

Tartalomjegyzék

1. Bevezetés	4
2. Előzmények	8
2.1. A PNS probléma	8
2.2. A PNS probléma redukciója	13
2.3. A maximális struktúra meghatározása	16
2.4. A súlyozott PNS modell	19
2.5. A PNS probléma NP teljessége	20
2.6. Döntési leképezések	26
2.7. Leszámlálási algoritmusok	30
2.7.1. Egyszerű leszámlálás	30
2.7.2. Teljes leszámlálás	34
3. A döntési leképezések száma	37
3.1. Általános eset	37

<i>TARTALOMJEGYZÉK</i>	2
3.2. Speciális esetek	44
3.2.1. Az Egyenes modell	45
3.2.2. A Lánc modell	47
3.2.3. Azonosságok	48
3.2.4. Szemléltetés	49
3.3. Következtetések	51
4. Összevonásos redukció	52
4.1. Az összevonás megvalósítása	52
4.2. Empirikus analízis	60
4.3. Következtetések	61
5. Előrettekintő B&B algoritmus	63
5.1. Az algoritmus bemutatása	64
5.2. Empirikus analízis	73
5.2.1. A vizsgálatok ismertetése	73
5.2.2. Konklúziók	74
6. Parciális leszámlálási eljárás	76
6.1. A parciális leszámlálás bemutatása	77
6.2. Empirikus analízis	79
6.2.1. Konklúziók	81

<i>TARTALOMJEGYZÉK</i>	3
7. Automataelméleti megközelítés	83
7.1. A módosított PNS probléma	83
7.2. A módosított PNS probléma megoldása	86
7.2.1. Automataelméleti alapfogalmak	86
7.2.2. A probléma megoldása	87
7.3. A PAT algoritmus	89
8. Köszönetnyilvánítás	99
9. A disszertáció összefoglalása	100
10. Summary of the doctoral thesis	104
10.1. Synopsis	104
10.2. The structural model of PNS	105
10.3. Results of the thesis	108
Tárgymutató	111
Irodalomjegyzék	114

1. fejezet

Bevezetés

A kémiában, biológiában, ipari alkalmazásokban gyakran felmerül olyan probléma, amelyben bizonyos „nyersanyagok” és „műveleti egységek” rendelkezésünkre állnak és előírt anyagokat kívánunk előállítani az adott műveleti egységek összekapcsolásával. A probléma egy lehetséges modellezését Fan és Friedler professzorok a 70-es években dolgozták ki, melyben strukturális tulajdonságokra épülő technikákat használtak a megoldások keresésére. Ebben a „Process Network Synthesis” (PNS)-nek nevezett modellben minden műveleti egység az anyagok egy részhalmazát inputként igényli és anyagok egy másik részhalmazát állítja elő. A gráfelméleti megközelítésben egy anyagtól irányított él vezet azokhoz a műveleti egységekhez, amelyek input anyagként felhasználják, illetve egy műveleti egységet irányított éllel kötünk össze azokkal az anyagokkal, amelyeket output anyagként termel. Így egy kétrészes (anyagok; műveleti egységek) irányított gráfot kapunk, a folyamat gráfját. Statikusan képzelve a termelési folyamatot, a műveleti egységek egy részrendszerének működésével a kívánt anyagokat bizonyos alapvető feltételek teljesülése esetén kaphatjuk meg. Ily módon a lehetséges megoldásoknak rendelkezniük kell bizonyos strukturális tulajdonságokkal, ami miatt ezeket megoldás struktúráknak is szokás nevezni. A Fan és Friedler professzorok által irányított kutatásokban kidolgozásra kerültek olyan eljárások, melyek lehetővé teszik adott problémára az összes lehetséges, vagy valamilyen szempontból optimális megoldás struktúra generálását ([15, 16, 18]). Ezek között kitüntetett szerepe van a maximális struktúrának, mely a lehetséges megoldás struktúrák uniója, és melynek ge-

nerálására a szerzőknek sikerült polinomiális idejű algoritmust adniuk ([11]). A 90'-es évek második felében Imreh Balázs vezetésével Szegeden is kialakult egy szakmai közösség, melynek szerencsére a szerző is részese lehetett, és amely első sorban a PNS probléma diszkrét, strukturális és kombinatorikai tulajdonságainak vizsgálatával foglalkozott. A csoport kutatómunkájának eredményeképpen számos publikált cikk született, melyek közül most csak a dolgozathoz kapcsolódó [2, 25, 3, 4, 5, 30, 20, 6, 7, 21, 31, 23, 8, 28, 22] cikkeket említeném meg.

A dolgozatban tárgyalt PNS modell csak strukturális szempontból tekinti a problémát, mivel annak leírása anyagmennyiségekre vonatkozó előírásokat nem tartalmaz. A rendelkezésre álló műveleti egységek viszont rendelkeznek bizonyos költséggel és az anyagelőállítás során használt műveleti egységek összköltségét szeretnénk optimalizálni: keressük a műveleti egységek azon legkisebb összköltségű részhalmazát, mely a rendelkezésre álló nyersanyagokból képes előállítani a kívánt végtermékeket. Mivel a minimum meghatározása még ebben a legegyszerűbb strukturális esetben is a halmazlefedési problémával ekvivalens ([2, 17, 27]), így a halmazlefedési probléma NP teljességéből ([32, 1]) következően a strukturális PNS-probléma is sajnos NP-teljes. Nem várható tehát hatékony megoldás rá. Ezért indokolt hatékonyabban megoldható speciális esetek, vagy csak az optimumhoz közeli megoldást szolgáltatató, de polinomiális idejű heurisztikák vizsgálata ([25, 30, 6, 7, 31, 8]), illetve exponenciális algoritmusok és azok különböző heurisztikákkal kombinált Branch-and-Bound jellegű változatainak kidolgozása ([13, 24, 21, 23]). Ezek felépítésében fontos szerepet játszik az úgynevezett döntési leképezés fogalma ([12]), mely lényegében meghatározza adott anyagra az őt gyártó műveleti egységek halmazát. Gyakorlati szempontból nyilván nagyon komoly előny, hogy csak bizonyos úgynevezett "konzisztens" döntési leképezéseket kell figyelembe venni, ami abból az észrevételből származik, hogy egy műveleti egység, ha működik, nem teheti meg, hogy bizonyos kimeneti anyagait gyártja, másokat pedig nem. Ezen belül további szűkítést eredményez az az észrevétel, hogy nem működhet olyan műveleti egység, amelyik valamelyik input anyagát egyetlen működő műveleti egységtől sem nyeri. Ezt felhasználva, a szitaformula segítségével felső korlát adható a lehetséges megoldás strukturák számára ([3, 4, 5]).

Jelen dolgozat második, Előzmények című fejezetében a teljesség igénye nélkül ismertetjük azokat az alapfogalmakat és technikákat, amelyekhez a dolgozat további részei kapcsolódni fognak. Bizonyos kapcsolódó al-

goritmusok teljes körű bemutatása viszont meghaladná a dolgozat kereteit, ezek esetében majd a megfelelő cikkekre fogunk hivatkozni, ezen fejezet célja inkább egy olyan átfogó képet nyújtani a probléma alapvető eredményeiről, mely lehetővé teszi a dolgozat további részeinek megértését, és amelyet a szerző kiindulópontként használhatott fel önálló kutatásai megvalósításához.

A PNS kutatásaiban történetileg legkorábbi, a szerzőhöz kapcsolódó kutatások az úgynevezett döntési leképezések tulajdonságainak feltárására irányultak. Lényegében azt vizsgáltuk, hogy hogyan és hányféleképpen lehet a köztes vagy végtermékekhez őket gyártó műveleti egységeket rendelni. Így születtek meg a döntési leképezésekről szóló, a dolgozat harmadik fejezetében bemutatott, közös kutatási eredmények, melyeket a [3], [4], [5] dolgozatokban publikáltunk és szerzőtársaimmal nem megosztható eredményeknek tekintjük őket.

A negyedik fejezetben is a probléma struktúráját fogjuk tanulmányozni, ebben az esetben azonban a probléma méretének csökkentése a cél, amitől azt reméljük, hogy a probléma megoldásának hatékonyságát is növelni tudja. A fejezet a [20] dolgozat eredményeire épül, amely közös dolgozat, de a szerzőnek meghatározó szerepe volt az eredmények elérésében. Hogy a probléma méretének csökkentésére kidolgozott módszer megfelelő alkalmazása valóban növelheti-e a megoldás hatékonyságát, arra választ kapunk az ötödik fejezetben, melyben az előzőleg már létező, az Előzményekben bizonyított létjogosultságú, exponenciális bonyolultságú korlátozás és szétválasztás típusú módszereket igyekszünk javítani a negyedik fejezetben már bemutatott ötletek és a módszerek egyes részeinek mélyebbre ható tanulmányozásának segítségével. A fejezet a szerző saját munkájának eredményeit tartalmazza, melyek a [21]-ben kerültek publikálásra.

