MÁTRIX-ALAPÚ, STRATÉGIAI PROJEKTTERVEZÉSI ELJÁRÁSOK[1]

KOSZTYÁN ZSOLT TIBOR
Pannon Egyetem, Kvantitatív Módszerek Tanszék

A hálós projekttervezési eljárások mellett, főleg stratégiai döntések támogatására mátrix-alapú projekttervezési módszereket is használhatunk. Ekkor lehetőségünk van az egyes tevékenységek végrehajtásának fontosságát vagy éppen megvalósítási valószínűségét is jellemezni. Ezáltal lehetőség van egy-egy projektváltozat megvalósítási valószínűségét is meghatározni, illetve a projektterveket fontosságuk alapján sorrendbe rakni. Itt a legnagyobb kihívást a lehetséges projektváltozatok, illetve projektstruktúrák gyors meghatározása, különböző szempontok szerinti sorrendbe rakása jelenti. Ez a tanulmány e probléma vizsgálatával foglalkozik. Gyors eljárásokat javaslunk, mellyel sorrendbe rakhatók a lehetséges projektváltozatok megvalósítási fontosság, idő, költség- vagy éppen erőforrás-igényeik szerint. A menedzsment kiválaszthatja, hogy egy adott költség-, idő- és erőforrás-igény esetén mely projektek valósíthatók meg. Egy ilyen eljárás alapja lehet egy projektszakértői rendszernek, amely a projektmenedzsment stratégiai döntéseit segítheti. A tanulmány végén a módszereket gyakorlati példán is teszteljük.

Kulcsszavak: projekttervezési módszerek, stratégiai döntéstámogatás, szakértői rendszerek.

1 Bevezetés

A projekttervezés során leggyakrabban alkalmazott hálótervezési módszerek [1,2,3], Gantt-diagramok [4], ciklogramok [5] elsősorban a tervezési folyamat operatív fázisát segítik [6]. Egy projektterv esetén képesek vagyunk e módszerekkel és az ezekre épülő költség- és erőforrás-tervezési eljárásokkal [7] meghatározni a projekt időtartamát, költségigényét, erőforrás-szükségletét. Azonban olyan kérdések megválaszolásában, ahol döntenünk kell, mely tevékenységet, részprojektet valósítjuk meg, milyen sorrendben végezzük el a tevékenységeket, ha arra több mód is kínálkozik, már nem adnak segítséget ezek az eljárások. Ebben az esetben mátrixos projekttervezési módszereket [8,9] alkalmazhatunk. Ekkor azonban a legnagyobb problémát a lehetséges projekttervek meghatározása illetve azok valamilyen szempont szerinti rendezése, rangsorolása jelenti. Mindezidáig a legtöbb problémára nem létezett gyors egzakt megoldás. A most következő tanulmányban olyan módszereket mutatok be, melyekkel lehetőség van a legrövidebb, legkevésbé költséges, vagy

[1]Beérkezett: 2013. március 20. E-mail: kzst@gtk.uni-pannon.hu.

éppen a legfontosabb projektváltozat kiválasztására. A módszer alkalmazása során figyelembe vesszük a felállított költség-, idő- és erőforráskorlátokat is.

2 Irodalmi áttekintés – mátrixos és hálós tervezési eljárások összehasonlítása

A hálótervezési módszerek mellett elsősorban termékfejlesztési [10] és szoftverfejlesztési [11] projektek kezelésénél egy másik eljárás-család is előtérbe került, amelyeket összefoglalóan mátrixos tervezési eljárásoknak nevezünk.

Mivel a projekthálókat reprezentáló gráfokhoz megadhatók a gráfok szomszédsági mátrixa, első ránézésre azt gondolhatnánk, hogy a tekintett eljárások csak a megjelenítésben különböznek a hálótervezési eljárásoktól. Ugyanakkor a megjelenítésen kívül a két módszertant az is megkülönbözteti, hogy a kifejlesztett algoritmusok a hálón vagy a mátrixon végzik-e a műveleteket.

A mátrixos megjelenítés esetén a sorokban és az oszlopokban a tevékenységek, a cellákban pedig a tevékenységek közötti kapcsolatok jelennek meg. A mátrixos módszerek első változataiban, melyet Design Structure Matrix-nak, illetve Dependency Structure Matrix-nak (vagy röviden csak DSM-nek) neveztek [10,12] még a tevékenységek közötti kapcsolatokat szigorú rákövetkezési relációnak tekintették. Ugyanakkor az eljárások újra kiemelten foglalkoztak a körfolyamatok figyelembevételével, amik fokozatosan kikerültek a hagyományos projekttervezési módszertanok látóköréből. A DSM-módszer legegyszerűbb változata is már 3 alapkapcsolatot kezel a tevékenységek között. Ezek a soros kapcsolatok, párhuzamos kapcsolatok, illetve az iteratív kapcsolatok. Bár az iteratív kapcsolatokkal, körfolyamatokkal már egy 1966-ban készült tanulmányban [13] foglalkoztak, ezek az eljárások a projektmenedzsment eszköztárába mégsem épültek be szervesen, pedig mind a termékfejlesztési, mind a karbantartási, mind pedig a szoftverfejlesztési projektek esetén időről-időre vissza kell térni egy korábbi fázisra.

Ezzel szemben a mátrixalapú megközelítés az iteratív kapcsolatok és a körök kezelésére különös figyelmet szentelt; mind a körök megtalálására, mind az esetleges feloldásukra külön tanulmányok készültek (lásd pl. Xiao et. al, 2007 [14] partícionáló algoritmusát, amelyet e tanulmány is alkalmaz). Mátrix-alapú módszereket alkalmaztak ütemezésre [15,16], valamint erőforrás-korlátos projektütemezési problémák megoldására is [17]. A hagyományos mátrix-alapú módszerek hátránya azonban, hogy a Steward-féle bináris DSM csak szigorú megelőzési kapcsolatokat kezel (egy tevékenység vagy függ, vagy nem függ más tevékenységtől), nem nyújt további információt az interakció/kölcsönhatás/kapcsolat természetéről.

Későbbi kutatások [18,19] azonban rámutattak, hogy a mátrixban nem csak biztos (determinisztikus) kapcsolatok jelölésére van mód, lehetőség van a kapcsolaterősség mértékének jelölésére is. Ezt a módszert *numerikus DSM-módszernek* (NDSM) nevezik, az „X”-ek helyett számokat, vagy csak a bizonytalanság jelölése esetén „?”-jelet írunk. Az NDSM-módszer alkalmazása során megjeleníthetjük két tevékenység közötti függőség fokát. Ez lehetővé teszi például egy visszacsatolási hurok valószínűségének megjelenítését, ezáltal prioritások képezhetők a fontos iterációk között a folyamat tervezésében (dsmweb.org). Ez a leírás tulajdonképpen kapcsolati szinten kezeli a rákövetkezési relációk közötti bizonytalanságot. Hogyan lehet a kapcsolatok bizonytalanságát felderíteni? E problémával is számos tanulmány foglalkozott: a tevékenységek függőségi viszonyát meghatározhatják korábbi projekttapasztalatok [19], de akár szakértői vélemények is [20]. Ezekkel a területekkel részletesebben a következő fejezetekben is foglalkozni fogok.

Informatikai és innovációs projektek esetén nagyon gyakran előfordul, hogy egy tevékenységet akár egymás után sorosan, vagy akár párhuzamosan is végre lehet hajtani. E probléma modellezésére alkalmas a szintén mátrixalapú eljárás, melyet sztochasztikus hálótervezési eljárásnak (angolul: Stochastic Network Planning Method, rövidítve: SNPM) neveztek el [9], utalva arra, hogy eredményül több projekthálót is kaphatunk.

A két elgondolás között az a legfőbb különbség, hogy az SNPM segítségével meg is határozzuk a lehetséges projektstruktúrákat. Ha pl. két tevékenység sorosan és párhuzamosan is végrehajtható, akkor két lehetséges projektstruktúrát kell tekintenünk (az egyik esetben a tevékenységek sorosan, a másikban párhuzamosan hajtódnak végre). A bizonytalan kapcsolatok megvalósításához fontossági, pont- vagy szkórértéket rendelhetünk, illetve a bekövetkezésükhöz valószínűségi érték rendelhető. A bizonytalan kapcsolatok pontértékeiből azután a projektstruktúrákra vonatkozó pontértékek is számíthatók [9]. A módszer segítségével egy sztochasztikus hálótervből, amelyben tevékenységek között bizonytalan kapcsolatok is szerepelhetnek, determinisztikus projekttervek adhatók meg, amelyeket már hagyományos (mátrixos és/vagy hálós) projekttervezési eljárásokkal kezelhetünk, ütemezhetünk, allokálhatunk.

Már az NDSM-módszernél is utaltak arra, hogy a tevékenységek közötti függőségi fokokat valamiféleképpen osztályozzák [19]. A numerikus DSM értékei a diagonálison kívüli cellákban többek között a tevékenységek közötti függőségek relatív fontosságát is reprezentálhatják [18,21]. (Üres cella értéke nulla, amely azt mutatja, hogy a tevékenységek között nincsen függőség.) Ebben a modellben a diagonális értékek a tevékenység elvégzésének idejét mutathatják.

Az SNPM-módszerben is 0-val vagy üres cellával jelölhetjük, ha két tevékenység között *nincs* függőség. 1-essel, ha két tevékenység között *biztos* rákövetkezési reláció van. Ha két tevékenység között a kapcsolat erőssége 0 és 1 között van, akkor azt mondjuk, hogy a tevékenységek között *bizonytalan* kapcsolat áll fent.

Valószínűségként akkor kezelhetjük a tevékenységek közötti kapcsolatot, ha a lehetséges (korábban már hasonló projekteknél megvalósított) technológiai sorrendekre vonatkozóan rendelkezésünkre áll valamilyen a priori információ (ebben az esetben objektív valószínűségekről beszélhetünk), illetve esetlegesen több szakértői vélemény alapján alakítottuk ki a lehetséges technológiai kapcsolatokat (ebben az esetben szubjektív valószínűségekkel dolgozunk).

Ha nem a valószínűségi leírás mellett döntünk, akkor is beszélhetünk a tevékenységek közötti kapcsolat függőségi fokáról, amelyet a *kapcsolat erősségének* is nevezünk. **A** és **B** tevékenység közötti függőségi fok megmutatja, hogy mennyire fontos, hogy **A** tevékenységet **B** kövesse.

A kapcsolat erőssége és/vagy valószínűsége 0 és 1 között bármilyen értéket felvehet. Bár a bemutatandó módszer szempontjából hasonlóképpen kezelhetők azok a mátrixok, amelyeknél a diagonálison kívüli 0 és 1 közötti számok a kapcsolatok valószínűségét, illetve a kapcsolat erősségét jelölik, mivel tartalmilag mást jelentenek, megkülönböztetjük ezt a két esetet egymástól és másképpen is jelöljük majd őket.

Az SNPM-módszer továbbfejlesztett változatában, mely a *projekt szakértői mátrix* (angolul: Project Expert Matrix, rövidítve: PEM) nevet kapta [11,22], már nem csak a tevékenységek közötti kapcsolatok lehetnek bizonytalanok, sztochasztikusak, hanem a projektben végrehajtandó tevékenységek előfordulása is.

Ekkor is meghatározhatók a lehetséges projekttervek, csak ekkor két lépésben (lásd részletesen: [11]): Első lépésben meg kell határozni, hogy mely tevékenységeket hajtjuk végre. Ezek lesznek az ún. *projektváltozatok*, ahol a tevékenységek megvalósításának bizonytalansága azt jelenti, hogy dönthet úgy a projektvezetés, hogy a tevékenységet megvalósítja, illetve úgy is, hogy a tevékenységeket (kapcsolataikkal együtt) elhagyja a projekttervből. A projektváltozatok meghatározása után második lépésként SNPM-módszer segítségével már meghatározhatók a lehetséges projekttervek. A két lépés után determinisztikus projektterveket kapunk, amelyek már hagyományos projekttervező eszközökkel kezelhetők. A két lépés lehetne egy szakértői modul része, amennyiben létezne egy gyors algoritmus, amely egy ilyen szakértői mátrixból meghatározza a peremfeltételeknek megfelelő (adott idő-, költség- és erőforrás-igényeket nem túllépő) projekttervek közül valamilyen szempont – pl. bekövetkezési valószínűség/megvalósítási fontosság, átfutási idő, költség- és erőforrásigények – szerinti sorrendben legjobbnak értékelt projektváltozatokat. E tanulmány pontosan ilyen keretalgoritmust mutat be.

Nagyszámú projektváltozat és projektstruktúra esetén hagyományos hálótervezési eljárásokkal a fenti probléma nehezen kezelhető, hiszen az ábrázoláshoz vagy meg kell határozni az összes lehetséges változatot, melyeket megfelelő operátorokkal kell összekapcsolni (lásd: GERT=(Graphical Evaluation and Review Technique) módszer [13]), vagy be kell érnünk a logikai tervezéssel, ahol szaggatott vonallal tudjuk jelezni a tevékenységek bizonytalanságát (PEG=Project Expert Graph [11]), illetve a tevékenységek közötti bizonytalan kapcsolatokat, de ütemezni nem tudunk a hálón.

