tevékenység megelőző tevékenysége vég-kezdet kapcsolatban volt megadva

A 3.1-7 táblázatban több olyan tevékenységet is látunk, ahol a megadott megelőző tevékenység száma után „FS-4 days” vagy „FS-7 days” áll. Tekintsük a 115-ös tevékenységet. Ez a tevékenység akkor kezdődhet el legkorábban, amikor a 114-es tevékenység időtartamából már csak 4 nap van hátra. A 114-es tevékenységet tehát két részre osztottuk: az első része 4.8 napig tart, a második része pedig 4 napig. Az erőforrásigény mindkét résztevékenységnél ugyanannyi maradt, mint az eredeti tevékenységnél, a változás csak a két résztevékenység időtartamában, illetve a befejezési és kezdési idejében jelentkezett. Ezeket az átalakításokat csak a program alkalmazhatósága érdekében végeztem el. Tényleges megszakítás nem történt a tevékenységek között.

Elmondható, hogy a kereskedelemben kapható projektmenedzsment-szoftverek legtöbb lehetőségét egy átlagos projektszervezet nem használja ki. Fontos, hasznos funkciók viszont a legtöbb kereskedelmi szoftverből hiányoznak, illetve kezelésük igen nehézkes (pl. optimális erőforrás-allokáció megtalálása, tevékenységek, költségigények, erőforrásigények bizonytalanságának meghatározása stb.)

Mivel a csúszások egyik oka lehet, hogy nem megfelelően használja fel a vállalat a rendelkezésére álló erőforrásokat, így fontos szempont – főleg nagyobb beruházások esetén – az erőforrások helyes felhasználása, valamint az egyes tevékenységekhez kapcsolódó erőforrásigények összevetése a rendelkezésre álló erőforrás-kerettel. Ahol sok párhuzamos tevékenység folyik egy időben, és ezek elvégzéséhez pl. több munkaerőre van szükség, mint amennyi a cég rendelkezésére áll, ott vagy más vállalat segítségét kell igénybe vennünk (pl. alvállalkozók bevonása), ha időben el akarjuk végezni a tevékenységeket (időkorlátos erőforrás-allokáció), vagy ha erre nincs mód, akkor időben későbbre kell beütemezni a tevékenységeket (erőforrás-korlátos erőforrás-allokáció).

Szintén nagy hiányossága a kereskedelemben használt szoftvereknek, hogy nagyon nehézkesen lehet csak velük kezelni az alvállalkozásoknak kiadott munkákat. Itt ugyanis arról van szó, hogy az erőforrásokkal nem az adott vállalat, hanem az alvállalkozók rendelkeznek. Szerződés szerint, meghatározott díj ellenében, adott időpontra elvégeznek egy (vagy több) tevékenységet. Többek között ez az oka, hogy azok a vállalkozások, akik alkalmaznak is valamely projektkezelő szoftvert, megelégszenek a tevékenységek ütemezésével.

120	Tömbalapok betonozása 1.	48	530	2320	530	18	▲ NextActivities ▲ NextID (6) <table><tr><td></td><td>Rbc</td><td>Text</td></tr><tr><td>1</td><td>121</td><td></td></tr><tr><td>2</td><td>122</td><td></td></tr><tr><td>3</td><td>123</td><td></td></tr><tr><td>4</td><td>124</td><td></td></tr><tr><td>5</td><td>125</td><td></td></tr><tr><td>6</td><td>128</td><td></td></tr></table>		Rbc	Text	1	121		2	122		3	123		4	124		5	125		6	128	
	Rbc	Text																										
1	121																											
2	122																											
3	123																											
4	124																											
5	125																											
6	128																											
121	Tömbalapok betonozása 2.	40	578	2368	578	18	▼ NextActivities																					
122	Előregyártott kehelynyak	40	670	2420	670	16	▼ NextActivities																					
123	Monolit vasbeton kehelytalp, tömbalapok	72	670	2420	670	16	▼ NextActivities																					
124	Szerelő, fagykizáró betonok	32	770	2490	770	18	▲ NextActivities ◀ NextID 212																					
125	Előregyártott talpgerendák 1.	2	820	2112	820	19	▲ NextActivities ▲ NextID (5) <table><tr><td></td><td>Rbc</td><td>Text</td></tr><tr><td>1</td><td>126</td><td></td></tr><tr><td>2</td><td>129</td><td></td></tr><tr><td>3</td><td>130</td><td></td></tr><tr><td>4</td><td>136</td><td></td></tr><tr><td>5</td><td>145</td><td></td></tr></table>		Rbc	Text	1	126		2	129		3	130		4	136		5	145				
	Rbc	Text																										
1	126																											
2	129																											
3	130																											
4	136																											
5	145																											
126	Előregyártott talpgerendák 2.	30	822	2114	822	19	▲ NextActivities ▲ NextID (3) <table><tr><td></td><td>Rbc</td><td>Text</td></tr><tr><td>1</td><td>127</td><td></td></tr><tr><td>2</td><td>132</td><td></td></tr><tr><td>3</td><td>133</td><td></td></tr></table>		Rbc	Text	1	127		2	132		3	133										
	Rbc	Text																										
1	127																											
2	132																											
3	133																											
127	Előregyártott talpgerendák 3.	40	852	2144	852	19	▼ NextActivities																					
128	Monolit talpgerendák	96	820	2370	820	19	▼ NextActivities																					
129	Visszatöltések	48	1020	2450	1020	16	▼ NextActivities																					

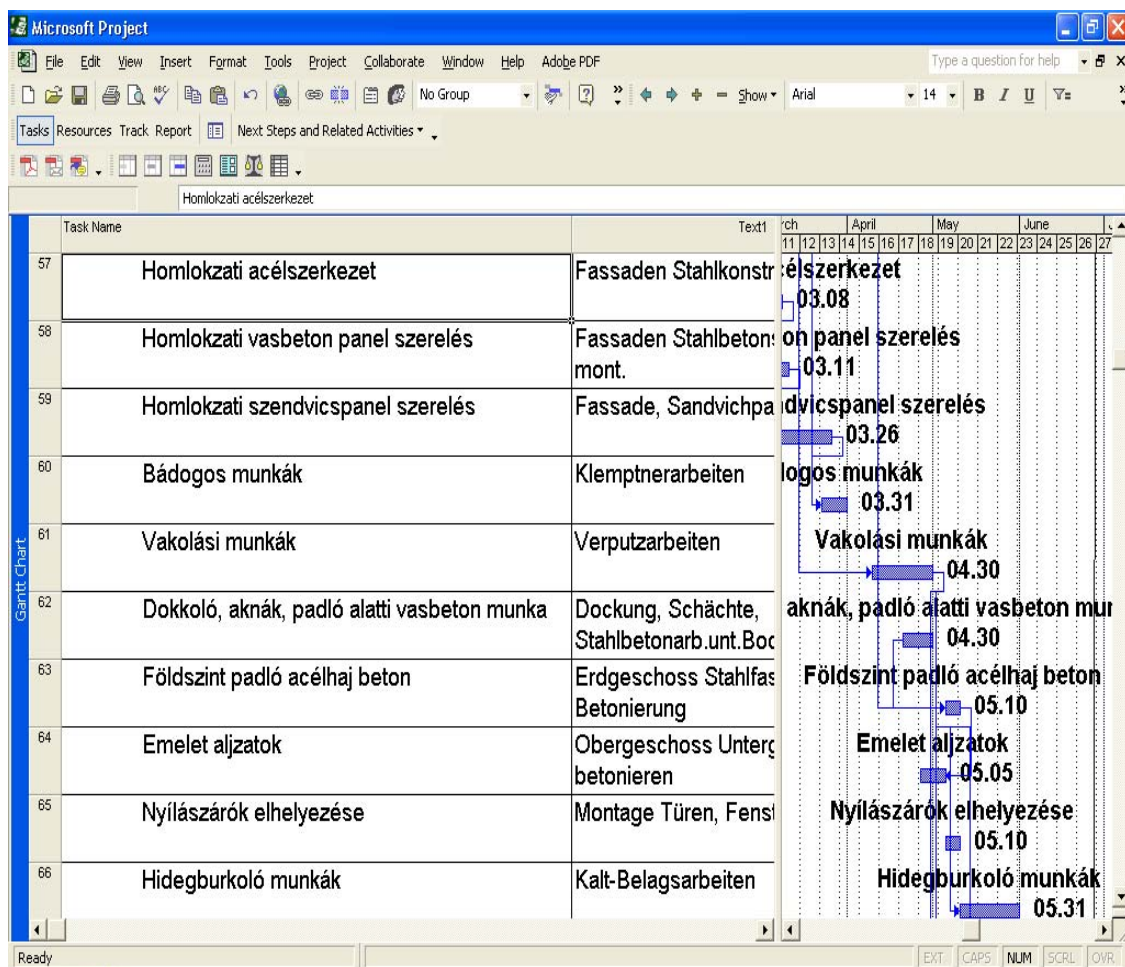
3.1-8 táblázat: az előző táblázat megvalósítása .XML állományban. Az egyes oszlopok balról jobbra a következők: tevékenység sorszáma, neve, időtartama, legkorábbi kezdésének ideje, legkésőbbi kezdésének ideje, tényleges kezdésének ideje, erőforrásigénye, követő tevékenységei. Néhány tevékenység követő tevékenységeit részletezve láthatjuk

3.2 Alkalmazott informatikai technológiák bemutatása

Egy projektszervezet dönthet úgy, hogy nem alkalmazza az ütemezés és erőforrás-allokációs módszer lehetőségeit, illetve olyan informatikai alkalmazásokat, melyekkel optimális erőforrás-allokációt határozhat meg. Ebben az esetben azonban – főleg nagyobb projektek esetén – nehezen biztosítható, hogy a projekt meghatározott időn belül befejeződjön. Kisebb projektek esetén gyakran eltekintenek az ütemezés és erőforrás-allokációs módszer lehetőségeitől, mert azt egy tapasztalt projektvezető sok éves tapasztalatából adódóan átlátja, és az egyes váratlan eseményeket kezelni tudja.

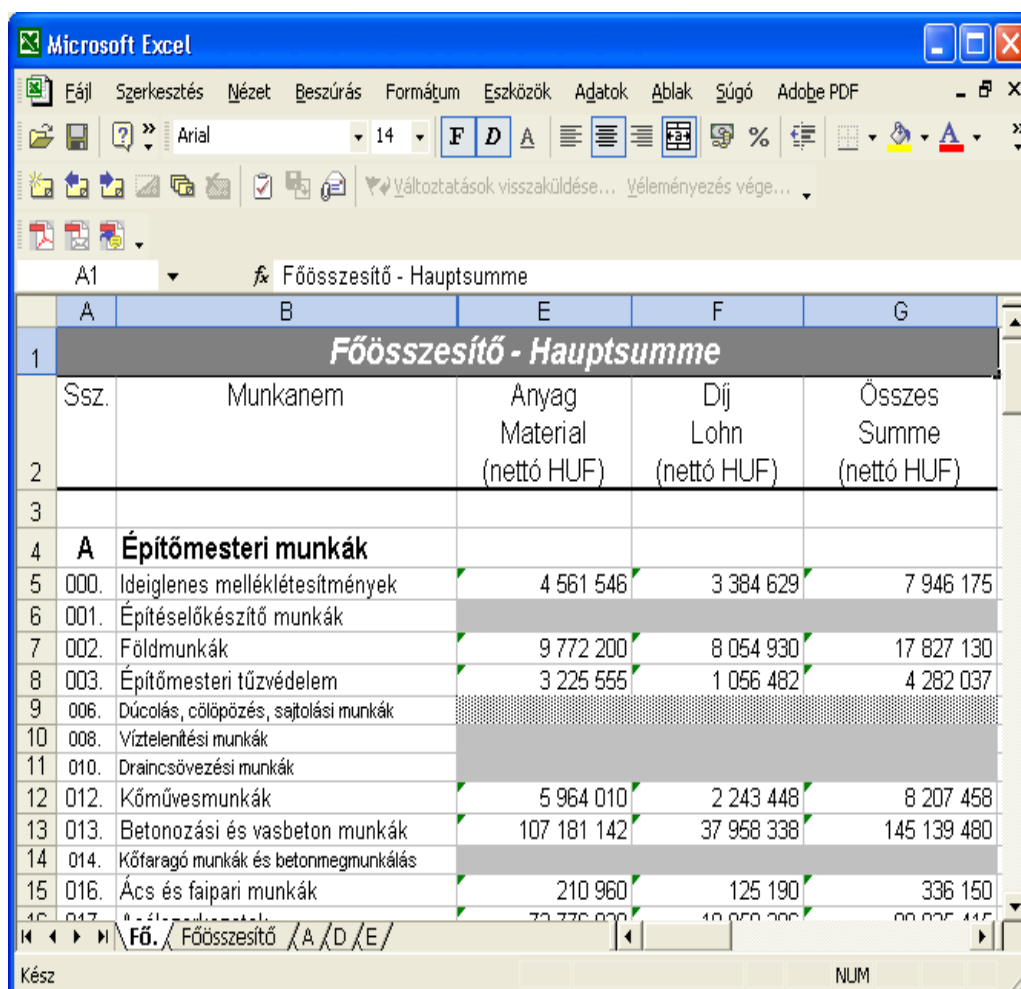
Ha egy vállalat csak ütemezi a projektben elvégzendő tevékenységeit, akkor is meg kell becsülnie az egyes tevékenységek várható időtartamát. Két lehetősége van: vagy fix időtartamként kezeli az egyes tevékenységek időtartamát, és az utólagos korrekciókat később végzi majd el a tervben, vagy eleve valószínűségi változóként kezeli a tevékenységek időtartamát, ezzel bizonyos határokon belül kezelni tudja a projekt átfutási idejének bizonytalanságát. Tervezni tudja, hogy adott valószínűségi szint mellett várhatóan mikor fog befejeződni a projekt. A bemutatandó projektben fix időtartamokkal dolgoztak. Ennek egyik oka, hogy a vállalat által használt Microsoft Project kezeli ugyan a tevékenységek időtartamának bizonytalanságát, azonban ezek a lehetőségek igen korlátozottak.

A vállalat által alkalmazott projektmenedzsment-szoftver széles körben alkalmazott ütemező és erőforrás-allokáló szoftver. Számos kényelmi funkciója (pl. projektnaptár, erőforrások, költségek időbeli felmerülésének nyomkövetése stb.) segíti a projektmenedzser munkáját.



3.2-1 ábra: a Microsoft Project kezelőfelülete

Rendelkezésre állt egy Microsoft Project által készített ütemterv. Ezzel a szoftverrel lehetőség van a tevékenységek logikai összerendelésére. Ebből a program automatikusan kiszámítja a tevékenységek legkorábbi és legkésőbbi kezdését, illetve befejezését. Természetesen lehetőség van egy-egy tevékenység kezdési idejét közvetlenül is megadni. A logikai összerendelések sem kötelező jellegűek. Ha viszont a meglévő logikai kapcsolatokat nem modellezzük, akkor a tevékenységek esetleges csúszása esetén a rákövetkezési relációban lévő, de a feladatban logikai kapcsolattal nem modellezett tevékenységek csúszása nehezen követhető nyomon. Ilyen csúszások sok esetben az erőforrások helytelen felhasználásából, külső környezeti hatásokból vagy előre nem várt okokból adódnak (pl. hosszú esőzés, hosszantartó fagy stb.). Mivel a vállalat ebben a programban sem a költségek felmerülésével, sem pedig az erőforrás-szükségletekkel nem számolt, így sok tevékenység logikai összerendelése elmaradt, illetve hiányosan állt rendelkezésre. Először össze kellett rendelni a tevékenységeket megfelelő logikai sorrendben.



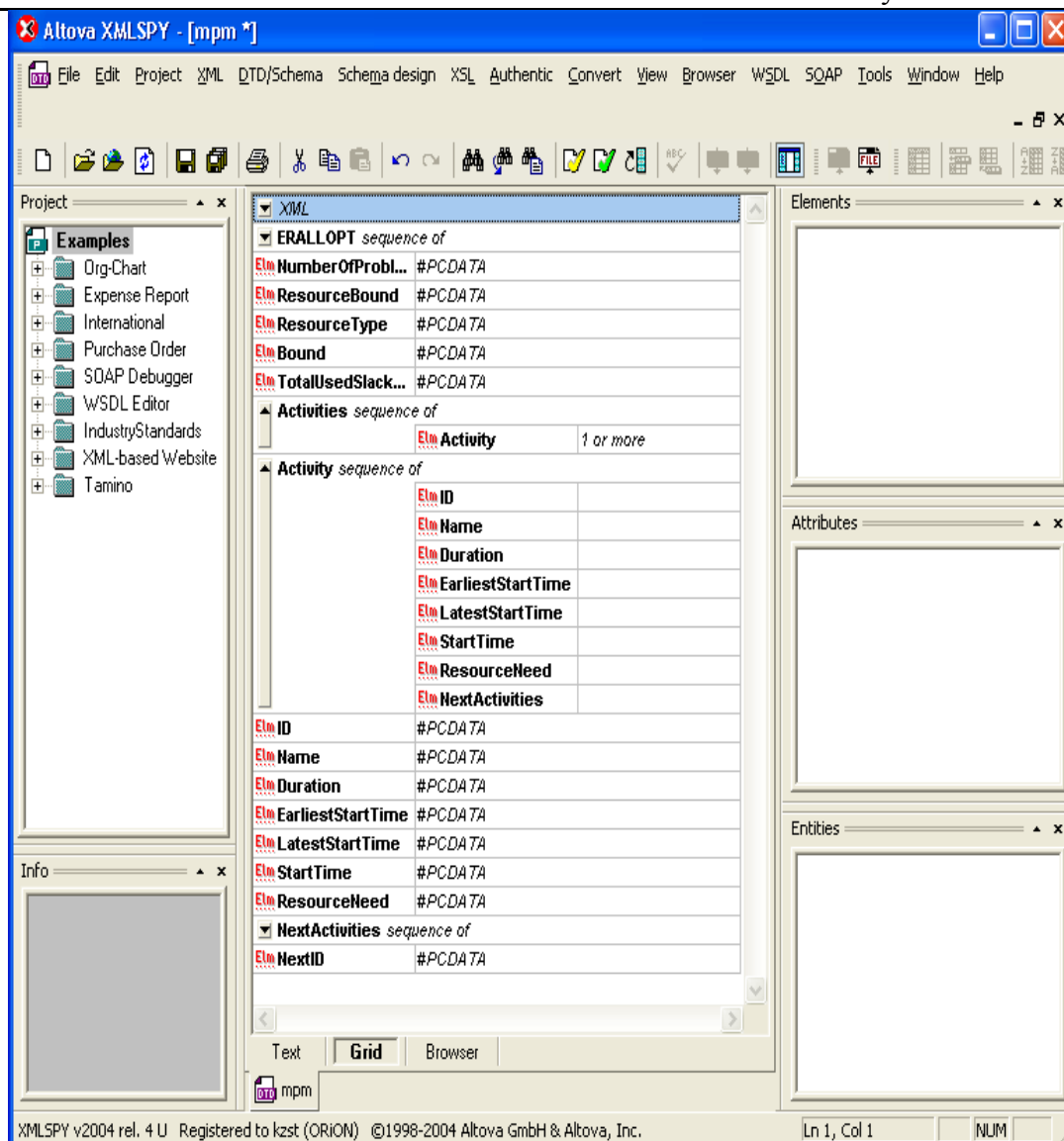
	A	B	E	F	G
1	Főösszesítő - Hauptsumme				
	Ssz.	Munkanem	Anyag Material (nettó HUF)	Díj Lohn (nettó HUF)	Összes Summe (nettó HUF)
2					
3					
4	A	Építőmesteri munkák			
5	000.	Ideiglenes melléklétesítmények	4 561 546	3 384 629	7 946 175
6	001.	Építésselőkészítő munkák			
7	002.	Földmunkák	9 772 200	8 054 930	17 827 130
8	003.	Építőmesteri tűzvédelem	3 225 555	1 056 482	4 282 037
9	006.	Dúcolás, cölöpözés, sajtólási munkák			
10	008.	Vízelenítési munkák			
11	010.	Draincsövezési munkák			
12	012.	Kőművesmunkák	5 964 010	2 243 448	8 207 458
13	013.	Betonozási és vasbeton munkák	107 181 142	37 958 338	145 139 480
14	014.	Kőfaragó munkák és betonmegmunkálás			
15	016.	Ács és faipari munkák	210 960	125 190	336 150
16	017.	Főösszesítő	72 776 800	10 000 000	82 776 800

3.2-2 ábra: a Microsoft Excel kezelőfelülete

A költségigényeket külön Microsoft Excel táblázatban kaptam meg. Az elvégzendő tevékenységek anyagköltségei, illetve bérköltségei is szerepelnek egy-egy tevékenység mellett. Ha tudjuk, hogy egy tevékenységnek mennyi a bérköltsége, az időtartama, valamint számolhatunk egy órabérrel, akkor hozzávetőlegesen meg tudjuk becsülni az emberi erőforrás-szükségletet a következő képlettel: $\text{bérköltség (adott tevékenységre) (Ft)} = \text{tevékenység időtartama (nap)} \times \text{munkaóra egy nap alatt (óra/nap)} \times \text{órabér (Ft/óra)} \times \text{munkások száma}$. Ha ezen adatokat meghatároztuk, akkor a feladatra kereshető egy megengedett erőforrás-allokáció adott erőforráskorlát esetén. Optimális megoldást egy általunk kifejlesztett párhuzamos Branch and Bound módszeren alapuló erőforrás-optimaló algoritmussal kerestünk. Ehhez át kellett az adatokat konvertálni .XML formátumú file-ba, mivel ez a program ilyen típusú file-okból olvassa be az adatokat.