A hatodik fejezetben az összes optimális megoldás felsorolására fogunk adni az egyetlen létező és nem kifejezetten erre a célra kidolgozott eljárásnál sokkal hatékonyabb algoritmust. A fejezet a szerző saját eredményeire épül, melyek a [23]-ben nyertek publikálást.

Végül, az utolsó fejezetben, a kombinatorikus optimalizálás és az automataelmélet érdekes és hasznos összekapcsolására fogunk példát mutatni, amikor a PNS probléma optimális megoldását egy hozzárendelt automata átmeneti gráfiában való legrövidebb út keresésével fogjuk meghatározni. A módosított PNS probléma és annak alapvető megoldása után, a 7.3. alfe-

jezetben egy komplexebb, a szerző által kidolgozott és a [22]-ban publikált megoldó eljárást fogunk bemutatni.

2. fejezet

Előzmények

2.1. A PNS probléma

Jelölje $\varphi(H)$ egy tetszőleges H halmaz összes részhalmazát, $\varphi'(H)$ pedig a H halmaz összes nemüres részhalmazát. Legyenek M és $O \subseteq \varphi'(M) \times \varphi'(M)$ véges, nemüres, és diszjunkt halmazok. Az M elemei az anyagok, míg az O elemei a műveleti egységek, melyek segítségével bizonyos bemenő anyagokból nyerünk előírt módon egy kimeneti anyaghalmazt. Hogy mi megy végbe a műveleti egységekben, azzal nem foglalkozunk. Figyelmen kívül hagyjuk továbbá azt is, hogy miként kezdett a rendszer működni, csak statikus „termeléssel” foglalkozunk. Formálisan bármely $u \in O$ műveleti egységre $u = (\alpha, \beta)$, ahol az α a bemeneti (nem üres) anyaghalmaz, β pedig a kimeneti (nem üres) anyaghalmaz. Azt fogjuk mondani, hogy az u műveleti egység az α anyaghalmazból a β anyaghalmazt gyártja.

2.1.1. Definíció. Az (M, O) párhoz egyérelműen hozzárendelhető egy gráf, amit a **folyamat gráfjának** nevezünk: $PG(M, O) = (M \cup O, A_1 \cup A_2)$, ahol az élhalmaz kétféle típusú élből áll,

$$A_1 = \{(X, Y) : Y = (\alpha, \beta) \in O \text{ és } X \in \alpha\},$$

$$A_2 = \{(Y, X) : Y = (\alpha, \beta) \in O \text{ és } X \in \beta\}.$$

2.1.2. Definíció. Egy (V', E') gráf egy $PG(M, O)$ folyamat gráf **részgráfja**, ha:

- $V' = M' \cup O'$, $M' \subseteq M$, $O' \subseteq O$
- $O' \subseteq \varphi'(M') \times \varphi'(M')$
- $E' = A'_1 \cup A'_2$, ahol

$$A'_1 = \{(X, Y) : Y = (\alpha, \beta) \in O' \text{ és } X \in \alpha\},$$

$$A'_2 = \{(Y, X) : Y = (\alpha, \beta) \in O' \text{ és } X \in \beta\}.$$

2.1.1. Megjegyzés. Vegyük észre, hogy adott (M, O) pár és a hozzá rendelt folyamat gráf kölcsönösen egyértelműen meghatározzák egymást. Ezért a továbbiakban az (M, O) párokat azonosítani fogjuk a hozzájuk rendelt folyamat gráfokkal.

Egy $X \in M$ anyag **forrás** (M, O) -ban, ha nem létezik (Y, X) él a folyamat gráfban. Ha léteznek X_1, X_2, \dots, X_n csúcsok a gráfban, melyekre $(X_1, X_2), (X_2, X_3), \dots, (X_{n-1}, X_n)$ élek az (M, O) folyamat gráfban, akkor az ezen csúcsok által meghatározott **utat** $[X_1, X_n]$ -el fogjuk jelölni.

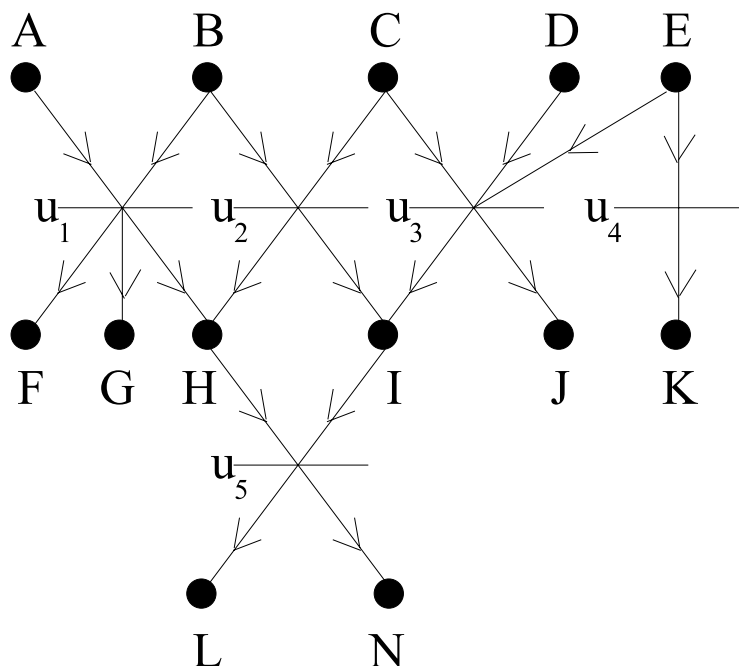
Legyen most $P \subseteq M$ és $R \subseteq M$ az **előállítandó anyagok** és a **felhasználható nyersanyagok** egymástól diszjunkt halmaza. Az előállítandó anyagokra szinonimaként fogjuk használni a **céltermék** vagy **végtermék** szavakat, a felhasználható nyersanyagokat pedig egyszerűen **nyersanyagoknak** nevezzük. Akkor az $\mathbf{M}=(P, R, O)$ hármast a tekintett PNS-probléma **strukturális modelljének** nevezzük.

2.1.1. Példa. Legyen $\mathbf{M}=(P, R, O)$, melyben

- $P = \{L, N\}$,
- $R = \{A, B, C, D, E\}$,
- $O = \{u_1, u_2, u_3, u_4, u_5\}$, ahol

- $u_1 = (\{A, B\}, \{F, G, H\})$,
- $u_2 = (\{B, C\}, \{H, I\})$,
- $u_3 = (\{C, D, E\}, \{I, J\})$,
- $u_4 = (\{E\}, \{K\})$, és
- $u_5 = (\{H, I\}, \{L, N\})$.

Ekkor $M = \{A, B, C, D, E, F, G, H, I, J, K, L, N\}$ és az (M, O) folyamat gráfot a 2.1. ábra szemlélteti.



2.1. ábra.

2.1.3. Definíció. Legyen adott egy $M = (P, R, O)$ strukturális modell és legyen $o \subseteq O$ műveleti egységek egy halmaza. Ekkor egy (m, o) részgráfot az M strukturális modell egy **lehetséges megoldás struktúrájának** nevezünk, ha teljesülnek a következő tulajdonságok:

$$(A1) \quad P \subseteq m,$$

(A2) $\forall X \in m, X \in R \Leftrightarrow$ nem létezik (Y, X) él (m, o) -ban,

(A3) $\forall Y_0 \in o, \exists [Y_0, Y_n]$ út, amelyre $Y_n \in P$,

(A4) $\forall X \in m, \exists(\alpha, \beta) \in o$ úgy, hogy $X \in \alpha \cup \beta$.

Jelölje $S(\mathbf{M})$ az \mathbf{M} strukturális modell lehetséges megoldás struktúráinak halmazát. Egy lehetséges megoldás struktúráját tehát úgy képzelhetünk el, mint a folyamat gráfjának egy olyan részhálózatát, melyben:

- szerepelnek az előállítandó anyagok,
- nyersanyagokat nem gyártunk, és minden nem nyersanyagot gyártja valamelyik műveleti egységünk,
- csak olyan műveleti egységet működtetünk, amely legalább közvetve részt vesz valamelyik előállítandó anyag gyártásában, illetve
- nincs izolált anyagi pont a részgráfban.