Ugyanakkor mátrixos módszerekkel is kihívást jelentett a lehetséges projektváltozatok és az ezen belül lehetséges projektstruktúrák nagy számának kezelése. A korábban bemutatott PEM- [24] és SNPM-módszerek [9] csak az összes lehetséges projektváltozatot, illetve projektstruktúrát adták meg, mely sok bizonytalan tevékenység, illetve kapcsolat esetén nagyon sok lehetséges projekttervet eredményez. Tegyük fel például, hogy t darab bizonytalan tevékenység van a projekt szakértői mátrixban. Attól függően, hogy e tevékenységeket megvalósítjuk, vagy elhagyjuk, összesen 2^t lehetséges projektváltozat létezik. Ami már önmagában is nagy szám, ugyanakkor a lehetséges projektváltozatokon belül még projektstruktúrákat is tekintenünk kell, amely k bizonytalan kapcsolat esetén szintén 2^k lehetséges projektstruktúrát eredményez. Sokáig reménytelennek tűnt, hogy a nagyszámú projektváltozatokat, projektstruktúrákat adott szempontok szerint (idő-, költség-, erőforrásszükséglet, fontosság stb.) sorrendbe tudjuk rakni. Ezért legtöbb esetben genetikus algoritmusok segítségével lehetett adott célfüggvénynek megfelelő megoldásokat keresni [23,24,25].

Egyetlen esetben tudtunk eddig egzakt, mohó algoritmust adni a fenti problémára, nevezetesen, amikor a cél a legvalószínűbb projektterv megtalálása volt. A [11] tanulmányban bemutatott APS (Agile Project Scheduling) módszer azonban nem volt alkalmazható más célfüggvények (pl. lehető legkisebb átfutási idő, legkisebb projektköltség, legkisebb erőforrás-igény stb.) alkalmazása esetén. Adós maradt továbbá a körfolyamatok kezelésével. Ebben a tanulmányban egy keretalgoritmust olvashatunk, ahol többfajta célfüggvény (pl. legvalószínűbb/legfontosabb, legkisebb költségű/átfutási idejű/erőforrás-igényű projektterv megtalálása) és korlátozó feltétel (pl. idő/költség/erőforrás-korlát) is megadható. Kezelhetők és feloldhatók a körfolyamatok. Így az itt bemutatott módszerek alapját képezhetik egy projektszakértői rendszernek.

3 Lehetséges projekttervek rendezése

Mielőtt kiválasztanánk egy adott célfüggvény szerinti legjobb megoldást, a lehetséges projektváltozatokat, illetve projektstruktúrákat megvalósítási valószínűségük, fontosságuk, átfutási idejük, költség- és erőforrás-igényeik alapján pontoznunk kell. Ezeket az értékeket fogjuk a továbbiakban együttesen szkóroknak, vagy pontértékeknek nevezni. A következő alfejezetben azt fogom bemutatni, hogy hogyan lehet ezeket az értékeket kiszámítani, majd ezután megmutatom, hogy a pontértékeket felhasználva hogyan lehet megtalálni egy adott célfüggvényre nézve, adott korlátokat nem túllépő legjobb megoldást.

3.1 Projektváltozatok, projektstruktúrák jellemzése

A pontértékek megállapításánál kulcsfontosságú lesz a bizonytalan tevékenység-előfordulások, illetve a bizonytalan kapcsolatok kezelése. Négyfajta pontértéket fogunk tekinteni. Ezekkel fogjuk tudni jellemezni a

1. Projektváltozat/projektstruktúra bizonytalanságát

    a. Előfordulási valószínűségét
    b. Megvalósítási fontosságát

2. Projektváltozat/projektstruktúra időigényét

3. Projektváltozat/projektstruktúra erőforrás-igényeit
4. Projektváltozat költségigényét

3.1.1 Bizonytalanság kezelése

Ahogyan azt korábban láthattuk, a projektszakértői mátrixban bizonytalanként tekinthetjük akár a tevékenységek előfordulását, akár a tevékenységek kapcsolatát is. A bizonytalanságot jellemezhetjük korábbi tapasztalatok alapján valószínűségekkel, de akár megvalósítási fontosságokkal is. Először feltételezzük, hogy korábbi tapasztalatok alapján ezek a bizonytalanságok valószínűségekként jellemezhetők. Ekkor feltételezhető, hogy egy i-edik tevékenység megvalósítási valószínűsége P_i-vel jellemezhető. Ekkor annak valószínűsége, hogy i-edik tevékenység nem valósul meg, $1-P_i$. (Itt az egységesség érdekében a projektváltozatokra vonatkozó értékeket mindig nagy, míg a projektstruktúrákra vonatkozó értékeket (pl. valószínűség, fontosság, átfutási idő stb.) mindig kis betűvel jelöljük.) Ha feltételezzük, hogy a tevékenységek megvalósulási valószínűsége egymástól független, akkor egy t tevékenységet tartalmazó projektváltozat megvalósulási valószínűsége: $P = \prod_{i=1}^{t} T_i$, ahol $T_i = P_i$, ha i-edik tevékenységet megvalósítjuk, és $T_i = Q_i = 1-P_i$, ha i-edik tevékenységet elhagyjuk a projektből. Ugyanígy meghatározhatók a projektstruktúrák megvalósulási valószínűsége is. Ha ugyanis egy $i \to j$ tevékenység közötti reláció valószínűségét $p_{i\to j}$-vel jellemezzük, akkor $1 - p_{i\to j}$-vel jellemezhetjük annak valószínűségét, hogy a két tevékenység között nincs kapcsolat. A két tevékenység esetén ez azt jelenti, hogy a két tevékenység sorosan, vagy párhuzamosan hajtódik végre.

A lehetséges projektstruktúrák birtokában a menedzsment dönthet arról, hogy a tevékenységeket sorosan vagy párhuzamosan valósítsa meg. Soros végrehajtás esetén kevesebb lesz az átlagos erőforrásigényünk, viszont a projekt időtartama hosszabb lesz, mint a párhuzamos végrehajtás esetén.

A kapcsolatokat ezek alapján két csoportra oszthatjuk: biztos (determinisztikus, kötelező) kapcsolatokra, illetve bizonytalan (sztochasztikus, nem kötelező) kapcsolatokra.

Le kell azonban szögezni, hogy a mátrixos tervezés során nem csupán egy másikfajta ábrázolásról van szó. A kapcsolatoknak értéket adva a lehetséges projektstruktúrák sorrendbe rakhatók. Tegyük fel, hogy ebben a példában a kapcsolatokhoz rendelt értékek valószínűségek lesznek. Ekkor teljesül, hogy annak a valószínűsége, hogy **A** tevékenységet **B** követ, $p_{A\to B}$, akkor annak a valószínűsége, hogy **A** és **B** között nincs rákövetkezési reláció, $1 - p_{A\to B}$.

Ha a relációkhoz rendelhető valószínűség, akkor meghatározható a projektstruktúra valószínűsége is, amely a bizonytalan relációk megvalósításának és meg nem valósításának szorzata lesz.

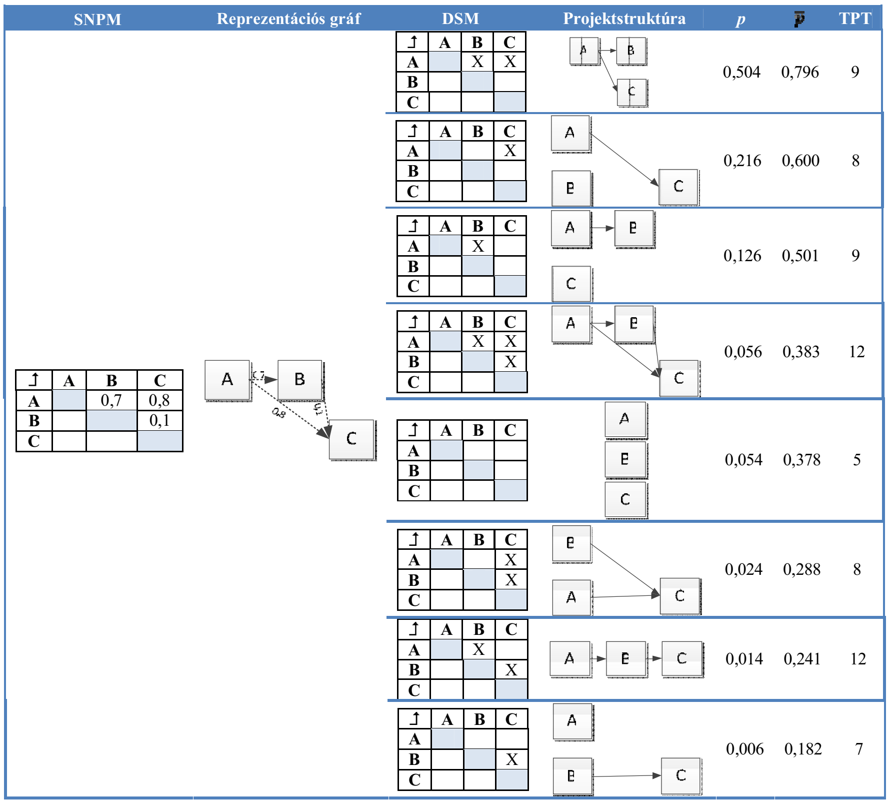
Ha a kapcsolat erőssége nem valószínűséget, hanem a kapcsolat, mint előírás fontosságát jelöli, akkor a projektstruktúra fontosságának, mint a kapcsolaterősségek fontosságainak összegét lehet tekinteni.

Mivel nem mindegy, hogy hány kapcsolaterősséget adunk/szorzunk össze, így értelmezhető a projektstruktúra átlagos pontértéke, illetve korrigált átlagos pontértéke. A projektstruktúra átlagos pontértéke

- kiszámítható a kapcsolatok pontértékeinek *geometriai átlag*ából, ha a kapcsolatok figyelembe vételét/elhagyását jellemző pontértékek valószínűségekként értelmezhetők (jelölése: $\overline{p}$);

- kiszámítható a kapcsolatok pontértékeinek *számtani átlag*ából, ha a kapcsolatok figyelembe vételét/elhagyását jellemző pontértékek fontosságként értelmezhetők (jelölése: $\overline{p}$).

Tekintsük az alábbi példát (*1. táblázat*), ahol **A**, **B** és **C** tevékenységet valósíthatunk meg. Legyen **A** tevékenység időtartama 5 hónap, **B** tevékenységé 4 hónap, míg **C** tevékenységé 3 hónap. Tegyük fel, hogy a kapcsolatok közötti pontértékek valószínűségekként értelmezhetőek. Számítsuk ki a projektstruktúra előfordulási valószínűségét (p) és átlagos pontértékét ($\overline{p}$)!



SNPM:

$\begin{array}{c|ccc} & A & B & C \\ \hline A & & 0{,}7 & 0{,}8 \\ B & & & 0{,}1 \\ C & & & \end{array}$

DSM	p	$\overline{p}$	TPT
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & X & X \\ B & & & \\ C & & & \end{array}$	0,504	0,796	9
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & & X \\ B & & & \\ C & & & \end{array}$	0,216	0,600	8
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & X & \\ B & & & \\ C & & & \end{array}$	0,126	0,501	9
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & X & X \\ B & & & X \\ C & & & \end{array}$	0,056	0,383	12
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & & \\ B & & & \\ C & & & \end{array}$	0,054	0,378	5
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & & X \\ B & & & X \\ C & & & \end{array}$	0,024	0,288	8
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & X & \\ B & & & X \\ C & & & \end{array}$	0,014	0,241	12
$\begin{array}{c\|ccc} & A & B & C \\ \hline A & & & \\ B & & & X \\ C & & & \end{array}$	0,006	0,182	7

1. táblázat. 3 bizonytalan kapcsolatot tartalmazó SNPM-mátrix kiértékelése

A projektstruktúra valószínűsége a bizonytalan kapcsolatok valószínűségeinek szorzata lesz. Az első projektstruktúra esetén ez az érték:

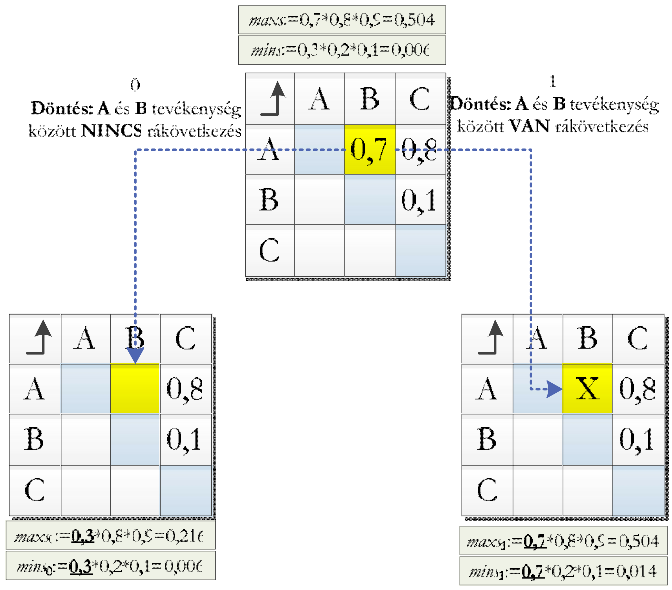
$$p_1 = p_{A\to B} * p_{A\to C} * (1 - p_{B\to C}) = 0{,}7 * 0{,}8 * (1 - 0{,}1) = 0{,}504\ ,$$

hiszen **A** és **B**, valamint **A** és **C** tevékenységek közötti relációk valószínűsége 0,7, illetve 0,8; **B** és **C** tevékenység közötti reláció elhagyásának valószínűsége pedig $1 - 0{,}1$. Kiszámítva valamennyi projektstruktúra valószínűségét és ezeket összeadva 1-et kapunk, hiszen az összes lehetséges projektstruktúra teljes eseményrendszert alkot. (Az egyes projektstruktúrák egymást kizárják, és valószínűségük összege a biztos esemény valószínűségét adja meg.)

