

# SZOFTVERFEJLESZTÉSI PROJEKTEK ÜTEMEZÉSÉNEK MODELLEZÉSE

Tézisek

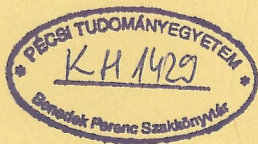
Kruzslicz Ferenc

Konzulens:

Dr. habil. Csébfalvi György

egyetemi docens PTE

Pécsi Tudományegyetem  
Közgazdaságtudományi Kar  
Gazdálkodástani Doktori Iskola  
2003.

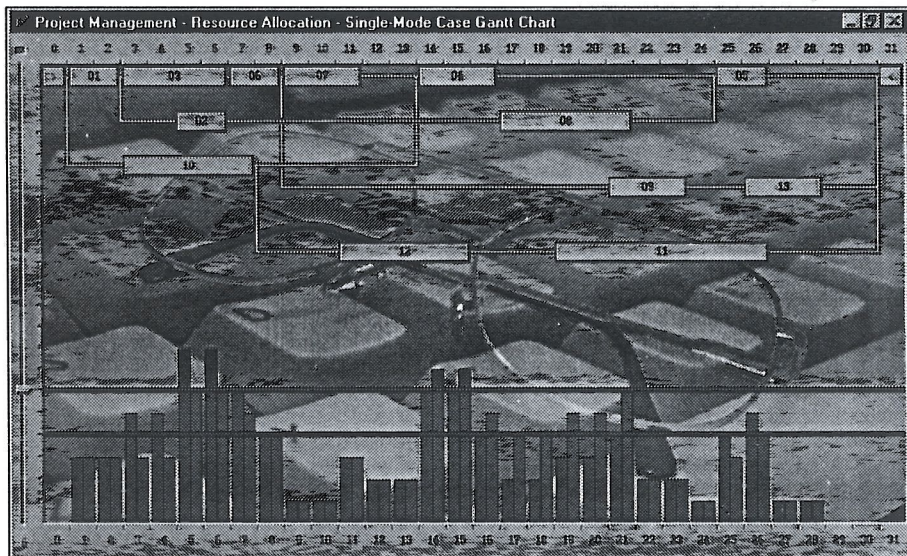




SZOFTVERFEJLESZTÉSI PROJEKTEK  
ÜTEMEZÉSÉNEK MODELLEZÉSE

Tézisek

Kruzslicz Ferenc



Konzulens:

Dr. habil. Csébfalvi György  
egyetemi docens PTE

## Háttér

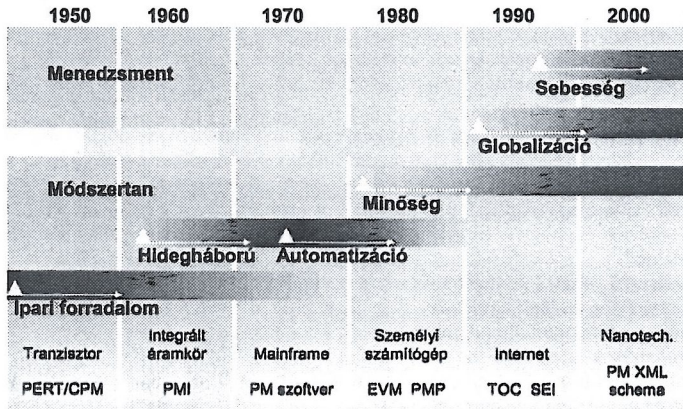
A projektmenedzsment mai modern formájának gyökerei mindössze néhány évtizedre vezethetők vissza, noha már az idősámítást megelőzően is kiviteleztek olyan hatalmas projekteket, mint a piramisok vagy a kínai fal építése. Az 1960-as évektől kezdve azonban a vállalatok fokozatosan rájöttek arra, hogy ha bizonyos tevékenységeiket projektek köré szervezik, és a különféle egységek és szakterületek közötti integrációt erősítik, az növeli a hatékonyságukat. Némely szervezet esetében ez egyszerűen kritikuss tényezővé vált.

Az ipari forradalom következményeként 1895 és 1923 között alakultak ki a klasszikus menedzsment tudományok alapjai, amely időszakot F.W. Taylor munkássága fémjelzi. Ekkor jelennek meg először az erőforrás-felhasználás mérésének és a termelési folyamatok ütemezésének fogalmai. Módszertani oldalról azonban ezen időszakból H. Gantt tevékenységét kell kiemelnünk, aki az első világháborúban a hadihajó-építés logisztikai problémáinak kezelésére fejlesztette ki a saját ütemezési és monitoring rendszerét. A róla elnevezett diagram, mint tervezési eszköz lényegében 1917 óta változatlan formában használatos. A második világháború nagyméretű és összetett projektjei tovább erősítették a menedzsment tevékenységek módszertani megalapozottságára iránti igényt. A menedzsment tudományok párhuzamos fejlődésének eredményeként eddigre kialakult a viselkedés elmélet, és kezdetét vette a két terület konvergenciája azáltal, hogy megjelenik a rendszer szemléletű megközelítés. A modern projektmenedzsment megszületésének 1940-es dátuma az atombomba előállítására kialakított Manhattan projekt nevéhez kötődik. A módszertan és menedzsment szülők gyermekeként létrejött projektmenedzsment kutatások területén ettől fogva lényegében 10 évenkénti ciklusban mindig történt valami lényeges változás.

Az 1950-es években megjelennek a folyamatábrák, 1957-ben publikálásra kerül a Critical Path Method (CPM), majd közvetlenül a rákövetkező évben a Program Evaluation and Review Technique (PERT) módszere. Az 1960-as évek hidegháborús korszakának hatalmas állami projektjei (űrprogram, atomerőművek) tovább gyorsították a tudományterület fejlődését. 1961-ben az IBM elsőként jelenik meg üzleti célú projektmenedzsment rendszerével. A számítási kapacitás növekedésének köszönhetően az 1970-es évektől kezdve a projektek időtervezés központi szemlélete kiegészül költség-haszon elemzési (Earned Value Management – EVM) szempontokkal is. Ezek az elemek 1969-ben alapított Project Management Institute (PMI) PMBOK nevű módszertanában jól tükröződnek, így erre a dolgozatom során többször is támaszkodtam. Ez a tendencia az 1980-as években a személyi számítógépek elterjedésével folytatódik, és a projektmenedzsment programok immár a kis- és közepes vállalatok számára is elérhetővé válnak. Erre égető szükség is volt, mert a technológiai fejlődés az eddig kisméretű projektek komplexitását is jelentősen megnövelte. A Project Management Professional (PMP) minősítés 1984-ben történt bevezetése egyben a projektmenedzsment szakma elfogadottságát is jelentette. A PRINCE strukturált projektvezetési módszertan 1991-es publikussá válásától kezdődően a projektmenedzsment szoftverek használata általánossá vált, és ennek következtében megnövekedtek azok az igények is, amelyek az általános célú irodai alkalmazásokra jellemzőek. Legyen a projektmenedzsment *gyorsabb, szebb, jobb és olcsóbb*.