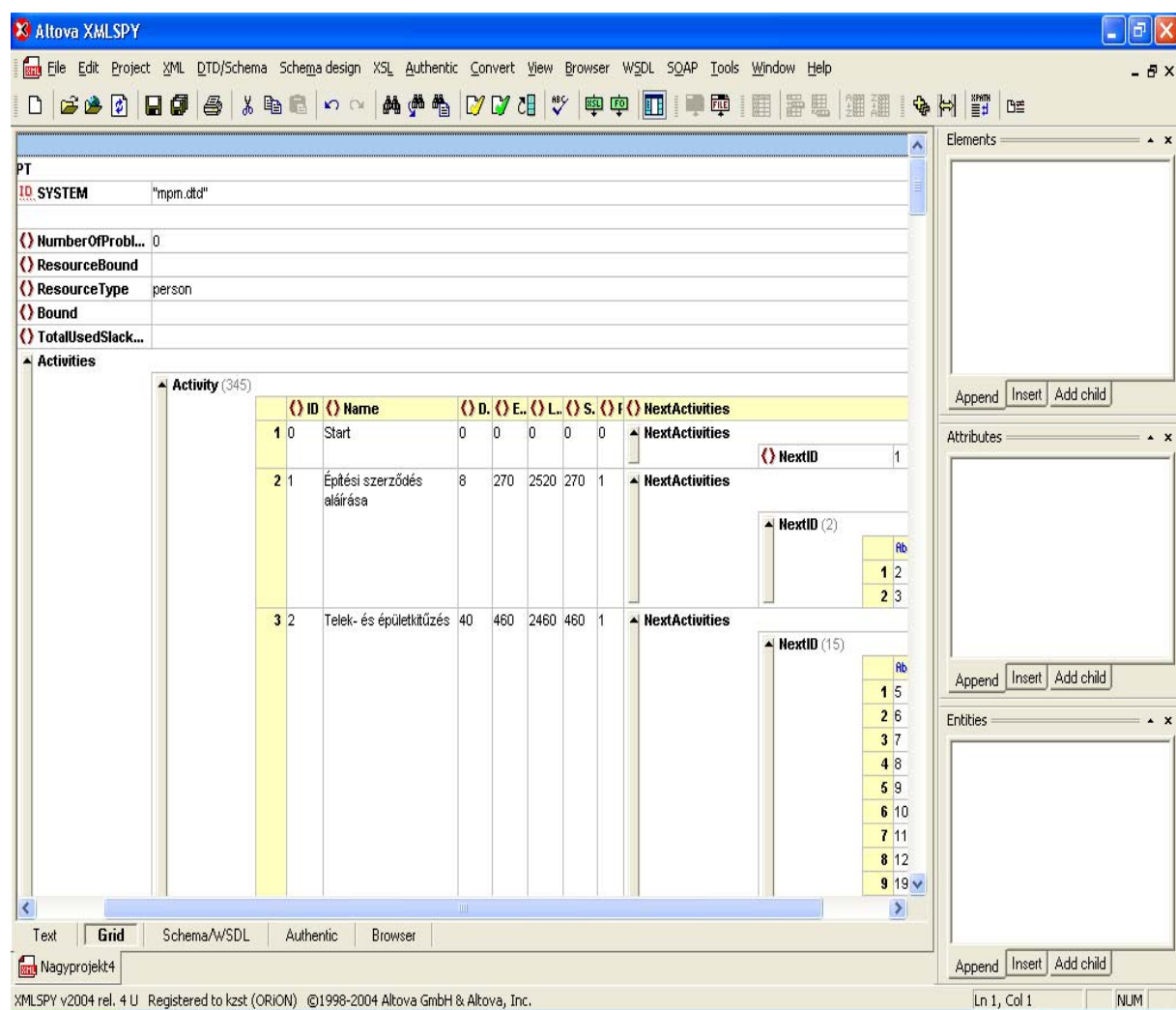
Az .XML (eXtended Markup Language) formátumú tárolás egy széles körben elterjedt tárolási módszer. A legtöbb szoftvernek – így a Microsoft Projectnek is – van .XML-file kimenete. Ennek a tárolási módszernek előnye, hogy lényegében bármilyen információt, szöveget, képet, videót, táblázatot, különböző speciális adatokat tárolhatunk. A tárolás mikéntjére egy úgynevezett definíciós file-t készíthetünk, mellyel az adatok helyes tárolását ellenőrizhetjük.

A feldolgozás során mi is kialakítottunk egy nagyon egyszerű tárolási szabályt, mely csak a legszükségesebb adatokat tartalmazza: erőforrás-korlát (Resource Bound), erőforrás-típus (Resource Type), valamint a tevékenységek csoportja (Activities). Ezenkívül tartalmaz olyan mezőket is a definíciós file, melyeket nem a felhasználóknak kell kitöltenie, hanem az erőforrás-optimaló program fogja ezeket futás közben kitölteni; ilyen pl. a Branch and Bound fában a problémák szétbontása részproblémákká (Number of Problem State), illetve a korlátszámító függvény értéke (Bound), valamint az összes felhasznált tartalékidő összege (Total Used Slack Time). A definíciós file-t az Altova XMLSPY program segítségével készítettük, mely egy nagyon könnyen használható .XML-file szerkesztő program.



3.2-3 ábra: az Altova XMLSPY használata .XML definíciós fájl készítéséhez

Egy-egy tevékenység esetében további adatokat is tárolunk: a tevékenység azonosítóját (ID), melyre összerendelés során hivatkozhatunk; a tevékenység nevét (Name); a tevékenység időtartamát (Duration); legkorábbi kezdési idejét (Earliest Start Time), mely a tevékenység tényleges kezdésének alsó korlátja lesz (ezt a logikai háló segítségével is meghatározhatjuk, illetve mi magunk is módosíthatjuk); tároljuk továbbá a tevékenység legkésőbbi kezdési idejét (Latest Start Time) is; ezen kívül tároljuk a tevékenységek erőforrásigényét, valamint a követő tevékenységek azonosítóját is. Ehhez hasonló .XML file-t kaphatunk, ha a Microsoft Project által használt projekt file-t .XML-formátumban mentjük ki. A projektben kiszámított adatokat – átalakító program híján – manuálisan vittük be az előre kialakított definíciós file-nak megfelelően. (A konvertáló program jelenleg fejlesztés alatt áll.)



3.2-4 ábra: az Altova XMLSPY használata .XML-file adatainak felviteléhez

A definíciós file-nak megfelelően a program minden mentésnél ellenőrzi, hogy az előre definiált szabályoknak megfelelően vittük-e be az adatokat. Az optimáló program jelenlegi megvalósítása egész számokkal dolgozik, így az Excel program segítségével a kezdési, lefutási és befejezési időket átalakítottam munkaórákra. 10 munkaórát számolva naponként könnyen visszaírható az eredmény az eredeti file-ba.

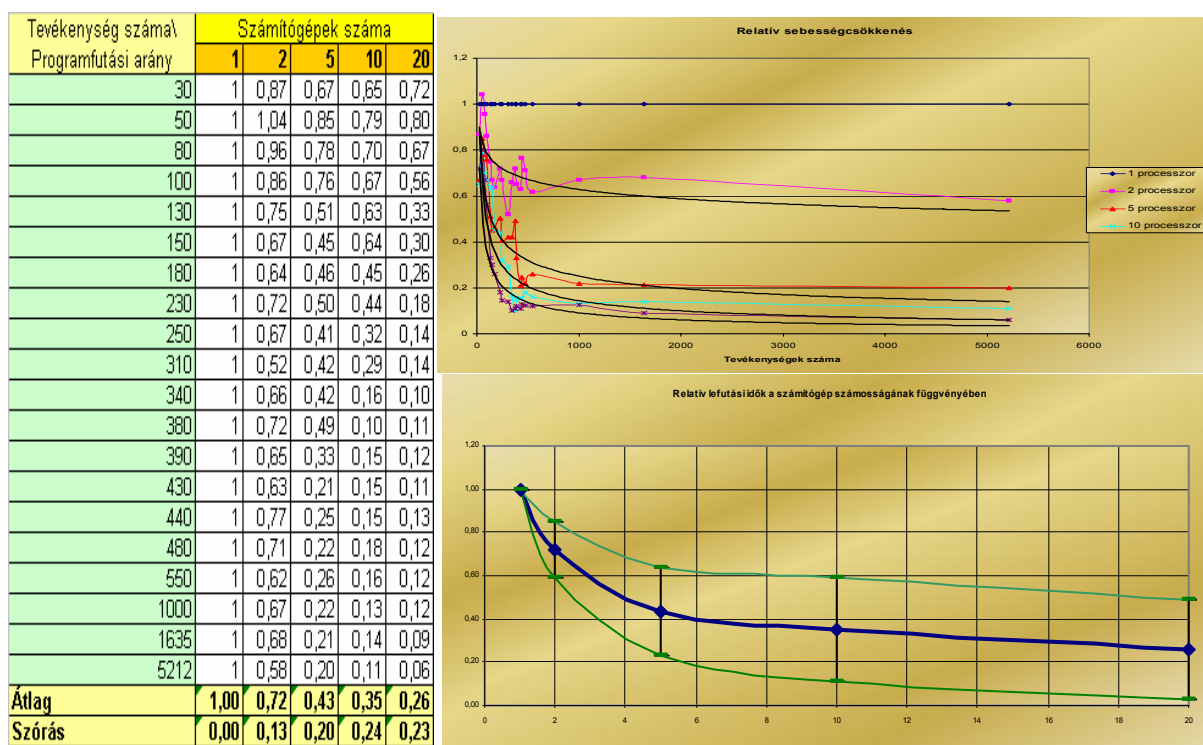
A kapott .XML-file-ban lévő adatokat ellenőrzésképpen kirajzoltattam egy általam készített programmal. Ennek a kirajzoláson kívül az volt a feladata, hogy a megoldóprogramhoz tesztfeladatokat gyártson, hogy annak helyes működését tesztelni lehessen. A tesztelés során a megoldóprogram sokkal gyorsabban oldotta meg a feladatokat, mint azon módszerekre készített szoftverek, melyeket az 1.6.2-es fejezetben bemutatam.

Szintek száma	10	20	20	30	30	40	40	40	40	40	40	40	40	40	40	40	50	50	60	100
Csúcsok száma	6	6	8	6	8	6	8	12	10	14	16	18	20	20	20	24	24	40	60	100
Max.r./lépés	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	3	4	4	5	5
Erőforrás korlát	28	28	49	30	40	38	40	50	45	55	60	80	75	80	70	85	85	100	175	300
Tevékenységek száma	30	50	80	100	130	150	180	230	250	310	340	380	390	430	440	480	550	1000	1635	5212
Elmozgatott tev. száma	6	4	3	14	12	13	18	24	24	20	51	21	25	36	23	57	63	92	112	254
1. mérés (ms)	15	22	28	32	36	40	42	44	45	50	52	54	58	62	78	120	188	342	812	4524
2. mérés (ms)	14	21	27	31	35	39	41	43	44	49	51	53	57	61	76	114	157	335	805	4507
3. mérés (ms)	12	20	27	30	35	39	40	43	43	49	51	52	57	60	76	109	152	335	801	4501
4. mérés (ms)	12	20	26	30	34	38	40	42	42	48	50	52	56	60	75	108	152	334	785	4421
5. mérés (ms)	12	19	26	30	34	38	39	42	42	47	50	52	56	58	75	107	151	330	775	4321
6. mérés (ms)	11	19	26	29	34	37	39	40	41	47	49	51	56	57	74	107	151	329	771	4212
7. mérés (ms)	10	18	26	29	33	37	39	40	41	46	49	50	55	57	72	106	142	327	770	4201
8. mérés (ms)	10	18	24	29	32	36	38	39	40	46	48	49	54	55	68	102	140	325	765	4136
9. mérés (ms)	10	17	24	28	32	36	37	39	39	46	48	48	54	54	64	102	138	320	764	4102
10. mérés (ms)	10	17	24	27	30	35	37	38	39	45	47	47	53	54	64	101	137	315	760	4024
Átlag	11,60	19,10	25,80	29,50	33,50	37,50	39,20	41,00	41,60	47,30	49,50	50,80	55,60	57,80	72,20	107,60	150,80	329,20	780,80	4294,90
Szórás	1,78	1,66	1,40	1,43	1,78	1,58	1,62	2,05	2,01	1,64	1,58	2,25	1,58	2,90	5,09	5,83	14,81	7,91	18,84	185,01
Relatív szórás	0,15	0,09	0,05	0,05	0,05	0,04	0,04	0,05	0,05	0,03	0,03	0,04	0,03	0,05	0,07	0,05	0,10	0,02	0,02	0,04

3.2-1 táblázat: a megoldó program lefutási sebessége ms-ban

A megoldó program 1000 tevékenység esetén $9000/329,2 = 27,28$ -szor gyorsabb, mint az eddig leggyorsabbnak tartott dinamikus programozáson alapuló szoftverek. (Ráadásul a JINI-alkalmazásnak köszönhetően a megoldás sebessége mérésről mérésre egyre kisebb, hiszen a rendszer maga is alkalmazkodik a feladathoz, és legközelebb már hatékonyabban osztja el a számítógép erőforrásait.) Látható, hogy nagy számú, akár 5 000 tevékenységet tartalmazó projektet is 5 másodperc alatt optimalizálhatunk.

A fenti adatok természetesen 1 számítógép használata esetén értendők. 300 tevékenység feletti projektek optimalizálása során már érdemes esetleg a vállalatban már meglévő hálózatot is kihasználni egy-egy nagyobb feladat megoldására.

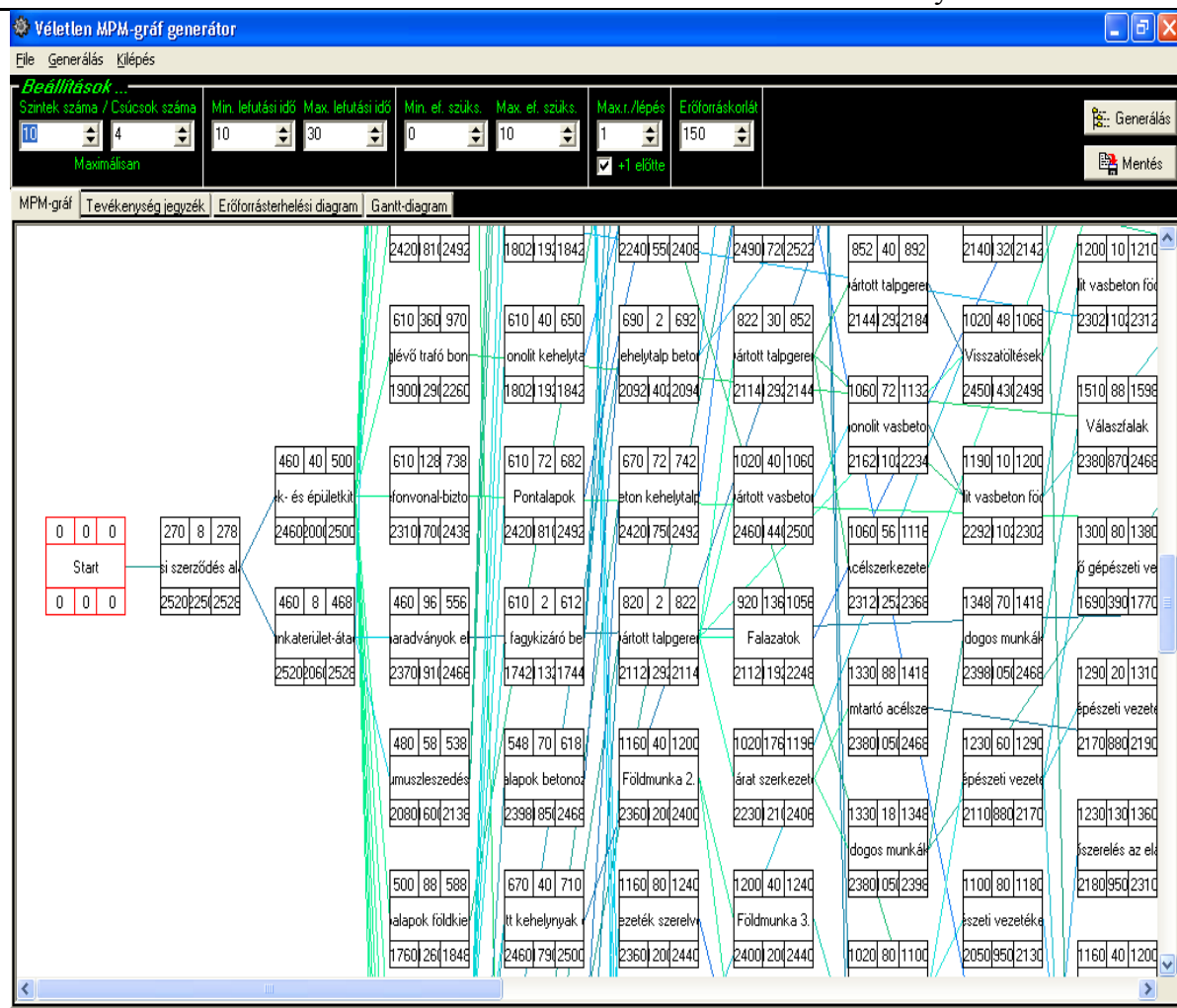


3.2-5 ábra: az optimalizálási idő gyorsítása több számítógép egyidejű használatával

A táblázatban a tevékenységek relatív futási idői szerepelnek az egy számítógépet használó lefutási időt 1-nek (100%-nak) tekintve. Látható, hogy jelentős gyorsítás csak megfelelő számú (legalább 300) tevékenység optimalizálásakor jelentkezik.