A projektstruktúra átlagos pontértéke megegyezik a korrigált pontértékkel, hiszen ebben a példában minden kapcsolatot bizonytalannak tekintettünk.

Érdemes megjegyezni, hogy amennyiben valamennyi kapcsolat megvalósulásához (p) / elhagyásához ($q = 1 - p$) 0 és 1 közötti számokat rendelünk, valamint a projektstruktúra pontértéke jellemezhető a bizonytalan kapcsolatok pontértékeinek szorzataként, akkor a fentiek szerint meghatározott projektstruktúra értékek összege 1-et fog adni még akkor is, ha ezek a számok nem értelmezhetők valószínűségekként.

A bizonytalan kapcsolatok, illetve bizonytalan tevékenység-előfordulások felhasználásával a projektszakértői mátrix kiértékelése során felső/alsó becslést lehet adni a projektváltozatokhoz, projektstruktúrákhoz tartozó pontértékekhez. Ehhez nem kell mást tennünk, mint a döntésünk eredményeként (mely szerint vagy végrehajtunk, vagy elhagyunk egy tevékenységet, illetve vagy figyelembe veszünk, vagy elhagyunk két tevékenység közötti kapcsolatot) kapható pontérték közül mindig a maximálisat/minimálisat tekintjük, akkor az így képzett pontérték éles felső, illetve alsó becslése lesz a megvalósítható projektváltozatok/projektstruktúrák pontértékeinek. Az éles szó itt azt jelenti, hogy a felső és alsó becsléshez tartozó pontérték konkrét projektváltozathoz kötődik. Példaként tekintsük az 1. táblázatban szereplő **SNPM** mátrixot. Legyen $\mathbf{p} := \mathbf{SNPM}$; $\mathbf{q} := \mathbf{1} - \mathbf{SNPM}$ 3×3-as mátrixok. Ekkor a projektstruktúra pontértékének (ami jelen esetben a projektstruktúra megvalósulási valószínűsége, ha a cellákban szereplő értékek a kapcsolatok megvalósulási valószínűségét jelölik és ezek az értékek függetlenek egymástól) maximális értéke: $maxs = 0{,}7 * 0{,}8 * 0{,}9 = 0{,}504$, amihez az 1. táblázatban szereplő első projektstruktúra tartozik; míg a projektstruktúrákhoz tartozó minimális pontérték: $mins = 0{,}3{*}0{,}2{*}0{,}1 = 0{,}006$, ami pont az 1. táblázatban szereplő utolsó projektstruktúra lesz. Ez a felismerés egy kiértékelés során kulcsfontosságú lesz, hiszen ha pl. egy kapcsolat előírásáról, vagy elhagyásáról döntünk, akkor azt a pontértéket fixnek tekintve, a még bizonytalan kapcsolatokat figyelembe véve, számolható egy elérhető maximális és egy minimális pontérték.



1. ábra. Lehetséges projektstruktúrák meghatározása (a kiértékelt bizonytalan kapcsolatot, ha elhagyjuk, üres cellával, ha előírjuk, ,,X"-szel jelöljük)

A kapcsolatok előírásáról, elhagyásáról hozott döntések az *1. ábra* szerint döntési fába rendezhetők. Ez a döntési fa bináris. Pontosan két döntést hozhatunk egy kapcsolatról: vagy előírjuk a kapcsolatot két tevékenység között, vagy elhagyjuk. A döntés után ezen már nem változtatunk, így a kapcsolati pontértékeket a projektstruktúrára vonatkozó pontértékek számításánál már fix értéknek tekintjük. A projektstruktúrára vonatkozó pontértékeket az alábbiak szerint számoljuk. Legyen adott egy $n \times n$-es $SNPM$-mátrix, melynek diagonálison kívüli elemei 0 és 1 közötti értékek lehetnek, ahol 1 a biztos kapcsolatot jelenti két tevékenység között. 0 jelenti, hogy két tevékenység között nincs rákövetkezési reláció. 0 és 1 közötti tetszőleges szám pedig azt jelenti, hogy két tevékenység közötti kapcsolat bizonytalan. Legyen $\mathbf{p}$, $\mathbf{q}$ szintén $n \times n$-es mátrixok, ahol $\mathbf{p}$ a kapcsolatok előírásának, $\mathbf{q}$ pedig elhagyásának pontértékét jelöli. (Ha a $\mathbf{p}$ a kapcsolatok előfordulásának valószínűségeit tartalmazó mátrix, akkor $\mathbf{q} = \mathbf{1} - \mathbf{p}$, ahol $\mathbf{1}$ is $n \times n$-es, 1-eseket tartalmazó mátrix.)

Ha a kapcsolati pontértékek valószínűségekként értelmezhetők:

$$maxscore(SNPM, \mathbf{p}, \mathbf{q}) = \prod_{\substack{i=1\ldots n, j=1\ldots n \\ p(i,j)\in(0,1)}} \begin{cases} \mathbf{p}(i,j), \text{ ha } SNPM(i,j) = 1 \\ \mathbf{q}(i,j), \text{ ha } SNPM(i,j) = 0 \\ \max\{\mathbf{p}(i,j), \mathbf{q}(i,j)\}, \text{ különben.} \end{cases} \tag{1}$$

Ha a kapcsolati pontértékek fontosságként értelmezhetők:

$$maxscore(SNPM, \mathbf{p}, \mathbf{q}) = \sum_{\substack{i=1\ldots n, j=1\ldots n\\ p(i,j)\in(0,1)}} \begin{cases} \mathbf{p}(i,j), \text{ ha } SNPM(i,j)=1\\ \mathbf{q}(i,j), \text{ ha } SNPM(i,j)=0\\ \max\{\mathbf{p}(i,j), \mathbf{q}(i,j)\}, \text{ különben.}\end{cases} \tag{2}$$

Ugyanígy az alsó becslést megadó minscore függvényt is megadhatjuk. Ha a kapcsolati pontértékek valószínűségekként értelmezhetők:

$$minscore(SNPM, \mathbf{p}, \mathbf{q}) = \prod_{\substack{i=1\ldots n, j=1\ldots n\\ p(i,j)\in(0,1)}} \begin{cases} \mathbf{p}(i,j), \text{ ha } SNPM(i,j)=1\\ \mathbf{q}(i,j), \text{ ha } SNPM(i,j)=0\\ \min\{\mathbf{p}(i,j), \mathbf{q}(i,j)\}, \text{ különben.}\end{cases} \tag{3}$$

Ha a kapcsolati pontértékek fontosságként értelmezhetők:

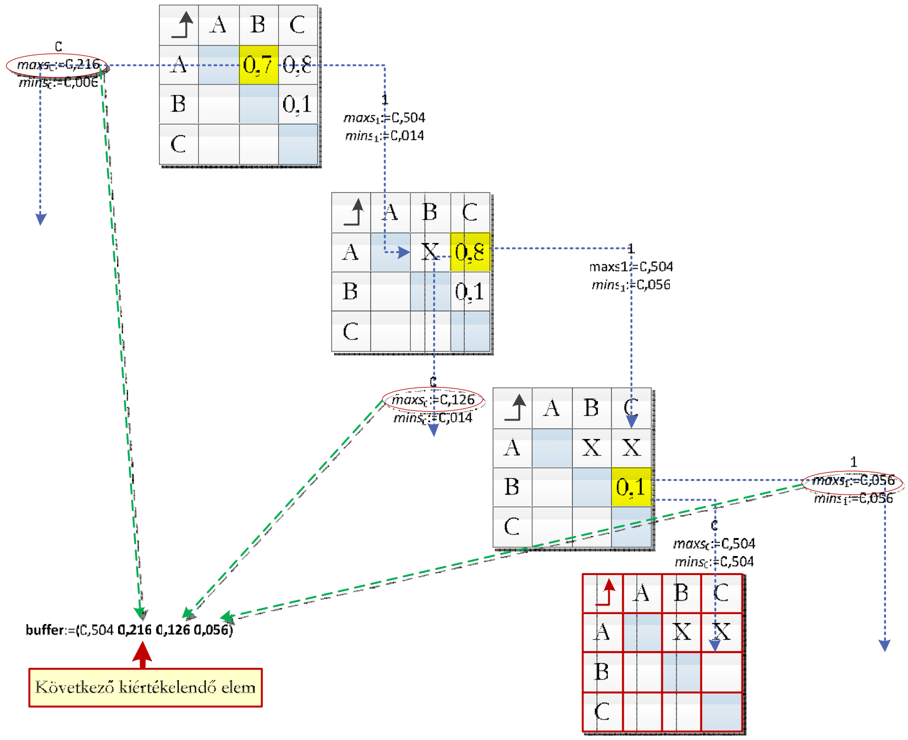
$$minscore(SNPM, \mathbf{p}, \mathbf{q}) = \sum_{\substack{i=1\ldots n, j=1\ldots n\\ p(i,j)\in(0,1)}} \begin{cases} \mathbf{p}(i,j), \text{ ha } SNPM(i,j)=1\\ \mathbf{q}(i,j), \text{ ha } SNPM(i,j)=0\\ \min\{\mathbf{p}(i,j), \mathbf{q}(i,j)\}, \text{ különben.}\end{cases} \tag{4}$$

Mivel a

$$\begin{aligned} mins &= minscore(\mathbf{SNPM}, \mathbf{p}, \mathbf{q}),\\ maxs &= maxscore(\mathbf{SNPM}, \mathbf{p}, \mathbf{q}) \end{aligned}$$

függvények (éles) alsó/felső becslését adják a projektstruktúrák pontértékeinek, így ezek felhasználhatók megfelelő korlátozó és szétválasztó algoritmusban. A kiértékelés során kapott (bináris) döntési fa tetején lévő $maxs$ érték lesz a lehető legnagyobb elérhető pontérték. A második legnagyobb érték legfeljebb a következő szinten jelenhet meg. Így a döntési fában a kiértékelés során mindig abba az irányba kell továbbhaladni, ahol ez a maximális pontérték szerepel. Így k bizonytalan kapcsolat esetén k lépésben meg lehet határozni a legnagyobb pontértékkel rendelkező projektstruktúrát (*2. ábra*). Közben a kiértékelés során a következő legnagyobb pontértékkel is találkozunk, amely legfeljebb a következő szinten szerepelhet.

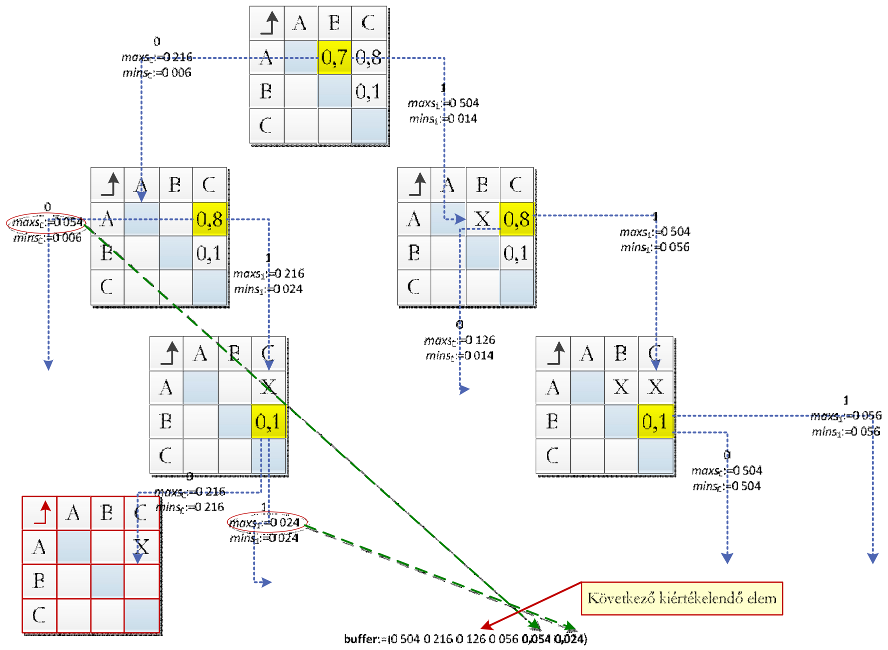
Ha elmentjük egy rendezett *buffer* halmazba a ki nem értékelt pontértéket, akkor bármilyen sorrendben is értékeljük ki a kapcsolatokat (bármilyen sorrendben is döntsünk róluk, hogy előírjuk-e vagy elhagyjuk-e őket), e halmaz következő eleme a következő legnagyobb pontértékkel rendelkező projektstruktúra lesz. Ennek oka a bináris döntési fa rendezettségében keresendő.