Módszertani szempontból az 1997-ben E. Goldratt tollából megjelent Theory of Constraints (TOC) című könyvét érdemes megemlíteni, amely a szűk keresztmetszetek elméletét alapozta meg. A 2000 évi ezredfordulótól a vállalatok projektmenedzsmentési képessége alapvető versenyelőnyezővé vált, ami az informatika és ezen belül a szoftverfejlesztés területén még fokozottabban érvényesül. A számítási kapacitás növekedését az alkalmazott matematikai módszerek változása követte: előtérbe kerültek a kombinatorikus optimalizálási problémák. Az erőforrás-felhasználáshoz kapcsolódó hozzárendelési és kiegyenlítési feladatok éppen ide, pontosabban az NP nehézségű problémák körébe tartozik. Ez azt jelenti, hogy a felmerülő problémákra egzakt választ csak a feladatok méretével exponenciális arányú időben lehet adni. Napjaink felhasználói „erről tudomást sem véve” *azonnali ütemezést, azonnali költségtervet és azonnali helyzetelemzést* várnak el a rendszerektől.





A projektmenedzsment fejlődésének korszakai

A dolgozatomban éppen ezen ellentmondás enyhítésére teszek javaslatot olyan új modellek készítésével és a hozzá tartozó megoldási algoritmusok megadásával, melyekkel a kapott ütemezések jobban megfelelnek a projektmenedzserek elvárásainak. A témakör aktualitását az eddig ismertetteken túl az is mutatja, hogy hosszú lemaradást pótolva 1999-ben hazánkban is megalakult a Magyar Projektmenedzsment Szövetség, melynek legfőbb célkitűzése a projektmenedzsment kultúra elméleti és gyakorlati fejlesztése.

## A dolgozat felépítése

A projektmenedzsment tipikusan határterületi kutatási téma, ezért a háttér információknál említett kettősséget a dolgozat is magán viseli. Az első három fejezet tárgyalja az erőforrás hozzárendelési és kiegyenlítési feladatok köréhez kapcsolódó gazdasági környezetet. A következő négy fejezetet a modellek illetve az algoritmusok ismertetése tölti ki. Az első rész inkább szakirodalmi tanulmány-nak tekinthető, míg a módszertani rész teljes egészében önálló kutatási munka eredménye. A módszertani és a menedzsment eredmények közelítéséből származó tárgyalásmódot a modellek lényegét képező erőforrás kategóriák kialakítása illetve ezek felhasználási hatékonyságának globális szemléletű mérése indokolja. A dolgozat struktúrája a fokozatos témakörreléltetés elvét követi.

Az első fejezet az általános projektmenedzsment alapfogalmait definiálja. A PMBOK módszertan alapján ismerteti az ide kapcsolódó menedzsment funkciókat, majd ebben a rendszerben kijelöli az erőforrás hozzárendelési és kiegyenlítési feladatokat helyét. A tevékenységek ütemezése a rendszer-szemléletű megközelítés szerinti tervezési alrendszerhez tartozik, a projekt életciklusban viszont az akvizíciós fázisban kap helyet. A projektek sikertényezőinek áttekintése is azt célozza, hogy a később bemutatandó modellek alkalmazási környezetét megfelelő pontossággal határozzuk meg.

A második fejezet már kizárólag csak az emberi erőforrás és időmenedzsment funkciókat tekinti át egy szinttel mélyebb és részletesebb kibontásban. Bemutatásra kerülnek az erőforrás-tervezéshez és az ütemezés készítéshez használatos alapvető technikák, majd pedig a gyenge illetve erős erőforrások általános jellemzésére kerül sor. Az egyes ismertető jegyek között megkülönböztetünk elsődleges jegyeket – melyek minden erőforrás fájtára alkalmazhatóak; valamint másodlagos jegyeket – melyek csak emberi erőforrások esetén értelmezhetőek.

A harmadik fejezet az informatikai, azon belül a szoftverfejlesztési projektek sajátosságait vizsgálja.

A tapasztalatok alapján az ilyen projektek esetén még mindig viszonylag alacsony a sikeres kivitelezések száma. Ez jórészt annak köszönhető, hogy a szoftverfejlesztés magán viseli mindhárom projekt típus (beruházási, innovációs és szellemi szolgáltatás) jegyeit. Ezen túlmenően megállapíthatjuk, hogy komplexitási és bizonytalansági szintjük magas, és ezért speciális kompetencia igényűek. Nagyobb hangsúlyt kell helyezni az erőforrások multitaskingolására, valamint a tanulási folyamatok beépítésére. A modellek tekintetében az itt megfogalmazott egyszerűsítő feltevések közül a három legfontosabb a következő:

1. Fel kell oldanunk a szoftverfejlesztés folyamatában rejlő rekurziót (lásd spirál modell) annak érdekében, hogy az egyes tevékenységek között megelőzési relációkat lehessen definiálni.
2. Feltételezzük, hogy a párhuzamosan fejlesztendő projektek (az erőforrás-felhasználástól eltekintve) függetlenek egymástól. A gyakorlatban az ilyen projektek szerkezete is igen hasonló, de ezt a modelljeinkben nem használjuk ki.
3. Az erőforrások rendelkezésre állási szintjét a projektek időtartama alatt konstansnak tekinthetjük. Kis és közepes méretű alkalmazások (small scale business application) esetén ez nem jelent lényeges megszorítást.

A dolgozat további részében az ábrázolhatóság kedvéért mindig csak a feladatlebontrási struktúra legmagasabb szintjét, a lineáris „rendszertervezés – programozás – tesztelés” felbontást alkalmaztam.

A negyedik fejezet készíti elő a modellek ismertetését azáltal, hogy pontosan definiálja az erőforrás hozzárendelési és kiegyenlítési problémák körét, a továbbiakban használatos globális állásidő mértéket, az alkalmazott jelölésrendszert, valamint itt kaptak helyet azok az alapvető eredmények, melyeket az algoritmusokban gyakran felhasználok. A projektek párhuzamos sorba kapcsolása után a kritikus út módszerének (CPM) bemutatása következik, majd legvégül a tartalék- és pufferidők kezelésének lehetőségeit vázolom fel.

Az ötödik fejezet a gyenge és erős erőforrásokkal kiegészített IT<sub>SH</sub> (MILP) determinisztikus modell teljes bemutatását tartalmazza. A maximális felhasználási szint (MU) és minimális állásidő (IT) fogalmához az erőforrásprofilok kvázi-konkváziálásán keresztül juthatunk el. Röviden összefoglalom, hogy az új globális mértékek alkalmazása milyen előnyökkel jár a tradicionális lokális mértékekhez képest. Ezek segítségével lépésről lépésre bemutatom a végső modell alapjául szolgáló vegyes egészértékű programozási (MILP) modellt, majd pedig erős és gyenge erőforrások kezelésére alkalmas formára alakítom. A részletes ismertetést az indokolja, hogy beláthassuk: a modell kialakításához vezető összes lépés algoritmizálható. A fejezet végén egy ilyen algoritmusra épülő programmal egy 5 projektből álló konkrét példa adataiból kiindulva állítottam elő a megfelelő MILP modell együtthatóit. Kisméretű feladatok esetében ez általános célú lineáris programozási szoftverekkel meg is oldható. A dolgozatban az optimális ütemezések előállítására a Lindo szoftvert alkalmaztam.

A hatodik fejezetben egy olyan új modellt mutatok be, amely már közepes méretű feladatok megoldására is alkalmas. A tárgyalás struktúrája megegyezik az előző fejezet felépítésével. Itt is előbb egy alapmodell kerül ismertetésre, majd ennek felhasználásával készül el a gyenge és erős erőforrásokat is kezelni képes kétfázisú implicit leszámolási algoritmus. Ez a fejezet még egy lényeges ponton kapcsolódik az előző részhez, miszerint az ott ismertetett MILP modell relaxált változatának segítségével kialakított LIT-LMU vágási szabály jelentős mértékben felgyorsítja a kereső algoritmust. Az algoritmus működésének szemléltetéséhez választott példa méretét (SmallSize) leginkább ábrázolhatósági szempontok korlátozták, ennél nagyobb méretű feladatok megoldása sem jelent problémát.