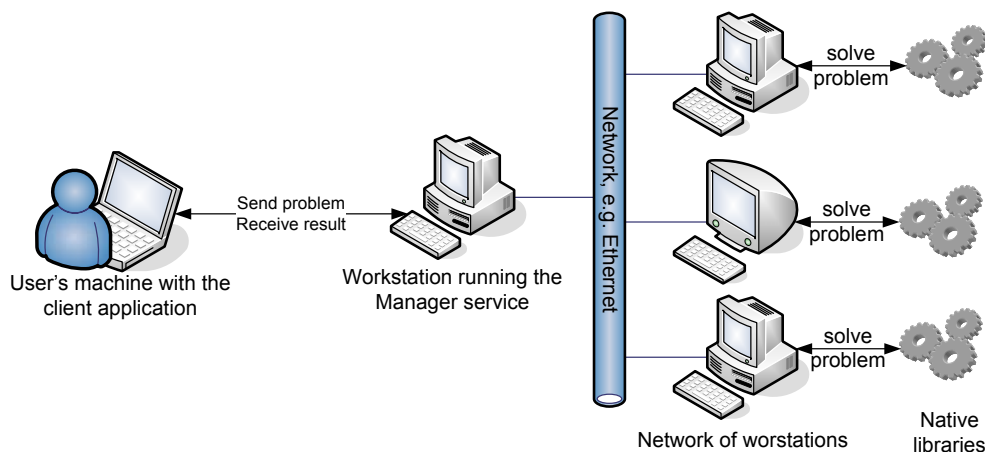
Magába a véletlengenerátor programba is számos funkciót beépítettem, mely az ellenőrzést szolgálta (pl. háló időadatainak, erőforrásadatainak vizsgálata, erőforrások, topológikusan rendezett hálós diagram megjelenítése stb.). Mivel a program topológikusan rendezi a hálót, így ellenőrizhető, hogy a gráfban nincs-e kör (ekkor ugyanis topológikusan nem rendezhető a gráf), valamint egy kezdő, és egy befejező tevékenysége van-e a hálónak. A programmal egy véletlengráfot lehetett generálni, mellyel a megoldóprogram helyességét lehetett ellenőrizni. A véletlengráf esetén be lehet állítani a topológikusan rendezett gráf szintjeinek számát, maximális szélességét, és a kapcsolatok maximális számát. (A beállításoknak megfelelő tesztfeladatok paramétereit a 3.2-1 táblázat tartalmazza.)

Ezekkel a beállításokkal tulajdonképpen bármilyen struktúrájú problémát lehetett generálni. Ennél egyszerűbb véletlengráf-generátort korábban már publikáltak. [95, 383] Ezzel a programmal azonban mind determinisztikus, mind sztochasztikus gráfot lehet generálni. Ezenkívül pedig költség-optimalizálást is lehet a hálón végezni mind determinisztikus, mind pedig sztochasztikus esetben.



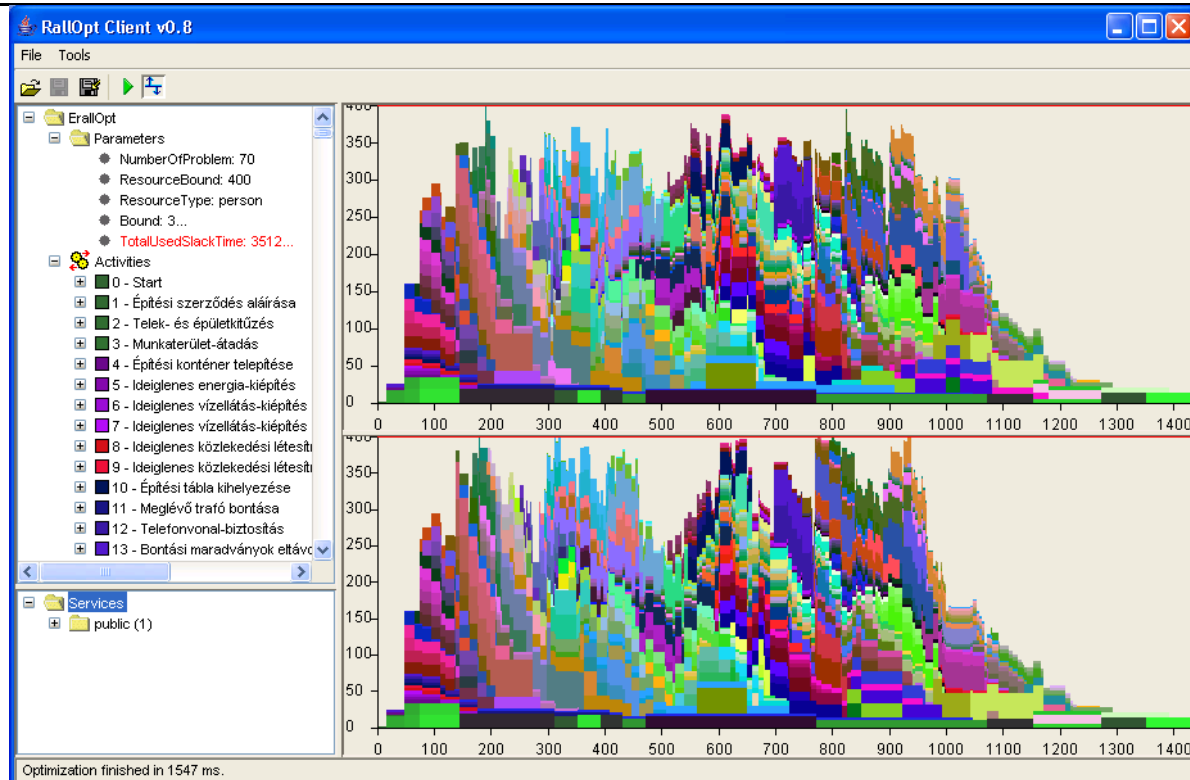
3.2-6 ábra: a feladat kirajzolása, a kapott MPM-háló ellenőrzése, az algoritmus tesztelése véletlenfeladat-generátor segítségével

A már .XML-file-ban elkészített adatokat ezután magának a megoldó programnak adtam át. Ez a program ugyanilyen file-ba írja vissza az eredményeket. Ez a program az általam kifejlesztett erőforrás-allokációs módszer alapján dolgozik. Az eredeti problémát több részproblémára bontja, majd ezeket kiküldi más számítógépekre. Ezek a számítógépek pedig visszaküldik az eredményt. Elsősorban ennek a szétbontásnak 200 tevékenység felett van értelme. Ekkor már számottevő a feladat megoldási idejének csökkenése. A programnak a JaBBa nevet adtuk, mely a Jini and Branch and Bound Algorithm szavak kezdőbetűit tartalmazza.



3.2-6 ábra: a JaBBA problémamegoldó-környezet felépítése

A problémamegoldó-környezet 4 részből áll. A felhasználó (user) elküldi az adatokat a menedzser-kiszolgáló állomásnak (manager service). Ennek a munkaállomásnak a feladata, hogy a feladatot több részfeladatra bontsa, valamint ezeket szétossza a hálózaton. A feladatok megoldásai háttérben több munkaállomáson (workstation) is egyidejűleg folyhatnak. Ennek koordinálása is a kiszolgáló szerver feladata. A munkaállomások olyan megoldófüggvényeket használnak (native libraries), melyekkel az adott részproblémát ki tudják értékelni. A programban alkalmazott JINI-technológia kifejezetten a hálózatos architektúrákra kifejlesztett párhuzamos adatfeldolgozást segítő függvénykönyvtár, mellyel a feladatok szétosztása, illetve a hálózat menedzselése sokkal könnyebben és megbízhatóbban megvalósítható.



3.2-8 ábra: a JaBBA problémamegoldó-környezet kliens oldali felülete

A kliens oldali felhasználói környezet megmutatja az eredeti megengedett erőforrás-allokációt, majd optimalás után felrajzolja az optimális erőforrás-allokáció terhelési diagramját. Mindegyik tevékenységet más színnel jeleníti meg a program, hogy a tevékenységek kezdésének változását nyomon tudjuk követni. Azon tevékenységeket, amelyeknek változott a lefutási ideje, a szoftver külön pirossal kiemeli. Láthatjuk, hogy a tevékenységek kezdési ideje mely értékről csökkent le az adott kezdési időre.

3.3 Eredmények

Az alkalmazott módszerrel jelentősen lehetett a tevékenységek kezdési idejét csökkenteni. Pusztán a megfelelő logikai összerendeléssel 2528-ról 1432 munkaórára lehetett csökkenteni a projekt átfutási idejét. Ez mintegy **43,75%-os** csökkenést jelent. Az optimális megoldás jóságának értékelésére egy mutatószámot dolgoztunk ki. Kíváncsiak voltunk, hogy átlagosan egy tevékenység mikor kezdődhet korábban a megengedett, illetve az optimális megoldásban. A mutatószám kialakítása során figyelembe vettük, hogy mennyivel kezdődhetek korábban a projekt kezdetéhez képest az egyes tevékenységek a megengedett, illetve az optimális megoldásban. Ha az eredeti és az optimált adatokat osztjuk egymással, akkor megkapjuk,

hogyan a tevékenység kezdési ideje hány százalékára csökkent a projekt kezdetéhez képest. Minden tevékenységre elvégezve, az adatokat esetlegesen a tevékenység időtartamával súlyozva megkaphatjuk, hogy egy átlagos tevékenység kezdési ideje hány százalékára csökkent a logikai összerendelések, illetve az erőforrás-optimalás után a projekt kezdetéhez képest. Erre a következő súlyozott értékeket használtuk:

$$Q_{EST}^{megeng.} = \frac{\sum_{i=1}^n d_i \frac{EST_i^{megeng.}}{EST_i}}{\sum_{i=1}^n d_i}, \quad Q_{EST}^{opt.} = \frac{\sum_{i=1}^n d_i \frac{EST_i^{opt.}}{EST_i}}{\sum_{i=1}^n d_i}, \quad Q_{AST}^{megeng.} = \frac{\sum_{i=1}^n d_i \frac{AST_i^{megeng.}}{AST_i}}{\sum_{i=1}^n d_i}, \quad Q_{AST}^{opt.} = \frac{\sum_{i=1}^n d_i \frac{AST_i^{opt.}}{AST_i}}{\sum_{i=1}^n d_i}. \quad (3.3-1)$$

A változást mind a legkorábbi kezdésre (EST = Earliest Start Time), mind pedig a tényleges kezdési időkre (AST = Actual Start Time) megvizsgáltuk, a megengedett és az optimális megoldások esetén is meghatároztuk ezeket az értékeket. Azt szeretnénk volna tudni, hogy átlagban hány százalékára csökkentek a tevékenységek kezdési ideje az eredeti adatokhoz képest a projekt kezdetéhez viszonyítva. Az átlagos érték kiszámításánál súlyoztuk a kezdési idejének hányadosát a tevékenységek időtartamaival.

Az eredményeket az alábbi táblázatban foglaltuk össze (az értékek %-ban értendők):

$Q_{EST}^{megeng.} = \frac{\sum_{i=1}^n d_i \frac{EST_i^{megeng.}}{EST_i}}{\sum_{i=1}^n d_i}$	$Q_{EST}^{opt.} = \frac{\sum_{i=1}^n d_i \frac{EST_i^{opt.}}{EST_i}}{\sum_{i=1}^n d_i}$	$Q_{AST}^{megeng.} = \frac{\sum_{i=1}^n d_i \frac{AST_i^{megeng.}}{AST_i}}{\sum_{i=1}^n d_i}$	$Q_{AST}^{opt.} = \frac{\sum_{i=1}^n d_i \frac{AST_i^{opt.}}{AST_i}}{\sum_{i=1}^n d_i}$
31,72%	31,72%	38,31%	37,42%

3.3-1 táblázat: a tevékenységek (legkorábbi/tényleges) kezdési időinek változása (súlyozott értékek)

Súlyozás nélkül az adatokat az alábbi táblázatban foglaltuk össze (az értékek %-ban értendők):

$Q_{EST}^{megeng.} = \frac{\sum_{i=1}^n \frac{EST_i^{megeng.}}{EST_i}}{n}$	$Q_{EST}^{opt.} = \frac{\sum_{i=1}^n \frac{EST_i^{opt.}}{EST_i}}{n}$	$Q_{AST}^{megeng.} = \frac{\sum_{i=1}^n \frac{AST_i^{megeng.}}{AST_i}}{n}$	$Q_{AST}^{opt.} = \frac{\sum_{i=1}^n \frac{AST_i^{opt.}}{AST_i}}{n}$
30,62%	30,62%	38,06%	37,24%

3.3-2 táblázat: a tevékenységek (legkorábbi/tényleges) kezdési időinek változása (súlyozatlan értékek)

Az adatokat tehát a következőképpen értelmezhetjük: egy átlagos tevékenység legkorábbi kezdési ideje 30,62%-ára csökkent (a projekt kezdetéhez viszonyítva) mind a megengedett, mind pedig az optimális megoldás esetén. Ha a tevékenységek időtartamát is figyelembe vesszük, akkor ez az érték 31,72%. Látható, hogy az optimális megoldás esetében a tényleges kezdési idő a megengedett megoldáshoz képest tovább csökkent.

További fontos érték az összes felhasznált tartalékidő számítása. Ennek számítását a következőképpen tehetjük:

$$TUST^{megeng.} = \sum_{i=1}^n AST_i^{megeng.} - EST_i^{megeng.} = 3879,7 \text{ nap},$$

$$TUST^{opt.} = \sum_{i=1}^n AST_i^{opt.} - EST_i^{opt.} = 3512,1 \text{ nap}.$$

Az összes felhasznált tartalékidő (Total Used Slack Time) értéke mind a megengedett, mind pedig az optimális megoldásban az összes tevékenységre a tevékenység tényleges kezdése mínusz a legkorábbi kezdése. A megengedett megoldásban, ahol az erőforráskorlátot (max. 400 fő) nem léptük túl, a tevékenységek összes felhasznált tartalékideje munkaóraban 31176 óra. Az optimaló algoritmusunk ezt az értéket 9,47%-kal csökkentette (azaz 25940 munkaórára) úgy, hogy az erőforráskorlátot továbbra is figyelembe vette. Ez az érték azt jelenti, hogy az alternatív úton lévő tevékenységek kezdési idői további 9,47%-kal csökkenthetők a megengedett megoldásban szereplő tevékenységek kezdéseikhez képest.

A megoldási idő 631 ms, ebből tisztán az algoritmus lefutási ideje (hálózati kommunikációs időt nem számítva) **142 ms**, tehát jóval kevesebb, mint egy másodperc. A teljes megoldási időhöz még hozzá kell adni a megengedett megoldás keresésének idejét is. Erre azonban már olyan heurisztikus algoritmusok is léteznek, amelyeknek a lefutási ideje

kevesebb, mint 1ms 1000 tevékenység esetén is (lásd 1.6.2 fejezet). Ezért ez a számítási idő gyakorlatilag elhanyagolható.

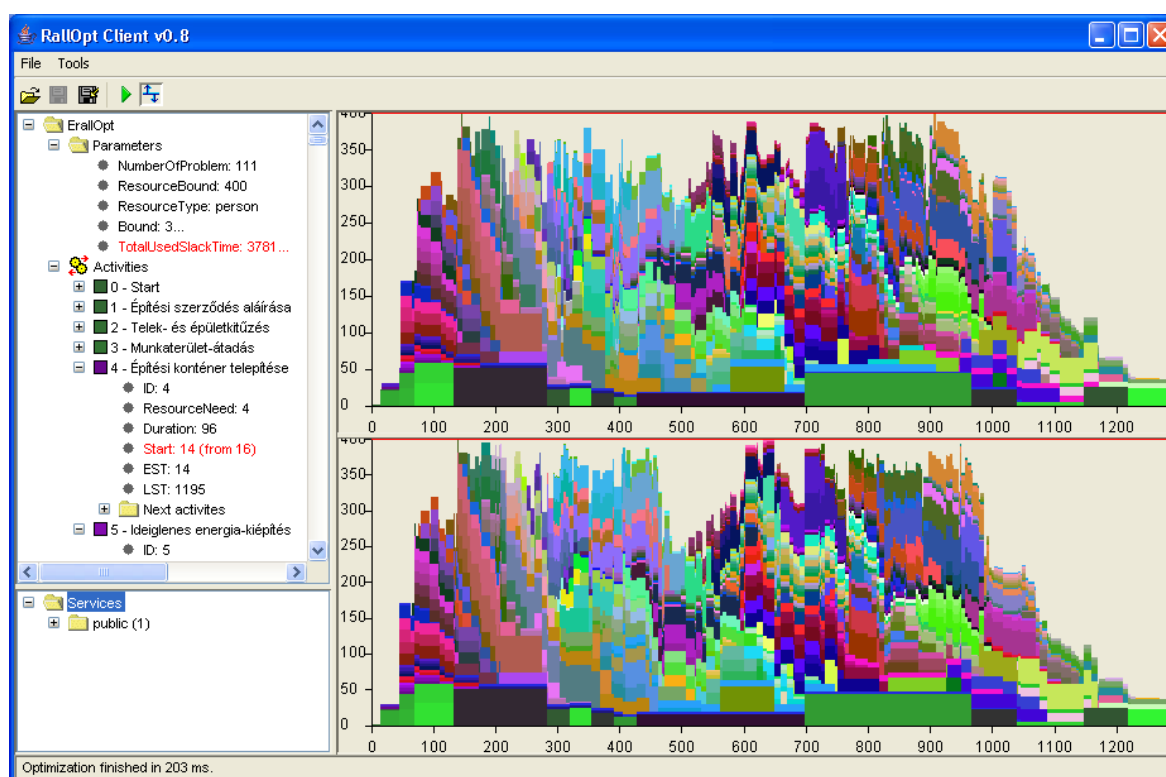
Amennyiben a vállalat kihasználja a projektmenedzsment-szoftverek nyújtotta lehetőségeket, jelentősen csökkentheti a projekt átfutási idejét pusztán azáltal, hogy a tevékenységek megfelelő logikai sorrendjét helyesen modellezi. Ebben az esetben az átfutási időt tekintve mintegy $100,00\%-56,25\%=43,75\%$ -os a csökkenés. Napokban mérve $252,8-143,2=109,6$ nappal előbb lehetett befejezni a projektet. Ha a szervezet az erőforrás, illetve költségigényeket is figyelembe veszi, akkor olyan ütemtervet határozhat meg, mely a rendelkezésre álló erőforrás-, illetve költségkorlátokat is figyelembe veszi. A módszer segítségével elsősorban fixköltséget (pl. bérleti díjak) lehet csökkenteni azáltal, hogy a projektet mintegy 109 nappal előbb be lehet fejezni. Ezenkívül a megvalósítási idő nagysága a pályázat odaítélésénél fontos szempont lehet. Ha egy cég ugyanannyi, vagy kevesebb összköltséggel, kevesebb idő alatt meg tudja valósítani a projektet, az a többi céggel szemben komoly versenyelőnyt jelenthet. Az erőforrás-szükségletek, erőforrás-korlátok nem álltak rendelkezésünkre. Költségigényekből megbecsülve az erőforrás-szükségletet az erőforrás-korlátot mintegy 2/3-ára lehetett csökkenteni úgy, hogy még megengedett (és ebből optimális) erőforrás-allokációt lehetett találni.