2. ábra. Legnagyobb pontértékkel rendelkező projektstruktúra megtalálása (a buffer-halmaz vastaggal szedett elemei a kiértékelés során újonnan bekerült elemek)

A következő pontértékkel rendelkező projektstruktúrát többféleképpen is meghatározhatjuk. Legegyszerűbb megoldás, hogy a buffer-halmazban eltárolt következő pontértékkel rendelkező elemet keressük. Felhasználjuk, hogy azt az ágat nem kell kiértékelnünk, ahol a *minscore* függvény értéke a keresendő érték felett, illetve a *maxscore* értéke a keresendő érték alatt van. A bejárásnál eltároljuk a ki nem értékelt ágak *maxscore* értékeit (*2-3. ábra*).

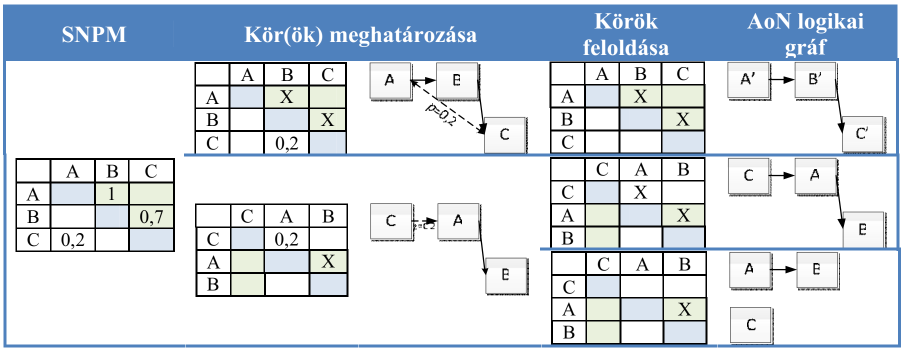
A kiértékelés gyorsítható, ha a bizonytalan függőségek, illetve projekt változatok esetén a bizonytalan tevékenység-előfordulások kiértékelését nem tetszőleges sorrendben végezzük, hanem azon kapcsolatok/tevékenységek kiértékelését végezzük el először, amelyeknél a kapcsolat (vagy tevékenység) megvalósításának (vagy elhagyásának) a pontértéke a legnagyobb.



3. ábra. Kiértékelés menete, ha a keresendő elem a második legnagyobb felvehető érték

Körök feloldása

Az előző példákban feltettük, hogy a projekt nem tartalmaz kört. Vagyis a mátrix felsőháromszög-mátrixba rendezhető. Azonban a lehetséges projektstruktúrák meghatározásánál ezt sehol nem használtuk ki. Az ütemezésnél is csak annyit kell kizárnunk, hogy 1 valószínűségű körfolyamat nem lehet a reprezentációs gráfban, illetve a projektstruktúrában. Éppen ezért először partícionáljuk az SNPM-mátrixot, majd a körfolyamatokat oldjuk fel úgy, hogy kiszámítjuk a visszacsatolás eredményeképpen a módosított átfutási időkkel a lehetséges projekttervek átfutási idejét. Lássunk erre egy példát. Legyen a partícionált SNPM-mátrixunk az alábbi *2. táblázatban* szereplő mátrix, ahol **A** tevékenység időtartama 5 hónap, **B** tevékenységé 4 hónap, míg **C** tevékenységé 3 hónap. Értékeljük ki először az előzőek alapján a lehetséges projektstruktúrákat úgy, hogy először csak a mátrix felső háromszög részét tekintjük. Tegyük fel, hogy a mátrixban szereplő értékek a kapcsolatok valószínűségét jelentik.



2. táblázat. Körök feloldása

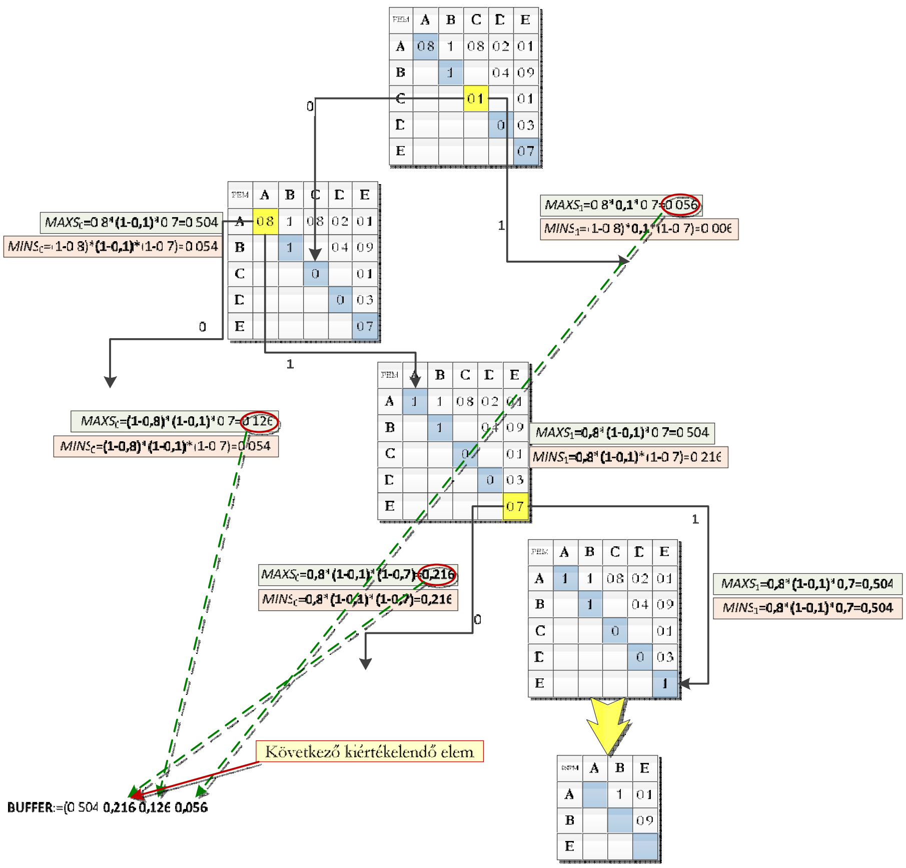
Ha a partícionált SNPM mátrix tartalmaz kört, akkor először meg kell határoznunk a lehetséges projektstruktúrákat a mátrix felsőháromszög értékeinek figyelembevételével. Ha a mátrix továbbra is tartalmaz kört, akkor a GERT-módszernél (Graphical Evaluation and Review Technique) [13] alkalmazott módon fel kell oldani a kört. Ekkor (ezt a projektstruktúrát '-vel jelölve) **A** tevékenység időtartama $d_{A'} = d_A/(1-p) = 5/(1-0{,}2) = 6{,}25$ hónap, $d_{B'} = 5$ hónap, míg $d_{C'} = 3{,}75$ hónap. Így ekkor a várható átfutási idő $TPT' = d_{A'} + d_{B'} + d_{C'} = 6{,}25 + 5 + 3{,}75 = 15$ hónap. Ez a projektstruktúra akkor teljesül, ha **B** és **C** tevékenység közötti kapcsolat is megvalósul, aminek a valószínűsége 0,7. Ha a **B** és a **C** tevékenység közötti kapcsolat elhagyható, akkor már nem lesz kör a hálózatban. Ebből adódóan meghatározható egy új $SNPM$-mátrix, mely esetében **C** és **A** tevékenység közötti kapcsolat valószínűsége 0,2 lesz. Ekkor a két lehetséges projektstruktúra megvalósulásának valószínűsége $(1-0{,}7) * 0{,}2 = 0{,}06$, illetve $(1-0{,}7) * (1-0{,}2) = 0{,}24$. Az átfutási idők $TPT_1 = d_C + d_A + d_B = 3 + 5 + 4 = 12$ hónap, míg a másik esetben $TPT_2 = \max(d_C, d_A + d_B) = \max(3, 5 + 4) = 9$. Ekkor is teljes eseményrendszert kapunk, hiszen $p' + p_1 + p_2 = 0{,}7 + 0{,}06 + 0{,}24 = 1$ lesz.

Ilyen módon tehát 2 lépésben a körök is kezelhetők, illetve feloldhatók a hálózatban.

Projektváltozatok meghatározása

Hasonlóan a projektstruktúrák pontértékeinek meghatározásához, a projektváltozatok pontértékei is megadhatók. A tevékenységek megvalósítási szintjén a pontérték jelenthet előfordulási valószínűséget, illetve a tevékenység megvalósításának fontosságát is. Ennek megfelelően a tevékenységek pontértékeiből származtatott projektváltozatokra vonatkozó pontérték jelenthet előfordulási valószínűséget, vagy egy, a tevékenységek megvalósításából származtatott megvalósítási prioritást.

A projektváltozatok meghatározása után az átlóban lévő üres cellák esetén az üres oszlopokat és üres sorokat elhagyjuk (*4. ábra*), így egy SNPM mátrixot kapunk, amit a fenti módon kiértékelhetünk.



4. ábra. Legnagyobb pontértékkel rendelkező projektváltozat meghatározása

A valószínűségek/fontosságok szerinti sorrend nem feltétlenül adja meg az időtartam minimalizálása szerinti sorrendet. Éppen ezért a kiértékelés során komplex célfüggvényeket határozhatunk meg, melyekben a valószínűségek/fontosságok mellett az időtartamok és erőforrás-igények is szerepet játszanak.

3.1.2 Átfutási idők, költség- és erőforrás-igények becslése

Időtervezés esetén (ha a követő tevékenység nem kezdődhet előbb, mint a megelőző), akkor a projekt átfutási idejének felső becslését az a projektstruktúra adja, amely esetén valamennyi bizonytalan kapcsolat megvalósul („soros végrehajtás"). Az átfutási idő alsó becslése pedig az a projektstruktúra lesz, ahol a tevékenységek párhuzamosan hajtódnak végre, vagyis a bizonytalan kapcsolatokat elhagyjuk a projekthálóból. Ekkor az átfutási idő becslésére vonatkozó $maxscoreTPT$ és $minscoreTPT$ értékek a következőképpen adhatók meg:

$$minscoreTPT := MPM(\lfloor \mathbf{SNPM} \rfloor, \mathbf{t}) ,$$
$$maxscoreTPT := MPM(\lceil \mathbf{SNPM} \rceil, \mathbf{t}) ,$$

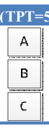
ahol $\lfloor\cdot\rfloor$ az alsó egészrészt, míg $\lceil\cdot\rceil$ a felső egészrészt jelölik, $\mathbf{t}$ pedig a tevékenységek időtartamait tartalmazó vektor. Az MPM függvény egy adjacencia mátrix segítségével leírt, kört nem tartalmazó tevékenység-csomópontú hálón számolja ki a $\mathbf{t}$ vektorban szereplő tevékenységidők alapján a projektstruktúra átfutási idejét.

Példaképpen tekintsük az előző feladatot. Itt $\mathbf{SNPM} = \begin{pmatrix} 0 & 0,7 & 0,8 \\ 0 & 0 & 0,1 \\ 0 & 0 & 0 \end{pmatrix}$.
Ekkor $\lfloor\mathbf{SNPM}\rfloor = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, ami a teljes párhuzamosítást, $\lceil\mathbf{SNPM}\rceil = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$, ami valamennyi kapcsolat megvalósítását jelenti. Mivel $d_A = 5$, $d_B = 4$, $d_C = 3$ hónap időtartamot igényel, ezért a maximális párhuzamosítás melletti átfutási idő 5 hónapot, míg valamennyi kapcsolat betartása esetén 12 hónapot igényelne a projekt megvalósítása.

Egy tevékenység-csomópontú hálótervezési, pl. MPM-módszerrel kiszámítva az átfutási időket, a legnagyobb értéket valamennyi kapcsolat figyelembevétele esetén kapjuk. Azonban még ekkor is elhagyhatók azok a kapcsolatok, amelyek nem kritikus úton vannak (*3. táblázat*).

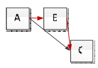
Párhuzamos végrehajtás (TPT=5)

	A	B	C
A			
B			
C			



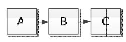
Valamennyi kapcsolat szerepeltetése (TPT=12)

	A	B	C
A		X	X
B			X
C			



Soros végrehajtás (TPT=12)

	A	B	C
A		X	
B			X
C			



3. táblázat. Nem kritikus úton lévő kapcsolatok elhagyása

Látható, hogy a projekt átfutási ideje nem változott, ugyanakkor a kapott hálóstruktúra egyszerűbb lett.

Amíg a maximális párhuzamosítás adja a legrövidebb időt, addig az is igaz, hogyha a tevékenységeket párhuzamosan hajtjuk végre, akkor igénylik a legtöbb erőforrást, míg a soros végrehajtás adja a legkevesebb átlagos erőforrásigényt. Az átlagos erőforrás-igényt itt sem befolyásolja, hogy a nem kritikus úton lévő bizonytalan kapcsolatokat megtartjuk vagy sem. Ugyanakkor minél kevesebb megkötésünk van, tehát minél kevesebb kapcsolatot tartalmaz a háló, annál könnyebb ezt az átlagos erőforrásigényt egy erőforrás-kiegyenlítési algoritmussal [7] elérni, vagy legalábbis megközelíteni.