Noha tudjuk, hogy nagyméretű feladatok esetében a kombinatorikus robbanás miatt nincs mód az optimális ütemezések gyors előállítására, a hetedik fejezet mégis azt vizsgálja meg, hogy miként



lehet legalább az ilyenek közelébe jutni. Az ilyen esetekben alkalmazható heurisztikus módszereket sorra véve megmutatom, hogy miként lehet ezeket az előző fejezetben ismertetett kétfázisú implicit leszámllási algoritmussal ötvözni annak érdekében, hogy a projektmenedzser igényeinek legjobban megfelelő ütemezésekhez juthassunk. Ezek közül a két legfontosabbnak a következőket tartom:

1. A modell tökéletesen alkalmas arra, hogy a projektmenedzser a saját szakértői preferenciáit is be tudja építeni a rendszerbe (a modell bármínemű változtatása nélkül).
2. A modell a kezdeti ütemezés előállításán túl kiválóan alkalmas arra, hogy az időközben a projektben bekövetkező változtatásokat visszavezessük. Ráadásul a módosított feladatok megoldása már gyorsabban kivitelezhető.

Az utolsó (nyolcadik) fejezet a dolgozat összefoglalásán túl a jövőbeli kutatási irányokat jelöli ki.

## Kutatási célkitűzések

A kutatási háttérben ismertetett tendenciák, valamint piacon jelenleg elérhető projektmenedzsmnt rendszerek ismeretében merült fel az igény olyan új modellek és algoritmusok keresésére, melyek jól illeszkednek ezen szoftverek menedzsmnt központú megközelítéséhez, azonban a rendelkezésre álló idő alatt egzakt megoldásokat állítanak elő. Ehhez egyrészt az eddigi lokális mértékek alkalmazásán alapuló szemléletet gyökeresen át kellett alakítani, másrészt a modellek matematikai szépségét valamelyest fel kellett áldozni a használhatóság oltárán. Céloom tehát olyan eszköz kifejlesztése volt, amely előrelépést jelent a projektmenedzsmnt szoftverek napjainkban aktuális „ütemezés minősége vs. előállításának sebessége” kihívásának területén. Ehhez az alábbi kérdések megválaszolását tűztem ki:

1. A menedzsmnt és a módszertani területek ismereteinek felhasználásával lehet-e olyan új, előre mutató fogalmakat alkotni, és eredményeket felmutatni, melyek megalapozottsága mindkét oldalról indokolt?
2. Hogyan lehet a projektmenedzserek ütemezésekkel szembeni elvárásainak jobban megfelelő modelleket felállítani?
3. Az elsősorban szoftverfejlesztési projektekre kialakított modelleket lehet-e, és hogyan úgy megfogalmazni, hogy alkalmasak legyenek másfajta projektek kezelésére is?
4. Milyen módszerekkel lehet áthidalni a problémaosztály eredendő NP nehézségét úgy, hogy az egyúttal alkalmas legyen szakértői preferenciák kezelésére is?
5. Miként lehet a jelenlegi rendszerekhez illeszthető, de azoktól mégis független egzakt ütemezési eszközt készíteni?

Azonban maradt néhány olyan kérdés, amely túlmutat a dolgozat keretein, de feltétlenül érdemesnek tartom megemlíteni.

Az első a publicitás kérdése. A projektmenedzsmnt területén számos eredmény jórészt üzletpolitikai megfontolásból nem kerül nyilvánosságra. Ezért nagyon fontosnak tartom, hogy ezek az algoritmusok napvilágot lássanak, hiszen működésük és hatékonyságuk csak ennek tükrében értelmezhetőek. Bár a szabad szoftverek egyik (a kutatások szempontjából legfontosabb) küldetése éppen ez, mégsem készült még ilyen rendszer. Ezt felismerve a fejlesztések csak nemrég indultak el, a kezdeti lépések biztatóak.

A második pedig a szabványosítás problémája. A gyakorlatban előforduló projektek kellően sokfélék ahhoz, hogy ne keressünk egyetlen sikerreceptet. Mégis a sokféle felfogás és létező rendszer közötti átjárás biztosításának hiánya mára a rendszerek fejlődésének gátját jelentheti. Ezért tartom nagyon fontosnak, hogy nemrégiben betervezésre került a projektek szabványos leírását szolgáló Project Management XML séma.

## Tézisek

A szakirodalom áttanulmányozása során szerzett ismerteket a saját kutatási eredményeimmel összevetve az alábbi feltételezéseket fogalmazhatjuk meg:

*1. Ha a rendelkezésre álló erőforrások korlátozottsága alapján eddig használt kategorikus (korlátos vs. korlátlan) besorolás helyett egy korlátozottsági skálát használunk, akkor az erőforrások hatékonyság felhasználásának modelljei pontosabbá tehetőek.*

Igazoltam, hogy az egyes erőforrások korlátozottsága nem tisztán az adott erőforrás belülről fakadó tulajdonsága, hanem korlátozottságának mértéke erősen függ a projektben betöltött szerepétől, és így végül magától a projekttől is. A gyengén illetve erősen korlátos erőforrások fogalmát matematikailag is pontosan definiálva, az így kialakított újszerű erőforrás-osztályok integráns részét képezik a kialakított ütemezési modelleknek. Végül javaslatot tettem arra, hogy milyen szempontok alapján lehet megítélni egy adott erőforrás rendelkezésre állásának korlátozottsági fokát, egy adott projekt tekintetében.

*2. A többféle (erősen és gyengén korlátos) erőforrás esetére, több célkritérium mellett, valamint több projekt környezetben is megadható egy általánosan használható determinisztikus ütemezési modell.*

A dolgozatban ismertetett ütemezési modell azon túl, hogy új elméleti eredmény, a gyakorlatban is közvetlenül alkalmazható. Használhatóságát a szoftverfejlesztési projektek esetére bizonyítottuk, de a módszer más (humán erőforrás intenzív) tevékenységek esetére is könnyen átvihető. Az ütemezések, illetve az erőforrás-kihasználtsági profilok hatékonyságának globális jellemzésére olyan kombinált célfüggvény rendszert alkalmaztunk, amely egyszerre minimalizálja a (gépütemezési feladatoknál ismert fogalom analógiájára épülő) tevékenységi állásidőket és maximalizálja az erőforrás kihasználtságot.

*3. Kisméretű feladatok esetére megadható egy nyolclépéses módszer, amellyel a modell egy vegyes egészértékű lineáris programozási feladat (MILP) megoldására vezethető vissza.*

A bemutatott módszer egyszerűsége és automatizálhatósága mellett rendelkezik azzal az előnyös tulajdonsággal is, hogy a visszavezetesként kapott MILP feladat egyíthető mátrixa igen ritka. Ennek következtében a kisméretű feladatok egzakt megoldása még viszonylag szerény számítógépes eszközökkel is kivitelezhető.

*4. Közepes méretű feladatok esetére megadható egy kétfázisú implicit leszámítási algoritmust.*

Az ismertetett kétfázisú algoritmus leglényegesebb tulajdonsága az, hogy sikeresen ötvözi az erőforrás hozzárendelési és kiegyenlítési feladatokra alkalmazható megoldási módszereket. Kiemelendő sajátossága továbbá az is, hogy képes az összes Pareto optimális megoldás előállítására, nem csak azok valamelyikének megkeresésére.