A programot további rövidítés után költségoptimalis erőforrásallokáció (**ERALL-OPT/KLTG-módszer** (lásd 5. fejezet)) segítségével több mint két héttel további 14,1 nappal 143,2 napról 129,1 napra lehetett redukálni. Ez további **9,85%-os** csökkenést jelent, ami viszont **145 200 Ft** további pótlólagos költséggel jár. A projekt teljes **1 066 500 000 Ft**-os költségvetéséhez képest (ebből közvetlen pl. berendezések használata, anyagfelhasználás stb. 401 979 924 Ft, a fennmaradó közvetett költségek, pl. bérleti díjak alvállalkozói teljesítések stb. 664 520 076 Ft-ot tesznek ki) ez a pótlólagos összeg jelentéktelennek tűnik.

A módszer lépései:

Lsz.	Tevékenység neve	Egységnyi költségnövekedési tényező	Csökkentés	Változó költség- igény növekedése	Átfutási idő
0.					1432
1.	Geberit Pluvia földbe kerülő része 2.	200	4	800	1428
2.	Telek- és épületkitűzés; Munkaterület-átadás	400	1	400	1427
3.	Monolit talpgerendák, alsó falak	600	17	10200	1410
4.	Geberit Pluvia földbe kerülő része 1.	600	4	2400	1406
5.	sprinkler földalatti vezeték I.	600	4	2400	1402
6.	szellőzésszerelés az eladótérben 1.	600	30	18000	1372
7.	nagy szellőző gép elhelyezése, bekötése	600	12	7200	1360
8.	Visszatöltések	1200	4	4800	1356
9.	szellőzésszerelés a szoc. és az ek. részben 1.	1200	30	36000	1326
10.	Építési szerződés aláírása	1400	1	1400	1325
11.	szellőzésszerelés Mall 1.	1600	8	12800	1317
12.	sprinkler csőszerelés a szoc. és az ek. részben II.	1800	8	14400	1309
13.	radiátorok felrakása a szoc. és az ek. részben	1800	8	14400	1301
14.	Humuszleszedés 1.; Beton a tömbalapba 1.	2000	3	6000	1298
15.	Humuszleszedés 2.; Beton a tömbalapba 2.	2000	7	14000	1291
Σ				145200	

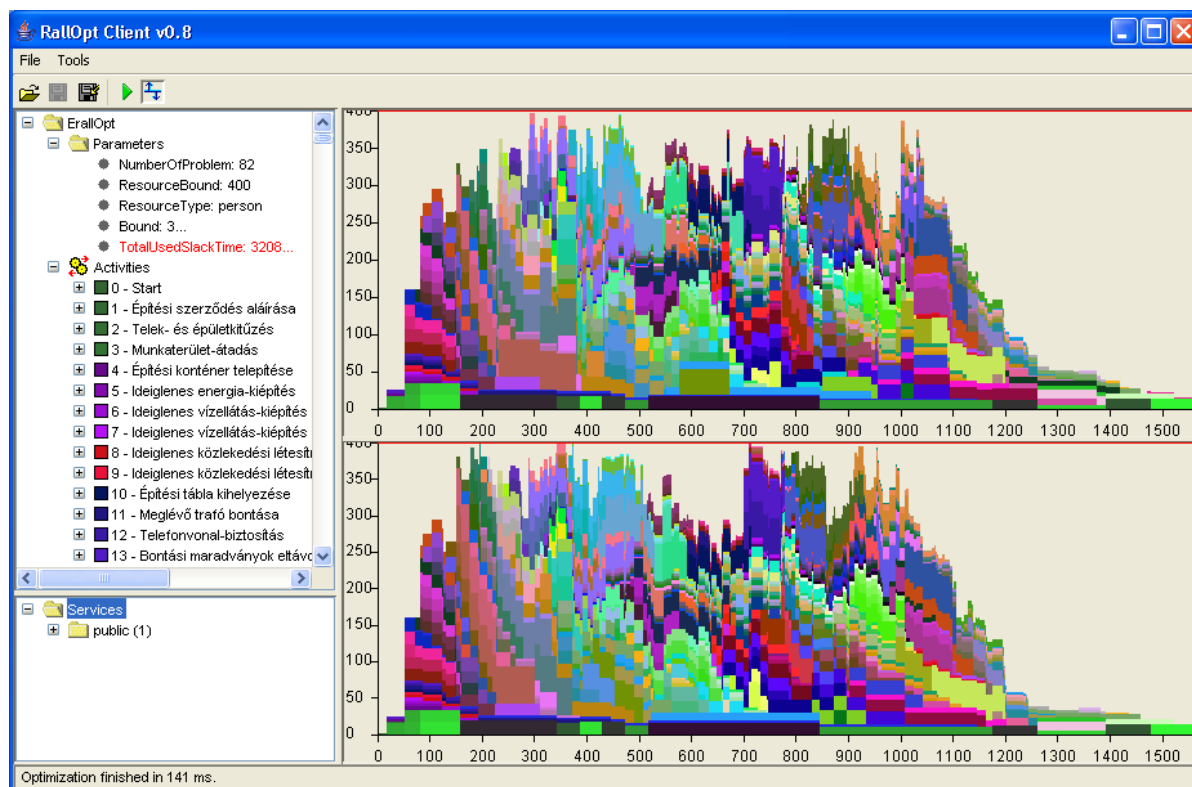
3.3-3 táblázat: a költségcsökkentés lépései



3.3-1 ábra: a projekt megengedett és optimális megoldása költségcsökkentés után

Mivel a közvetett költségek a program előbbi befejezése miatt előre láthatólag 30 000 Ft költségcsökkenéssel járnak naponta, így a minimális átfutási idő egyben minimális összköltségigénnyel is jár. A teljes megtakarítás = $(252,8 - 129,1) \times 30\,000 \text{ Ft} = 3\,565\,800 \text{ Ft}$. Így az összes költség **1 062 934 200 Ft**-ra csökken. Tehát a költségeket mintegy 3,6 MFt-tal az átfutási időt pedig majdnem 50%-kal lehetett csökkenteni a szervezési, idő-költség és erőforrás-optimáló módszerek segítségével.

Ha a projekt időtartamának bizonytalanságát is figyelembe akarjuk venni, akkor sztochasztikus optimáló módszert kell alkalmaznunk. A vállalat eddigi tapasztalatai alapján arra a következtetésre jutott, hogy a tervezett időtartamokat általában sikerül megfelelően betartani. Előfordul, hogy a tervezett időtartamnál 10%-kal hamarabb, vagy 30%-kal később fejezik be a tevékenységet, de ennek valószínűsége igen alacsony. Ez a megfigyelés indokolta, hogy a 3 paraméteres **PERT-módszerrel** sztochasztikus időtartamokkal is megvizsgáljam a projekt átfutási idejét. Ezért valamennyi tevékenység esetén $m_{(i,j)}$ -nek a tervezett időtartamokat választottam. Az optimista becslés ($a_{(i,j)}$) az időtartamok 90%-a, (ahol a technológia megengedte, és nem írtak elő kötött időtartamot). A pesszimista becslés ($b_{(i,j)}$) pedig az időtartamok 130%-a lett. Biztonsági szintnek magas 90%-os biztonsági szintet választottam. Az átfutási idők meghatározásra Fatemi Ghomi – Teimouri [131] módszerét alkalmaztam. E módszer segítségével a projekt átfutási ideje: 155,6 nap lett 90%-os biztonsági szint mellett. Ebben az esetben is található megengedett és optimális erőforrásallokáció:



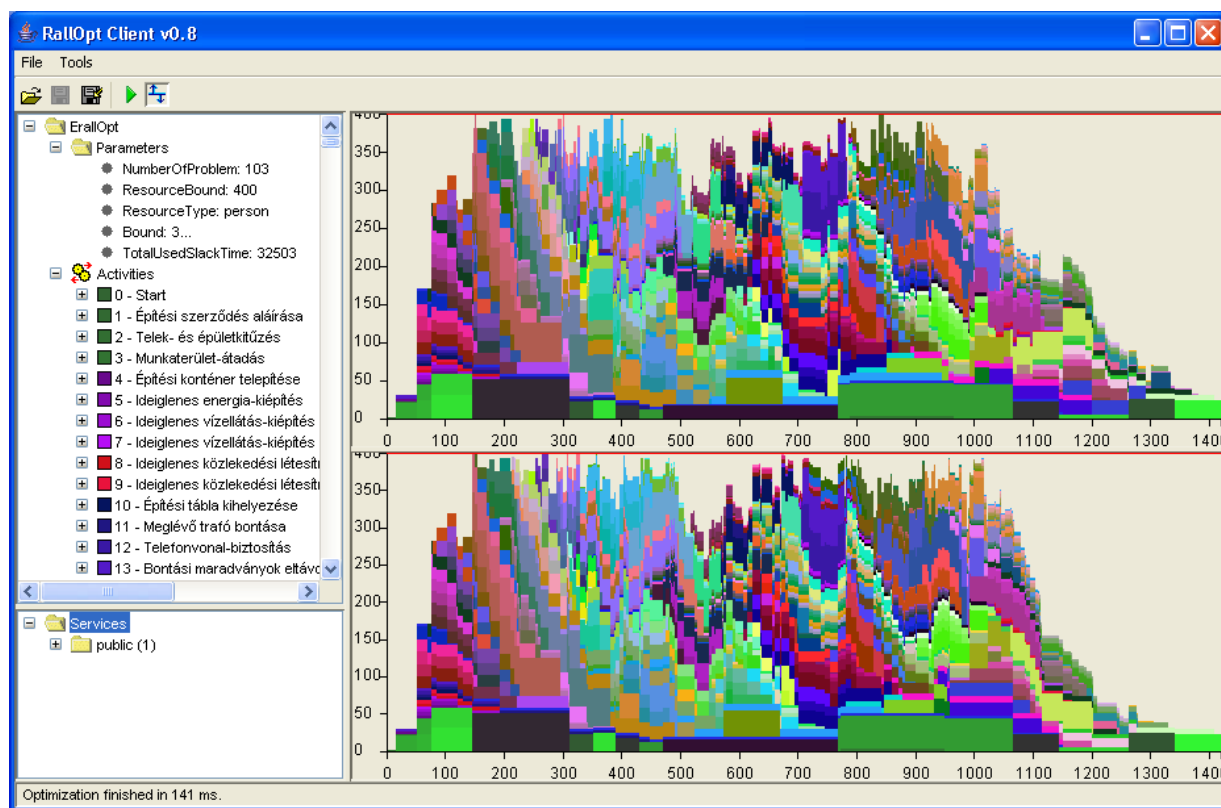
3.3-2 ábra: a projekt sztochasztikus időtervezés esetén

Természetesen sztochasztikus időtervezés mellett is lehet költségtervezést végezni, ehhez a 9. fejezetben leírt sztochasztikus módszeremet alkalmaztam. A költség-idő optimalizációs részfeladatot pedig Chrétienne – Sourd [35] módszerének segítségével végeztem, mely egy

PERT-hálókra alkalmazható költségminimalizáló eljárás (konvex függvényeket feltételezve).

Az átfutási idő 90%-os biztonsági szint mellett **142,1 nap**.

A megengedett és az optimális megoldás:



3.3-3 ábra: a projekt sztochasztikus idő- és költség-tervezés esetén

A gyorsítás során a változó költségek (várható) növekedése **138 200 Ft**. Az összes költségsökkenés: $(252,8 - 142,1) \times 30\,000 \text{ Ft} - 138\,200 \text{ Ft} = \underline{\underline{3\,182\,800 \text{ Ft}}}$.

A sztochasztikus tervezés során a tevékenységek időtartamainak bizonytalansága figyelembevételével azt is meg lehet határozni, hogy egy projekt előreláthatólag pl. 90%-os valószínűség mellett hány nap alatt hajtódik végre. Így a projektmenedzser már nemcsak a megvalósítási idővel, várható költségigénnyel, illetve erőforrás-szükséglettel, de a megvalósítás bizonytalanságával (kockázatával) is előre számolhat.

A módszer gyakorlati alkalmazhatóságához, illetve idő-, költség- és erőforrás-csökkentéshez szükség van arra, hogy a vállalat megbecsülje a tevékenységek időtartamát, illetve erőforrás- és költségigényét. Továbbá szükség van arra, hogy elkészítse a tevékenységek logikai hálóját. Ezután az erőforrásokra/költségekre/átfutási időre tűzzön ki egy korlátot, valamint határozzon meg egy célt (pl. lehető legkorábbi kezdés, egyenletes erőforrás-felhasználás stb.), melyre optimalni kívánja a feladatot. A bemutatott szoftverek, illetve módszerek alkalmazásával az optimális megoldás megtalálható.

Azon kívül, hogy a tevékenységek esetén pontosan nyilván kell tartani a tevékenységek időtartamát, erőforrás-, illetve költségigényét, számítástechnikai beruházást is igényel a vállalat számára.

A módszert egy számítógépen is lehet futtatni, de főleg nagyobb (legalább 300 tevékenység esetén) a hálózati elosztott optimalizálás jelentősen gyorsíthatja a feldolgozást. Egy szerver-számítógép esetén a szükséges hardverberuházás kb. 150 eFt - 250 eFt.

Természetesen a vállalatban már kiépített hálózatot, terminálokat is lehet használni elosztott optimalizálásra, mely optimalizálás a háttérben futtatva nem zavarja a vállalat mindennapi munkáját. Új hálózat kiépítése esetén hálózati csatlók, illetve kábelezéssel együtt ez kb. 10 eFt-ot jelent a cégnek gépenként.

A hardverberuházások mellett szoftverberuházásra is szükség van. Különböző projektmenedzsment-szoftverek különböző áron vásárolhatók meg. Néhány ingyenes egyszerű projektmenedzsment-szoftveren kívül a kereskedelemben kapható alkalmazások az 50 eFt-tól a több milliós szoftvercsomagokig terjedhetnek. Ezért az összegért más-más szolgáltatást nyújtanak. A beruházási költségek ellenére egy több milliós költségvetésű, több mint száz tevékenységet tartalmazó projekt esetén mindenképpen megéri meggondolni, hogy az idő-, költség- és erőforrás-tervezés adta lehetőségeket számítógépes projektmenedzsment-szoftverekkel minél jobban kihasználjuk.

4. Tézisek

- T1. Bizonyítottam, hogy egy erőforrás-korlátos erőforrás-allokációs probléma tetszőleges megengedett megoldásából véges lépésben el lehet érni egy adott célfüggvény(ek)re (lehető legkorábbi kezdés, lehető legkésőbbi kezdés) nézve az adott erőforráskorlátokat figyelembe vevő legjobb megoldást (amennyiben ezek az erőforrások megújuló erőforrások), melyet alkalmazni lehet akkor, ha:**
- az erőforráskorlát időben állandó/nem állandó (szakaszonként konstans),
 - az idő-, költség- és erőforrásigény együttes optimalizálása a cél,
 - többfajta erőforrás egyidejű, illetve párhuzamosan működő projektek közötti erőforrás-megosztás kezelése a cél, vagy
 - a fenti eseteket egyidejűleg kívánjuk kezelni.
- T2. Bizonyítottam, hogy egy megváltozott erőforrás-korlátos erőforrás-allokációs probléma tetszőleges megengedett megoldásából véges lépésben el lehet érni adott célfüggvény(ek)re nézve az adott korlátokat figyelembe vevő legjobb megoldást (amennyiben ezek az erőforrások megújuló erőforrások). Az eljárást alkalmazni lehet, ha változik:**
- az erőforráskorlát,
 - a tevékenységek időtartama vagy
 - az erőforrásigénye, illetve
 - ezen változások közül egyszerre több is bekövetkezik.
- T3. Igazoltam, hogy minden olyan esetben, ahol meghatározható vagy megbecsülhető egy tevékenység időtartamának, erőforrás-szükségleteinek várható értéke illetve szórása, valamint létezik az erőforrás-allokációs probléma egy adott biztonsági szintre vonatkozó, adott korlátokat figyelembe vevő megengedett megoldása, akkor véges lépésben található - egy adott, a menedzsment által meghatározott, célfüggvényre nézve - az adott korlátokat figyelembe vevő, adott biztonsági szintre nézve optimális megoldás.**
- T4. Igazoltam, hogy a kidolgozott módszerrel egy megváltozott erőforrás-allokációs problémából véges lépésben található adott biztonsági szintre nézve, adott**

korlátokat figyelembe vevő, adott célfüggvényre nézve optimális megoldás. A módszert alkalmazni lehet, ha változik:

- az erőforráskorlát,
- a tevékenységek időtartama vagy
- az erőforrásigénye, illetve
- ezen változások közül egyszerre több (akár mindegyik) is bekövetkezik.

T5. Igazoltam, hogy minden olyan esetben, amikor meghatározható a tevékenységek időtartamai és változó költségei között egy determinisztikus vagy sztochasztikus függvénykapcsolat, akkor megengedett megoldás létezése esetén meghatározható (feltéve, hogy a becsült valószínűségi változók (erősen) stationer ergodikus valószínűségi változók):

- az összköltségminimális optimális erőforrás-allokáció,
- minimális átfutási idővel rendelkező optimális erőforrás-allokáció.