2.1.2. Példa. Tekintsük a 2.1.1 példát. Akkor

- $(m_1, o_1) = (\{A, B, F, G, H, C, D, E, I, J, L, N\}, \{u_1, u_3, u_5\})$,
- $(m_2, o_2) = (\{B, C, H, I, L, N\}, \{u_2, u_5\})$
- $(m_3, o_3) = (\{A, B, F, G, H, C, I, L, N\}, \{u_1, u_2, u_5\})$,
- $(m_4, o_4) = (\{B, C, D, E, H, I, J, L, N\}, \{u_2, u_3, u_5\})$
- $(m_5, o_5) = (\{A, B, C, D, E, F, G, H, I, J, L, N\}, \{u_1, u_2, u_3, u_5\})$

az összes lehetséges megoldás struktúrák, míg például

- $(m_6, o_6) = (\{C, D, E, H, I, J, L, N\}, \{u_3, u_5\})$ és
- $(m_7, o_7) = (\{B, C, H, I, E, K, L, N\}, \{u_2, u_4, u_5\})$

nem lehetséges megoldás struktúrák az $(\mathcal{A}2)$, illetve az $(\mathcal{A}3)$ feltételek nem teljesülése miatt.

2.1.2. Megjegyzés. Az M és O halmazok végeessége miatt $S(\mathbf{M})$ is véges halmaz.

2.1.3. Megjegyzés. Általános esetben az $(\mathcal{A}1) - (\mathcal{A}4)$ feltételeket teljesítő lehetséges megoldás struktúrák halmaza üres halmaz is lehet: a 2.1.1. példában, ha $B \notin R$ lenne (azaz nem lenne nyersanyag), akkor $S(\mathbf{M}) = \emptyset$ -t kapnánk.

2.1.1. Lemma. ([15]) *Legyen $\mathbf{M}=(P, R, O)$ egy PNS probléma strukturális modellje. Ha (m, o) és (m', o') lehetséges megoldás struktúrái \mathbf{M} -nek, akkor $(m, o) \cup (m', o')$ is lehetséges megoldás struktúrája \mathbf{M} -nek.*

2.1.1. Következmény. *A 2.1.2. megjegyzés és 2.1.1. lemma alapján $S(\mathbf{M})$ összes lehetséges megoldás struktúráinak egyesítése is lehetséges megoldás struktúra lesz.*

2.1.4. Definíció. *Legyen $\mathbf{M}=(P, R, O)$ egy PNS probléma strukturális modellje. \mathbf{M} **maximális struktúrája** alatt a*

$$\mu(\mathbf{M}) = \bigcup_{(m,o) \in S(\mathbf{M})} (m, o)$$

*megoldás struktúrákat értjük. Ha $S(\mathbf{M}) = \emptyset$, akkor $\mu(\mathbf{M}) = \emptyset$ és $\mu(\mathbf{M})$ -et **degeneráltnak** nevezzük.*

2.1.4. Megjegyzés. $S(\mathbf{M}) \neq \emptyset$ akkor és csakis akkor, ha $\mu(\mathbf{M}) \neq \emptyset$.

2.1.5. Definíció. *Legyen $o \subseteq O$ műveleti egységek egy halmaza. Defináljuk a mat^{in} , mat^{out} , és mat függvényeket a következőképpen:*

$$mat^{in}(o) : \varphi'(O) \rightarrow \varphi'(M), \quad mat^{in}(o) = \bigcup_{(\alpha, \beta) \in o} \alpha,$$

$$mat^{out}(o) : \varphi'(O) \rightarrow \varphi'(M), \quad mat^{out}(o) = \bigcup_{(\alpha, \beta) \in o} \beta,$$

és

$mat(o) : \varphi''(O) \rightarrow \varphi'(M), \quad mat(o) = mat^{in}(o) \cup mat^{out}(o).$
Szemléletesen a műveleti egységek egy o halmazához tartozó input anyagok, illetve output anyagok egyesítéséről van szó.

2.1.2. Lemma. ([15]) *Legyen $\mathbf{M}=(P, R, O)$ egy PNS probléma strukturális modellje. Ha $(m, o) \in S(\mathbf{M})$, akkor $m = mat(o)$.*

2.1.2. Következmény. ([15]) *Egy (m, o) lehetséges megoldás struktúra egyértelműen meghatározott az o műveleti egység halmazzal.*

2.2. A PNS probléma redukciója

Az eddigiekből nyilvánvaló, hogy egy $\mathbf{M}=(P, R, O)$ strukturális modell egyértelműen meghatározza az $S(\mathbf{M})$ lehetséges megoldás struktúra halmazt. Ez fordítva azonban nem igaz: különböző strukturális modellek rendelkezhetnek azonos megoldás struktúra halmazzal. Például ha egy modellhez felvesszünk olyan további műveleti egységeket, melyeknek bemeneti és kimeneti anyaghalmazaik diszjunktak az eredeti modell anyaghalmazaitól, akkor az így kapott strukturális modell ugyanazzal a megoldás struktúra halmazzal fog rendelkezni. Gyakorlati szempontból fontos lenne tehát megtalálni azt a legkisebb méretű, és így legkönnyebben kezelhető megoldás struktúrát, mely tartalmazza az eredeti probléma összes lehetséges megoldás struktúráját.

2.2.1. Definíció. *Jelöljük \mathcal{M} -el a PNS problémák strukturális modelljeinek halmazát. Azt mondjuk, hogy az $\overline{\mathbf{M}} = (\overline{P}, \overline{R}, \overline{O}) \in \mathcal{M}$, és az $\mathbf{M}' = (P', R', O') \in \mathcal{M}$ strukturális modellek **ekvivalensek**, és ezt $\overline{\mathbf{M}} \sim \mathbf{M}'$ -vel jelöljük, ha $\overline{P} = P'$ és $S(\overline{\mathbf{M}}) = S(\mathbf{M}')$.*

A \sim reláció reflexív, tranzitív és szimmetrikus, azaz ekvivalencia reláció. Mivel gyakorlatilag csak a \mathcal{M} nem üres megoldás struktúra halmazzal rendelkező ekvivalencia osztályai érdekesek, ezért a továbbiakban ilyen

osztályokat fogunk tanulmányozni. Tetszőleges G' ilyen osztályra és $\mathbf{M} = (P, R, O) \in G'$, illetve $\mathbf{M}' = (P', R', O') \in G'$ strukturális modellekre definiáljuk a \triangleleft relációt a következőképpen: $\mathbf{M} \triangleleft \mathbf{M}'$ akkor és csak akkor, ha $(M, O) \subseteq (M', O')$ és $R \subseteq R'$, ahol (M, O) és (M', O') az \mathbf{M} illetve \mathbf{M}' strukturális modellek folyamat gráfjai. Nyilvánvalóan a \triangleleft reláció reflexív, antiszimmetrikus és tranzitív, azaz részben rendezés G' -n.

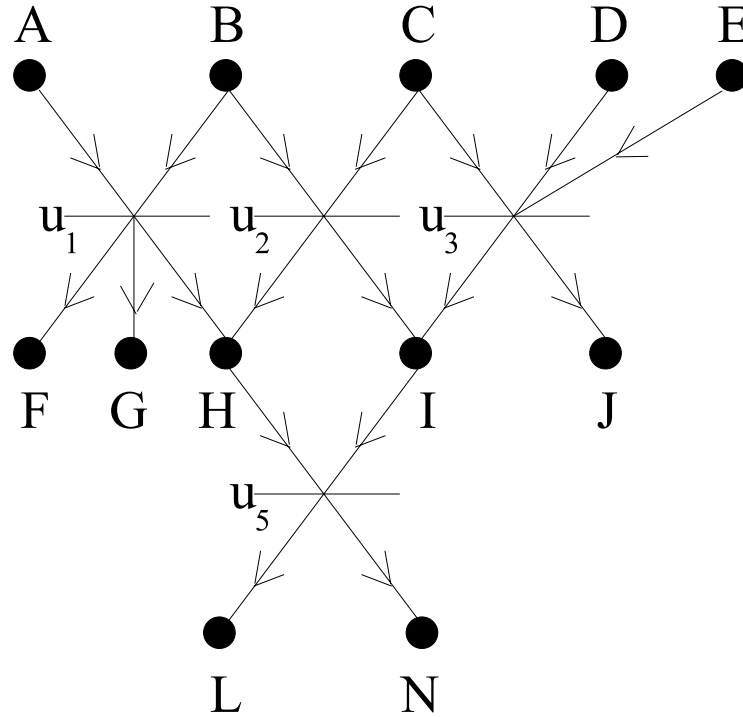
2.2.1. Példa. Legyen $\mathbf{M}' = (P', R', O')$, melyben

- $P' = \{L, N\}$,
- $R = \{A, B, C, D, E\}$,
- $O' = \{u_1, u_2, u_3, u_5\}$, ahol
 - $u_1 = (\{A, B\}, \{F, G, H\})$,
 - $u_2 = (\{B, C\}, \{H, I\})$,
 - $u_3 = (\{C, D, E\}, \{I, J\})$, és
 - $u_5 = (\{H, I\}, \{L, N\})$.