*5. Nagyméretű feladatok esetén megmutatható, hogy a közepes méretű feladatok megoldására kidolgozott kétfázisú módszer sikeresen ötvözhető a hagyományosan alkalmazott heurisztikus eljárások többségével.*

Bemutattam, hogy a keresési fa speciális felépítési, vágási és korlátozási szabályainak kialakításával hogyan valósíthatók meg a különféle heurisztikus megközelítések. Bebizonyítottam, hogy az algoritmus (speciális halmazainak segítségével) lényegében változatlan formában alkalmas a bázis ütemezés folyamatos karbantartására, finomítására. Példákon keresztül igazoltam, hogy az általam ajánlott módszerek valós környezetben is hatékonyan segítik az ütemezési problémák kezelését.



## Eredmények

A téziseknél megfogalmazott feltételezésekre a dolgozat megfelelő fejezetei pozitív válaszokat adnak. Az alábbi összefoglalóban főleg az új eredményeket szeretném kiemelni.

### *Gyenge és erős erőforrások*

A projektek kivitelezése során valamilyen erőforrás mindig felhasználásra kerül, ha más nem – az idő biztosan. A kivitelező általában rendelkezik ezekből bizonyos mennyiségekkel, de időnként többlet erőforrás-egységek bevonására lehet szüksége. A pótlólagos erőforrások bevonásának költsége az erőforrás természetétől függően a  $[0, \infty]$  intervallumban más és más értéket vehet fel. Az adott projekt szempontjából azonban meghatározhatunk egy olyan  $K$  küszöbértéket, mely alatt az adott erőforrás gyengének  $[0, K]$ , illetve felette erősnek minősíthető  $[K, \infty]$ . A rendelkezésre álló erőforrások tétlenségéből származó kieső idő költségére is hasonló megállapítás tehető.

Az  $R$  erőforrást erősen korlátos erőforrásnak (röviden erős erőforrásnak) nevezzük, ha a projekt időtartama alatt adott az erőforráshoz tartozó  $L$  rendelkezésre állási korlát. Ezt a korlátot egyetlen ütemezés sem lépheti túl, vagyis pótlólagos egységek bevonására nincs mód. Ellenkező esetben az erőforrást gyengén korlátos erőforrásnak nevezzük, és hozzá tartozó rendelkezésre állási korlátokat figyelmen kívül hagyhatjuk.

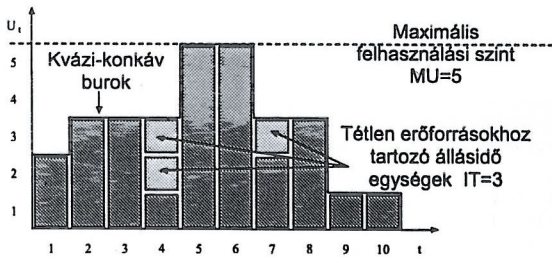
Az erőforrások adott projekten belüli tipizálása a következő jellemvonások elemzésével is megoldható. A megkülönböztető tulajdonságok közül itt most csak a legfontosabbakat sorolom fel, az elsődleges jegyeket kiemelésével:

<i>Ismertető jegy</i>	<i>Erősen korlátos erőforrások esetén</i>	<i>Gyengén korlátos erőforrások esetén</i>
Rendelkezésre álló erőforrás egység	Gyakorlatilag véges	Elméletileg végtelen
Projekt szempontjából kulcs erőforrás	Igen	Nem
Megújítható erőforrás	Nem	Igen
Létszám- vs. időigény átváltási ráta	Alacsony	Magas
Létszámbővítés időigénye / költsége	Hosszabb / Magas	Rövidebb / Alacsony
Kieső erőforrás pótolhatósága	Nehézkés	Viszonylag könnyű
Tanulási görbe	Lapos	Morodok

### *Minimális állásidő és maximális felhasználás mértéke*

Ha a projekt időtartama alatt egy adott erőforrásból szükséges mennyiségeket grafikonon ábrázoljuk, akkor megkapjuk az ütemezés által meghatározott erőforrás felhasználási profilt. A grafikon legmagasabb pontja határozza meg a maximális felhasználás mértékét. Amennyiben olyan ütemezéseket keresünk, amelyeknél egyik erőforrás profil sem lépi túl a hozzá tartozó rendelkezésre állási korlátot – úgy *hozzárendelési feladatról* beszélünk.

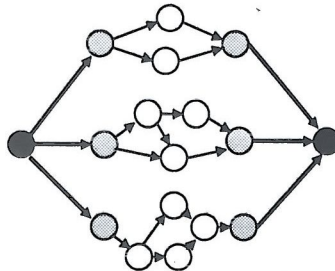
Ha a profil alakjára nézve vannak elvárásaink, akkor *kiegyenítési feladattal* állunk szemben. A projektmenedzser szempontjából a kvázi-konkáv alak tekinthető ideálisnak, lévén ez támogatja legjobban az erőforrások átlapolt felhasználását a különféle tevékenységek között. A kvázi-konkáv profilon alapuló globális megközelítés további előnye, hogy kiküszöböli az erőforrás felhasználás mértékének ingadozását, és a pótlólagos egységek bevonását is megszakítás nélküli módon teszi lehetővé.



Egy erőforrás profil, a hozzá tartozó kvázi-konkáv burok, az állásidők és a maximális felhasználási szint feltüntetésével

Mindkét esetben fontos követelmény a projekt időtartamának minimalizálása úgy, hogy tevékenységek kezdési időpontjai eleget tegyenek a közöttük lévő közvetlen megelőzési relációknak. Ilyen sorrendiségi megkötéseket leggyakrabban technológiai eljárások illetve preferenciák figyelembe vételével kaphatunk.

A több projekt esete az egyes projektek párhuzamos sorba kapcsolásával mindig visszavezethető az egy projekt esetre. Ehhez mindössze egy üres kezdő és egy záró tevékenységet kell bevezetni a megfelelő közvetlen megelőzési relációk felvételével.



Három AON (activity on node) jelöléssel felírt projekt párhuzamos sorba kapcsolása

### Gyenge és erős erőforrások ütemezésének $IT_{SH}$ determinisztikus MILP modellje

A többféle optimalitási célkritériumot helyettesítjük azok súlyozott összegével. A súlyok a projektmenedzser preferenciái alapján kerülnek kialakításra, a modell illusztrálásakor egységnyinek tekintjük őket. Ha  $IT^r(S)$  jelöli az  $r$ . erőforrás  $S$  ütemezéséhez tartozó tétlen időegységek számát, valamint  $MU^r(S)$  a maximális felhasználási szintjét, akkor a következő kiindulási modellt írhatjuk fel (a továbbiakban mindig feltételezzük, hogy a gyenge erőforrások indexei az alacsonyabbak):

$$\sum_{r=1}^{k+m} IT^r(S) + \sum_{r=1}^k MU^r(S) \rightarrow Min! \quad S \in \mathfrak{R}$$

ahol  $\mathfrak{R}$  jelöli a lehetséges ütemezések terét - vagyis azokat, amelyek nem sértik meg a közvetlen megelőzési relációkat. Az ezt leíró korlátozó feltételek két csoportba sorolhatóak aszerint, hogy az időre, vagy más egyéb erőforrásra jelentenek megszorításokat.