5. Függelék

5.1 Felhasznált irodalom:

- [1] Abbasi, G.Y., Mukattash, A.M. (2001). Crashing PERT networks using mathematical programming. *International Journal of Project Management*, Volume 19, Issue 3, April 2001, pp. 181-188.
- [2] Adlakha, V.G., Kulkarni, V.G. (1989). A classified bibliography of research on stochastic PERT networks: 1966-1987. *INFOR*, Volume 27, pp. 272-296.
- [3] Afonja, B., (1972). The moments of the maximum of correlated normal and t-variates. *Journal of Royal Statistical Society*, Volume 34, pp. 251-262.
- [4] Aggteleki, B., Bajna, M., (1994). *Projekttervezés, projektmenedzsment*, KözDok Rt. Budapest
- [5] Ahn, T., Erengüc, S.S. (1995). Resource constrained project scheduling problem with multiple crashable modes. Technical Report, College of Business Administration, University of Florida, Gainesville, USA
- [6] Ahuja, R.K., Magnanti, T.L., Orlin, J.B. (1998). *Network Flows. Theory, Algorithms and Applications*. Prentice-Hall, Upper Saddle River, New Jersey 07458, pp. 24-46.
- [7] Akçay, Y. (December, 2002). Three Essays on Resource Allocation Problems: Inventory Management in Assemble-To-Order Systems and Online Assignment of Flexible Resources. A Thesis in Business Administration. Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy. The Pennsylvania State University, The Graduate School, Department of Management Science and Information Systems.
- [8] Akkan, C., Drexl, A., Kimms, A. (2000). Network decomposition for the discrete time/cost trade-off problem-Part 1: Models and bounding methods. Extended

-
- Abstracts of the Seventh International Workshop on Project Management and Scheduling, April 17–19, Osnabrück, pp. 29–31.
- [9] Akkan, C., Drexl, A., Kimms, A. (2000). Network decomposition for the discrete time/cost trade-off problem-Part 2: Network decomposition and computational results. Extended Abstracts of the Seventh International Workshop on Project Management and Scheduling, April 17–19, Osnabrück, pp. 32–34.
- [10] Alcaraz, J., Maroto, C. (1998). A genetic algorithm for the resource-constrained project scheduling problem. Proceedings of the Sixth International Workshop on Project Management and Scheduling. Bogaziçi University, 7-9 July, pp. 7-10.
- [11] Ali, S., Kim, J-K., Siegel, H. J., Maciejewski, A.A., Yu, Y., Gundala, Sh. B., Gertphol, S., Prasanna, V. (2002). Greedy Heuristics for Resource Allocation in Dynamic Distributed Real-Time Heterogeneous Computing Systems. PDPTA '02 International Conference
- [12] Alvarez-Valdes, R., Tamarit, J.M. (1989). Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis. Slowinski and J. Weglarz (eds.), *Advances in Project Scheduling, Studies in Production and Engineering Economics*, Volume 9, Elsevier, Amsterdam, pp. 113-134.
- [13] Andrásfai, B. (1997). *Gráfelmélet*. Polygon, Szeged, pp. 1-131.
- [14] Ang, A.H-S., Abdelnour, J., Chaker, A.A. (1975). Analysis of activity networks under uncertainty. *Journal of the Engineering Mechanics Division, Proceedings of American Society of Civil Engineers*, Volume 101, EM4, pp. 373-387.
- [15] Arisawa, S., Elmaghraby, S.E. (1972). Optimal time-cost trade-offs in GERT networks. *Management Science*, Volume 18, Issue 11, pp. 589–599.
- [16] Arkin, E., Silverberg, E. (1987). Scheduling Jobs with Fixed Start and End Times. *Discrete Applied Mathematics*, Volume 18, pp. 1-8.
- [17] Audsley, N.C., Burns, A., Richardson, M.F., Wellings, A.J. (1991). Hard Real-Time Scheduling: The Deadline-Monotonic Approach. 8th RTOSS 1991, május
-

-
- [18] Babu, A.J.G., Suresh, N. (1996). Project management with time, cost and quality considerations. *European Journal of Operational Research*, Volume 88, pp. 320-327.
- [19] Bartusch, M., Möhring, R.H., Radermacher, F.J. (1988). Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, Volume 16, pp. 201–240.
- [20] Bazaraa, M.S., Shetty, C.M. (1979). *Nonlinear programming theory and algorithms*. Wiley, New York
- [21] Beard, C.C. (1999). Dynamic agent based prioritized resource allocation for stressed networks. Ph.D. dissertation, University of Kansas, Lawrence
- [22] Beard, C.C., Frost, V.S. (2001). Prioritized Resource Allocation for Stressed Networks. *IEEE/ACM Transactions on Networking*, Volume 9, Numero 5, pp. 618-633.
- [23] Beasley, J.D., Springer, S.G. (1977). The percentage points of the normal distribution. *Applied Statistics*, Volume 26, pp. 118-120.
- [24] Bell, C.E., Han, J. (1991). A new heuristic solution method in resource-constrained project scheduling. *Naval Research Logistics*, Volume 38, pp. 315-331.
- [25] Bencsik, A. (1998) *Szervezősmódszertan, szervezési technikák*. Veszprémi Egyetemi Kiadó, Veszprém
- [26] Bertsekas, D.P. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific: Belmont, MA.
- [27] Bertsekas, D.P., Castanon, D.A. (1999). Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics*, Volume 5, Issue 1, pp. 89-108.
- [28] Bianco, L., Speranza, M.G. (1990). Resource management in project scheduling. *Proceedings of the Second International Workshop on Project Management and Scheduling*, June 20–22, Compiègne.

-
- [29] Billstein, N., Radermacher, F.J. (1977). Time-cost optimization. *Methods of Operations Research*, Volume 27, pp. 274-294.
- [30] Bitran, G.R., Hax, A.C. (1981). Disaggregation and Resource Allocation Using Convex Knapsack Problems with Bounded Variables. *Management Science*, Volume 27, Numero 4, pp. 431-441.
- [31] Blazewicz, J., Lenstra, J.K., Rinnooy Kan, A.H.G. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, Volume 5, Issue 1, pp. 11-24.
- [32] Boctor, F.F. (1990). Some efficient multi-heuristic procedures for resource-constrained project scheduling. *European Journal of Operational Research*, Volume 49, pp. 3-13.
- [33] Boctor, F.F. (1996). A new and efficient heuristic for scheduling problems with resource restrictions and multiple execution modes. *European Journal of Operational Research*, Volume 90, pp. 349-361.
- [34] Bouleimen, K., Lecoque, H. (1998). A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple mode version. Technical Report, Service de Robotique et Automatisation, University of Liège
- [35] Bowman, R.A. (1994). Stochastic gradient-based time-cost tradeoffs in PERT networks using simulation. *Annals of Operations Research*, Volume 53, pp. 533-551.
- [36] Bowman, R.A. (1995). Efficient estimation of arc criticalities in stochastic activity networks. *Management Science*, Volume 41, pp. 58-67.
- [37] Brinkmann, K., Neumann, K. (1996). Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: The resource-leveling and minimum project duration problems. *Journal of Decision Systems*, Volume 5, pp. 129-155.
- [38] Brucker, P. (1998). *Scheduling Algorithms*. 2nd edition. Springer, Berlin, Germany
-

-
- [39] Brucker, P., Drexl, A., Möhring, R.H., Neumann, K., Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models and methods. *European Journal of Operational Research*, Volume 112, Issue 1, pp. 3–41.
- [40] Brucker, P., Jurisch, B., Sievers, B. (1994). A Branch and Bound algorithm for the job-shop problem. *Discrete Applied Mathematics*, Volume 49, pp. 107-127.
- [41] Brucker, P., Knust, S. (2000). A linear programming and constraint propagation-based lower bound for the RCPSP. *European Journal of Operational Research*, Volume 127, pp. 355–362.
- [42] Brucker, P., Knust, S. (2001). Resource-constrained project scheduling and timetabling. Burke, E., Erben, W. (Eds.), *The Practice and Theory of Automated Timetabling III*, Springer Lecture Notes in Computer Science, vol. 2079
- [43] Brucker, P., Knust, S. (2003). Lower bounds for resource-constrained project scheduling problems. *European Journal of Operational Research*, Volume 149, pp. 302–313.
- [44] Brucker, P., Knust, S., Schoo, A., Thiele, O. (1998). A Branch and Bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, Volume 107, pp. 272–288.
- [45] Buckley, J.J. (1989). Fuzzy PERT. *Applications of Fuzzy Set Methodologies in Industrial Engineering*, pp. 103-114.
- [46] Burt, J.M., Garman, M. (1971). Monte Carlo techniques for stochastic PERT network analysis. *INFOR*, Volume 9, pp. 248-262.
- [47] Burt, J.M., Garman, M.B. (1971). Conditional Monte Carlo: a simulation technique for stochastic network analysis. *Management Science*, Volume 18, Issue 3, pp. 207-217.
- [48] Burt, J.M., Gaver, D.P., Perias, M. (1971). Simple stochastic networks: some problems and procedures. *Naval Research Logistics Quarterly*, Volume 17, pp. 439-460.
-

-
- [49] Chae, K.C. (1990). A geometric interpretation of the PERT assumptions on the activity time. *International Journal of Mathematics Education for Science and Technology*, Volume 21, pp. 283-288.
- [50] Chanas, S., Kamburowski, J. (1981). The use of fuzzy variables in PERT. *Fuzzy Sets and Systems*, Volume 5, pp. 11-19.
- [51] Chanas, S., Zieliński, P. (2002). The computational complexity of the critical problems in a network with interval activity times. *European Journal of Operational Research*, Volume 136, pp. 541-550.
- [52] Chen, B., Munson, Ch.L. (2001). Resource Allocation with Lumpy Demand: To Speed or Not to Speed? *Proceedings of the Twelfth Annual Conference of the Production and Operations Management Society, POM-2001*, March 30-April 2, 2001, Orlando, FL.
- [53] Chen, J., Bugacov, A., Szekely, P., Frank, M., Cai, M., Kim, D., Neches, R. (2003). Distributed Resource Allocation: Knowing When To Quit. *AAMAS2003 Melbourne, Australia*.
- [54] Chen, M.-S., Yu, R.S., Wu, K.-L. (1992). Scheduling and Processor Allocation for Parallel Execution of Multi-Join Queries. *Proceedings of the 8th International Conference on Data Engineering*, Phoenix, Arizona
- [55] Chesley, G. (1975). Elicitation of subjective probabilities: A review. *The Accounting Review*, Volume 60, Issue 2, pp. 325-337.
- [56] Cho, J.G., Yum, B.J. (1997). An uncertainty importance measure of activities in PERT networks. *International Journal of Production Researches*, Volume 35, Issue 10, pp. 2737–2757.
- [57] Cho, J.H., Kim, Y.D. (1997). A simulated annealing algorithm for resource constrained project scheduling problems. *Journal of Operational Research Society*, Volume 48, pp. 736–744.
- [58] Choueiry, B.Y., Faltings, B. (1994). Interactive Resource Allocation by Problem Decomposition and Temporal Abstractions. *Current Trends in AI Planning*, (C. Bäckström and E. Sandewall eds.), IOS Press, Amsterdam, pp. 87-104
-

-
- [59] Choueiry, B.Y., Faltings, B. (1994). A Decomposition Heuristic for Resource Allocation. ECAI 94. 11th European Conference on Artificial Intelligence, Published in 1994 by John Wiley & Sons, Ltd.
- [60] Chrétienne, Ph., Sourd, F. (2003). PERT scheduling with convex cost functions. Theoretical Computer Science, Volume 292, pp. 145–164.
- [61] Christofides, N., Alvarez-Valdés, R., Tamarit, J.M. (1987). Project scheduling with resource constraints: A Branch and Bound approach. European Journal of Operational Research, Volume 29, pp. 262-273.
- [62] Christophides, N., (1975). Graph theory: An Algorithmic Approach. Academic Press, New York.
- [63] Clark, C.E. (1962). The PERT model for the distribution of an activity time. Operations Research, Volume 10, pp. 405-406.
- [64] Clingen, C.T. (1964). A modification of Fulkerson's PERT algorithm. Operations Research, Volume 12, Issue 4, pp. 629–632.
- [65] Cohen, T.H., Leiserson, Ch.E., Rivest, R.L. (1997). Algoritmusok. Műszaki könyvkiadó, Budapest, pp. 392-543.
- [66] Cormen, T., Leiserson, C., Rivest, R. (1990). Introduction to Algorithms. The MIT Press, 1990. ISBN 0-262-03141-8 (MIT Press), ISBN 0-07-013143-0 (McGraw-Hill)
- [67] Cox, M.A.A. (1991). The implementation of functions to evaluate percentage points of the normal and Student's t distributions on a spreadsheet. Statistician, Volume 40, pp. 87-94.
- [68] Cox, M.A.A. (1995). Simple normal approximation to the completion time distribution for a PERT network. International Journal of Project Management, Volume 13, Numero 4, pp. 265-270.
- [69] Crespo-Abril, F., Maroto, C., Montesinos, A. (1998). An exact parallel Branch and Bound algorithm for the resource-constrained project scheduling problem.
-

-
- Proceedings of the Sixth International Workshop on Project Management and Scheduling. Bogaziçi University, 7-9July, pp. 27-30.
- [70] Crowston, W. (1970). Network reduction and solution. *Operations Research Quarterly*, Volume 21, pp. 435–450.
- [71] Crowston, W., Thompson, G.L. (1967). Decision CPM: A method for simultaneous planning, scheduling and control of projects. *Operations Research*, Volume 15, pp. 407–426.
- [72] Cruz, S., García, J., Herrerías, R. (1999). Stochastic models alternative to the classical PERT for the treatment of the risk: mesokurtic and of constant variance. *Central European Journal of Operation Research*, Volume 7, Issue 3, pp. 159–175.
- [73] Davis, E.W. (1975). Project: network summary measures constrained-resource scheduling. *AHE Transactions*, Volume 7, pp. 132-142.
- [74] Davis, E.W., Heidorn, G.E. (1971). An algorithm for optimal project scheduling under multiple resource constraints. *Management Science*, Volume 17, Issue 12, pp. 803-816.
- [75] Davis, E.W., Patterson, J.H. (1975). A comparison of heuristic and optimum solutions in resource-constrained project scheduling. *Management Science*, Volume 21, pp. 944-955.
- [76] Dawson, C.W., Dawson, R.J. (1993). Simulation of stochastic and generalised activity networks. *Proceedings of the International ASME Conference on Systems*, Volume 3, pp. 41-55.
- [77] Dawson, C.W., Dawson, R.J. (1994). A clarification of mode representation in generalised activity networks for practical project management. *International Journal of Project Management*, Volume 12, pp. 81 88.
- [78] Dawson, C.W., Dawson, R.J. (1995). Generalised activity-on-the node networks for managing uncertainties in projects. *International Journal of Project Management*, Volume 13, pp. 353-362.
-

- [79] De Boer, R. (1998). Resource-constrained multi-project management - a hierarchical decision support system. PhD Dissertation, Institute for Business Engineering and Technology Application, University of Twente, Enschede.
- [80] De Bruin, A., Kindervater, G.A.P., Trienekens, H.W.J.M. (1989). Parallel Branch and Bound and Anomalies. No 95 in Discussion Paper from Erasmus University Rotterdam, Faculty of Economics.
- [81] De Reyck, B. (1995). On the use of the restrictiveness as a measure of complexity for resource-constrained project scheduling. Onderzoeks-rapport Nr. 9535, Department of Applied Economics, Katholieke Universiteit Leuven.
- [82] De Reyck, B. (1998). Scheduling projects with generalized precedence relations: Exact and heuristic procedures. PhD Dissertation, Katholieke Universiteit Leuven, Belgium.
- [83] De Reyck, B., Demeulemeester, E., Herroelen, W. (1997). Local search methods for the discrete time/resource trade-off problem in project networks. Technical Report, Department of Applied Economics, Katholieke Universiteit Leuven
- [84] De Reyck, B., Herroelen, W. (1996). On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research*, Volume 91, pp. 347–366.
- [85] De Reyck, B., Herroelen, W. (1998). A Branch and Bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, Volume 111, pp. 152–174.
- [86] De Reyck, B., Herroelen, W. (1998). An optimal procedure for the resource-constrained project scheduling problem with discounted cash flows and generalized precedence relations. *Computers and Operations Research*, Volume 25, pp. 1–17.
- [87] De Reyck, B., Herroelen, W. (1999). The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, Volume 119, pp. 538–556.

- [88] De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E. (1992). Complexity of the discrete time-cost tradeoff problem for project networks. Tech. Report, Dept. MIS and Dec. Sci., University of Dayton, Dayton, OH 45469-2130.
- [89] De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E. (1995). The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research*, Volume 81, pp. 225-238.
- [90] De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E. (1997). Complexity of the discrete time-cost trade-off problem for project networks. *Operations Research*, Volume 45, pp. 302-306.
- [91] De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E. (1993). The discrete time/cost trade-off problem revisited. Working Paper 93-04, Dept. MIS and Dec. Sci., University of Dayton, Dayton, OH 45469-2130.
- [92] Dean, T., Kaelbling, L.P., Dirman, J., Nicholson, A. (1995). Planning Under Time Constraints in Stochastic Domains. *Artificial Intelligence*, Volume 76, Number 1-2, pp. 35-74.
- [93] Deckro, R., Winkofsky, E., Hebert, J., Gagnon, R. (1991). A decomposition approach to multi-project scheduling. *European Journal of Operational Research*, Volume 51, Issue 1, pp. 110–118.
- [94] Demeulemeester, E.L., De Reyck, B., Foubert, B., Herroelen, W., Vanhoucke, M. (1998). New computational results for the discrete time/cost trade-off problem in project networks. *Journal of the Operational Research Society*, Volume 49, pp. 1153–1163.
- [95] Demeulemeester, E.L., Dodin, B., Herroelen, W.S. (1993). A random activity network generator. *Operations Research*, Volume 41, Issue 5, pp. 972-980.
- [96] Demeulemeester, E.L., Elmaghraby, S.E., Herroelen, W. (1996). Optimal procedures for the discrete time/cost trade-off problem in project networks. *European Journal of Operational Research*, Volume 88, pp. 50–68.

- [97] Demeulemeester, E.L., Herroelen, W. (1992). A branch and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, Volume 38, Issue 12, pp. 1803-1818.
- [98] Demeulemeester, E.L., (1995). Minimizing resource availability costs in time-limited project networks. *Management Science*, Volume 41, pp. 1590-1598.
- [99] Demeulemeester, E.L., Herroelen, W. (1997). New benchmark results for the resource-constrained project scheduling problem. *Management Science*, Volume 43, pp. 1485-1492.
- [100] Demeulemeester, E.L., Herroelen, W.S., Elmaghraby, S.E. (1996). Optimal procedures for the discrete time/cost trade-off problem in project networks. *European Journal of Operational Research*, Volume 88, pp. 50-68.
- [101] Denardo, E.V., (1982) *Dynamic programming. Models and Applications*. Prentice Hall, Englewood Cliffs, N.J.
- [102] Dennis L. (1998). *Modern Gazdasági Ismeretek - Projekt menedzsment*, Panem Könyvkiadó, Budapest, pp. 89-114.
- [103] Devroye, L.P. (1979). Inequalities for the completion times of stochastic PERT networks. *Mathematics of Operations Research* 4, pp. 441-447.
- [104] Dodin, B.M. (1984). Determining the K most critical paths in PERT network. *Operations Research*, Volume 32, Issue 4, pp. 859-877.
- [105] Dodin, B.M. (1985). Approximating the distribution function in stochastic network. *Computers and Operations Research*, Volume 12, Issue 3, pp. 251-264.
- [106] Dodin, B.M. (1985). Bounding the project completion time distribution in PERT networks. *Journal of Operations Research*, Volume 33, Issue 4, pp. 826-881.
- [107] Dodin, B.M., Elmaghraby, S.E. (1985). Approximating the criticality indices of the activities in PERT networks. *Management Science*, Volume 31, Issue 2, pp. 207-223.

- [108] Dolgov, D., Durfee, E.H. (2002). Satisficing strategies for resource-limited policy search in dynamic environments. Proceedings of the first international joint conference on Autonomous agents and multiagent systems. Bologna, Italy.
- [109] Donaldson, W. (1965). Estimation of the mean and variance of a PERT activity time. Operations Research, Volume 13, pp. 382-385.
- [110] Dorndorf, U., Pesch, E., Phan-Huy, T. (2000). A time-oriented Branch and Bound algorithm for project scheduling with generalised precedence constraints. Management Science, Volume 46, pp. 1365–1384.
- [111] Dorndorf, U., Phan-Huy, T., Pesch, E. (1998). A survey of interval capacity consistency tests for time- and resource-constrained scheduling. Węglarz, J. (Ed.), Handbook on Recent Advances in Project Scheduling, Kluwer Academic Publishers, Dordrecht
- [112] Downey, P.J. (1990). Distribution-free bounds on the expectation of the maximum with scheduling applications. Operations Research Letters, Volume 9, pp. 189-201.
- [113] Drexl, A., Grünewald, J. (1993). Nonpreemptive multi-mode resource-constrained project scheduling. IIE Transactions, Volume 25, Issue 5, pp. 74-81.
- [114] Edwards, Keith, W. (1999). Core Jini. Prentice Hall PTR, Upper Saddle River, NJ07458.
- [115] Elmaghraby, S.E. (1967). On the expected duration of PERT type network. Management Science, Volume 13, Issue 5, pp. 299–306.
- [116] Elmaghraby, S.E. (1977). Activity Networks: Project Planning and Control by Network Models. Wiley, New York
- [117] Elmaghraby, S.E. (1989). The estimation of some network parameters in the PERT model of activity networks: Review and critique. Słowiński, R., Węglarz, J. (Eds.), Advances in Project Scheduling, Chapter 1, Part III, Elsevier, Amsterdam, pp. 371-432.