A folyamat gráfját a 2.2. ábra szemlélteti. Könnyen ellenőrizhető, hogy $P' = P$, $R' = R$, $O' \subseteq O$, továbbá a folyamatok lehetséges megoldás strukturái azonosak a 2.1.2. példában leírtakkal. Mindebből az következik, hogy $\mathbf{M}' \triangleleft \mathbf{M}$. Érdekes továbbá megfigyelni azt is, hogy a példák maximális strukturái is azonosak: $\mu(\mathbf{M}) = \mu(\mathbf{M}') = (m_5, o_5)$.

Szeretnénk meghatározni adott ekvivalencia osztály egy minimális elemét a \triangleleft részben rendezésre nézve. Legyen G' egy nem üres lehetséges megoldás struktúra halmazzal rendelkező ekvivalencia osztály. Legyen $\mathbf{M} = (P, R, O) \in G'$. Mivel $S(\mathbf{M}) \neq \emptyset$, ezért a 2.1.4. megjegyzés alapján $\mu(\mathbf{M})$ nem degenerált. Akkor $\mu(\mathbf{M}) = (\overline{M}, \overline{O})$ -re képezzük az $\overline{\mathbf{M}} = (P, \overline{R}, \overline{O})$ hármast, ahol $\overline{R} = R \cap \overline{M}$.

2.2.1. Lemma. A fenti módon meghatározott $\overline{\mathbf{M}} = (P, \overline{R}, \overline{O})$ egy PNS probléma olyan strukturális modellje, mely ekvivalens \mathbf{M} -el és $\overline{\mathbf{M}} \triangleleft \mathbf{M}$.



2.2. ábra.

2.2.1. Következmény. ([15]) Minden $M \in G'$ -re $\overline{M} \triangleleft M$ teljesül, ami azt jelenti, hogy \overline{M} az ekvivalencia osztály legkisebb eleme.

Az ekvivalencia osztály legkisebb eleme a folyamat gráf olyan minimális részgráfjának felel meg, mely tartalmazza az összes lehetséges megoldás struktúrát, de nem tartalmaz szükségtelen információt. Ennek megtalálása azért hasznos, mert csökkenti a probléma méretét.

2.2.2. Definíció. Egy PNS probléma $\mathbf{M} = (P, R, O)$ strukturális modelljét a tekintett PNS probléma **redukált strukturális modelljének** nevezzük, ha $S(\mathbf{M}) \neq \emptyset$, és bármely más $\mathbf{M}' \sim \mathbf{M}$ strukturális modellre $\mathbf{M} \triangleleft \mathbf{M}'$.

Tehát egy G' ekvivalencia osztály egy nem üres lehetséges megoldás struktúra halmazzal rendelkező $\mathbf{M} = (P, R, O)$ strukturális modelljéből

kiindulva, a $\mu(\mathbf{M}) = (\overline{M}, \overline{O})$ maximális struktúra ismeretében, az \overline{M} redukált struktúrát az $\overline{M} = (P, \overline{R}, \overline{O})$ módon kaphatjuk meg, ahol $\overline{R} = R \cap \overline{M}$. Ehhez azonban tetszőleges strukturális modellről el kellene tudnunk dönteni, hogy a lehetséges megoldás struktúrák halmaza üres halmaz-e, és ha nem, akkor meg kellene tudnunk határozni a maximális struktúrát. Ennek megvalósítását fogjuk bemutatni a következő alfejezetben.

2.3. A maximális struktúra meghatározása

A vizsgálatok során bebizonyosodott, hogy a lehetséges megoldás struktúra halmaz ürességének eldöntésére és a maximális struktúra generálására adható egy hatékony, polinomiális idejű algoritmus ([16, 11]). További empirikus vizsgálatok megmutatták, hogy véletlenszerűen generált feladatokra a redukciós eljárás bizonyos, átlagosnak tekinthető feladatosztályoknál közelítőleg 47%-ára csökkenti a feladat méretét ([25]). Ebben az alfejezetben a ([11])-ben leírt algoritmmust fogjuk ismertetni.

Legyen $\mathbf{M} = (P, R, O)$ egy PNS probléma strukturális modellje.

Maximális struktúra generáló algoritmus (MSG)

1. Redukció

Inicializálás

- 1.1. Legyen $O_0 = O \setminus \{(\alpha, \beta) : (\alpha, \beta) \in O \ \& \ \beta \cap R \neq \emptyset\}$ és $M_0 = \text{mat}(O_0)$.
- 1.2. Ha $P \not\subseteq M_0$, akkor nem létezik \mathbf{M} -re maximális struktúra és az algoritmus véget ér. Egyébként folytassuk az eljárást a következő lépéssel.
- 1.3. Legyen $T_0 = \{X : X \in M_0 \setminus R \ \& \ ((\alpha, \beta) \in O_0 \longrightarrow X \notin \beta)\}$.
- 1.4. Legyen $r = 0$.

Iteráció

- 1.5. Ha $T_r = \emptyset$, akkor folytassuk az eljárást a 2.1. lépéssel.
- 1.6. Válasszunk egy $X \in T_r$ anyagot.
- 1.7. Legyen $O_X = \{(\alpha, \beta) : (\alpha, \beta) \in O_r \ \& \ X \in \alpha\}$.
- 1.8. Legyen $O_{r+1} = O_r \setminus O_X$ és $M_{r+1} = \text{mat}(O_{r+1})$.
- 1.9. Ha $P \not\subseteq M_{r+1}$, akkor nem létezik \mathbf{M} -re maximális struktúra és az algoritmus véget ér. Egyébként folytassuk az eljárást a következő lépéssel.
- 1.10. Határozzuk meg a
- $$T'_r = \{Y : Y \in \text{mat}^{out}(O_X) \ \& \ Y \notin \text{mat}^{out}(O_{r+1}) \ \& \ Y \in \text{mat}^{in}(O_{r+1})\}$$
- halmazt.
- 1.11. Legyen $T_{r+1} = (T_r \cap M_{r+1}) \cup T'_r$.
- 1.12. Növeljük eggyel az r iterációs számot.
- 1.13. Kezdjünk egy új iterációt az 1.5. lépéssel.

2. Építés

Inicializálás

- 2.1. Legyen $W_0 = P$, $m_0 = \emptyset$, $o_0 = \emptyset$ és $s = 0$.

Iteráció

- 2.2. Ha $W_s = \emptyset$, akkor vége: kaptunk egy megoldás struktúrát \mathbf{M} -re.
 Ha $\bar{m} = \text{mat}(o_s)$, akkor (\bar{m}, o_s) az \mathbf{M} maximális struktúrája.
 Ha $W_s \neq \emptyset$, akkor folytassuk az eljárást a következő lépéssel.

- 2.3. Válasszunk egy teszőleges X anyagot W_s -ből.
- 2.4. Legyen $m_{s+1} = m_s \cup \{X\}$.
- 2.5. Alkossuk meg az $O_X^* = \{(\alpha, \beta) : (\alpha, \beta) \in O_r \ \& \ X \in \beta\}$ halmazt.
- 2.6. Legyen $o_{s+1} = o_s \cup O_X^*$.
- 2.7. $W_{s+1} = (W_s \cup \text{mat}^{in}(O_X^*)) \setminus (R \cup m_{s+1})$.
- 2.8. Növeljük eggyel az s iterációs számot.
- 2.9. Kezdjük egy új iterációt a 2.2. lépéssel.

Az algoritmus két fő részből áll. Az első redukciós részben töröljük azokat a műveleti egységeket, melyek vagy nyersanyagot is termelnek, vagy valamely nyersanyagtól különböző, egyetlen műveleti egység által sem termelt bemeneti anyag hiányában nem tudnának működni. Ha közben azt tapasztaljuk, hogy valamely célterméket egyetlen megmaradt műveleti egységünk sem gyártja, ez azt jelenti, hogy nincs lehetséges megoldás struktúra, azaz $S(\mathbf{M}) = \emptyset$, de akkor maximális struktúra sincs, így az algoritmust megállíthatjuk.

A második részben a rendelkezésre álló műveleti egységekből felépítjük azt a hálózatot, mely kizárólag hasznos anyagokat termelő műveleti egységeket tartalmaz. Először kiindulunk abból, hogy a céltermékeket le kell gyártani. Ehhez minden olyan műveleti egység hasznos lehet, mely célterméket gyárt, tehát ezeket bevesszük a hálózatba. De a hálózatba bevett műveleti egységek bemeneti anyagait is le kellene gyártani, tehát bevesszük az azokat gyártó műveleti egységeket is, és így tovább. Természetesen általános esetben megtörténhet, hogy a céltermékek gyártása az ily módon meghatározott műveleti egységek akár több különböző valódi rész-halmazával is legyárthatók, most az volt a cél, hogy összegyűjtsük azokat a műveleti egységeket, melyek részt vehetnek valamely lehetséges megoldás struktúrában.