Input paraméterek jelentése és ...	jelölése.
A projektben előforduló tevékenységek száma	$N$
A projekt időtartama alatt az $r$ . erőforrás rendelkezésre állási szintje (készlet)	$L^r$
Gyengén korlátos erőforrások száma	$k$
Erősen korlátos erőforrások száma	$m$
Az $i$ . tevékenység végrehajtásához szükséges idő	$D_i$
Az $i$ . tevékenység végrehajtásához az $r$ . erőforrásból szükséges mennyiség	$R_i^r$
Tevékenységek közötti függőségek megadása közvetlen megelőzési relációkkal (az $i$ . tevékenységnek a $j$ . tevékenység megkezdése előtt be kell fejeződnie)	$i \rightarrow j$

Modell változóinak jelentése és ...	jelölése.
Az $S$ ütemezés mellett az $i$ . tevékenység megkezdésének időpontja	$S_i$
Az adott ütemezés $t$ . időpontjában az $r$ . erőforrásból szükséges mennyiség	$U_i^r$
Az $U$ erőforrás-felhasználási profil kvázi-konkáv burkolója	$\bar{U}$
Az $i$ . tevékenység kezdetének CPM szerinti legkorábbi időpontja	$ES_i$
Az $i$ . tevékenység kezdetének CPM szerinti legkésőbbi időpontja	$LS_i$
Az $i$ . tevékenység a $t$ . időpontban kezdődik-e vagy sem (logikai 0/1 változó)	$AS_{i,t}$

A fenti jelölésekkel a célfüggvény és a korlátozó feltételek a következő alakot öltik:

Célfüggvény: 
$$\sum_{r=1}^{k+m} \sum_{t=1}^T \bar{U}_i^r - U_i^r + \sum_{r=1}^k \max_{1 \leq t \leq T} U_i^r \rightarrow \text{Min!}$$

**Idő-feltételek (Strong-X felírással):**

Minden tevékenység pontosan egyszer kerül végrehajtásra,

$$\sum_{i=ES_i}^{LS_i} AS_{i,t} = 1 \quad \text{ahol } 1 \leq i \leq N \text{ és } AS_{i,t} \in \{0,1\}$$

és az ütemezés megtartja a közvetlen megelőzési relációkat.

$$\sum_{i=1}^{LS_i} AS_{i,t} + \sum_{i=ES_i}^{t+D_i-1} AS_{i,t} \leq 1 \quad \text{ahol } ES_i \leq t \leq LS_i \text{ és } i \rightarrow j$$

**Erőforrás-feltételek:**

Meghatározzuk az erőforrás felhasználási szinteket,

$$U_i^r = \sum_{i=1}^N \left( R_i^r \cdot \sum_{l=t-D_i+1}^t AS_{i,l} \right) \quad \text{ahol } 1 \leq t \leq T \text{ és } 1 \leq r \leq k+m$$

melyek közül az erősen korlátosak nem léphetik túl a rendelkezésre állási korlátokat.

$$U_i^r \leq L^r \quad \text{ahol } 1 \leq t \leq T \text{ és } k+1 \leq r \leq k+m$$

A felírásból látható, hogy a MILP alaknak egyedül a célfüggvény nem felel csak meg.

Ha azonban a célfüggvényben a maximális felhasználási szint  $\sum_{r=1}^k \max_{1 \leq t \leq T} U_i^r$  képlete helyett a  $\sum_{r=1}^k M^r$  kifejezést használjuk, és a feltételrendszer minden  $1 \leq t \leq T$  és  $1 \leq r \leq k$  esetén kiegészítjük az  $U_i^r \leq M^r$  egyenlőségekkel, akkor az eredeti felírással ekvivalens alakot kapunk, amely már egy lépéssel közelebb áll célunkhoz.

Alkalmazzunk hasonló átalakítást a célfüggvény tétlen állásidőket mérő első tagjára is. Most a

$$\sum_{r=1}^{k+m} \sum_{i=1}^T \bar{U}_i^r - U_i^r \quad \text{összeg helyett a } \sum_{r=1}^{k+m} \sum_{i=1}^T I_i^r \quad \text{kifejezést használhatjuk, ám a feltételrendszer ez esetben}$$

sokkal több sorral kell bővíteni. Az erőforrás felhasználási profil kvázi-konkáv burkolójának rekurziómentes kifejtésével kaphatjuk a következő addicionális feltételeket:

Minden  $1 \leq t \leq T$  és  $1 \leq r \leq k + m$  esetén:  $U'_t + I'_t \geq \text{MIN}'_t$  és  $\text{MIN}'_t = \min(\text{MAX}'_{t-1}, \text{MAX}'_{t+1})$

$$\begin{array}{ll} \text{MAX}'_t = U'_t & \text{MAX}'_r = U'_r \\ \text{MAX}'_t \geq \text{MAX}'_{t-1} & \text{MAX}'_r \geq \text{MAX}'_{r+1} \\ \text{MAX}'_t \geq U'_t & \text{MAX}'_r \geq U'_r \end{array}$$

Ebben az alakban a minimum függvény kivételével már minden feltétel megfelel a MILP formának. Ha azonban az  $M = \min(A, B)$  feltételek mindegyikét helyettesítjük a következő, vele ekvivalens felírással:  $M \leq A$ ,  $M \leq B$ ,  $M = A \cdot x + B \cdot (1-x)$  ahol  $x \in \{0,1\}$ , akkor már csak az egyenlőségnél szereplő kétfágú szorzatok rontják el az összképet.

A Big-M módszer segítségével éppen az ilyen kéttényezős szorzatokra tudunk MILP kompatibilis felírást adni. Minden  $M = A \cdot x + B \cdot (1-x)$  alakú egyenlőségi feltételt helyettesíthetünk az alábbi feltételrendszerrel:  $M = PL + PR$ ,  $PL \geq \underline{A} \cdot x$ ,  $PL \geq A + \bar{A} \cdot x - \bar{A}$ ,  $PR \geq -\underline{B} \cdot x + \underline{B}$ ,  $PR \geq B - \bar{B} \cdot x$ .

Ebben a felírásban az alul- illetve felülvonással jelölt értékek olyan konstansok, melyek az adott változó alsó illetve felső korlátait jelentik. Ezen konstansok értékei a közvetlen megelőzési relációk segítségével határozhatóak meg.

Mindent összevetve megállapíthatjuk, hogy az átalakítások során valóban MILP feladatot kaptunk, és a felíráshoz vezető út teljes mértékben algoritmizálható. A kapott modell mérete ellenére jól kezelhető, mert az együttható mátrixának sűrűsége igen kicsi.

### Gyenge és erős erőforrások Pareto optimális ütemezésének ITMU<sub>SH</sub> determinisztikus modellje

A közvetlen megelőzési relációkat megtartó (ún. *lehetséges*) ütemezéseken túl be kell vezetnünk a elfogadható ütemezés fogalmát is. Ha egy  $S$  lehetséges ütemezés végrehajtásakor egyetlen időpontban sem lépünk túl az erős erőforrásokból rendelkezésre álló mennyiséget, akkor az  $S$  ütemezést *elfogadható* ütemezésnek nevezzük. Amennyiben a problémát az eredeti, több célkritériumos alakban akarjuk kezelni, úgy a következő feladatot fogalmazhatjuk meg:

Állapítsuk meg azt a minimális  $T$  időtartamot, amely mellett létezik elfogadható ütemezés, majd az elfogadható ütemezések  $\mathcal{R}^H$  terében keressük meg az  $(IT^1(S), \dots, IT^{k+m}(S), MU^1(S), \dots, MU^{k+m}(S))$  vektor szerinti összes hatékony (Pareto optimális)  $S \in \mathcal{R}^H$  megoldást. Az  $S \in \mathcal{R}^H$  ütemezést Pareto optimális ütemezésnek nevezzük, ha nem létezik olyan  $S' \in \mathcal{R}^H$  ütemezés, amelyre minden  $1 \leq r \leq k + m$  esetén teljesülnek az  $IT^r(S') \leq IT^r(S)$  és  $MU^r(S') \leq MU^r(S)$  egyenlőtlenségek úgy, hogy az esetek legalább egyikében az egyik egyenlőtlenség szigorú.