- [118] Elmaghraby, S.E. (1992). Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research*, Volume 88, pp. 50-86.
- [119] Elmaghraby, S.E. (1995). Activity nets: a guided tour through some recent developments. *European Journal of Operational Research*, Volume 82, pp. 383-408.
- [120] Elmaghraby, S.E., Herroelen, W.S. (1980). On the measurement of complexity in activity networks. *European Journal of Operational Research*, Volume 5, pp. 223-234.
- [121] Elmaghraby, S.E., Herroelen, W.S. (1990). The scheduling of activities to maximize the net present value of projects. *European Journal of Operational Research*, Volume 49, pp. 35-49.
- [122] Elmaghraby, S.E., Kamburowski, J. (1992). The analysis of activity networks under generalised precedence relations (GPRs). *Management Science*, Volume 38, pp. 1245-1263.
- [123] Elmaghraby, S.E., Salem, A. (1980). Optimal project compression under convex cost functions I: Quadratic cost with continuous derivative. *Operations Research Technical Report 158*, North Carolina State University at Raleigh.
- [124] Elmaghraby, S.E., Salem, A. (1980). Optimal project compression under convex cost functions II. *Operations Research Technical Report 157*, North Carolina State University at Raleigh.
- [125] Elmaghraby, S.E., Salem, A. (1981). Optimal linear approximation in project compression. *Operations Research Technical Report 171*, North Carolina State University at Raleigh.
- [126] Elmaghraby, S.E. (1978). *Activity Networks. Project Planning and Control by Network Models*. Wiley – Interscience, New York
- [127] Erenguç, S.S., Tufekçi, S., Zappe, C.J. (1993). Solving time/cost trade-off problems with discounted cash flows using generalised Benders decomposition. *Naval Research Logistics*, Volume 40, pp. 25-50.

-
- [128] Falk, J.E., Horowitz, J.L. (1972). Critical path problems with concave cost-time curves. *Management Science*, Volume 19, pp. 446–455.
- [129] Farnum, N.R., Stanton, L.W. (1987). Some results concerning the estimation of beta distribution parameters in PERT. *Journal of the Operational Research Society*, Volume 38, pp. 287-290.
- [130] Fatemi Ghomi, S.M.T., Rabbani, M. (2003). A new structural mechanism for reducibility of stochastic PERT networks. *European Journal of Operational Research*, Volume 145, Issue 2, 1 March 2003, pp. 394-402.
- [131] Fatemi Ghomi, S.M.T., Teimouri, E. (2002). Path critical index and activity critical index in PERT networks. *European Journal of Operational Research*, Volume 141, Issue 1, 16 August 2002, pp. 147-152.
- [132] Ferguson, D.F., Nikolaou, C., Sairamesh, J., Yemini, Y. (1996). *Economic Models for Allocating Resources in Computer Systems*. Clearwater, S. (ed.), *Market Based Control of Distributed Systems*. World Scientific Press.
- [133] Fest, A., Möhring, R.H., Stork, F., Uetz, M. (1998). Resource-constrained project scheduling with time windows: A branching scheme based on dynamic release dates. Technical Report 596, Technische Universität Berlin
- [134] Fisher, D.L., Saisi, D., Goldstein, W.M. (1985). Stochastic PERT networks: op diagrams, critical paths and the project completion time. *Computers and Operations Research*, Volume 12, Issue 5, pp. 471–482.
- [135] Fleszar, K., Hindi, K.S. (2004). Solving the resource-constrained project scheduling problem by a variable neighbourhood search. *European Journal of Operational Research*, Volume 155, pp. 402–413.
- [136] Foldes, S., Soumis, F. (1993). PERT and crashing revisited: Mathematical generalizations. *European Journal of Operational Research*, Volume 64, pp. 286-294.
- [137] Franck, B., Neumann, K. (1997). Resource-constrained project scheduling with time windows - structural questions and priority-rule methods. Report WIOR-492, Universität Karlsruhe
-

- [138] Frank, M., Bugacov, A., Chen, J., Dakin, G., Szekely, P., Neches, B. (2001). The marbles manifesto: A definition and comparison of cooperative negotiation schemes for distributed resource allocation. Proceedings of the 2001 AAAI (American Association for Artificial Intelligence) Fall Symposium on Negotiation Methods for Autonomous Cooperative Systems, November 2001, pp. 36-45.
- [139] Fulkerson, D.R. (1961). A network flow computation for project cost curves. Management Science, Volume 7, pp. 167-178.
- [140] Fulkerson, D.R. (1962). Expected critical path length in PERT network. Operations Research, Volume 10, Issue 6, 808–817.
- [141] Fulp, E.W., Ott, M., Reininger, D., Reeves, D.S. (1998). Paying for QoS: An Optimal Distributed Algorithm for Pricing Network Resources. Proceedings of the IEEE Sixth International Workshop on Quality of Service, pp. 75–84.
- [142] Gallagher, C. (1987). A note on PERT assumptions. Management Science, Volume 33, Issue 10, p. 1360.
- [143] Garavelli, A.C., Pontrandolfo, P. (1995). A heuristic method for the estimation of the project duration in a stochastic network scheduling. Recherche operationnelle/Operations Research, Volume 29, Issue 3, pp. 285-298.
- [144] Garey, U.R., Johnson, D.C. (1979). Computer and Intractability: a guide to the theory of NP-completeness. Freeman, San Francisco, USA
- [145] Garofalakis, M.N., Ioannidis, Y.E. (1996). Multi-dimensional Resource Scheduling for Parallel Queries. SIGMOD '96, 6/96 Montreal, Canada, pp. 365-376.
- [146] Gaul, W. (1982). On stochastic analysis of project-networks. Dempster, M.A.H., Lenstra, J.K., Rinnoy Kan, A.H.G. (Eds.), Deterministic and Stochastic Scheduling, Reidel, Dordrecht, pp. 297-309.
- [147] Gazdik, I. (1983). Fuzzy-network planning-FNET. IEEE Transactions Reliability, Volume 32, Issue 3, pp. 304-313.

- [148] Geoffrion, A., Marsten, R.E. (1972). Integer programming algorithms: A framework and state-of-the-art survey. *Management Science*, Volume 18, Issue 9, pp. 465-491.
- [149] Ginis, R. (1999). Optimal Distributed Resource Allocation. Thesis. Partial Fulfillment of the Requirements for the Degree of Master of Science, California Institute of Technology, Pasadena, California
- [150] Golenko-Ginzburg, D. (1988). On the distribution of activity time in PERT. *Journal of Operation Research Society*, Volume 39, Issue 8, pp. 767-771.
- [151] Golenko-Ginzburg, D. (1989). PERT assumptions revisited. *Omega*, Volume 17, pp. 393-396.
- [152] Golenko-Ginzburg, D., Gonik, A. (1998). A heuristic for network project scheduling with random activity duration's depending on the resource allocation. *International Journal of Production Economics*, Volume 55, Issue 22, pp. 149-162.
- [153] Golenko-Ginzburg, D., Gonik, A., Kesler, S. (1996). Hierarchical decision-making model for planning and controlling stochastic projects *International Journal of Production Economy*, Volume 46, Issue 47, pp. 39-54.
- [154] Gong, D., Hugsted, R. (1993). Time-uncertainty analysis in project networks with a new merge event time-estimation technique. *International Journal of Project Management*, Volume 11, Issue 3, pp. 165-74.
- [155] Gonik, A. (1995). Planning and controlling multi-level stochastic projects. Thesis submitted for the PhD. Degree, Ben-Gurion University of the Negev, Beer-Sheva, Israel
- [156] Goyal, S.K. (1975). A note on the paper: A simple CPM time/cost trade-off algorithm. *Management Science*, Volume 21, pp. 718-722.
- [157] Görög, M. (1999). Általános projektmenedzsment. Aula Kiadó, Budapest, pp. 11-18, 89-108.
- [158] Görög, M. (2001). Bevezetés a projektmenedzsmentbe. Aula Kiadó, Budapest, pp. 47-80.

- [159] Graham, R.J., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. (1979). Optimization and approximation in deterministic sequencing and scheduling theory: A survey. *Annals of Discrete Mathematics*, Volume 5, pp. 287-326.
- [160] Gratch, J., Chien, S. (1996). Adaptive Problem-Solving for Large-Scale Scheduling Problems: A Case Study. *Journal of Artificial Intelligence Research*, Volume 4, pp. 365-396.
- [161] Grubbs, F.E. (1962). Attempts to validate certain PERT statistics or a 'Picking on PERT'. *Operations Research*, Volume 10, pp. 912-915.
- [162] Hagstrom, J.N. (1990). Computing the probability distribution of project duration in a PERT network. *Networks*, Volume 20, pp. 231-244.
- [163] Harthley, H.O., Wortham, A.W. (1966). A statistical theory for PERT critical path analysis. *Management Science*, Volume 12, Issue 10, pp. 469-481.
- [164] Hartmann, S. (1998). A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*, Volume 45, Issue 7, pp. 733-750.
- [165] Hartmann, S., Drexl, A. (1998). Project scheduling with multiple modes: a comparison of exact algorithms. *Networks*, Volume 32, pp. 283-297.
- [166] Hartmann, S., Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, Volume 127, Issue 2, pp. 394-407.
- [167] Harvey, R.T., Patterson, J.H. (1979). An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis. *Found. Control Eng.*, Volume 4, pp. 107-117.
- [168] Heally, T.L. (1971). Activity subdivision and PERT probability statements. *Operations Research*, Volume 9, pp. 341-348.
- [169] Heilmann, R. (1998). A Branch and Bound procedure for MRCPSP/max. Report WIOR-512, Universität Karlsruhe

- [170] Heilmann, R. (2003). A Branch and Bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, Volume 144, pp. 348–365.
- [171] Heilmann, R., Schwindt, C. (1997). Lower bounds for RCPSP/max. Report WIOR-511, Universität Karlsruhe
- [172] Heller, U. (1981). On the shortest overall duration in stochastic project networks. *Methods of Operations Research*, Volume 42, pp. 85-104.
- [173] Herroelen, W., De Reyck, B., Demeulemeester, E. (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers and Operations Research*, Volume 25, Issue 4, pp. 279–302.
- [174] Herroelen, W., Demeulemeester, E., De Reyck, B. (1998). A classification scheme for project scheduling. Weglarz, J. (Ed.), *Project Scheduling - Recent Models, Algorithms and Applications*, Chapter 1. Kluwer Academic Publishers, Dordrecht, pp. 197–212.
- [175] Herroelen, W., Demeulemeester, E., De Reyck, B. (1999). A classification scheme for project scheduling problems. Weglarz, J. (Ed.), *Project Scheduling: Recent Models, Algorithms and Applications*, Chapter 1. Kluwer Academic Publishers, Dordrecht, pp. 1–26.
- [176] Herroelen, W., Demeulemeester, E., De Reyck, B. (1999). An integrated classification scheme for resource scheduling. Research Report 9905, Department of Applied Economics, K.U. Leuven
- [177] Herroelen, W., Demeulemeester, E., De Reyck, B. (2001). A note on the paper „Resource-constrained project scheduling: Notation, classification, models and methods” by Brucker et al. *European Journal of Operational Research*, Volume 128, pp. 679-688.
- [178] Herroelen, W., Demeulemeester, E.L., Van Dommelen, P. (1996). An optimal recursive search procedure for the deterministic unconstrained max-npv project scheduling problem. Research Report 9603, Department of Applied Economics, Katholieke Universiteit Leuven

- [179] Hillier, F.S., Lieberman, G.J. (1986). Introduction to Operations Research. Holden-Day, San Francisco CA.
- [180] Hindelang, T.J., Muth, J.F. (1979). A dynamic programming algorithm for Decision CPM networks. Operations Research, Volume 27, pp. 225-241.
- [181] Hoogeveen, J.A., Lenstra, J.K., Van de Velde, S.L. (1997). Sequencing and scheduling. Dell'Amico, M., Maffioli, F., Martello (Eds.), S., Annotated Bibliographies in Combinatorial Optimization. Wiley, New York, pp. 181-197.
- [182] Ibaraki, T., Katoh, N. (1988). Resource Allocation Problems: Algorithmic Approaches. MIT Press, 1988. ISBN 0-262-09027-9.
- [183] Icmeli, O., Erengüc, S.S. (1996). A Branch and Bound procedure for the resource constrained project scheduling problem with discounted cash flows. Management Science, Volume 42, pp. 1395-1408.
- [184] Icmeli, O., Erengüc, S.S., Zappe, C.J. (1993). Project scheduling problems: A survey. International Journal of Operations and Production Management, Volume 13, Issue 11, pp. 80–91.
- [185] Igelmund, G., Radermacher, F.J. (1983). Algorithmic approaches to preselective strategies for stochastic scheduling problems. Networks, Volume 13, pp. 29-48.
- [186] Igelmund, G., Radermacher, F.J. (1983). Preselective strategies for the optimization of stochastic project networks under resource constraints. Networks, Volume 13, pp. 1-28.
- [187] Ignizio, J.P. (1982). Linear Programming in Single and Multiple Systems. Prentice-Hall, Englewood Cliffs, NJ
- [188] Johnson, G.A., Schou, C.D. (1990). Expediting projects in PERT with stochastic time estimates. Project Management Journal, Volume 21, Issue 2, pp. 29–33.
- [189] Johnson, T.J.R. (1967). An algorithm for the resource-constrained project scheduling problem. PhD Dissertation, MIT, Boston, USA

- [190] Kaimann, R.A. (1971). Coefficient of network complexity. *Management Science*, Volume 21, Issue 2, pp. 172-177.
- [191] Kamburowski, J. (1985). An upper bound on the expected completion time of PERT network. *European Journal of Operational Research*, Volume 21, Issue 2, pp. 206–212.
- [192] Kamburowski, J. (1985). Normally distributed activity durations in PERT networks. *Journal of Operational Research Society*, Volume 36, pp. 1051-1057.
- [193] Kamburowski, J. (1993). Project crashing in CPM-networks. *Proceedings of the 1993 Annual Meeting of Decision Science Institute*, Washington, D.C., pp. 1099-1101.
- [194] Kamburowski, J. (1995). On the Minimum Cost Project Schedule. *Omega*, International Journal of Management Science, Volume 23, Numero 4, pp. 463-465.
- [195] Kamburowski, J. (1997). New Validations of PERT Times. *International Journal of Management Science*, Volume 25, Numero 3, pp. 323-328.
- [196] Kamburowski, J., Michael, D.J., Stallmann, M.F.M. (1992). Optimal construction of project activity networks. *Proceedings of the 1992 Annual Meeting of the Decision Sciences Institute*, San Francisco, pp. 1424–1426.
- [197] Kapur, K.C. (1973). An algorithm for the project cost/duration analysis problem with quadratic and convex cost functions. *IIE Transactions*, Volume 5, pp. 314–322.
- [198] Karaa, F.A., Nasr, A.Y. (1986). Resource management in construction. *Journal of Construction Engineering and Management*, Volume 112, pp. 346-357.
- [199] Kaufmann, A. (1968). Az operációkutatás módszerei és modelljei. Műszaki könyvkiadó, Budapest, pp. 11-123.
- [200] Kaufmann, A., Desbazeille G. (1972). A kritikus út módszerének matematikai alapjai, Műszaki könyvkiadó, Budapest, pp. 7-203.
- [201] Keefer, D.L., Verdini, W.A. (1993). Better estimation of PERT activity time parameters. *Management Science*, Volume 39, Issue 9, pp. 1086-1091.

- [202] Keith L., James G. (2000) Projektmenedzsment és hálós tervezési technikák. Kossuth Kiadó, Budapest
- [203] Kelley Jr., J.E. (1961). Critical path planning and scheduling: mathematical basis. Operations Research, Volume 9, Issue 3, pp. 296–320.
- [204] Kelley Jr., J.E. (1963). The critical path method: Resource planning and scheduling. Muth, J.F., Thompson (Eds.), G.L., Industrial Scheduling, Prentice Hall, NJ, pp. 347-365.
- [205] Kelley Jr., J.E., Walker, M.R. (1959). Critical Path Planning and Scheduling: An Introduction. Mauchly Associates, Ambler, PA
- [206] Kendall, M., Stuart, A. (1969). The Advanced Theory of Statistics, Hafner, New York.
- [207] Kidd, J.B. (1990). Project Management software – are we being overpersuaded? International Journal of Project Management, Volume 8, pp. 109-115.
- [208] Kidd, J.B. (1991). Do today's projects need powerful network planning tools? International Journal of Production Research, Volume 29, pp. 1969-1978.
- [209] Klein Haneveld, W.K. (1986). Robustness against dependence in PERT: An application of duality and distributions with known marginals. Mathematical Programming Study, Volume 27, pp. 153-182.
- [210] Klein, R., Scholl, A. (1999). Computing lower bounds by destructive improvement: An application to resource-constrained project scheduling. European Journal of Operational Research, Volume 112, pp. 322–346.
- [211] Kochenberger, G.A., McCarl, B.A., Wyman, F.P. (1974). A Heuristic for General Integer Programming, Decision Sciences, Volume 5, pp. 36-44.
- [212] Kolisch, R. (1995). Project Scheduling under Resource Constraints: Efficient Heuristics for Several Problem Classes. Physica, Heidelberg

- [213] Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, Volume 90, pp. 320–333.
- [214] Kolisch, R., Drexl, A. (1996). Adaptive search for solving hard project scheduling problems. *Naval Research Logistics*, Volume 43, pp. 23-40.
- [215] Kolisch, R., Drexl, A. (1997). Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, Volume 29, pp. 987-999.
- [216] Kolisch, R., Hartmann, S. (1999). Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. Węglarz, J. (Ed.), *Project Scheduling. Recent Models, Algorithms and Applications*. Kluwer Academic Publishers, Boston, pp. 147–178.
- [217] Kolisch, R., Schwindt, C., Sprecher, A. (1998). Benchmark instances for project scheduling problems. Węglarz, J. (Ed.), *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht, pp. 197–212.
- [218] Kolisch, R., Sprecher, A. (1996). PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, Volume 96, pp. 205-216.
- [219] Kolisch, R., Sprecher, A., Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, Volume 41, Issue 10, pp. 1693–1703.
- [220] Kosztyán Zs.T., Bencsik, A., Hógyor, A. (2002). *Működő projektek optimális erőforrás elosztása*. Műszaki Vezető, Verlag Dashöfer 2002.
- [221] Kosztyán Zs.T., Bencsik, A., Hógyor, A. (2002). *Működő projektek optimális erőforrás elosztása*. Gyakorlati Logisztikai Füzetek, Verlag Dashöfer 2002.
- [222] Kosztyán Zs.T., Bencsik, A., Mátrai, R. (2004). Handling the Deterministic and Stochastic Time, Resources and Costs in Project Management and Logistics. *microCAD 2004*, Miskolc, 2004. March. 17-19., pp. 59-64.