Összesítve, az algoritmus eldönti az " $S(\mathbf{M}) = \emptyset$?" kérdést, és amennyiben létezik lehetséges megoldás struktúra, az eredményeképpen kapott hálózat azokat és csakis azokat a műveleti egységeket és anyagokat tartalmazza, amelyek részt vehetnek valamely lehetséges megoldás struktúrában,

így a modell maximális struktúráját szolgáltatja. Lényeges továbbá az is, hogy az algoritmus polinomiális idő alatt oldja meg ezt a feladatot ([11]).

2.4. A súlyozott PNS modell

Legyen $\mathbf{M} = (P, R, O)$ egy PNS probléma strukturális modellje. Gyakorlati szempontból természetes igény, hogy szeretnénk meghatározni azt a lehetséges megoldás struktúrát, mely valamilyen szempontból a leggazdaságosabban állítja elő a céltermékeket. Definiálunk tehát a lehetséges megoldás struktúrák halmazán egy költségfüggvényt és keresünk egy legkisebb költségű lehetséges megoldást. Legyen $z : S(\mathbf{M}) \rightarrow R_+$ egy ilyen költségfüggvény. Akkor a megoldandó feladat:

$$(PNS-1) \quad \min\{z((m, o)) : (m, o) \in S(\mathbf{M})\}.$$

Egy megoldás struktúra költségfüggvényét többféleképpen definiálhatjuk. Mi most ennek egy egyszerű és természetes definícióját fogjuk adni.

Legyen $w : O \rightarrow R_+$ egy költségfüggvény a műveleti egységeken. Egy lehetséges megoldás struktúra költségét a benne levő műveleti egységek összköltségéként fogjuk definiálni. Így a PNS probléma azon változatát vizsgáljuk, amikor a feladat egy olyan lehetséges megoldás struktúra meghatározása, melyre a benne működő műveleti egységek súlya minimális, vagyis

$$(PNS-2) \quad \min\{\sum_{u \in o} w(u) : (m, o) \in S(\mathbf{M})\}.$$

Az a kérdés, hogy meg tudjuk-e oldani ezt a feladatot, ha igen hogyan, és milyen hatékonysággal ?

Mivel az M és O halmazok végeessége miatt a lehetséges megoldás struktúrák száma is véges, továbbá adott lehetséges megoldás struktúra fentiekben definiált költségének kiszámítása is véges idő alatt elvégezhető, ezért nyilván a legkisebb költségű lehetséges megoldás struktúra meghatározása is véges időben megoldható feladat. Például egy lehetőség, hogy felsoroljuk a strukturális modellben szereplő O halmaz összes részalmazát, mindegyikről megvizsgáljuk, hogy lehetséges megoldás struktúra-e, és a lehetsé-

ges megoldás struktúrák közül kiválasztjuk a legkisebb költségűt. Könnyen belátható azonban erről a módszerről, hogy nagyon kicsi a hatékonysága, hiszen túl sok részhalmazt vizsgálunk meg feleslegesen. Az előzőekben tárgyaltak ismeretében azonnal látjuk, hogy csak a maximális struktúra rész-halmazzaival érdemes foglalkozni, de még ezek felsorolása is nagyon sok, az O műveleti egység halmaz számosságának függvényében exponenciális számú részhalmaz megvizsgálását igényli. A következő fejezetekben látni fogjuk, hogy ennél hatékonyabb megoldást ugyan lehet találni, de sokkal hatékonyabb, polinomiális idejű megoldás megtalálása nem várható.

2.5. A PNS probléma NP teljessége

A [2, 17, 27] cikkekben igazolást nyert a a PNS-2 probléma NP teljessége. A bizonyítás alapötlete az volt, hogy megmutatták a PNS-2 problémák egy részosztályának ekvivalenciáját a halmazlefedési problémával, mely egy jól ismert NP teljes probléma ([32, 1]). Az eredmény fontossága és a két probléma ekvivalenciájának érdekessége miatt az alábbiakban ismertetjük a bizonyítások lényegét.

Először a PNS-2 probléma NP nehézségét fogjuk megmutatni. Tekintsük a PNS-2 problémák azon PNS_{RP} részosztályát, melyben $O \subseteq \varphi'(R) \times \varphi'(P)$, azaz melyben minden műveleti egység nyersanyagokból céltermékeket gyárt, és ezek párhuzamosan dolgoznak. Legyen tehát $O = \{u_1, \dots, u_n\}$, $u_j = (\alpha_j, \beta_j) \in \varphi'(R) \times \varphi'(P)$, $j = 1, \dots, n$. Ha a β_j halmazok súlyozását a következőképpen definiáljuk: $w'(\beta_j) = w(u_j)$, akkor könnyen belátható, hogy a PNS_{RP} probléma ekvivalens a P halmaz β_j halmazokkal való halmazlefedési problémájával.

Fordítva, tekintsünk egy tetszőleges halmazlefedési problémát. Ez azt jelenti, hogy egy tetszőleges nem üres, véges, P halmazt le szeretnénk fedni súlyozott β_j , $j = 1, \dots, n$ halmazokkal, minél kisebb összköltséggel. Jelölje a halmazok súlyozását $w'(\beta_j)$, $j = 1, \dots, n$. Legyen $R \neq \emptyset$ egy tetszőleges véges halmaz, melyre $R \cap P = \emptyset$. Legyenek továbbá $u_j = (R, \beta_j)$, $j = 1, \dots, n$, és $O = \{u_1, \dots, u_n\}$. Definiáljunk egy w súlyfüggvényt O -n a következőképpen: $w(u_j) = w'(\beta_j)$, $j = 1, \dots, n$. Ha a P és R halmazokat anyaghalmazoknak, az O -t pedig műveleti egységek halmazának tekintjük,

akkor a (P, R, O) által meghatározott PNS_{RP} probléma ekvivalens a tekintett halmazlefedési problémával.

A fentiekből következik, hogy:

2.5.1. Lemma. ([2, 17]) *A PNS_{RP} problémaosztály ekvivalens a halmazlefedési problémával.*

Mivel azonban a PNS_{RP} problémaosztály a PNS-2 problémának egy részosztálya, a halmazlefedési probléma pedig NP teljes, ezért a következő állítást kapjuk.

2.5.1. Tétel. ([2, 17]) *A PNS-2 probléma NP nehéz.*

Egy NP nehéz probléma NP teljességéhez azt kellene megmutatni, hogy eleme az NP osztálynak. Ha veszünk egy feltételezett optimális megoldást, annak az ellenőrzése, hogy valóban megfelelő költségű lehetséges megoldás struktúra-e, polinomiális időben elvégezhető. A ([27]) cikkben azonban a szerzők egy sokkal szebb, konstruktív megoldást adtak, melyet az alábbiakban vázolunk.

Abból indulunk ki, hogy a lehetséges megoldás struktúrák mind a maximális struktúra részei, és így módon, ha $(\overline{M}, \overline{O})$ az $\mathbf{M} = (P, R, O)$ struktúrális modell maximális struktúrája, akkor az eredeti súlyozott PNS probléma ekvivalens a

$$(PNS-3) \quad \min \left\{ \sum_{u \in o} w(u) : (m, o) \in S(P, R \cap \overline{M}, \overline{O}) \right\}$$

feladattal.

Ha $S(P, R, O) = \emptyset$, akkor a PNS probléma ekvivalens azzal a halmazlefedési problémával, melyben a P halmazt kell lefedni annak egy valódi, tetszőleges súlyú részhalmazával: egyiknek sincs lehetséges megoldása. Továbbá, az " $S(P, R, O) = \emptyset$?" kérdés, a már említettek szerint, polinomiális időben eldönthető ([11]).

Ha $S(P, R, O) \neq \emptyset$, akkor felhasználva a 2.1.2. következménybeli megállapítást, mi szerint egy műveleti egység halmaz egyértelműen meghatároz egy lehetséges megoldás struktúrát, a lehetséges megoldás struktúrát egy logikai konjunktív normál formával fogjuk jellemezni.