Projektváltozatok költségbecslése

Ha nem a fontosság vagy a valószínűség alapján rangsoroljuk az egyes projektváltozatokat, hanem pl. a költségigényeik szerint, akkor elmondható, hogy azok a projektváltozatok tartalmazzák a legkevesebb költséget, ahol a lehető legkevesebb tevékenységet valósítjuk meg (a bizonytalan tevékenységeket elhagyjuk). Ugyanígy elmondható, hogy a legtöbb költséget az a projekt eredményezi, ahol valamennyi (bizonytalan) tevékenységet megvalósítjuk. Ezek alapján adható egy $MAXSCORETPC$ és egy $MINSCORETPC$ függvény az alábbiak szerint. Jelölje n tevékenység esetén $\mathbf{PEM}^{\wedge}$-mátrix azt az $n\times n$-es mátrixot, ahol a **PEM**-mátrix diagonálisában lévő értékek felső egész részét vesszük. Ha tehát $PEM(i,j)$ $(i,j := 1\ldots n)$ a **PEM**-mátrix eleme, akkor $i \neq$ esetén $PEM(i,j)$ $\mathbf{PEM}^{\wedge}$-mátrix eleme is, ugyanakkor $i = j$ esetén $\lceil PEM(i,i)\rceil$ eleme $\mathbf{PEM}^{\wedge}$-mátrixnak. Ugyanígy **PEM**-mátrix módosítását $\mathbf{PEM}^{\vee}$-mátrixszal jelöljük, ahol az eredeti **PEM**-mátrix diagonálisában lévő elemek alsó egész részét fogja tartalmazni az eredményül kapott $\mathbf{PEM}^{\vee}$-mátrix. Ezek alapján az előző példára:

$$\mathbf{PEM} = \begin{pmatrix} 0{,}8 & 1 & 0{,}8 & 0{,}2 & 0{,}1 \\ 0 & 1 & 0 & 0{,}4 & 0{,}9 \\ 0 & 0 & 0{,}1 & 0 & 0{,}1 \\ 0 & 0 & 0 & 0 & 0{,}3 \\ 0 & 0 & 0 & 0 & 0{,}7 \end{pmatrix},$$

$$\mathbf{PEM}^{\wedge} = \begin{pmatrix} 1 & 1 & 0{,}8 & 0{,}2 & 0{,}1 \\ 0 & 1 & 0 & 0{,}4 & 0{,}9 \\ 0 & 0 & 1 & 0 & 0{,}1 \\ 0 & 0 & 0 & 0 & 0{,}3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{PEM}^{\vee} = \begin{pmatrix} 0 & 1 & 0{,}8 & 0{,}2 & 0{,}1 \\ 0 & 1 & 0 & 0{,}4 & 0{,}9 \\ 0 & 0 & 0 & 0 & 0{,}1 \\ 0 & 0 & 0 & 0 & 0{,}3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Legyen $TPC(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS})$ a **PEM**-mátrixban szereplő biztos tevékenység-előfordulásokra vonatkozó költségigények összegét megadó függvény, ahol **COSTS** a tevékenységek költségigényét tartalmazó vektor. Ekkor

$$MAXSCORETPC(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS}) = \mathbf{TPC}(\mathbf{PEM}^{\wedge}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS}),$$

$$MINSCORETPC(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS}) = \mathbf{TPC}(\mathbf{PEM}^{\vee}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS}).$$

Projektváltozatok átfutási idejének és erőforrás-igényének becslése

Átfutási időt, átlagos erőforrás-igényt projektstruktúrák szintjén számolhatunk. Ugyanakkor az átfutási idő, valamint az átlagos erőforrásigény alsó és felső becslését már projektváltozatok szintjén is elvégezhetjük.

Az átfutási idő felső becslése az a projektstruktúra lesz, ahol valamennyi bizonytalan tevékenységet megvalósítunk és valamennyi bizonytalan kapcsolatot előírunk. Az átfutási idő minimuma (legkorábbi kezdésre való ütemezés esetén) az a projektstruktúra lesz, ahol valamennyi bizonytalan tevékenységet elhagyunk és valamennyi bizonytalan kapcsolatot feloldunk. Az átlagos erőforrás-igény felső becslését akkor kapjuk, ha egy **PEM**-mátrixban valamennyi (bizonytalan) tevékenységet megvalósítjuk, de valamennyi (bizonytalan) kapcsolatot feloldunk. A legkisebb átlagos erőforrás-igényt akkor kapjuk, ha valamennyi (bizonytalan) tevékenységet elhagyjuk, ugyanakkor valamennyi erőforrás-igényt előírjuk. Jelölje $\mathbf{T}$ a tevékenységek időtartamát tartalmazó n elemű vektort, $\mathbf{R}$ az erőforrás-igényeket tartalmazó $n \times r$-es mátrixot, ahol n a tevékenységek, r az erőforrások számát jelöli. Jelölje $\mathbf{P}$ $n \times n$-es mátrix a tevékenységek és a kapcsolatok megvalósításának fontosságát/valószínűségét, $\mathbf{Q}$ $n \times n$-es mátrix a tevékenységek és kapcsolatok elhagyásának valószínűségét/fontosságát. Jelölje továbbá $t := \mathit{synchronize\_t}(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{T})$ azt a függvényt, amely a **PEM**-mátrix diagonálisában lévő üres cellák alapján $\mathbf{T}$ vektorból törli a meg nem valósított tevékenységeket és adja vissza a $\mathbf{t}$ vektorba.

Ekkor legyen

$$\mathbf{t}_1 := \mathit{synchronize\_t}(\lceil \mathbf{PEM} \rceil, \mathbf{P}, \mathbf{Q}, \mathbf{T})\ ;$$
$$\mathbf{t}_0 := \mathit{synchronize\_t}(\lfloor \mathbf{PEM} \rfloor, \mathbf{P}, \mathbf{Q}, \mathbf{T})\,.$$

Tekintsük kiindulásképpen az alábbi **PEM** mátrixot. Legyen példánkban $\mathbf{P} := \mathbf{PEM}$, $\mathbf{Q} = \mathbf{1} - \mathbf{PEM}$, ahol $\mathbf{1}$ egy $n \times n$-es 1-eseket tartalmazó mátrix.

$$\mathbf{PEM} = \begin{pmatrix} 0{,}8 & 1 & 0{,}8 & 0{,}2 & 0{,}1 \\ 0 & 1 & 0 & 0{,}4 & 0{,}9 \\ 0 & 0 & 0{,}1 & 0 & 0{,}1 \\ 0 & 0 & 0 & 0 & 0{,}3 \\ 0 & 0 & 0 & 0 & 0{,}7 \end{pmatrix}, \quad \mathbf{SNPM}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$\mathbf{SNPM}_0 = (1)$. Legyenek adottak a tevékenységek idő- és erőforrásadatai. Az időadatokat napokban tartalmazó $\mathbf{T}$ vektor legyen a következőképpen megadva: $\mathbf{T} = (4; 2; 4; 10; 3)$ hét. Az erőforrásigényeket megadó $\mathbf{R}$ vektor: $\mathbf{R} = (2; 3; 8; 14; 1)$ fő. Ezek alapján határozzuk meg a maximális prioritású projekttervet, ami 8 hét alatt 4 fővel elvégezhető. (A szakemberek heti bére legyen egységesen 300 EUR/hét). (Az egyszerűség kedvéért a költségek tekintetében csak erőforrásköltségekkel számolunk.) Ebből adódóan az erőforrásköltségeket **COSTS** vektor jelöli EUR-ban. $\mathbf{COSTS} = (2400; 1800; 9600; 42000; 900)$ EUR. Ekkor $\mathbf{t}_1 = (4; 2; 4; 3)$ nap. $\mathbf{t}_0 := 2$ nap, hiszen 4. tevékenységet, melyeknek időtartama 10 nap, mindenképpen elhagyunk, míg az összes bizonytalan kapcsolat elhagyása esetén csak **B** tevékenységet fogjuk megvalósítani. A tevékenységidők felső becslése $MAXT := MPM(\mathbf{SNPM}_1, \mathbf{t}_1)$, alsó becslése $MINT := MPM(\mathbf{SNPM}_0, \mathbf{t}_0)$.

Ugyanígy becsülhetők az erőforrás-igények átlagai. Ez az érték akkor minimális, ha valamennyi bizonytalan tevékenységet elhagyjuk, és valamennyi kapcsolatot előírjuk. Ezt a $\lceil \mathbf{PEM}^{\vee} \rceil$ mátrix adja meg. A legtöbb átlagos erőforrás-igény akkor jelentkezik, ha valamennyi tevékenységet megvalósítjuk, és valamennyi bizonytalan kapcsolatot feloldjuk. Ezt a $\lfloor \mathbf{PEM}^{\wedge} \rfloor$ mátrix adja meg. Ha az eredményül kapott projektterven erőforrás-kiegyenlítést szeretnénk végezni, akkor az erőforrás-igény(ek)re vonatkozó mutatók az erőforrás-igény(ek) átlagát jelölik. Ha az eredményül kapott projekttervekből egy olyan erőforrás-tervet szeretnénk meghatározni, amely az erőforrás-korlátot nem lépi túl, de a tevékenységek a lehető legkorábbi időpontban kezdődnek, akkor az erőforrás-igények jellemzésére a legmegfelelőbb mutató az erőforrás-igények maximuma (**MAXR**). Ha ugyanis ez a mutató az erőforrás-korlát alatt van, akkor nincs is szükségünk erőforrás-optimalizálásra. Ha pedig felette van, akkor is csak a tényleges projektterv ismeretében kell az erőforrás-tervezést elvégezni. Ebből adódóan a kiértékeléskor egy lehetséges projekt-változatot, projektstruktúrát nem zárunk ki pusztán azért, mert a maximális erőforrás-igény túllépi az erőforrás-korlátot. Csak akkor zárunk ki egy projekttervet a lehetséges megoldások közül, ha nem található rá megengedett, illetve egy optimáló, pl. ERALL-OPT módszerrel [26] számítható optimális megoldás.

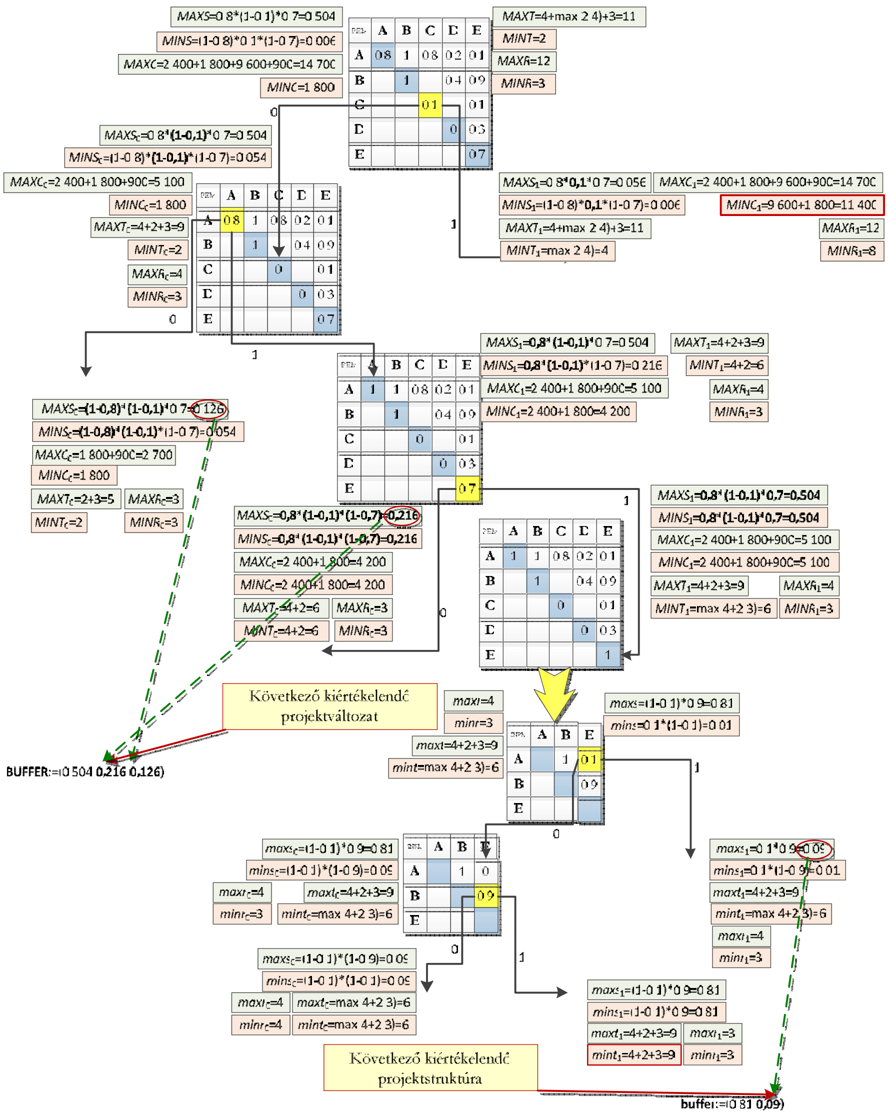
3.2 Projektváltozatok és projektstruktúrák gyors meghatározása

Legyen az erőforrás-korlát 4 fő, az időkorlát 8 hét, a költségkorlát pedig legyen 6000 EUR. Határozzuk meg a korlátokat kielégítő legjobb (legnagyobb pontértékkel rendelkező/legnagyobb valószínűséggel bekövetkező) megoldást! Az átlagos erőforrás-igény helyett számítsuk ki az erőforrás-igények maximumát és a pontértékeket e szerint számoljuk ki. A *4. táblázatban* szereplő jelöléseket használjuk.