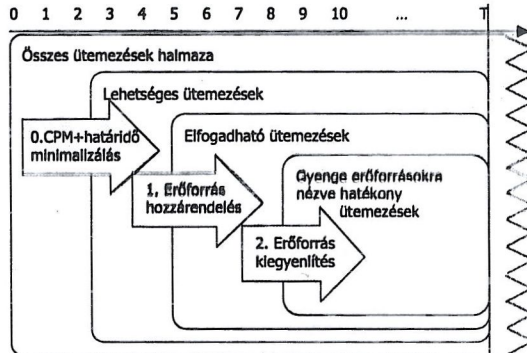
A hatékony megoldások előállítására ebben az esetben a következő kétfázisú implicit leszámolás algoritmust adhatjuk meg:

**0. fázis - Kiindulás:** Megállapítjuk azt a minimális időtartamot, amelytől kezdve létezik lehetséges megoldás. Ezt a kiindulási állapotot a CPM módszer segítségével határozzuk meg úgy, hogy az erőforrásainkat figyelmen kívül hagyjuk.



1. fázis - Hozzárendelés: Az elfogadható ütemezések terét úgy állíthatjuk elő, hogy a gyenge erőforrások figyelmen kívül hagyásával kapott problémát, mint erőforrás hozzárendelési feladatot oldjuk meg. Figyelembe kell vennünk azonban azt, hogy ez esetleg csak a 0. fázisban megállapított minimális határidő növelésével lehetséges.

2. fázis – Kiegészítés: Az előző fázis végeredményeként rendelkezésünkre áll az elfogadható ütemezések halmaza. Ha most ebben a térben az összes erőforrás figyelembe vételével kapott erőforrás kiegyenlítési feladatot oldjuk meg, akkor az így kapott Pareto optimális megoldások egyben az eredeti feladat hatékony megoldásai is lesznek.



A kétfázisú, implicit leszámllási algoritmus menetének vázlata

Ahhoz, hogy az előbbieken felvázolt lépéseket végre is tudjam hajtani, olyan eszközt kellett találni, amely alkalmas ütemezés-halmazok előállítására és tárolására, sőt mindezt hatékonyan oldja meg. Az implicit leszámllási probléma jellegéből adódóan következik, hogy a keresési fákon alapuló módszerek között kell a megoldást keresni. Az ütemezési halmazokat reprezentáló keresőfa gyors bejárásához egyrészt alkalmas fabejárési stratégiára, másrészt pedig hatékony vágási és korlátozási szabályokra van szükségünk.

#### Ütemezés-halmazok reprezentációja az első (hozzárendelési) fázis keresőfájában

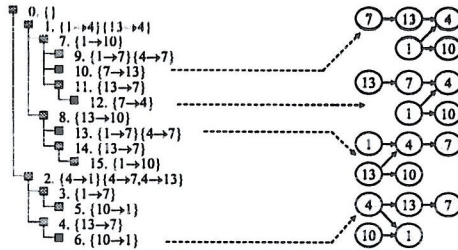
A keresőfa minden csomópontjához egy közvetlen megelőzési relációkat tartalmazó halmazt rendelünk hozzá. Ennek megfelelően egy adott csomóponthoz azokat az ütemezéseket rendeljük hozzá, amelyek az eredeti közvetlen megelőzési relációkon túl, a csomóponthoz rendelt újabb relációkat is megtartják. A keresőfa még ki nem bontott ágait aktív ágaknak nevezzük.

1. Kiindulásként a fának egyetlen aktív csomópontja van (a gyökere), amelyhez az üres halmazt rendeljük; hiszen ez felel meg a 0. fázisban kapott állapotnak. Majd a következő lépéseket folytatjuk mindaddig, amíg van feloldható konfliktust tartalmazó aktív csomópont a fában:

2. Minden újonnan létrehozott aktív csomóponthoz meghatározunk egy hozzá tartozó minimális összeférhetlenségi halmazt. Bár a csomóponthoz tartozó ütemezések eleget tesznek a közvetlen megelőzési relációkon túl a csomóponthoz rendelt halmaz relációinak is; mégis előfordulhat, hogy valamely erős erőforrásból felhasznált mennyiség egy adott időpontban túllépi a rendelkezésre álló mennyiséget. Az ilyen helyzetet *erőforrás konfliktusnak*, az adott időpillanatban végrehajtás alatt álló tevékenységeket pedig *összeférhetetlennek* nevezzük. Az algoritmus szempontjából azonban ezeknek egy speciális típusa, a minimális összeférhetlenségi halmaz fogalma bizonyult kulcsfontosságúnak. Tevékenységek egy halmazát akkor nevezzük *minimális összeférhetlenségi halmaznak*, ha egymással összeférhetetlenek, de számuk minimális abban az értelemben, hogy bármely valódi részhalmazuk már összeférhető. Amennyiben egy csomóponthoz több minimális összeférhetlenségi halmaz is található, akkor időben a legelsőnek előfordulót választjuk. Ha ilyenből is

több van, akkor azok közül egy tetszőlegesen. Ezzel a sorrendkijelöléssel biztosítjuk, hogy egyetlen konfliktus se kerülje el a figyelmünket.

3. A keresőfa aktív csomópontjai közül egy tetszőlegesen bontunk ki. A kibontás jelen esetben azt jelenti, hogy ebből a csomópontból annyi új leágazást készítünk, ahányféleképpen a kiválasztott csomóponthoz rendelt minimális összeférhetetlenségi halmaz konfliktusait feloldhatjuk. Vegyük észre, hogy ha egy  $K$  minimális összeférhetetlenségi halmaz bármely két  $i, j$  eleme között közvetlen megelőzési relációt definiálunk, akkor a halmaz a konfliktus időpontjára nézve összeférhetővé válik. Az ilyen  $i \rightarrow j$ , illetve  $j \rightarrow i$  típusú feloldások száma tehát legfeljebb  $|K| \cdot (|K| - 1)$ , ugyanis némely feloldás ellentmondásba kerülhet az eddigiekkel. A kiválasztott csomópont halmazát kibővítjük a leágazatához tartozó közvetlen megelőzési relációval, és az így keletkezett halmazt rendeljük az új csomóponthoz. A kibontandó aktív csomópont véletlenszerű kiválasztása helyett alkalmazhatunk valamilyen súlyozás szerinti irányított kiválasztást is.



Egy keresési fa, levelein a kibontáskor hozzáadott új konfliktus feloldási relációk feltüntetésével.

A keresőfa kibontása akkor ér véget, ha minden levél vagy konfliktusmentes, vagy az adott határidő alatt feloldhatatlan konfliktust tartalmaz. Ha vannak konfliktusmentes levelek, akkor az ezekhez tartozó ütemezések adják ki az elfogadható ütemezések halmazát. Ha azonban minden levél feloldhatatlan konfliktust tartalmaz, akkor az erős erőforrások figyelembe vételével az adott idő alatt a hozzárendelési feladat nem oldható meg. Ekkor a projekt határidejét egy időegységgel meghosszabbítva új hozzárendelési feladatot definiálunk. Ehhez az új feladathoz tartozó fagyőkeret az előző (sikertelen) gyökér alá fűzzük, és az első fázis lépéseit újra kezdjük.