Legyen $\overline{O} = \{(\alpha_1, \beta_1), \dots, (\alpha_l, \beta_l)\}$ és $J = \{1, \dots, l\}$. Akkor az $(\overline{M}, \overline{O})$ minden (m, o) folyamat részgráfjához hozzárendelhetünk egy y_1, \dots, y_l logikai vektort úgy, hogy minden $j \in J$ -re $y_j = IGAZ \iff (\alpha_j, \beta_j) \in o$. Könnyen belátható, hogy ez egy bijektív leképezés az $(\overline{M}, \overline{O})$ folyamat $(\mathcal{A4})$ -et kielégítő részgráfjai és a hozzájuk rendelt logikai vektorok között. Egy \mathbf{y} logikai vektornak megfelelő (m, o) folyamat részgráf a következőképpen határozható meg:

$$m = \bigcup_{j \in T(\mathbf{y})} (\alpha_j \cup \beta_j) \text{ és } o = \{(\alpha_j, \beta_j) : j \in T(\mathbf{y})\},$$

ahol $T(\mathbf{y}) = \{j : j \in J \text{ és } y_j = IGAZ\}$. Azonban tetszőleges \mathbf{y} vektorhoz rendelt folyamat részgráf nem feltétlen lehetséges megoldás struktúra is. Az alábbiakban meghatározunk egy olyan Φ konjunktív normál formát, amelyet egy \mathbf{y} logikai vektor akkor és csak akkor elégít ki, ha a hozzárendelt folyamat részgráf lehetséges megoldás struktúra is. Legyenek:

- $\Phi_0 = \bigwedge_{X \in P} \bigvee_{\substack{j \in J \\ X \in \beta_j}} y_j$
- $\Phi_1 = \bigwedge_{\substack{j \in J \\ X \in \alpha_j \setminus R}} (\neg y_j \vee \bigvee_{\substack{h \in J \\ X \in \beta_h}} y_h)$
- $\Phi_2 = \bigwedge_{\substack{j \in J \\ P \cap \beta_j = \emptyset}} (\neg y_j \vee \bigvee_{\substack{h \in J \\ \beta_j \cap \alpha_h \neq \emptyset}} y_h)$
- $\Phi = \Phi_0 \wedge \Phi_1 \wedge \Phi_2$

2.5.2. Lemma. ([27]) *Egy \mathbf{y} logikai l -vektor akkor és csak akkor elégíti ki a Φ konjunktív normál formát, ha az \mathbf{y} -hoz rendelt folyamat gráf egy lehetséges megoldás struktúra.*

Bizonyítás Legyen (m, o) egy tetszőleges $(\overline{M}, \overline{O})$ -beli lehetséges megoldás struktúra és \mathbf{y} a hozzá rendelt logikai vektor. Mivel $m = mat(o)$, $(\mathcal{A1})$ és

($\mathcal{A}2$) alapján minden P -beli anyagot gyárt valamilyen o -beli műveleti egység, így $\Phi_0(\mathbf{y}) = IGAZ$. ($\mathcal{A}2$)-ből nyilván következik $\Phi_1(\mathbf{y}) = IGAZ$ is.

Bizonyítani szeretnénk Φ_2 teljesülését is. Legyen $j \in J$ úgy, hogy $\beta_j \cap P = \emptyset$. Akkor $\neg y_j \vee \bigvee_{\substack{h \in J \\ \beta_j \cap \alpha_h \neq \emptyset}} y_h$ tagja a Φ_2 -nek. Ha $u_j \notin o$, akkor $y_j = HAMIS$ és az előbbi diszjunkció logikai értéke $IGAZ$. Meg kell még mutatnunk, hogy akkor is igaz, ha $u_j \in o$, azaz $y_j = IGAZ$. Ebben az esetben ($\mathcal{A}3$) alapján létezik egy (m, o) -beli út u_j -ből P -be, ami miatt létezik olyan $u_h \in o$, melyre $\beta_j \cap \alpha_h \neq \emptyset$. Ez viszont azt jelenti, hogy $y_h = IGAZ$, amiből az következik, hogy az őt tartalmazó diszjunkció is igaz. Mivel a $j \in J$ -t teszőlegesen választottuk a $\beta_j \cap P = \emptyset$ feltétel mellett, így a Φ_2 konjunkció minden tagja $IGAZ$ lesz, amiből következik Φ_2 teljesülése.

A fentiekben megmutattuk tehát, hogy Φ_0 , Φ_1 és Φ_2 mindegyike $IGAZ$, amiből következik Φ teljesülése.

A másik irány bizonyításához most tegyük fel, hogy \mathbf{y} kielégíti Φ -t. Legyen (m, o) az $(\overline{M}, \overline{O})$ \mathbf{y} -hoz rendelt folyamat részgráfja. Igazolni szeretnénk, hogy (m, o) lehetséges megoldás struktúra, azaz kielégíti az ($\mathcal{A}1$) - ($\mathcal{A}4$) feltételeket.

($\mathcal{A}4$) az \mathbf{y} definíciójából közvetlenül következik.

$\Phi_0(\mathbf{y}) = IGAZ$ miatt minden $X \in P$ -re létezik olyan $u \in O$ műveleti egység, mely X -et közvetlenül termeli, így ($\mathcal{A}1$) is teljesül.

($\mathcal{A}2$) bizonyításához legyen $X \in m \cap R$. Mivel $o \subseteq \overline{O}$ és az $(\overline{M}, \overline{O})$ maximális struktúrában a nyersanyagokat egyetlen \overline{O} -beli műveleti egység sem gyártja, így nem létezhet (Y, X) él az (m, o) -ban. Most feltételezzük, hogy $X \in m$ és nem létezik (Y, X) él (m, o) -ban. Bizonyítani szeretnénk, hogy $X \in R$. Feltételezzük az ellenkezőjét, azt hogy $X \notin R$. Mivel $X \in m$, a Φ -hez rendelt (m, o) definíciója miatt léteznie kell $u_i = (\alpha_i, \beta_i) \in o$ -nak úgy, hogy $X \in \alpha_i$, ami azt jelenti, hogy $y_i = IGAZ$. Másfelől az $(\overline{M}, \overline{O})$ maximális struktúra is a 2.1.1. következmény alapján egy lehetséges megoldás struktúra, melyben feltételezésünk szerint X nem nyersanyag, így létezniük kell olyan $u_h \in \overline{O}$ műveleti egységeknek, melyekre $X \in \beta_h$. Mivel a Φ_1 a maximális struktúra minden nem csak nyersanyag bemenettel rendelkező műveleti egységére tartalmaz egy tagot, ezért Φ_1 tartalmazni fogja

a $(\neg y_i \vee \bigvee_{\substack{h \in J \\ X \in \beta_h}} y_h)$ tagot is. Mivel feltételezésünk szerint $\Phi = IGAZ$, ezért

$\Phi_1 = IGAZ$, de akkor az $(\neg y_i \vee \bigvee_{\substack{h \in J \\ X \in \beta_h}} y_h)$ tagnak is $IGAZ$ -nak kell lennie.

Ugyanakkor a fentiekben azt kaptuk, hogy $y_i = IGAZ$. Ez azt jelenti, hogy $\bigvee_{\substack{h \in J \\ X \in \beta_h}} y_h$ is $IGAZ$, amiből az következik, hogy létezik olyan h_0 , melyre $u_{h_0} \in o$ és $X \in \beta_{h_0}$. Ez viszont azt jelenti, hogy létezik (Y, X) él (m, o) -ban, ami ellentmondás a feltételezésünkkel. Következésképpen $X \in R$, és (A2) teljesül.

(A3) bizonyításához vegyünk egy tetszőleges $u_i \in o$ műveleti egységet. Ha $\beta_i \cap P \neq \emptyset$, akkor nyilván létezik út u_i -ből P -be. Egyébként, felhasználva azt a tényt, hogy a maximális struktúra egy lehetséges megoldás struktúra, a maximális struktúrában létezik út u_i -ből P -be, így létezik olyan $u_h \in \bar{O}$ műveleti egység, melyre $\beta_i \cap \alpha_h \neq \emptyset$. Mivel Φ_2 a maximális struktúra minden célterméket nem gyártó műveleti egységére tartalmaz egy tagot, ezért tartalmazni fogja a $\neg y_i \vee \bigvee_{\substack{h \in J \\ \beta_j \cap \alpha_h \neq \emptyset}} y_h$ tagot is. Mivel $u_i \in o$ miatt $y_i = IGAZ$, ezért az előbbiekhöz hasonló módon létezik h_0 úgy, hogy $u_{h_0} \in o$ és $\beta_i \cap \alpha_{h_0} \neq \emptyset$. Most u_{h_0} -al indulva ugyanezt a gondolatmenetet megismételve, véges számú ismétlődő lépés után kapunk egy u_i -ből P -be vezető (m, o) -beli utat, ami azt mutatja, hogy (A3) igaz. \square

A 2.5.2. lemma alapján felírhatjuk a PNS probléma egy ekvivalens formáját:

$$(PNS-4) \quad \min \left\{ \sum_{j \in T(\mathbf{y})} w_j : \mathbf{y} \text{ kielégíti } \Phi\text{-t} \right\},$$

ahol $w_j = w(u_j)$, $j = 1, \dots, l$.