Jelölés	Magyarázat
$MAXS$	A projektváltozatokra vonatkozó legnagyobb elérhető pontérték
$MINS$	A projektváltozatokra vonatkozó legkisebb elérhető pontérték
$maxs$	A projektstruktúrákra vonatkozó legnagyobb elérhető pontérték
$mins$	A projektstruktúrákra vonatkozó legkisebb elérhető pontérték
$MAXC$	A projektváltozatokra vonatkozó legnagyobb elérhető költségigény
$MINC$	A projektváltozatokra vonatkozó legkisebb elérhető költségigény
$MAXT$	A projektváltozatokra vonatkozó legnagyobb elérhető időigény
$MINT$	A projektváltozatokra vonatkozó legkisebb elérhető időigény
$maxt$	A projektstruktúrákra vonatkozó legnagyobb elérhető időigény
$mint$	A projektstruktúrákra vonatkozó legkisebb elérhető időigény
$MAXR$	A projektváltozatokra vonatkozó legnagyobb erőforrás-igény maximuma
$MINR$	A projektváltozatokra vonatkozó legkisebb erőforrás-igény maximuma
$maxr$	A projektstruktúrákra vonatkozó legnagyobb erőforrás-igény maximuma
$minr$	A projektstruktúrákra vonatkozó legkisebb erőforrás-igény maximuma

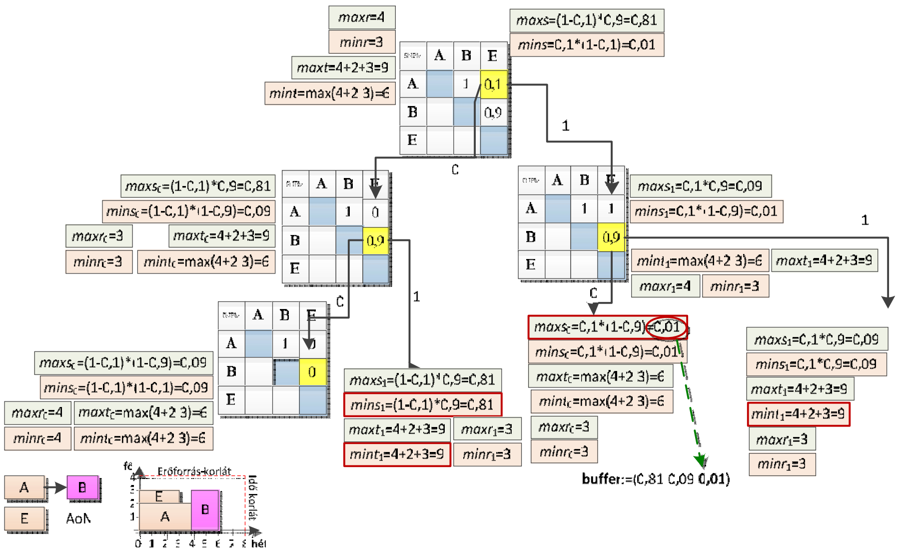
4. táblázat. Pontértékek jelölései és magyarázata

Ahogyan a 4. táblázatból is látható, nagybetűk mutatják a projektváltozatok, míg kisbetűk a projektstruktúrák jellemzőit.



5. ábra. Korlátoknak megfelelő projektterv meghatározása

Az *5. ábrán* látható, hogy a legvalószínűbb projektstruktúra nem felel meg a feltételeknek, ebből adódóan a következő projektstruktúrát kell megkeresnünk, mely pontértéke 0,09. Ebből kettő is található, ugyanakkor csak egy felel meg a korlátoknak (*6. ábra*). A következő projektváltozatot csak akkor kell tekinteni, ha az itt található projektstruktúrához egyetlen megfelelő projektváltozat sem tartozik.



6. ábra. Korlátoknak megfelelő projektterv meghatározása

Ha a bizonytalan tevékenységek száma t, a bizonytalan kapcsolatok száma pedig k, akkor $O(k+t)$ lépésben lehet meghatározni a legnagyobb pontértékkel rendelkező megoldást. Az n-edik legjobb megoldást pedig $O(n(k+t))$ lépésben. Az alábbi pszeudokód mutatja a fenti ábrán bemutatott algoritmus vázát. A fontosabb alkalmazott eljárásokat tartalmazza az *5. táblázat*.

Függvény	Magyarázat
$maxscore(\mathbf{SNPM}, \mathbf{p}, \mathbf{q})$	A projektstruktúrákra vonatkozó legnagyobb elérhető pontértéket adja meg.
$minscore(\mathbf{SNPM}, \mathbf{p}, \mathbf{q})$	A projektstruktúrákra vonatkozó legkisebb elérhető pontértéket adja meg.
$maxscoreTPT(\mathbf{SNPM}, \mathbf{p}, \mathbf{q}, \mathbf{t})$	A leghosszabb lehetséges projektstruktúra időtartamát adja vissza.
$minscoreTPT(\mathbf{SNPM}, \mathbf{p}, \mathbf{q}, \mathbf{t})$	A legrövidebb lehetséges projektstruktúra időtartamát adja vissza.
$ERALL-OPT(\mathbf{SNPM}, maxRES, \mathbf{t}, \mathbf{r})$	Egy kiértékelt projektstruktúrára vonatkozó optimális erőforrás-allokációt adja meg.
$MAXSCORE(\mathbf{PEM}, \mathbf{P}, \mathbf{Q})$	A projektváltozatokra vonatkozó legnagyobb elérhető pontértéket adja meg.
$MINSCORE(\mathbf{PEM}, \mathbf{P}, \mathbf{Q})$	A projektváltozatokra vonatkozó legkisebb elérhető pontértéket adja meg.
$MAXSCORETPC(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS})$	A projektváltozatokra vonatkozó legnagyobb költségigényt adja meg.
$MINSCORETPC(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{COSTS})$	A projektváltozatokra vonatkozó legkisebb költségigényt adja meg.
$MAXSCORETPT(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{T})$	A leghosszabb lehetséges projektváltozat időtartamát adja vissza.
$MINSCORETPT(\mathbf{PEM}, \mathbf{P}, \mathbf{Q}, \mathbf{T})$	A legrövidebb lehetséges projektváltozat időtartamát adja vissza.

5. táblázat. Fontosabb függvények felsorolása és magyarázata

```
function DSMs:=DSMfind(SNPM,DSMs,p,q,c,t,r,maxTPT,maxRES):
   [i,j]=argmax(p,q)where 0<SNPM<1 //bizonytalan kapcsolatok közül keressük meg
   if i≠0 and j≠0: //a maximális p, vagy q értékűt és adjuk vissza a találat helyét
      SNPM(i,j):=0 //A bizonytalan ka pcsolat értéke legyen 0.
      maxs0:=maxscore(SNPM,p,q)| mins0:=minscore(SNPM,p,q)
      maxt0:=maxscoreTPT(SNPM,p,q,t)
      mint0:=minscoreTPT(SNPM,p,q,t)
      if maxs0>=c and mins0<=c and mint0<=maxTPT then:
         DSMs:=DSMfind(SNPM,DSMs,p,q,t,r,maxTPT,maxRES)
      else: //Ekkor csak a maxscore értéket mentjük el.
         if maxs0∉BUFFER then: BUFFER=insertdescend(maxs0)
         SNPM(i,j):=1 //A bizonytalan kapcsolat értéke legyen 1.
         maxs1:=maxscore(SNPM,p,q)| mins1:=minscore(SNPM,p,q)
         maxt1:=maxscoreTPT(SNPM,p,q,t)
         mint1:=minscoreTPT(SNPM,p,q,t)
         if maxs1>=c and mins1<=c and mint1<=maxTPT then:
            DSMs:=DSMfind(SNPM,DSMs,p,q,c,t,r,maxTPT,maxRES)
         else: //Ekkor csak a maxscore értéket mentjük el.
            if maxs1∉BUFFER then: BUFFER=insertdescend(maxs1)
   else: if ERALL-OPT(SNPM,maxRES,t,r) is feasible than : DSMs:=DSMs∪SNPM
return DSMs

function SNPMs:=SNPMfind(PEM,SNPMs,P,Q,C,T,R,COSTS,maxTPT,maxTPC):
   i=argmax(diag(P),diag(Q))where 0<diag(PEM)<1 //bizonytalan tevékenységek közül
   if i≠0: //keressük meg a maximális P, vagy Q értékűt és adjuk vissza a találat helyét
      PEM(i,i):=0 //A bizonytalan tevékenység értéke legyen 0.
      MAXS0:=MAXSCORE(PEM,P,Q)| MINS0:=MINSCORE(PEM,P,Q)
      MAXC0:=MAXSCORETPC(PEM,P,Q,COSTS)| MINC0:=MINSCORETPC(PEM,P,Q,COSTS)
      MAXT0:=MAXSCORETPT(PEM,P,Q,T)| MINT0:=MINSCORETPT(PEM,P,Q,T)
      if MAXS0>=C and MINS0<=C and MINC0<=maxTPC and MINT0<=maxTPT then:
         SNPMs:=SNPMfind(PEM,SNPMs,P,Q,C,T,R,COSTS,maxTPT,maxTPC)
      else: //Ekkor csak a MAXSCORE értéket mentjük el.
         if MAXS0∉BUFFER then: BUFFER=insertdescend(maxs0)
         PEM(i,i):=1 //A bizonytalan tevékenység értéke legyen 1.
         MAXS1:=MAXSCORE(PEM,P,Q)| MINS1:=MINSCORE(PEM,P,Q)
         MAXC1:=MAXSCORETPC(PEM,P,Q,COSTS)| MINC1:=MINSCORETPC(PEM,P,Q,COSTS)
         MAXT1:=MAXSCORETPT(PEM,P,Q,T)| MINT1:=MINSCORETPT(PEM,P,Q,T)
         if MAXS1>=C and MINS1<=C and MINC0<=maxTPC and MINT0<=maxTPT then:
            SNPMs:=SNPMfind(PEM,SNPMs,P,Q,C,T,R,COSTS,maxTPT,maxTPC)
         else: //Ekkor csak a maxscore értéket mentjük el.
            if MAXS1∉BUFFER then: BUFFER=insertdescend(maxs1)
   else: SNPMs:=SNPMs∪delete_empty_cells(PEM) //A kiértékelt PEM-et vegyük fel
return SNPMs

function DSMs=EPR(PEM,P,Q,T,R,COSTS,maxTPT,maxRES,maxTPC)
   global BUFFER,buffer
   BUFFER:=MAXSCORE(PEM,P,Q)
   SNPMs:=∅| DSMs:=∅
   i:=1
   do
      SNPMs=SNPMs∪SNPMfind(PEM,SNPMs,P,Q,BUFFER(i),T,R,COSTS,maxTPT,maxTPC)
      p:=synchronize(SNPMs(i),P); //A kitörölt sorokat/oszlopokat P-ből is töröljük
      q:=synchronize(SNPMs(i),Q); //A kitörölt sorokat/oszlopokat Q-ból is töröljük
      t:=synchronize_t(SNPMs(i),T); //A kitörölt sorokat/oszlopokat T-ből is töröljük
      r:=synchronize_r(SNPMs(i),R); //A kitörölt sorokat/oszlopokat R-ből is töröljük
      buffer:=maxscore(SNPMs(i),p,q)
      j:=1
      do
         DSMs=DSMs∪DSMfind(SNPMs(i),DSMs,p,q,buffer(j),t,r,maxTPT,maxRES)
         j:=j+1
      while j<=numel(buffer) and DSMSs=∅:
      i:=i+1
   while i<=numel(BUFFER) and DSMSs=∅:
return
```

1. pszeudokód: EPR (Expert Project Ranking)

Ezzel a módszerrel a korlátoknak megfelelő legnagyobb pontértékkel rendelkező projektváltozatot, projektstruktúrát lehet meghatározni.

Azonban ez a keretalgoritmus pontértékeket számoló függvények változtatásával átalakítható lehető legrövidebb, lehető legkisebb költséggel rendelkező, vagy akár lehető legkevesebb erőforrást igénylő projektváltozatok, projektstruktúrák meghatározására is. Ehhez csak a pontértékek figyelembe-vételét kell átalakítanunk. Ami most célfüggvényként szerepelt (pl. lehető legfontosabb projektváltozat, projektstruktúra megtalálása) az lehet most akár korlátozó feltétel, és ami eddig korlátozó feltétel volt (pl. átfutási idő, költségigény), az most célfüggvényként kerül figyelembe.

4 Szimulációs eredmények

A szimulációk célja annak bemutatása, hogy nagyszámú lehetséges projektváltozat, illetve projektstruktúra esetén is lehetőség nyílik az első m legjobb pontértékkel rendelkező, adott korlátoknak megfelelő lehetséges projektterv meghatározására. A tesztelés során először egy projektváltozatot jellemző **SNPM**-mátrixot generáltunk, ahol az átló feletti cellaértékek $(0,1)$ között bármilyen számot felvehettek. Ekkor n tevékenység esetén $k = n*(n-1)/2$ bizonytalan kapcsolatot kapunk, amely $2^k = 2^{n*(n-1)/2}$ lehetséges projektstruktúrát jelent, ahol k a bizonytalan kapcsolatokat, n pedig a tevékenységek számát jelöli. Ekkor a feladat az első m legjobb pontértékkel rendelkező projektváltozat meghatározása volt.