### Ütemezés-halmazok reprezentációjá a második (kiegyenlítési) fázis keresőfájában

A második fázis annyi kiegyenlítési feladat megoldását jelenti, ahány konfliktusmentes levelet találunk az első fázis végeredményeként kapott keresőfában. Minden egyes levélhez egy olyan kiegyenlítési feladatot rendelünk, amelyet az eredeti feladat következő módosításaival kapunk:

- A projekt határidejének az első fázisban kapott határidőt jelöljük ki.
- Az erős erőforrásokat „visszaminősítjük” gyenge erőforrásokká, vagyis ennek megfelelően kezeljük őket is.
- Az eredeti feladat közvetlen megelőzési relációihoz hozzávesszük az adott levélhez tartozó konfliktus feloldási relációkat is.
- A feladat céljául az  $(IT^1(S), \dots, IT^{k+m}(S), MU^1(S), \dots, MU^{k+m}(S))$  vektort minimalizálását választjuk.

Vegyük észre, hogy az első fázis tetszőleges konfliktusmentes leveléhez tartozó ütemezések az eredeti feladatnak elfogadható, a fentiek szerint módosított feladatnak azonban lehetséges megoldásai lesznek. Mivel az erőforrás konfliktusokat lényegében új közvetlen megelőzési relációk



megadásával oldottuk fel, ezért a kiegyenlítési feladatok megoldása nem vezet ki az eredeti feladat elfogadható ütemezéseinek köréből. A második fázis végrehajtása után kapott hatékony ütemezések tehát egyben az eredeti feladat egzakt megoldásai lesznek.

A hozzárendelési fázisban használt, keresőfán alapuló implicit leszámológási módszer ebben a kiegyenlítési fázisban is alkalmazható. Sőt technikailag az első fázis eredményeként kapott fa kibontását folytathatjuk tovább, csak a kibontás módszere és jelentése fog megváltozni. Azon részfák mérete, melyekkel a kiegyenlítési fázisban dolgozunk, jelentősen nagyobbak, mint amilyenekkel a hozzárendelési fázisban találkozhattunk. Ezért most elengedhetetlenül fontos egy jó bejárás stratégia és hatékony vágási szabályok alkalmazása.

A második fázis feladataihoz tartozó fák gyökerei tehát az első fázis keresőfájának konfliktusmentes levelei lesznek. Az egyes csomópontokhoz a második fázisban is halmazokat rendelünk, de ezek a halmazok most nem konfliktusfeloldó halmazok lesznek, hanem bizonyos tevékenységek kezdeteit rögzítik adott időpontokban. Egy ilyen  $F = \{S_{i_1} = t_{i_1}, \dots, S_{i_n} = t_{i_n}\}$  halmaz lényegében egy részütemezést határoz meg azáltal, hogy az  $i_1$ -dik tevékeny kezdetül a  $t_{i_1}$  időpontot jelöli meg, és így tovább. Ezzel a keresőfa egyes csomópontjaihoz azokat az  $\mathfrak{R}(F)$  ütemezéseket rendeljük hozzá, amelyek egyrészt kielégítik az eredeti feladat közvetlen megelőzési relációinak konfliktusfeloldó relációkkal kibővített halmazát; másrészt a hozzá tartozó halmaz tevékenységeinek kezdetei az ott megadott időpontokban explicit módon le vannak rögzítve.

1. Első lépésben a részfa gyökeréhez most is az üres halmazt rendeljük, melynek jelentése szerint még minden tevékenység „mozgatható”, egyik sincs rögzítve. Ezek után az alábbi lépéseket ismételjük mindaddig, amíg valamely aktív csomópontban még található mozgatható elem.

2. A mozgatható tevékenységet tartalmazó aktív csomópontok közül kiválasztunk egy tetszőlegeset. Ezen csomópontoz tartozó mozgatható tevékenységek közül egyet kiválasztva annyiféle leágazást készítfünk, ahányféleképpen az a kiszemelt tevékenység kezdési időpontját rögzíteni lehet. A szülő ág halmazát kibővítjük a leágaztatáshoz tartozó tevékenységkezdet rögzítésének leírásával, és az így kapott rögzítési halmazt rendeljük hozzá az újonnan leágaztatott csomópontoz.

A második fázis egy-egy részfájának kibontása akkor ér véget, ha a részfa egyetlen levele sem tartalmaz már mozgatható elemet. Ennek megfelelően az összes részfa kibontása után a leveleken megkapjuk az eredeti feladat összes elfogadható ütemezését, már csak ezek közül kell a Pareto minimálisakat kiválogatni.

Vegyük észre azonban, hogy a Pareto optimumok keresésével nem kell a legvégső pillanatig várni. Ugyanis ha a részfák bejárása közben egy adott csomópontoz tartozó ütemezések célfüggvényértékeire sikerül megfelelő alsó becslést adnunk, akkor az egész ág kifejtését elhagyhatjuk. Ilyen megfelelő alsó becslés a dolgozatban bemutatott LIT-LMU (Lower bounds of Idle Times and Lower bounds of Maximal Usage) vágási szabály, melynek értelmében:

Az  $\mathfrak{R}^*$  ütemezés-halmaz dominálja az  $\mathfrak{R}(F)$  halmazt, ha minden  $1 \leq r \leq k+m$  erőforrásra nézve teljesül mind az  $IT^r(S^*) \leq LIT^r(F)$ , mind a  $MU^r(S^*) \leq LMU^r(F)$  egyenlőtlenség, ahol  $S^*$  egy tetszőleges  $\mathfrak{R}^*$ -beli ütemezés. Ilyen esetben, ha  $\mathfrak{R}^*$  már szerepel csomópontként a keresőfában, akkor az  $\mathfrak{R}(F)$ -hez tartozó csomópont kibontása elhagyható. Az egyenlőtlenségekben szereplő becslések kiszámításának pontos definícióit az alábbiak szerint adhatjuk meg:

$$LMU^r(\emptyset) = \max_{1 \leq i \leq N} R_i^r, \text{ valamint } LMU^r(F) = \max(MU^r(S'), LMU^r(\emptyset)), \text{ ahol } S' \text{ jelöli azt}$$

a „csonka” ütemezést, amelyben csak az explicit módon rögzített tevékenységek erőforrásigényét vesszük figyelembe.

$$LIT^r(F) = \sum_{i \in C(F)} \max(\min(MAXCL_{i-1}^r, MAXCR_{i+1}^r) - U_i^r, 0), \text{ ahol}$$

minden  $1 \leq i \leq T-2$  esetén  $MAXCL_i^r = \max(C_1^r(F), C_2^r(F), \dots, C_i^r(F))$ , valamint

minden  $3 \leq i \leq T$  esetén  $MAXCR_i^r = \max(C_i^r(F), C_{i+1}^r(F), \dots, C_T^r(F))$ .

Továbbá minden  $1 \leq i \leq T$  és  $1 \leq r \leq k+m$  esetén  $C_i^r(F) = \min(U_i^r(S) : S \in \mathfrak{R}(F))$ ,

valamint  $C(F) \subset \{1, 2, \dots, T\}$  a projekt időegységeinek egy olyan halmaza, amely időpontokban az erőforrás-kihasználtsági profil a mozgatható tevékenységek segítségével nem változtatható meg.

A részfák méretének további jelentős mértékű csökkenését érhetjük el azáltal, hogy a leágaztatás során a CPM szerinti kritikus út tevékenységeinek kezdési időpontjait azonnal rögzítjük, lévén ezek amúgy sem mozgatható tevékenységek.

Ami a bejárési stratégiát illeti, ott is érdemes a kifejtést mindig a „legígéretebb” csomóponttal folytatni. A legígéretebb alatt azt értjük, amely várhatóan a legközelebb visz egy optimális megoldáshoz, és/vagy a lehető legtöbb ágat vágja le. Egy viszonylag gyorsan kiértékelhető súlyfüggvény

lehet a  $QC(S) = \sum_{r=1}^{k+m} IT^r(S) + MU^r(S)$ , melynél a legkisebb költségű csomópont kibontása ajánlott.