A ([17]) alapján bevezetve a $z_j^+, z_j^- \in \{0, 1\}$, $j = 1, \dots, l$ változókat úgy, hogy $z_+^j = 1$ akkor és csakis akkor ha $y_j = IGAZ$, továbbá $z_j^- = 1 - z_j^+$, a fenti feladatot az alábbi formában is felírhatjuk:

$$\sum_{\substack{j \in J \\ X \in \beta_j}} z_j^+ \geq 1, \text{ minden } X \in P\text{-re,}$$

$$z_j^- + \sum_{\substack{h \in J \\ X \in \beta_h}} z_h^+ \geq 1, \text{ minden } j \in J, X \in \alpha_j \setminus R\text{-re,}$$

$$z_j^- + \sum_{\substack{h \in J \\ \beta_j \cap \alpha_h \neq \emptyset}} z_h^+ \geq 1, \text{ minden } j \in J, P \cap \beta_j = \emptyset\text{-re,}$$

$$z_j^+ + z_j^- = 1, \text{ minden } j \in J\text{-re,}$$

$$z_j^+, z_j^- \in \{0, 1\}, \text{ minden } j \in J\text{-re,}$$

$$\min \sum_{j \in J} w_j \cdot z_j^+$$

A ([17])-ban bemutatott bizonyítás alapján könnyen belátható, hogy tetszőleges $L > \sum_{j \in J} w_j$ -re a fenti feladat optimális megoldásai azonosak az alábbi halmazlefedési probléma optimális megoldásaival:

$$\sum_{\substack{j \in J \\ X \in \beta_j}} z_j^+ \geq 1, \text{ minden } X \in P\text{-re,}$$

$$z_j^- + \sum_{\substack{h \in J \\ X \in \beta_h}} z_h^+ \geq 1, \text{ minden } j \in J, X \in \alpha_j \setminus R\text{-re,}$$

$$z_j^- + \sum_{\substack{h \in J \\ \beta_j \cap \alpha_h \neq \emptyset}} z_h^+ \geq 1, \text{ minden } j \in J, P \cap \beta_j = \emptyset\text{-re,}$$

$$z_j^+ + z_j^- \geq 1, \text{ minden } j \in J\text{-re,}$$

$$z_j^+, z_j^- \in \{0, 1\}, \text{ minden } j \in J\text{-re,}$$

$$\min \sum_{j \in J} [(w_j + L) \cdot z_j^+ + L \cdot z_j^-]$$

Összegezve az eddigieket, azt kaptuk, hogy a PNS probléma visszavezethető egy halmazlefedési feladatra, ugyanakkor a 2.5.1. lemma értelmében a halmazlefedési feladat ekvivalens egy speciális PNS problémaosztállyal. Ebből az következik, hogy:

2.5.2. Tétel. [27] *A PNS-2 probléma ekvivalens a halmazlefedési problémával.*

A halmazlefedési probléma NP teljessége (ld. [32, 1]) alapján ebből adódik, hogy:

2.5.3. Tétel. [27] *A PNS-2 probléma NP teljes.*

A 2.5.3. tétel alapján feltételezhető, hogy a PNS-2 probléma hatékony, polinomiális idejű megoldása nem várható. Ez indokolja azt, hogy exponenciális idejű megoldó algoritmusokat elfogadhatóknak tekintsünk és alkalmazzuk az ilyen esetekben szokásos korlátozás és szétválasztás jellegű módszereket.

2.6. Döntési leképezések

Fontos szerepet játszik a PNS-problémának a korlátozás és szétválasztás módszerével történő, különböző ([24, 21]) megoldásaiban a ([13, 14])-ban bevezetett döntési leképezés fogalma.

Legyen O a műveleti egységek egy halmaza. Definiáljuk a $\Delta : M \setminus R \rightarrow \wp(O)$, függvényt a következő módon. Minden $X \in M \setminus R$ -re legyen $\Delta(X) = \{(\alpha, \beta) : (\alpha, \beta) \in O \ \& \ X \in \beta\}$. Szemléletesen a Δ minden nem nyersanyaghoz hozzárendeli azokat a műveleti egységeket, melyek azt az anyagot gyárthatják. Mi viszont általános esetben a műveleti egységeknek csak egy olyan részhalmazát szeretnénk használni, mely a legkisebb költséggel képes a céltermékek legyártására. Mivel általában egy anyagot több műveleti egység is legyárthat, ezért egy megoldás felépítése során minden anyagra vonatkozóan dönteniünk kell arról, hogy azt az anyagot mely műveleti egységekkel kívánjuk legyártani. Ezen döntések leírására használjuk a döntési leképezéseket. Legyen $m \subseteq M \setminus R$ és $\delta(X) \subseteq \Delta(X)$, minden $X \in m$ -re. A $\delta[m] = \{(X, \delta(X)) : X \in m\}$ leképezést **reguláris döntési leképezésnek**, vagy egyszerűen csak **döntési leképezésnek** nevezzük.

Egy döntési leképezés **konzisztens**, ha $\delta(X) \cap \Delta(Y) \subseteq \delta(Y)$, bármely $X, Y \in m$ -re. Ez azt fejezi ki, hogy ha egy műveleti egység több

anyagot is gyárt, akkor vagy azt feltételezzük, hogy nem működik, azaz nem gyárt semmit és akkor nem rendeljük hozzá egyetlen anyaghoz sem, vagy azt feltételezzük, hogy működik, de akkor mindegyik kimeneti anyagát gyártja és így mindhez hozzá kell rendelni. Az \mathbf{M} strukturális modell konzisztens döntési leképezéseinek halmazát $\Omega_{\mathbf{M}}$ -el fogjuk jelölni.

Most tegyük fel, hogy egy $\delta[m]$ döntési leképezés által leszögeztük a műveleti egységek egy részhalmazát abból a célból, hogy az m -beli anyagokat közvetlenül gyártsa. Ha veszünk egy további $Y \in M \setminus (m \cup R)$ anyagot és ennek közvetlen gyártására konzisztens módon hozzárendeljük az u'_1, \dots, u'_r $\Delta(Y)$ -beli műveleti egységeket, akkor egy nagyobb részfolyamatot kapunk. Az ennek megfelelő

$$\delta'[m \cup \{Y\}] = \delta[m] \cup \{(Y, \{u'_1, \dots, u'_r\})\}$$

döntési leképezésre azt mondjuk, hogy a $\delta[m]$ **reguláris kiterjesztése** vagy egyszerűen csak a $\delta[m]$ **kiterjesztése**.

A kiterjesztés függvény $\Omega_{\mathbf{M}}$ -en egy reflexív, antiszimmetrikus és tranzitív, tehát részben rendezés relációt határoz meg. Jelöljük a részben rendezett halmazt $(\Omega_{\mathbf{M}}, \leq)$ -el. Nyilvánvalóan $\delta[\emptyset]$ az $(\Omega_{\mathbf{M}}, \leq)$ halmaz legkisebb eleme. A fentiek értelmében a kiterjesztés relációt általánosíthatjuk úgy, hogy azt mondjuk, hogy $\delta_2[m_2]$ (**reguláris**) **kiterjesztése** $\delta_1[m_1]$ -nek és ezt ugyancsak $\delta_1[m_1] \leq \delta_2[m_2]$ -vel jelöljük, ha $m_1 \subseteq m_2$, $\delta_1[m_1]$ és $\delta_2[m_2]$ konzisztens döntési leképezések, valamint $\delta_1(X) = \delta_2(X)$ minden $X \in m_1$ -re. Az $(\Omega_{\mathbf{M}}, \leq)$ halmaz maximális elemeit $\Omega_{\mathbf{M}}^{\max}$ -val fogjuk jelölni.

Könnyen beláthatjuk, hogy teljesül az alábbi tulajdonság:

2.6.1. Lemma. *Egy $\delta[m] \in \Omega_{\mathbf{M}}$ döntési leképezés az $(\Omega_{\mathbf{M}}, \leq)$ részben rendezett halmaz maximális eleme akkor és csakis akkor, ha $m = M \setminus R$.*

Bizonyos esetekben egy $\delta[m] \in \Omega_{\mathbf{M}} \setminus \Omega_{\mathbf{M}}^{\max}$ döntési leképezésből maximális elemet képezhetünk a következő módon. Ha

$$W = (\text{mat}^{in}(\text{op}(\delta[m])) \cup P) \setminus (R \cup m) = \emptyset$$

akkor legyen

$$\delta'(X) = \{(\alpha, \beta) : (\alpha, \beta) \in \text{op}(\delta[m]) \text{ and } X \in \beta\}$$

minden $X \in M \setminus R$ -re. Akkor könnyen belátható, hogy $\delta' \in \Omega_{\mathbf{M}}^{\max}$. A δ' döntési leképezést a $\delta[m]$ (**reguláris**) **lezárásának** fogjuk nevezni.