A második szimuláció esetén egy n tevékenységből álló **PEM**-mátrixot határoztunk meg, ahol az átló és az átló feletti értékek $(0,1)$ közötti számokat vehetnek fel. Ebben az esetben a lehetséges projektváltozatok száma $t = n$ esetén $2^t = 2^n$, ahol t a bizonytalan tevékenység-előfordulások számát, n pedig a lehetséges tevékenységek számát jelöli. Mindegyik lehetséges projektváltozatot lehet kódolni t biten ábrázolható bináris számmal. A bináris számban, j-edik helyi értéken szereplő bináris 0 azt jelöli, hogy azt a j-edik tevékenységet elhagyjuk. Az 1-es pedig azt, hogy megvalósítjuk. Így pl. $t = 4$ esetén a bináris 0101 azt jelöli, hogy a második és a negyedik bizonytalan tevékenységet fogjuk megvalósítani, a többit elhagyjuk. Ez lesz a lehetséges projektváltozatok közül decimálisan az 5-ös számú projektváltozat, hiszen a bináris 0101 decimálisan 5-öt jelöl.

A lehetséges projektváltozatokon belül kell azután meghatározni a lehetséges projektstruktúrák számát, ami $\sum_{i=0}^{2^t-1} 2^{bin_1(i)*(bin_1(i)-1)/2}$, ahol $bin_1(i)$ az i bináris számrendszerben szereplő 1-eseinek számát adja meg. A formula egyszerűsíthető, ha figyelembe vesszük, hogy $j = 0,1,2,\dots,t$ tevékenységet kell tetszőlegesen kiválasztanunk, amit, ha minden tevékenység között lehet kapcsolat, akkor $2^{j(j-1)/2}$-féleképpen tudunk elrendezni, hiszen pontosan enynyi projektstruktúra létezhet. Ekkor az összes lehetséges projektstruktúra száma a következőképpen számítható: $\sum_{j=0}^{t} \binom{t}{j} 2^{j(j-1)/2}$. Látható, hogy viszonylag könnyű nagyon sok projektváltozatot, projektstruktúrát tartalmazó projekttervet megadni. Egy 10×10-es PEM-mátrix esetén a lehetséges projektstruktúrák száma már több mint $3{,}588 * 10^{13}$. Belátható, hogy itt nem megoldható, hogy kiszámoljuk az összes lehetőséget, majd azokat sorrendbe rakjuk valamely rendező algoritmussal. Éppen ezért a fent bemutatott módszerekkel fogjuk meghatározni az első m darab projektstruktúrát úgy, hogy a lehető legnagyobb pontértékkel rendelkező projektváltozatokon belül keressük a lehető legnagyobb pontértékkel rendelkező projektstruktúrákat.

A harmadik szimulációban már a tevékenységekhez költség-, idő- és erőforrás-adatokat is rendelünk. Megvizsgáljuk, hogy az adott idő-, költség- és erőforrás-korlátokat nem túllépő lehetséges projektváltozatok közül melyek lesznek a legnagyobb pontértékűek.

Az olvasóban jogosan felmerülhet a kérdés, ha már 10 tevékenység esetén ekkora mennyiségű lehetséges projektstruktúra közül kell válogatnunk, akkor egy több ezer tevékenységet tartalmazó projektben hogyan lehetne bármilyen algoritmikus módszerrel a lehető legnagyobb pontértékkel rendelkező (legfontosabb, legvalószínűbb) projekttervet megtalálni.

A fenti pontértékek becslése pont a lépésszám csökkentését szolgálta. Láthattuk, hogy $O(k+t)$ lépésben megtaláltuk a legnagyobb pontértékkel rendelkező projekttervet (projektváltozaton belül lehetséges projektstruktúrát). Ez pl. egy $n = 1000$, $t = n$, $k = n * (n-1)/2$, $n \times n$-es **PEM**-mátrixban legfeljebb $O(k+t) = O(t + t * (t-1)/2) = O(t^2)$ lépést jelent. A lépésszám ráadásul nem a tevékenységek számától, hanem a bizonytalan tevékenység-előfordulások és a bizonytalan kapcsolatok számától függ. Így ez az érték is csak egy felső becslése a lehetséges lépésszámnak.

Lehetséges projektstruktúrák meghatározása

A fenti módon, egy **SNPM**-mátrix átló feletti celláit véletlen számokkal töltöttük fel. Kezdetben feltételeztük, hogy $\mathbf{p} = \mathbf{SNPM}$, $\mathbf{q} = \mathbf{1} - \mathbf{SNPM}$. Ebben az esetben az első 1, 10, 100, illetve 1000 projektstruktúrát kellett meghatározni. Arra is kíváncsiak voltunk, hogy az első m legnagyobb pontértékkel rendelkező projektváltozathoz hány további pontértéket kellett eltárolni. Ahogy az előző példákból láthattuk, ezeket a pontértékeket kell majd megkeresnünk, ha a keresést tovább folytatnánk. Ha a *buffer*-halmaz mérete túlságosan nagy lenne, akkor ez egyrészt arra utal, hogy sok pontértéket kellett kiszámítani, vagyis sok projektstruktúrát kellett meghatározni, másrészt a számítás és a *buffer*-halmaz rendezése sok időt venne igénybe.

n	Projektváltozatok száma	m	Buffer-halmaz mérete
5	$2^{10} = 1024$	1	11
		10	22
		100	126
		1000	1009
10	$2^{45} \approx 3,52 * 10^{13}$	1	46
		10	54
		100	162
		1000	1267
25	$2^{300} \approx 2,04 * 10^{90}$	1	301
		10	322
		100	351
		1000	1807
50	$2^{1225} \approx 5,78 * 10^{368}$	1	1226
		10	1410
		100	1821
		1000	2237

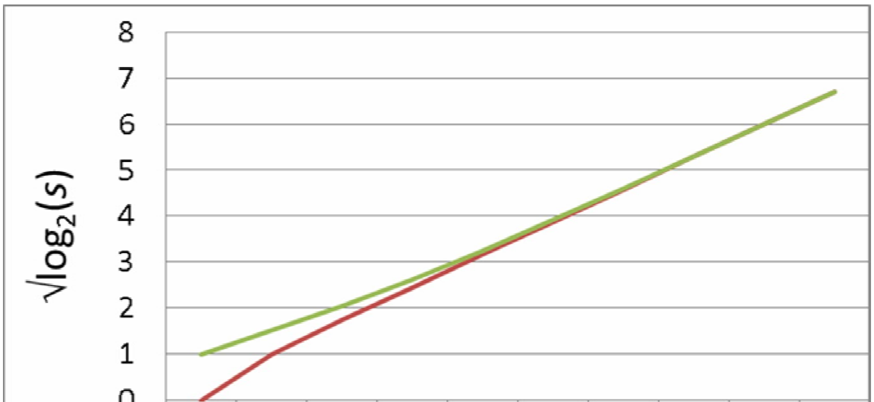
6. táblázat. Projektstruktúrák meghatározása

A kiértékelés során legfeljebb a buffer-halmazban szereplő pontértékekhez tartozó projektterveket értékeljük ki. A gyors kiértékelő algoritmusnak köszönhetően azonban még ilyen sok lehetséges projektstruktúra közül is ki lehet választani az első akár 1000 legnagyobb pontértékkel rendelkezőt.

Lehetséges projekttervek meghatározása

Egy szakértői mátrixból a lehetséges projekttervek meghatározása két lépésben történik. Első lépésben a legnagyobb pontértékkel rendelkező projektváltozatokat kell meghatározni. Majd ezen belül kell megtalálni a legnagyobb pontértékkel rendelkező projektstruktúrákat. A legtöbb lehetséges projektstruktúra akkor jelenik meg, ha valamennyi tevékenységet megvalósítjuk. Ekkor a lehetséges projektstruktúrák száma (amennyiben minden tevékenység között értelmezhetünk sztochasztikus kapcsolatokat) $2^{t(t-1)/2}$.

Ugyanakkor, ha csak egy tevékenységet nem valósítunk meg, a lehetséges projektstruktúrák száma már drasztikusan csökken. Ezt láthatjuk a *7. ábrán* is. A feladat itt is az első m darab projektterv (legjobb projektváltozaton belül megvalósítható legjobb projektstruktúra) meghatározása. Ennek maximális értéke $s = \sum_{j=0}^{t} \binom{t}{j} 2^{j(j-1)/2}$, ahol t a tevékenységek számát, s a lehetséges struktúrák számát jelöli. Az ábrán a lehetséges projektstruktúrák számának 2-es alapú logaritmus négyzetgyökét szemléltettük, hiszen egy t tevékenységet tartalmazó projektstruktúra száma $s = 2^{t(t-1)/2}$, így $\sqrt{\log_2 s} = \sqrt{t(t-1)/2}$, ami $\sqrt{2}(t/2 - 1/4)$-hez tart, így jó közelítéssel lineáris függvénye a tevékenységek számának (*7. ábra*). Látható, hogy a projektstruktúrák számát jó közelítéssel a legtöbb tevékenységet tartalmazó (ún. maximális) projektváltozat lehetséges projektstruktúráinak száma határozza meg.



	1	2	3	4	5	6	7	8	9	10
SNPMs	0	1	1,732	2,449	3,162	3,873	4,583	5,292	6	6,708
PEMs	1	1,524	2,042	2,612	3,241	3,91	4,6	5,3	6,004	6,71

7. ábra. Lehetséges projektstruktúrák száma

A fenti *6. táblázathoz* hasonló eredményeket kaptunk a szakértői mátrix kiértékelése során is. 50 tevékenységet tartalmazó szakértői mátrix esetén, ahol a lehetséges projektstruktúrák száma: $5{,}78 * 10^{368}$, a kiértékelt projekttervek (projektváltozatok, és azon belül kiszámított projektstruktúrák) 2458 volt, ami alig magasabb az előző feladatban meghatározott projektstruktúrák számánál. Az első m projektstruktúra futásigénye: $mO(t + k)$, ahol k maximálisan $t(t-1)/2$ lehet, így a futásidő ekkor $mO(t(t+1)/2) \approx mO(t^2)$-tel becsülhető. Annak köszönhetően, hogy azokat a projektstruktúrákat, amelyek egy adott korlát alatt vannak, nem kell kiértékelnünk, a véletlen pontértékeket tartalmazó szimulációkban jóval kevesebb, kvázi lineáris lépésben is meghatározhattuk az első legnagyobb pontértékkel rendelkező projektstruktúrát.

Korlátokat figyelembe vevő projekttervek meghatározása

A logikai tervezés után következik a költség- és erőforrás-tervezés, melyet vagy a kiértékelt projektstruktúrákon végezhetünk el, vagy már a kiértékelés közben kiszámíthatjuk a maximális átfutási időket, költség- és erőforrásigényeket. Ehhez tudnunk kell, hogy a projektstruktúra költségének kiszámítása $O(t)$, hiszen itt egyszerűen csak össze kell adni a tevékenységeket. Az átfutási idő meghatározása topológikus rendezésen alapuló leghosszabb út keresésével $O(t+k)$ [27]. Az átlagos erőforrásigények meghatározása (legkorábbi időpontra ütemezett tevékenységek erőforrás-igényét figyelembe véve): $rO(t + k)$, ahol r az erőforrások számát jelöli. Mivel azonban a leggyorsabb kiegyenlítési algoritmus is $O(t^4)$ rendű, az optimális erőforrás-allokációs problémák pedig általában már ún. NP-teljes problémák [7], melyekre gyors algoritmus mindezidáig nem létezik, így az erőforrás-tervezést csak a lehetséges projektstruktúrák meghatározása után fogjuk elvégezni.

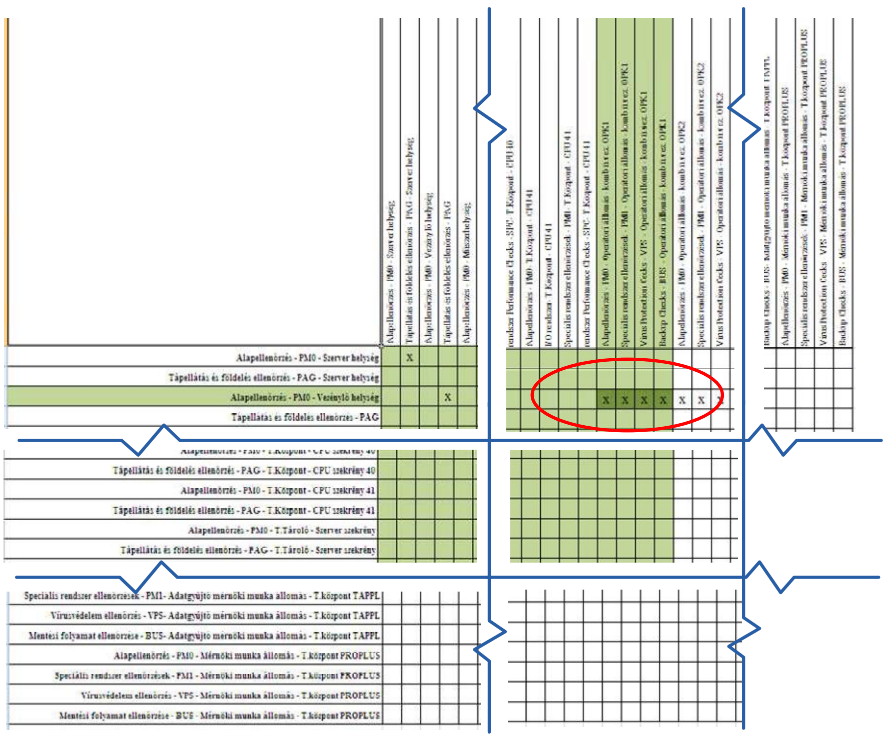
Ebben a feladatban véletlenszerűen generáltunk tevékenységekhez átfutási időket, költség- és erőforrás-igényeket. Az átfutási idők 1-10 nap között, a költségigények 100 és 1000 EUR között, a négyfajta erőforrásigény, mely jelölhetett emberi erőforrást és berendezéseket is, 1 és 5 diszkrét érték között bármilyen értéket felvehetett egyenlő valószínűséggel. A költségkorlát az összes költség (valamennyi tevékenység végrehajtása esetén a teljes költség) $tpc = (60, 70, 80, 90, 100)$%-a, az átfutási idő az összes tevékenység soros végrehajtásának átfutási idejének $tpt = 25, 50, 75$%-a volt. Az erőforrás-korlát minden esetben 20 egységnyi volt. A tevékenységek pontértékei 0,5 és 1 között vehettek fel értéket. Ezzel szimulálva, hogy a projekttervben szereplő tevékenységeket, ha lehetőségünk nyílik, akkor inkább valósítsuk meg. A kapcsolatok 0 és 1 között vehettek fel értékeket. Ebben a feladatban 50×50-es mátrixot tekintettünk, ahol valamennyi tevékenységet és valamennyi kapcsolatot is bizonytalannak tekintettünk. A lehetséges projektstruktúrák száma $5{,}78 * 10^{368}$. Ennek ellenére a legfeljebb 1815 lépésben meghatároztuk a korlátokat (idő és költségkorlátot) nem túllépő projekttervet. Ezután következhetett az idő- és erőforrás-korlátokat nem túllépő lehető legkorábbi kezdési időpontokat megadó optimális megoldás megtalálása. Mivel az eredményül kapott projektstruktúráknak volt megengedett megoldása, így az ERALL-OPT [26] módszerrel a lehető legkorábbi kezdésre vonatkozó optimális megoldás is megadható volt.