Az itt felsorolt szabályok alkalmazásával lehetővé válik immár közepes méretű feladatok egzakt megoldása is.

## Heurisztikus elemek alkalmazása

Amikor egy feladat mérete eléri a kombinatorikus robbanást, akkor az eddig alkalmazott módszerek számítási igénye jelentősen megnő. Ezért az algoritmus gyorsítása érdekében az alábbi lehetőségeket (vagy ezek kombinációit) vehetjük sorra:

1. Az optimális megoldások helyett megelégszünk annak valamilyen közelítésével, vagy
2. az eredeti feladat helyett annak gyengített (relaxált) változatát oldjuk meg, vagy
3. az összes optimális megoldás megkeresése helyett megelégszünk a legelső optimum megtalálásával, vagy
4. újabb vágási szabályok és fabejárési stratégiák felállításával csökkentjük a keresési fa méretét, vagy
5. speciális heurisztikát alkalmazunk, amely kihasználja a feladat sajátosságainak szerkezetét, vagy
6. a fontosabb elágazásoknál „kézi vezérlésre kapcsolunk”, azaz szakértői döntésre bizzuk az algoritmus folytatását.

A dolgozat utolsó fejezetében megmutatom, hogy miként lehet a fenti technikákat akár egymással ötvözve is megvalósítani a kétfázisú implicit leszámítási algoritmus keretein belül. Legvégül pedig arra hozok példát, hogy megfelelő költségfüggvény alkalmazása esetén a mindenkor legígéretebb csomópont kibontásának stratégiája biztosítja számunkra azt, hogy akár rábízhatjuk magunkat az időkorlátos keresés módszerére is.



## Konklúzió

A dolgozatban olyan új erőforrás ütemezési modellek kerültek bemutatásra, melyek egyrészt globális szemléleten alapulnak, másrészt képesek kezelni a gyenge illetve erős kategóriákba tartozó erőforrásfajtákat is. A kisméretű feladatokra felírt MILP modell relaxált változatai felhasználhatóak a közepes méretű feladatok esetére kialakított implicit leszámítási algoritmusok felgyorsítására, hiszen megfelelő alsó becsléseket lehet velük készíteni. A kis és közepes méretű feladatok megoldására közölt algoritmusok egzakta, mert az összes optimális ütemezést képesek előállítani.

A dolgozatban szerepelnek olyan apróbb eredmények is, amelyek elméleti szempontból talán nem, de gyakorlati szempontból érdekesek lehetnek. Ezeket szeretném a következőkben röviden ismertetni:

A negyedik fejezetben említésre kerül, de különösebb hangsúlyt nem kap az a tény, miszerint a modelleink és módszereink változtatás nélkül képesek kezelni a puffert és tartalék idő problémáját is. Inkább érdekes megemlíteni azt is, hogy modelleink akkor is változtatás nélkül alkalmazhatóak, ha a projekt időtartama alatt az erőforrások rendelkezésre állására vonatkozó konstans megkötést feloldjuk. A több projektből alkotott úgynevezett superprojekt pedig lehetőséget ad arra, hogy több projektet is átfogó mérföldköveket lehessen kijelölni.

A projektmenedzsment ütemezési feladatai azonban nem érnek véget azzal, hogy a projekt megkezdése előtt elkészül a kiindulási hálótér – a bázis terv. Az esetlegesen előforduló határidő-csúszások miatt, vagy egyszerűen az idő előre haladtával a bázis tervet időről-időre felül kell vizsgálni. Szerencsére a dolgozatban bemutatott megoldási módszerek mindegyike könnyen módosítható úgy, hogy az ilyen menetközbeni ütemezés-finomításoknak is eleget tegyen. A kisméretű feladatokra kialakított MILP módszernél ehhez mindössze bizonyos korlátok egyenlőtlenség jeleit kell egyenlőség jelre cserélni. Az implicit leszámítási modell esetében pedig az idő során megváltozott helyzetet könnyen leírhatjuk új, addicionális közvetlen megelőzési halmazok, valamint tevékenységkezdési rögzítési halmazok segítségével. Ezt a lehetőséget a heurisztikákat tárgyaló fejezetben, a projektvezetői preferencia beépítésekor illusztráltuk is. Az így kapott feladatok megoldása általában még gyorsabb is lesz, hiszen a feladat mérete az idő előre haladtával exponenciálisan csökken. Az eredeti modellben elvégzett számítások számos részlete a finomított ütemezésnél is újra hasznosítható, hiszen a keresőfák közös részei változatlanok maradnak.

A dolgozatban közölt eredmények alkalmasak arra, hogy implementálásukkal olyan „plugin” eszköz készülhessen, amely sikeresen beépíthető a különféle projektmenedzsment szoftverekbe.

Dolgozatomban megmutattam, hogy a kutatás alapjául szolgáló újfajta globális szemlélet alkalmazásával, valamint az erőforrások újszerű kategorizálásának bevezetésével használható modelleket lehet készíteni. Fontos kérdés maradt azonban továbbra is az, hogy a hagyományosan lokális szemléletű eredményeket miként lehet átültetni ebbe az új rendszerbe. A projekt sikerességének hármas (határidőn illetve költségvetésen belüli, és megfelelő minőségű) kritérium rendszeréből kizárólag az elsőre koncentráltunk. Biztató eredmények vannak azonban arra nézve, hogy a modell jól alkalmazható a pénzügyi tervezés során is.

Amennyiben ismertek egy adott projekt típus jellegzetességei, akkor az ebből fakadó specialitások kihasználásának érdekében érdemes feláldozni a modellek általános jellegét. A dolgozat címében szereplő szoftverfejlesztés területe számos ilyen lehetőséget biztosít. Az ilyen projektek esetében például kiemelt jelentőségű az, hogy az egyes tevékenységek időtartamának pontos becslése nehézségekbe ütközik. Természetes módon merül fel tehát az igény, hogy a determinisztikus modelleink analógiájára kialakítsuk a tevékenységi időkre vonatkozó sztochasztikus illetve fuzzy modelleket is.

## **Publikációk**

### **A disszertációhoz kapcsolódó publikációk jegyzéke**

**Sramó-Kruzslicz:**

Rendszerfejlesztés CASE eszközökkel. Egyetemi jegyzet 1996 (100 oldal)

**Kruzslicz:**

New global resource leveling measures (2002) Working paper

**Csébfalvi-Kruzslicz:**

A New Resource Allocation Model with Hard and Soft Resource Constraints. MOPTA Conference (2002) Hamilton

**Kruzslicz:**

Szoftverfejlesztési tevékenységek egzakt ütemezése erős és gyenge erőforrás korlátok mellett. Szigma (2002) (közlésre elfogadva)

**Kruzslicz:**

A New Exact Resource Allocation Algorithm with Hard and Soft Resource Constraints. Operations Research (2002) Klagenfurt (Operations Research Proceedings 2002, p.223-228)

**Kruzslicz:**

Szoftverfejlesztési projektek ütemezésének optimalizálása ("Töltsük meg tudással" konferencia Pécs PAB-KTK 2003)

**Kruzslicz:**

Projektütemezés erős és gyenge erőforrás korlátok mellett. (2003) Szigma (megjelenés alatt)

**Csébfalvi-Kruzslicz:**

Project scheduling with hard and soft resource constraints (2003) European Journal of Operational Research (referálás alatt)