- [223] Kosztyán, Zs.T., Bencsik, A., (2001). Erőforrás-optimalizálás, egy új módszer alkalmazása a projektmenedzsmentben (A cikk az MLBKT évente meghirdetett logisztikai diákpályázatának nyertes dolgozata alapján készült). Logisztikai Híradó 2001/6 5-7 oldal.
- [224] Kosztyán, Zs.T., Bencsik, A., (2003). Bizonytalan átfutási idejű projektek optimális erőforrás elosztása. Műszaki Vezető, Verlag Dashöfer 2003.
- [225] Kosztyán, Zs.T., Bencsik, A., (2003). Bizonytalanság kezelése különböző logisztikai feladatok esetén. Műszaki Vezető, Dashöfer 2003.
- [226] Kosztyán, Zs.T., Bencsik, A., Hgyor, A. (2002). Egy új módszer alkalmazása erőforrás tervezésben. 2002 Logisztikai Évkönyv 99-109 oldal.
- [227] Kosztyán, Zs.T., Bencsik, A., Mátrai, R. (2003). Uncertainty in Optimal Resource Allocation. The 4th International Conference of PhD Students, University of Miskolc, 11-17 August 2003, pp. 115-120.
- [228] Kosztyán, Zs.T., Bencsik, A., Mátrai, R., (2004). Bizonytalanság kezelése kissorozatgyártás termelésirányításában, Műszaki Vezető, Verlag Dashöfer 2004.
- [229] Kosztyán, Zs.T., Bencsik, A., Mátrai, R., (2004). Költségcsökkentés alternatív megoldások segítségével. Műszaki Vezető, Verlag Dashöfer 2004.
- [230] Kosztyán, Zs.T., Póta, Sz., (2004). Optimal Deterministic and Stochastic Resource Allocation in a Distributed System. CSCS (The Fourth Conference of PhD Students in Computer Science) 2004, Szeged, 2004.July. 1-4.
- [231] Kotiah, T.C.T., Wallace, N.D. (1973). Another look at the PERT assumptions. Management Science, Volume 20, pp. 44-49.
- [232] Krämer, A. (1997). Branch and bound methods for scheduling problems with multiprocessor tasks on dedicated processors. Operations Research Spektrum, Volume 19, pp. 219-227.
- [233] Kulkarni, V.G., Adlakha, V.G. (1980). Markov and Markov regenerative PERT network. Journal of Operations Research, Volume 34, Issue 5, pp. 769–781.

- [234] Kurihara, K., Nishiuchi, N. (2002). Efficient Monte Carlo simulation method of GERT-type network for project management. *Computers and Industrial Engineering*, Volume 42, pp. 521-531.
- [235] Kurtulus, I., Davis, E.W. (1982). Multi-project scheduling: categorization of heuristic rules performance. *Management Science*, Volume 28, pp. 161-172.
- [236] Kurtulus, I.S., Narula, S. (1985). Multi-project.scheduling: analysis of project performance. *HE Transactions*, Volume 17, pp. 58-65.
- [237] Lai, T., Sahni, S. (1984), Anomalies in parallel branch and bound algorithms. *Communications of the ACM*, Volume 27, Issue 6, pp. 594-602.
- [238] Laird, P. (1992). Dynamic Optimization. *Proceedings of the Ninth International Conference on Machine Learning*.
- [239] Lamberson, L.R., Hocking, R.R. (1970). Optimum time compression in project scheduling. *Management Science*, Volume B 16, pp. 597–606.
- [240] Lau, A., Lau, H-S., Zhang, Y. (1996). A simple and logic alternative for making PERT time estimates. *IIE Transaction*, Volume 28, Issue 3, pp. 183–192.
- [241] Lau, H-S., Somarajan, C. (1995). A proposal on improved procedures for estimating task-time distributions in PERT. *European Journal of Operational Research*, Volume 85, pp. 39-52.
- [242] Lawler, E.L. (1976). *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York
- [243] Lee, J.K., Kim, Y.D. (1996). Search heuristics for resource constrained project scheduling. *Journal of the Operational Research Society*, Volume 47, pp. 678–689.
- [244] Leon, V.J., Ramamoorthy, B. (1995). Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling. *Operations Research Spektrum*, Volume 17, pp. 173–182.

- [245] Lewin, R.I., Kirkpatrick, Ch.A., D.C.S. (1966). Planning and control PERT/CPM. Mc. Graw-Hill Book Company, New York.
- [246] Li, R.K.Y., Willis, J. (1992). An iterative scheduling technique for resource-constrained project scheduling. *European Journal of Operational Research*, Volume 56, pp. 370–379.
- [247] Lindsey, J.H. (1972). An estimate of expected critical path length in PERT network. *Journal of Operations Research*, Volume 20, Issue 4, pp. 800–812.
- [248] Littlefield Jr., T.K., Randolph, P.H. (1987). An answer to Sasieni's question on PERT times. *Management Science*, Volume 33, pp. 1357-1359.
- [249] Littlefield Jr., T.K., Randolph, P.H. (1991). PERT duration times: Mathematics or MBO. *Interfaces*, Volume 21, p. 9295.
- [250] Lova, A., Maroto, C., Tormos, P. (1998). Resource constrained multiproject scheduling: A multicriteria heuristic algorithm. *Proceedings of the Sixth International Workshop on Project Management and Scheduling*. Bogaziçi University, 7-9 July, pp. 90-93.
- [251] Luenberger, D.G. (1973). *Introduction to Linear and Non-linear Programming*, Addison-Wesley, Massachusetts, USA
- [252] MacCrimmon, K.R., Rayvec, C.A. (1964). An analytical study of the PERT assumptions. *Operations Research*, Volume 12, Issue 1, pp. 16-37.
- [253] Magott, J., Skudlarski, K. (1993). Estimating the mean completion time of PERT networks with exponentially distributed durations of activities. *European Journal of Operational Research*, Volume 71, pp. 70-79.
- [254] Maniezzo, V., Mingozzi, A. (1999). A heuristic for the multimode project scheduling problem based on Bender's decomposition. Węglarz, J. (Ed.), *Project Scheduling: Recent Models, Algorithms and Applications*. Kluwer, Boston, pp. 179–196.

- [255] Maroto, C., Tormos, P., Lova, A. (1998). The evolution of software quality in project scheduling. Weglarz, J. (Ed.), Project Scheduling - Recent Models, Algorithms and Applications, Chapter 11. Kluwer Academic Publishers, Dordrecht
- [256] Martin, N.G., Allen, J.F. (1993). Statistical Probabilities for Planning. Technical Report 474, Department of Computer Science, University of Rochester, Rochester, NY.
- [257] Mátrai, R., Kosztyán, Zs.T., Bencsik, A., (2004). Erőforrás-tervezés egy létesítményprojekt megvalósítása esetén. Műszaki Vezető, Verlag Dashöfer 2004.
- [258] McKeown, G.P., Rayward-Smith, V.J. and Rush, S.A (1994). Advances in Parallel Algorithms, Parellel Branch-and-Bound. IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam, The Netherlands, pp. 112-150.
- [259] Meilijson, I., Nadas, A. (1979). Convex majorization with an application to the length of critical paths. Journal of Applied Probability, Volume 16, pp. 671-677.
- [260] Menipaz, E., Ben-Yair, A. (2002). Harmonization simulation model for managing several stochastic projects. Mathematics and Computers in Simulation, Volume 61, pp. 61–66.
- [261] Menipaz, E., Ben-Yair, A. (2002). Three-parametrical harmonization model in project management by means of simulation. Mathematics and Computers in Simulation, Volume 59, Issue 5, pp. 431–436.
- [262] Mieghem, J.A.V. (1998). Investment strategies for flexible resources. Management Science, Volume 44, Issue 8, pp. 1071-1078.
- [263] Miller, R.W. (1962). How to plan and control with PERT. Harvard Business Review Mar/Apr., pp. 93-104.
- [264] Mingozzi, A., Maniezzo, V., Ricciardelli, S., Bianco, L. (1998). An exact algorithm for the resource-constrained project scheduling based on a new mathematical formulation. Management Science, Volume 44, pp. 714-729.

- [265] Moder, J.J., Phillips, C.R., Davis, E.W. (1983). Project Management with CPM, PERT and Precedence Diagramming, 3rd ed., Van Nostrand Reinhold, New York, USA
- [266] Moder, J.J., Rodgers, E. (1968). Judgement estimates of the moments of PERT type distributions. Management Science, Volume B 15, pp. 76-83.
- [267] Moitra, S.D. (1990). Skewness and the beta distribution. Journal of the Operational Research Society, Volume 41, pp. 953-961.
- [268] Mon, D.L., Cheng, C.H. (1994). Evaluating weapon system by analytical hierarchy process based on fuzzy scales. Fuzzy Sets and Systems, Volume 62, pp. 1-10.
- [269] Mon, D.L., Cheng, C.H. (1994). Fuzzy system reliability analysis for components with different membership functions. Fuzzy Sets and Systems, Volume 62, pp. 145-157.
- [270] Mon, D-L., Cheng, Ch-H., Lu, H-Ch. (1995). Application of fuzzy distributions on project management. Fuzzy Sets and Systems, Volume 73, pp. 227-234.
- [271] Mongalo, M.A., Lee, J.A. (1990). A comparative study of methods for probabilistic project scheduling. Computers Ind. Engineering, Volume 19, pp. 505-509.
- [272] Moodie, C.L., Mandeville, D.E. (1965). Project resource balancing by assembly line balancing techniques. Journal of Industrial Engineering, Volume 17, Issue 7, pp. 377-383.
- [273] Möhring, R.H., Stork, F., Uetz, M. (1998). Resource constrained project scheduling with time windows: A branching scheme based on dynamic release dates. Arbeitspapier 596, Fachbereich Mathematik, Technical University of Berlin.
- [274] Möhring, R.H. (1984). Minimizing costs of resource requirements in project networks subject to a fixed completion time. Operations Research, Volume 32, pp. 89-120.
- [275] Möhring, R.H., Müller, R. (1998). A combinatorial approach to bound the distribution function of the makespan in stochastic project networks. Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany

- [276] Möhring, R.H., Radermacher, F.J. (1984). Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals of Discrete Mathematics*, Volume 19, pp. 257-356.
- [277] Möhring, R.H., Radermacher, F.J. (1989). The order-theoretic approach to scheduling: The stochastic case. Sowiński, R., Węglarz (Eds.), J., *Advances in Project Scheduling*, Elsevier, Amsterdam, pp. 497-531.
- [278] Möhring, R.H., Radermacher, F.J., Weiss, G. (1984). Stochastic scheduling problems I - General strategies. *Operations Research*, Volume A 28, pp. 193-260.
- [279] Möhring, R.H., Radermacher, F.J., Weiss, G. (1985). Stochastic scheduling problems II - Set strategies. *Operations Research*, Volume A 29, pp. 65-104.
- [280] Möhring, R.H., Schulz, A.S., Stork, F., Uetz, M. (1999). Resource-constrained project scheduling: Computing lower bounds by solving minimum cut problems. Nešetřil, J. (Ed.), *Proceedings of the 7th Annual European Symposium on Algorithms*, Lecture Notes in Computer Science 1643, Springer, pp. 139–150.
- [281] Möhring, R.H., Stork, F. (1998). Linear preselective policies for stochastic project scheduling. Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany
- [282] Möhring, R.H., Stork, F., Uetz, M. (1998). Resource constrained project scheduling with time windows: A branching scheme based on dynamic release dates. Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany
- [283] Mummolo, G. (1994). PERT-path network technique: a new approach to project planning. *International Journal of Project Management* 12, 89-99.
- [284] Mummolo, G. (1997). Measuring uncertainty and criticality in network planning by PERT-path technique. *International Journal of Project Management*, Volume 15, Numero 6, pp. 377-387.
- [285] Mummolo, G., Iavagnih, R. (June 1994). Managing planners' uncertainties by the PERT-path technique: an Italian case study INTERNET'94 12th World congress, Oslo

-
- [286] Munro, M.C., Huff, S.L., Marcolin, B.L., Compeau, D.R. (1997). Understanding and measuring user competence. *Information and Management*, Volume 33, pp. 45-57.
- [287] Naphade, K.S., Wu, S.D., Storer, R.H. (1997). Problem space search algorithms for resource-constrained project scheduling. *Annals of Operations Research*, Volume 70, pp. 307–326.
- [288] Németh, L. (1965). Hálótechnikai termelésprogramozó és erőforrásallokáló eljárás (ERALL-1). É. M. SZÁMGÉP
- [289] Németh, L. (1966). Organizational and control methods of building processes (ERALL-2). É. M. SZÁMGÉP
- [290] Neumann K., Zhan J. (1998). Heuristics for the minimum project duration problem with minimal and maximal time lags under fixed resource constraints. *Journal of Intelligent Manufacturing*, Volume 6, pp. 145-154.
- [291] Neumann, K., Schwindt, C. (1995). Projects with minimal and maximal time lags: construction of activity-on-node networks and applications. Report WIOR-447, Universität Karlsruhe
- [292] Neumann, K., Schwindt, C. (1997). Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production. *Operations Research Spektrum*, Volume 19, pp. 205-217.
- [293] Neumann, K., Steinhardt, U. (1979). GERT Networks and the Time-Oriented Evaluation of Projects. *Lecture Notes in Economics and Mathematical Systems*, Volume 172, Springer, Berlin.
- [294] Neumann, K., Zhan, J. (1995). Heuristics for the minimum project-duration problem with minimal and maximal time-lags under fixed resource constraints. *Journal of Intelligent Manufacturing*, Volume 6, pp. 145-154.
- [295] Neumann, K., Zimmermann, J. (1998). Methods for resource-constrained project scheduling with regular and nonregular objective functions and schedule-dependent time windows. Węglarz, J. (Ed.), *Handbook on Recent Advances in Project Scheduling*, Kluwer Academic Publishers, Dordrecht
-

- [296] Nonobe, K., Ibaraki, T. (2002). Formulation and tabu search algorithm for the resource constrained project scheduling problem. Ribeiro, C.C., Hansen, P. (Eds.), *Essays and Surveys in Metaheuristics*, Kluwer, pp. 557–588.
- [297] Nuijten, W.P.M. (1994). *Time and Resource Constrained Scheduling: A Constraint Satisfaction Approach*. PhD Dissertation, Eindhoven University of Technology.
- [298] Nübel, H. (1998). A branch-and-bound procedure for the resource investment problem with generalized precedence constraints. Report WIOR-516, Universität Karlsruhe
- [299] Oguz, O., Bala, H. (1994). A comparative study of computational procedures for the resource constrained project scheduling problem. *European Journal of Operational Research*, Volume 72, pp. 406-416.
- [300] Ord, J.K. (1991). A simple approximation to the completion time distribution for a PERT network. *Journal of Operational Research Society*, Volume 42, pp. 1011-1017
- [301] Özdamar, L. (1996). A genetic algorithm approach to a general category project scheduling problem. Research Report, Marmara University, Istanbul
- [302] Özdamar, L., Ulusoy, G. (1994). A local constraint based analysis approach to project scheduling under general resource constraints. *European Journal of Operational Research*, Volume 79, pp. 287-298.
- [303] Özdamar, L., Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. *IIE Transactions*, Volume 27, pp. 574-586.
- [304] Özdamar, L., Ulusoy, G. (1996). An iterative local constraint based analysis for solving the resource constrained project scheduling problem. *Journal of Operations Management*, Volume 14, pp. 193–208.
- [305] Padman, R., Smith-Daniels, D.E., Smith-Daniels, V.L. (1997). Heuristic scheduling of resource-constrained projects with cash flows: An optimization approach. *Naval Research Logistics Quarterly*, Volume 44, pp. 365-381.
- [306] Papadimitriou, H., Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover

- [307] Papp, O. (1969). A hálós programozási módszerek gyakorlati alkalmazása. Közgazdasági és Jogi Kiadó
- [308] Papp, O., (1985). Hálótervezés az ipari gyakorlatban. Közgazdasági és Jogi Könyvkiadó, Budapest, 1985
- [309] Papp, O., (2001). Projektmenedzsment – Projektek tervezése, szervezése, irányítása. BMGE Mérnöktovábbképző Intézet, Budapest, 2001 (5. kiadás)
- [310] Papp, O., (2002). Projektmenedzsment a gyakorlatban. LSI, Budapest, 2002
- [311] Parks, W.H., Ramsing, K.D. (1969). The use of the compound poisson in PERT, Management Science, Volume B 15, pp. 397-402.
- [312] Pascoe, T.L. (1966). Allocation of resources C.P.M. Revue Française Recherche Opérationnelle, Volume 38, pp. 31–38.
- [313] Patterson, J.H. (1976). Project scheduling: The effect of problem structure on heuristic performance. Naval Research Logistics Quarterly, Volume 23, pp. 95-124.
- [314] Patterson, J.H. (1984). A comparison of exact approaches for solving the multiple constrained resource project scheduling problem. Management Science, Volume 30, Issue 7, pp. 854-867.
- [315] Patterson, J.H., Harvey, R.T. (1979). An implicit enumeration algorithm for the time/cost trade-off problem in project network analysis. Foundations of Control Engineering, Volume 6, pp. 107–117.
- [316] Patterson, J.H., Huber, W.D. (1974). A horizon-varying zero-one approach to project scheduling, Management Science, Volume 20, Issue 6, pp. 990-998.
- [317] Patterson, J.H., Sowiński, R., Talbot, F.B., Węglarz, J. (1989). An algorithm for a general class of precedence and resource constrained scheduling problems. Sowiński, R., Węglarz, J. (Eds.), Advances in Project Scheduling, Elsevier, Amsterdam, pp. 3-28.

- [318] Patterson, J.H., Sowiński, R., Talbot, F.B., Węglarz, J. (1990). Computational experience with a backtracking algorithm for solving a general class of resource constrained scheduling problems. *European Journal of Operational Research*, Volume 90, pp. 68-79.
- [319] Patterson, J.H. (1984). A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Management Science*, Volume 30, pp. 854-867.
- [320] Phillips Jr., S., Dessouky, M.I. (1977). Solving the project time/cost tradeoff problem using the minimal cut concept. *Management Science*, Volume 24, pp. 393-400.
- [321] Phillips Jr., S., Dessouky, M.I. (1979) The cut search algorithm with arc capacities and lower bounds. *Management Science*, Volume 25, pp. 396-404.
- [322] Pirkul, H. (1987). A heuristic solution procedure for the multiconstraint zero-one knapsack problem, *Naval Research Logistics*, Volume 34, pp. 161-172.
- [323] Pirkul, H., Narasimhan, S. (1986). Efficient algorithms for the multiconstraint general knapsack problem. *IIE Transactions*, pp. 195-203.
- [324] Pleguezuelo, R.H., Pérez, J.G., Rambaudo, S.C. (2003). A note on the reasonableness of PERT hypotheses. *Operations Research Letters*, Volume 31, Issue 1, January 2003, pp. 60-62.
- [325] Pontrandolfo, P. (2000). Project duration in stochastic networks by the PERT-path technique. *International Journal of Project Management*, Volume 18, pp. 215-222.
- [326] Prade, H. (1979). Using fuzzy set theory in a scheduling problem: a case study. *Fuzzy Sets and Systems*, Volume 2, pp. 153-165.
- [327] Pritsker, A.A.B., Happ, W.W. (1966). GERT: graphical evaluation and review technique part I. Fundamentals. *Journal of Industrial Engineering*, Volume 17, pp. 267-274.