5 Gyakorlati alkalmazás

A következő esetpéldában egy számítástechnikai infrastruktúra karbantartásának támogatását láthatjuk, melyen keresztül ismertetem a kifejlesztett szakértői rendszer alkalmazásának lehetőségeit.

A hardverek és szoftverek karbantartása komplex és bonyolult kapcsolati rendszerrel rendelkezik, ezt a gyakorlatból kiemelt *8. ábra* is alátámasztja. A korábban bemutatott mátrixos tervezési módszer alkalmazásával a karbantartás tervezhetővé válik. A tevékenységek kockázat alapú értékelését követően karbantartási tervet állíthatunk össze, amely az idő-, erőforrás- és költségkorlátnak megfelel.

Az informatikai infrastruktúra karbantartási szolgáltatást megrendelő ügyfelek modulokat, ezzel együtt tevékenységeket vásárolnak, amelyek a karbantartási műveleteket tartalmazzák. A szolgáltató vállalattal megegyezik a vevő a végrehajtási időkorlátban és költségráfordításban. Példánkban a megkötött szerződésbe 49 db modult foglaltak, ezeken belül pedig a tevékenységek esetenként sorosan, de legtöbb esetben párhuzamosan is végrehajthatók.



8. ábra. Előzetes karbantartási terv moduláris felépítettsége - részlet

A karbantartási szolgáltatást megrendelő ügyfél a 8. ábra felső részén jelölt (1 db) operátori állomás karbantartási tevékenységeire szeretne kevesebb időt és költséget fordítani. Korlátként állomásonként 4 mérnökórát és 100 000 Ft/állomás költségkeretet adott meg, így a karbantartást végző szolgáltató vállalat munkatársainak (4 fő) ezekhez a korlátokhoz kell tartaniuk magukat a terület ellenőrzése és javítása során. Abban az esetben, ha minden egyes tevékenységet elvégezne, 4,85 mérnökórára lenne szüksége 1 karbantartónak, és 101 850 Ft-ba kerülne 1 állomás teljes átvizsgálása és javítása.

Az említett operátori állomás karbantartásához köthető modulkombinációt emeltem ki, amelyet tevékenységekre bontunk le. A tevékenységeket megvizsgálva, kockázati kategóriákba sorolhatjuk, így magas ($1{,}0 \geq$ kock.csop. $>$ $0{,}8$), közepes ($0{,}8 \geq$ kock.csop. $> 0{,}6$) illetve alacsony ($0{,}6 \geq$ kock.csop. ≥ 0) kockázati csoportokat állapítottunk meg a vállalati szakemberek bevonásával. A felállított projekt szakértői mátrix átlójába foglaltuk össze a meghibásodási valószínűségeket. Ezek után már a karbantartás tevékenységeinek sorrendje könnyen megállapítható. A magas kockázatú karbantartási tevékenységek mellett esetünkben a közepes kockázatú tevékenységeket is szerepeltettük a karbantartási tervünkben. Majd a tevékenységek közti kapcsolatok figyelembevételével a tevékenységek sorrendjét is megállapítottuk. A modulok végrehajtását a szigorúan előírt tevékenységekkel kezdték meg, majd ezt a többi tevékenység végrehajtása követte. Egyszerre több tevékenység volt végrehajtható, hiszen számítógépes diagnosztika alkalmazásával egy időben több ellenőrző program is futtatható. Az erőforrás minden egyes operátori állomás esetében 1 karbantartó személyre korlátozódik.

Modulok	Tevékenységek	Standard Diagnosztika futtatása	Riasztások és események figyelése	Lemezhasznosítás ellenőrzése	VDU'k ellenőrzése	Számítógép ellenőrzése	Nyomtató ellenőrzése	Operációs Rendszer Diagnosztika futtatása	Adatgyűjtő alkalmazás ellenőrzése	Power down	Vírusvédelem ellenőrzése	Mentési folyamat ellenőrzése
Alapellenőrzés - PMO	Standard Diagnosztika futtatása	1,00										
	Riasztások és események figyelése		1,00									
	Lemezhasznosítás ellenőrzése			0,70								
	képernyő ellenőrzése				0,85							
	Számítógép ellenőrzése	1,00	1,00	1,00	1,00	1,00	1,00					
	Nyomtató ellenőrzése						0,65					
Speciális rendszer ellenőrzések - PM1	Operációs Rendszer Diagnosztika futtatása							0,90				
	Adatgyűjtő alkalmazás ellenőrzése								0,65			
	Power Down - lekapcsolás							1,00	1,00	1,00		
Vírusvédelem ellenőrzése - VPS	Vírusvédelem ellenőrzése										1,00	
Mentési folyamat ellenőrzése - BUS	Mentési folyamat ellenőrzése, rendszeres mentések elvégzése (automatizált, manuális)	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,85

9. ábra. operátori állomás alacsony kockázatú tevékenységekkel redukált kapcsolati mátrixa

Az idő- és költségkorlátokat figyelembe véve pedig a legnagyobb prioritással rendelkező projektváltozatot kerestük. Kimenetként olyan karbantartási tervet kaptunk (*9. ábra*), amelyben az alacsony kockázatú tevékenységek már nem szerepeltek. A költségek redukálódnak 33 850 Ft-tal. A mátrixos tervezési módszer alkalmazásával így a megrendelő által támasztott korlátoknak megfelelt az így összeállított karbantartási terv.

6 Összefoglalás

A bemutatott módszerek a lehetséges projekttervek meghatározásában, különböző szempontok szerinti meghatározásában nyújthatnak segítséget. A módszer végeredményeként egy olyan projekttervet kapunk, amely a megfogalmazott céloknak leginkább megfelel, tartalmazza a legfontosabb elvégzendő feladatokat, ugyanakkor a rendelkezésre álló idő- költség- és erőforráskeretet sem lépi túl. A módszer alkalmazásának eredményeképpen projektterveket kapunk, amelyek már hagyományos projektmenedzsment szoftverekkel is könnyen menedzselhetők, nyomon követhetők. Ugyanakkor a megfelelő projektterv kiválasztásának szerepét egy erre épülő projekt szakértői rendszer végezheti el. Az itt bemutatott eljárás egy ilyen rendszer kialakításához is segítséget nyújthat.

Köszönetnyilvánítás

Jelen tanulmány a Bolyai János Kutatási Ösztöndíj támogatásával készült.

Irodalom

1. Eisner H. (1962): A Generalized Network Approach to the Planning and Scheduling of a Research Project. *Operation Research*, 1962, vol 10. no. 1. pp. 115-125.
2. Roy, B. (1962): Graphes et ordonnancements. *Revue Française de recherche opérationnelle.* no. 25, 6 (10), page 323.
3. Kelley J. – Walker, M. (1959): Critical-Path Planning and Scheduling. *Proceedings of the Eastern Joint Computer Conference.*
4. Gantt, H. L. (1974): Work, Wages and Profit. *The Engineering Magazine*, New York, 1910; republished as Work, Wages and Profits, Easton, Pennsylvania, Hive Publishing Company, 1974, ISBN 0-87960-048-9
5. Al Sarraj, Z. M. (1990): Formal development of line-of-balance technique. *Journal of Construction Engineering and Management ASCE* 116 4, pp. 689–704
6. Project Management Institute (2006): *Projekt menedzsment útmutató – PMBOK GUIDE*, Budapest, Akadémiai Kiadó, pp. 21–25, ISBN 963 05 8401 8
7. 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.

8. Danilovic, Mike – Browning, Tyson R. (2007): Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management* 25:300–314.
9. Kosztyán, Zs. T. – Fejes, J. – Kiss, J. (2008): Sztochasztikus hálótervezési módszerek. *Szigma* 39. pp. 85–103.
10. Steward D. (1981): *System Analysis and Management: Structure, Strategy, and Design*. New York: Petrocelli Books.
11. Kosztyán Zsolt Tibor, Kiss Judit (2011): Mátrixalapú projekttervezési módszer. *Vezetéstudomány*. 2011/10 pp. 28–43.
12. Eppinger, Steven D. – Whitney, Daniel E. – Smith, Robert P. – Gebala, David A. (1994): A model-based method for organizing tasks in product development. *Research in Engineering Design*, 6:1–13.
13. Pritsker A. A. (1966): GERT: Graphical Evaluation and Review Technique. MEMORANDUM, RM-4973-NASA
14. Xiao, Renbin – Chen, Tinggui – Tao, Zhenwu (2007): Information modeling and reengineering for product development process. *International Journal of Management Science and Engineering Management* 2, no. 1, 64–74.
15. Minogue, P. (2011): Gantt-Like DSMs. Invest on Visualization. *Proceedings of the 13th International DSM Conference,* pp. 259–271.
16. Chen, Chun-Hsien – Ling, Shih Fu – Chen, Wei (2003): Project scheduling for collaborative product development using DSM. *International Journal of Project Management* 21:291–299.
17. Yan H, Wang Z, Jiang M. (2002): A Quantitative Approach to the Process Modelling and Planning in Concurrent Engineering. *Concurrent Engineering.* 10:97–111.
18. Yassine, A. – Falkenburg, D. – Chelst, K. (1999): Engineering design management: An information structure approach. *International Journal of Production Research*, 37(13). pp. 2957–75.
19. Tang, D. – Zhu, R. – Tang, J. – Xu, R. – He, R. (2010): Product design knowledge management based on design structure matrix. *Advanced Engineering Informatics*, pp. 159–166.
20. Chen, Shi-jie – Lin, Li (2002): A Project Task Coordination Model for Team Organization in Concurrent Engineering. *Concurrent Engineering.* 10:187–202.
21. Browning, T. R. – Eppinger, S. D. (2002): Modelling Impacts of Process Architecture on Cost and Schedule Risk in Product Development. *IEEE Transactions on Engineering Management*, 49(4), pp. 428–442
22. Kiss, J. – Kosztyán, Zs. T. (2010): Using PEM as a knowledge management tool – How can be used earlier experience at new IT and innovation projects? KMO (Knowledge Management in Organizations) 2010, Veszprém, May 18-19, 2010. pp. 204–217
23. Kosztyán, Zs. T. (2012): Challanges of the Project Planning Methods in the 21st Century 2012. *Problems of Management in the 21st Century* 2(5):46–60.
24. Kiss, J. (2012): Next generational applications – Supporting the planning phase of projects. World Congress on Information Technology. Barcelona, Spain, 2012. november 14–16. (proceedings megjelenés alatt)
25. Kosztyán Zs. T. (2013): Projekttervezési módszerek kihívásai a XXI. században. *Vezetéstudomány*, 2013. VII (megjelenés alatt)

26. Kosztyán, Zs. T. – Bencsik, A. – Póta, Sz.: Resource Allocation and its Distributed Implementation. *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, (ed. Tarek Sobh), Springer 2007, ISBN: 978-1-4020-6267-4, pp. 511–518
27. Ahuja, R. K. – Magnanti, T. L. – Orlin, J. (1993): *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall.

MATRIX-BASED STRATEGIC PROJECT PLANNING

Beside network planning methods matrix-based methods can also be used in project management, primarily for supporting strategic decision-making. In this case either importance or probability of task completions can be described, and thus either importance or probability of possible project scenarios and project structures can be determined and ranked by theirs importance or probabilities. When using matrix-based project planning the main challenge is to select the project scenario and project structures regarding the management claims. In this study fast, exact algorithms are introduced to select the most important project scenarios or the least cost/time demanded project structures. This algorithm is a framework algorithm which can be a fundamental basis of a project expert system.