- [328] Pritsker, A.A.B., Watters, W.D., P.Wolfe, M. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, Volume 16, pp. 93–108.
- [329] Pritsker, A.A.B., Whitehouse, G.E. (1966). GERT: Graphical evaluation and review technique part II. Probabilistic and industrial engineering applications. *Journal of Industrial Engineering*, Volume 17, Issue 6, pp. 293-301.
- [330] Rajkumar, R., Lee, Ch., Lehoczky, J.P., Siewiorek, D.P. (1998). Some Solutions for QoS-based Resource Allocation. Technical Report, Department of Computer Science, Carnegie Mellon University.
- [331] Rhiel, G.S. (1990). A refinement of the PERT assumptions. *Proceedings of the Decision Sciences Institute National Meeting*, San Diego, pp. 773-775.
- [332] Ringer, L.J. (1969). Numerical operators for statistical PERT critical path analysis. *Management Science*, Volume 16, Issue 2, pp. 136–143.
- [333] Robillard, P., Trahan, M. (1976). Expected completion time in PERT network. *Operations Research*, Volume 24, Issue 1, pp. 177–182.
- [334] Robillard, P., Trahan, M. (1977). The completion time of PERT network, *Journal of Operations Research*, Volume 25, Issue 1, pp. 15–29.
- [335] Robinson, D.R. (1975). A Dynamic Programming solution to cost-time tradeoff for CPM. *Management Science*, Volume 22, Issue 2, pp. 158-166.
- [336] Rónyai, L., Ivanyos, G., Szabó, R. (1998). *Algoritmusok*. Typotex, Budapest, pp. 110-183.
- [337] Roşu, D., Schwan, K., Yalamanchili, S., Jha, R. (December, 1997). On Adaptive Resource Allocation for Complex Real-Time Applications. 18th IEEE Real-Time Systems Symposium
- [338] Russell, R.A. (1986). A comparison of heuristics for scheduling projects with cash flows and resource restrictions. *Management Science*, Volume 32, pp. 1291-1300.

- [339] Sairamesh, J., Ferguson, D.F., Yemini, Y. (1995). An Approach to Pricing, Optimal Allocation and Quality of Service Provisioning in High-speed Packet Networks. Proceedings of the IEEE INFOCOM, pp. 1111-1119.
- [340] Salewski, F., Schirmer, A., Drexl, A. (1997). Project scheduling under resource and mode identity constraints: Model, complexity, methods and application. European Journal of Operational Research, Volume 102, pp. 88-110.
- [341] Sampson, S.E., Weiss, E.N. (1993). Local search techniques for the generalized resource constrained project scheduling problem. Naval Research Logistics, Volume 40, pp. 665-675.
- [342] Sasieni, M.W. (1986). A note on PERT times. Management Science, Volume 32, Issue 12, pp. 1652-1653.
- [343] Schirmer, A., Drexl, A. (1996). Partially renewable resources – a generalization of resource-constrained project scheduling. Paper presented at the IFORS Triennial Meeting, Vancouver, BC, 8-12 July.
- [344] Schrijver, A. (1986). Theory of Linear and Integer Programming. Wiley, New York.
- [345] Schwindt, C. (1996). Generation of resource-constrained project scheduling problem subject to minimal and maximum time lags. Report WIOR-489, University of Karlsruhe.
- [346] Schwindt, C. (1998). Generation of resource-constrained project scheduling problems subject to temporal constraints. Report WIOR-543, University of Karlsruhe
- [347] Scott, C.H., Jefferson, T.R. (1995). Allocation of resources in project management. International Journal of Systems Science, Volume 26, pp. 413-420.
- [348] Sculli, D. (1983). The completion times of PERT network, Journal of Operations Research Society, Volume 34, Issue 2, pp. 155–158.
- [349] Sculli, D. (1989). A historical note on PERT times. Omega, Volume 17, pp. 195-200.

- [350] Shogan, A.W. (1977). Bounding distributions for a stochastic PERT network. *Networks*, Volume 7, pp. 359-381.
- [351] Siemens, N. (1971). A simple CPM time/cost trade-off algorithm. *Management Science*, Volume B 17, pp. 354–363.
- [352] Siemens, N., Gooding, C. (1975). Reducing project duration at minimum cost: A time/cost trade-off algorithm. *OMEGA*, Volume 3, pp. 569–581.
- [353] Sigal, C.E., Pritsker, A.A.B., Solberg, J.J. (1979). The use of cutsets in Monte Carlo analysis of stochastic networks. *Mathematics and Computers in Simulation*, Volume 21, pp. 376–384.
- [354] Simpson, III, W.P., Patterson, J.H. (1992). Modeling the performance of and exact solution procedure for the resource-constrained project scheduling problem. *Decision Sciences Institute Proceedings*, Volume 3, pp. 1430-1433.
- [355] Simpson, III, W.P., Patterson, J.H. (1996). A multiple-tree search procedure for the resource-constrained project scheduling problem. *European Journal of Operational Research*, Volume 89, pp. 525-542.
- [356] Skutella, M. (1998). Approximation algorithms for the discrete time–cost tradeoff problem. *Mathematics of Operations Research*, Volume 23, pp. 909–929.
- [357] Skutella, M. (1998). Approximation and randomization in scheduling. PhD Thesis, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany
- [358] Somarajan, R.C., Girardeau, C., Li, J. (1992). A proposal on improved procedures for estimating task-time distributions in PERT. *Proceedings of the Decision Sciences Institute National Meeting*, San Francisco, pp. 1109-1111.
- [359] Soroush, H.M. (1993). Risk taking in stochastic PERT networks. *European Journal of Operational Research*, Volume 67, pp. 221-241.
- [360] Soroush, H.M. (1994). The most critical path in a PERT network. *Journal of the Operational Research Society*, Volume 45, pp. 287-300.

-
- [361] Sox, C.R., Muckstadt J.A. (1996). Multi-Item, Multi-Period Production Planning with Uncertain Demand. *HE Transactions*, Volume 28, pp. 891-900.
- [362] Sprecher, A. (1994). *Resource-Constrained Project Scheduling - Exact Methods for the Multi-Mode Case*. Springer, Berlin
- [363] Sprecher, A. (1996). Solving the RCPSP efficiently at modest memory requirements. Technical Report 425, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel.
- [364] Sprecher, A., Drexl, A. (1998). Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, Volume 107, pp. 431-450.
- [365] Sprecher, A., Hartmann, S., Drexl, A. (1997). An exact algorithm for project scheduling with multiple modes. *Operations Research Spektrum*, Volume 19, pp. 195–203.
- [366] Sprecher, A., Kolisch, R., Drexl, A. (1995). Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research*, Volume 80, pp. 94–102.
- [367] Stinson, J.P., Davis, E.W., Khumawala, B.M. (1978). Multiple resource-constrained scheduling using branch and bound. *AIIE Transactions*, Volume 10, pp. 252-259.
- [368] Stork, F. (1998). A branch and bound algorithm for minimizing expected makespan in stochastic project networks with resource constraints. Technical Report, Technische Universität Berlin, Fachbereich Mathematik, Berlin, Germany
- [369] Sullivan, R.S., Hayya, J.C. (1980). A comparison of the method of bounding distributions (MBD) and Monte Carlo simulation for analyzing stochastic acyclic networks. *Operations Research*, Volume 28, pp. 614-617.
- [370] Swanson, L.A., Pazer, H.L. (1971). Implications of the underlying assumptions of PERT. *Decision Science*, Volume 2, pp. 461-480.
- [371] Szendrei, Á. (1996). *Diszkrét matematika*. Polygon, Szeged, pp. 63-83, 295-324.
-

- [372] Talbot, F.B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science*, Volume 28, pp. 1197-1210.
- [373] Talbot, F.B., Patterson, J.H. (1978). An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems. *Management Science*, Volume 24, Issue 11, pp. 1163-1174.
- [374] Thesen, A. (1976). Heuristic scheduling of activities under resource and precedence restrictions. *Management Science*, Volume 23, pp. 412-422.
- [375] Tomii, N., Zhou, L.J., Fukumura, N. (1999). An Algorithm for Station Shunting Scheduling Problems Combining Probabilistic Local Search and PERT. Springer-Verlag, Berlin, Heidelberg, pp. 788-797.
- [376] Tóth, I. (1987). *Matematika üzemgazdászoknak - Operációkutatás I.* Tankönyvkiadó, Budapest, pp. 167- 189.
- [377] Ulusoy, G., Özdamar, L. (1994). A constrained-based perspective in resource constrained project scheduling. *International Journal of Production Research*, Volume 32, pp. 693-705.
- [378] Valls, V., Laguna, M., Lino, P., Pérez, A., Quintanilla, S. (1999). Project Scheduling with Stochastic Activity Interruptions. Węglarz, J. (Ed.), *Project Scheduling. Recent Models, Algorithms and Applications*. Kluwer Academic Publishers, Boston, pp. 333–354.
- [379] Valls, V., Quintanilla, S., Ballestín, F. (2003). Resource-constrained project scheduling: A critical activity reordering heuristic. *European Journal of Operational Research*, Volume 149, pp. 282–301.
- [380] Van Hove, J.C., Deckro, R. (1998). Multi-modal project scheduling with generalized precedence constraints. *Proceedings of the PMS Workshop, Istanbul*, pp. 137–140.
- [381] Van Slyke, R.M. (1963). Monte Carlo methods and the PERT problem. *Operations Research*, Volume 11, Issue 5, pp. 839–860.

-
- [382] Vanhoucke, M., Demeulemeester, E., Herroelen, W. (2002). Discrete time/cost trade-offs in project scheduling with time-switch constraints. *Journal of the Operational Research Society*, Volume 53, pp. 1–11.
- [383] Vanhoucke, M., Demeulemeester, E., Herroelen, W. (2003). A random network generator for activity-on-the-node networks. *Journal of Scheduling*, Volume 6, pp. 13–34.
- [384] Vatter, P., Bradley, S., Frey, S., Jackson, B. (1978). *Quantitative Methods in Management*, Irwin, IL.
- [385] Weiss, G. (1986). Stochastic bounds on distributions of optimal value functions with applications to PERT, network flows and reliability. *Operations Research*, Volume 34, pp. 595-605.
- [386] Whitehouse, G.E., Pritsker, A.A.B. (1969). GERT: Part III-- further statistical results; counters, renewal times and correlations. *AIIE Transactions*, Volume 1, Issue 1, pp. 45-50.
- [387] Williams, T.M. (1995). What are PERT estimates? *Journal of the Operational Research Society*, Volume 46, pp. 1498-1504.
- [388] Wu, W-Z., Zhang, W-X., Li, H-Z. (2003). Knowledge acquisition in incomplete fuzzy information systems via rough set approach. *Expert Systems*, Volume 20, Issue 5, pp. 280–286.
- [389] Wu, Y., Li, C. (1994). Minimal cost networks: the cut set parallel difference method. *Omega*, Volume 22, pp. 401-407.
- [390] Yang, H.H., Chen, Y.L. (2000). Finding the critical path in an activity network with time-switch constraints. *European Journal of Operational Research*, Volume 120, pp. 603–613.
- [391] Yang, K.K., Tay, L.C., Sum, C.C. (1995). A comparison of stochastic scheduling rules for maximizing project net present value. *European Journal of Operational Research*, Volume 85, pp. 327-339.
-

- [392] Younis, M.A., Saad, B. (1996). Optimal resource leveling of multi-resource projects. Computers and Industrial Engineering, Volume 31, pp. 1-4.
- [393] Zhan, J. (1994). Heuristics for scheduling resource-constrained projects in MPM networks. European Journal of Operational Research, Volume 76, pp. 192-205.
- [394] Zhao, W-X., Ramamritham, K., Stankovic, J.A. (1987). Preemptive scheduling under time and resource constraints. IEEE Transactions on Computers, Volume C-36, Issue 8, pp. 949-960.
- [395] Zimmermann, J. (1997). Heuristics for resource-leveling problems in project scheduling with minimum and maximum time lags. Report WIOR-491, Universität Karlsruhe
- [396] Zimmermann, J., Engelhardt, H. (1998). Lower bounds and exact algorithms for resource levelling problems. Report WIOR-517, Universität Karlsruhe
- [397] Guide to the Expression of Uncertainty in Measurement, International Organization for Standardization 1993
- [398] Sun Microsystems Inc. Jini connection technology overview. (1999). Technical report, Sun Microsystems Inc.
- [399] Microsoft Project home: <http://office.microsoft.com/hu-hu/assistance/CH790018101038.aspx>
- [400] Primavera Project Planner® (P3®) Features and Specifications: http://www.primavera.com/solutions/ec_p3_features.html
- [401] Computer Associates Super Project: <http://ca.com/>

Megjelenés alatt lévő publikációk:

Zs. Kosztyán, Sz. Póta, A. Bencsik, R. Mátrai: Distributed Deterministic and Stochastic Optimal Resource Allocation, Logistics Networks - models, methods and applications, University of Miskolc, 2004, ISBN 963 661 641 8., Eds. T. Bányai, J. Cselényi.

Zs. Kosztyán, Z. Perjés, A. Bencsik: Cost Reduction By Means Of Alternative Solutions, *Aston Times Office*, Aston University, Birmingham

5.2 A dolgozatban használt jelölések

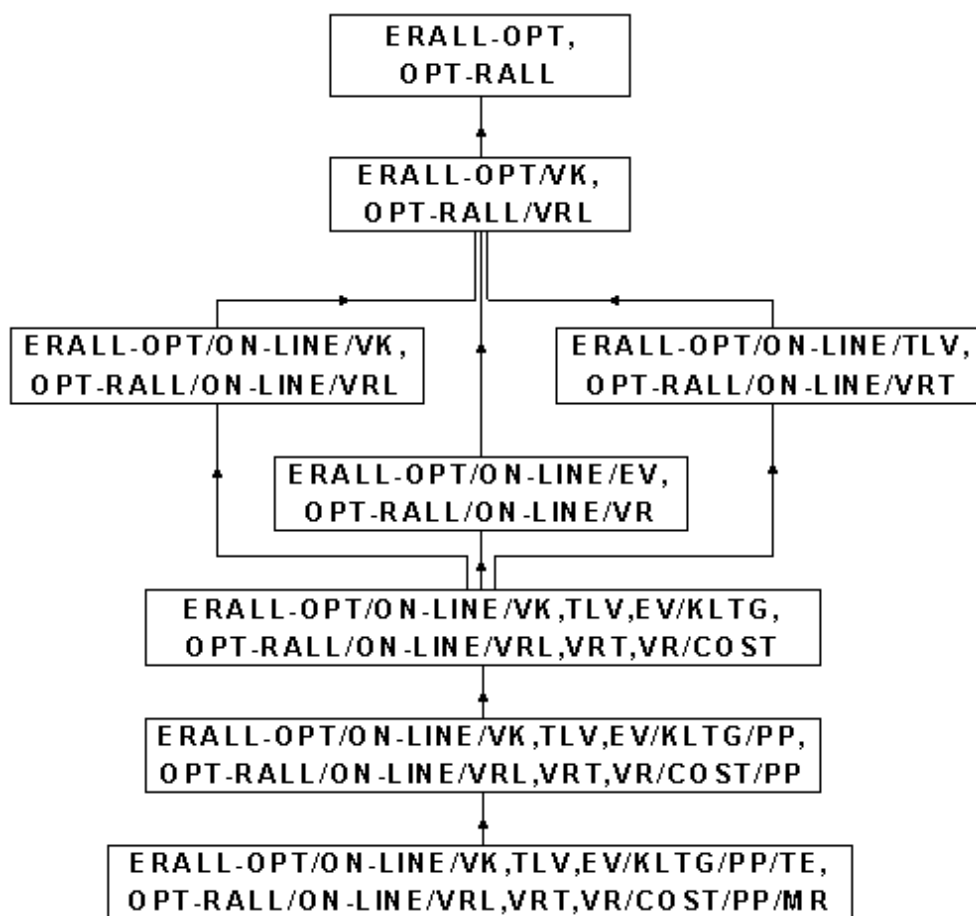
Jelölés	Magyarázat
(i,j)	Tevékenység, ahol i esemény a tevékenység kezdete, j a tevékenység befejezése.
A	Az alaphalmaz, amely valamennyi tevékenységet tartalmazza.
a,b,m	Optimista-, pesszimista-, legvalószínűbb időtartambecslés
$d_{(i,j)}$	(i,j) tevékenység időtartama.
EET_i	i esemény legkorábbi bekövetkezési ideje.
$EFT_{(i,j)}$	(i,j) tevékenység legkorábbi befejezési ideje.
$EST_{(i,j)}$	(i,j) tevékenység legkorábbi kezdési ideje.
LET_i	J esemény legkésőbbi bekövetkezési ideje.
$LFT_{(i,j)}$	(i,j) tevékenység legkésőbbi befejezési ideje.
$LST_{(i,j)}$	(i,j) tevékenység legkésőbbi kezdési ideje.
P	A megengedettség érdekében elmozgatott tevékenységek halmaza.
Q	Az egy lépésben optimalizálandó tevékenységek halmaza.
$r(t)$	Tevékenység erőforrás-szükséglete adott időtartam esetén.
T	Töréspontok halmaza.
$T_{(i,j)}$	(i,j) tevékenységre vonatkozó töréspontok halmaza.
$t_{f(k,l,l,m)}$	(k,l) (rész)tevékenység (tényleges) befejezése és (l,m) (rész)tevékenység (tényleges) kezdése között eltelt idő.
$t_{fmax(k,l,l,m)}$	(k,l) és (l,m) (rész)tevékenység közötti maximális megszakíthatósági idő.
$t_{i(i,j)}$	(i,j) tevékenységhez legközelebb eső töréspont távolsága a tevékenység kezdéséhez viszonyítva.
$t_{i,j}$	Tevékenység várható időtartama.
t_l	Azt az időt mutatja meg, hogy az (i,j) tevékenységet mennyivel lehet visszatolni (egy lépésben), hogy az erőforráskorlátot nem túllépve a rákövetkezési relációkat ne sértsük meg.
TPT	A projekt átfutási ideje.
T_s	Rákövetkezési relációk halmaza.
$t_{s(i,j)}$	Azt az időt mutatja meg, hogy az (i,j) tevékenységet mennyivel lehet visszatolni úgy, hogy a rákövetkezési relációkat ne sértsük meg.
$v(t), t_v$	ON-LINE ütemezés esetén az újraoptimalizálás kezdete.
$Vc(t)$	Tevékenység változó költség-igénye adott időtartam esetén.
$W_{(i,j)}$	Kezdési idő a megengedett megoldásban.
$x_{(i,j)}$	A megengedett megoldásban (i,j) tevékenység felhasznált tartalékideje.
$z_{(i,j)}$	Legkorábbi kezdési idő a maximálisan párhuzamosított megoldásban. Kezdetben: $z_{(i,j)} = EST_{(i,j)}$.
α, β	β -eloszlás paraméterei.
$\sigma_{i,j}$	Az időtartam szórása.

5.2-1 táblázat: a dolgozatban használt, illetve bevezetett jelölések

5.3 Az algoritmusok szerkezete, tulajdonságok öröklődése

Az egyes algoritmusok egy-egy plusz tulajdonsággal rendelkeznek; amelyik algoritmusból öröklődnek, annak az algoritmusnak a tulajdonságait bővíthetik tovább. Az

ábrán látható az algoritmusok egy lehetséges felépítése. A diagram legalján a legbonyolultabb, de egyben a legtöbb helyen használható algoritmus látható.



5.3-1 ábra: egy lehetséges felépítése az algoritmusoknak