Mivel minden $\delta[m] \in \Omega_{\mathbf{M}}^{\max}$ -re $m = M \setminus R$, ezért ismert $\mathbf{M} = (P, R, O)$ strukturális modell esetén adott maximális döntési leképezést egyszerűen δ -val fogunk jelölni. Érdemes definiálni a műveleti egységek azon halmazát, amely egy konzisztens döntési leképezés által van meghatározva. Nevezetesen, legyen

$$op(\delta[m]) = \cup \{\delta(X) : X \in m\}.$$

A maximális konzisztens döntési leképezések és a lehetséges megoldás struktúrák közötti összefüggések tanulmányozására, minden $(m, o) \in S(\mathbf{M})$ lehetséges megoldás struktúrához hozzárendelünk egy maximális konzisztens döntési leképezést a következőképpen.

2.6.1. Definíció. Legyen $\rho : S(\mathbf{M}) \longrightarrow \Omega_{\mathbf{M}}^{\max}$ függvény, melyre $\rho(m, o) = \delta$ úgy, hogy

$$\delta(X) = \{u : u = (\alpha, \beta) \in o \ \& \ X \in \beta\}, \text{ ha } X \in m \setminus R, \text{ és}$$

$$\delta(X) = \emptyset, \text{ ha } X \notin M \setminus (R \cup m).$$

2.6.2. Lemma. ([21]) ρ egy injektív leképezés $S(\mathbf{M})$ -ről $\Omega_{\mathbf{M}}^{\max}$ -ba, továbbá

$$\rho^{-1}(\delta) = (mat(op(\delta)), op(\delta))$$

igaz minden olyan δ -ra, mely egy $S(\mathbf{M})$ -beli elem ρ általi leképezése.

Bizonyítás Először megmutatjuk, hogy $(m, o) \in S(\mathbf{M})$ -ből következik $\rho((m, o)) \in \Omega_{\mathbf{M}}^{\max}$. Mivel a $\rho((m, o)) = \delta[M \setminus R]$ az $M \setminus R$ halmazon definiált, továbbá egy $\delta[m]$ döntési leképezés akkor és csakis akkor maximális, ha $m = M \setminus R$, ezért azt kell belátnunk, hogy δ konzisztens. Ebből a célból legyenek $X, Y \in M \setminus R$ tetszőleges anyagok. Igazoljuk, hogy $\delta(X) \cap \Delta(Y) \subseteq \delta(Y)$. Ha $\delta(X) \cap \Delta(Y) = \emptyset$, akkor a konzisztenciához szükséges tartalmazás nyilvánvalóan teljesül. Most tegyük fel, hogy $\delta(X) \cap \Delta(Y) \neq \emptyset$ és $(\alpha, \beta) \in \delta(X) \cap \Delta(Y)$. Akkor δ definíciója alapján $X \in m \setminus R$ és $(\alpha, \beta) \in o$.

Másfelől $(\alpha, \beta) \in \Delta(Y)$ -ből következik, hogy $Y \in \beta$. Mivel (m, o) egy megoldás struktúra, ezért a 2.1.2. lemma alapján $m = \text{mat}(o) \supseteq \beta \ni Y$, és így $Y \in m$. Az $Y \in M \setminus R$ feltételezés azt vonja maga után, hogy $Y \notin R$, és ezért $Y \in m \setminus R$. Azt kapjuk tehát, hogy $(\alpha, \beta) \in \delta(Y)$, amiből következik az előírt tartalmazás.

Most legyen (m, o) és (m', o') két különböző megoldás struktúra. Megmutatjuk, hogy a $\rho((m, o)) = \delta$ és $\rho((m', o')) = \delta'$, egymástól különböző döntési leképezések. A 2.1.2. lemma és 2.1.2. következmény alapján az (m, o) és (m', o') egyértelműen meghatározottak az o illetve o' halmazok által, ezért $o \neq o'$. Akkor az általánosság megszorítása nélkül feltételezhetjük, hogy létezik olyan $(\alpha, \beta) \in O$, melyre $(\alpha, \beta) \in o$, de $(\alpha, \beta) \notin o'$. Mivel $(m, o) \in S(\mathbf{M})$, ezért a 2.1.2. lemma alapján $m = \text{mat}(o)$ és így $\beta \subseteq m$. Az $(\mathcal{A}3)$ feltételből $|\beta| \geq 1$, vagyis létezik legalább egy olyan X anyag, melyre $X \in \beta$, és ezért $X \in m$. Mivel (M, O) egy megoldás struktúra és $X \in \beta$, továbbá $(\alpha, \beta) \in O$, ezért az $(\mathcal{A}2)$ feltételből $X \notin R$ és így $X \in m \setminus R$. Továbbá a δ definíciójából $(\alpha, \beta) \in \delta(X)$. Két esetet különböztethetünk meg.

Ha $X \notin m'$, akkor a δ' definíciója alapján $\delta'(X) = \emptyset$, következésképpen $\delta(X) \neq \delta'(X)$.

Ha $X \in m'$, akkor $(\alpha, \beta) \notin o'$ -ből δ' definíciója alapján következik, hogy $(\alpha, \beta) \notin \delta'(X)$.

Ily módon igazoltuk, hogy $\delta \neq \delta'$.

Végül legyen $\delta = \rho((m, o))$ valamely $(m, o) \in S(\mathbf{M})$ megoldás struktúrára. A ρ definíciója alapján könnyen belátható, hogy $o = \text{op}(\delta)$. Akkor a 2.1.2. lemmából azt kapjuk, hogy $(m, o) = (\text{mat}(\text{op}(\delta)), \text{op}(\delta))$, ami azt jelenti, hogy igazoltuk az állítást. \square

2.6.1. Megjegyzés. Vegyük észre, hogy ρ -ra a szürjektívitas, és ily módon a bijektívitas sem teljesül. Például a $\delta^*(X) = \emptyset, \forall X \in M \setminus R$ döntési leképezés eleme az $\Omega_{\mathbf{M}}^{\max}$ halmaznak, de nem létezik olyan $(m, o) \in S(\mathbf{M})$ lehetséges megoldás struktúra, melyre $\rho((m, o)) = \delta^*$ teljesülne.

Jelölje $S'(\mathbf{M})$ az $S(\mathbf{M})$ ρ melletti képét, vagyis legyen

$$S'(\mathbf{M}) = \{\rho((m, o)) : (m, o) \in S(\mathbf{M})\}.$$

Vegyük észre, hogy minden $(m, o) \in S(\mathbf{M})$ -re ha $\rho((m, o)) = \delta[M \setminus R]$, akkor $o = op(\delta[M \setminus R])$. Ezen észrevételből és a 2.6.2. lemmából azt kapjuk, hogy a PNS-3 feladat helyett megoldhatjuk az alábbi feladatot:

$$(PNS-5) \quad \min \left\{ \sum_{u \in op(\delta)} w(u) : \delta \in S'(\mathbf{M}) \right\}.$$

Az egyszerűség kedvéért az $S'(\mathbf{M})$ elemeit **lehetséges megoldásoknak** fogjuk nevezni.

A későbbiekben látni fogjuk, hogy a konzisztens döntési leképezések lehetővé teszik a lehetséges megoldás struktúrák hatékonyabb leírását és generálását.

2.7. Leszámlálási algoritmusok

Ebben a részben a PNS-5 feladat megoldására fogunk kidolgozni alapvető algoritmusokat. A 2.5.3 tétel alapján tudjuk, hogy nem várható hatékony polinomiális idejű megoldás. Ezért először kidolgozunk egy olyan Branch and Bound jellegű eljárást, mely általános keretet nyújt majd az ilyen jellegű algoritmusok későbbiekben történő tárgyalására, majd egy olyan algoritmust fogunk megnézni, mely a feladat összes lehetséges megoldásait, köztük az összes optimális megoldást is felsorolja.

2.7.1. Egyszerű leszámolás

Legyen \mathbf{M} egy PNS probléma strukturális modellje és jelölje (M, O) a maximális struktúrát. Célunk az, hogy megoldjuk a PNS-5 problémát. Mivel az $S'(\mathbf{M})$ nem üres, véges halmaz, ezért léteznie kell legalább egy optimális megoldásnak. A továbbiakban megmutatjuk, hogy hogyan alkalmazható az általános korlátozás és szétválasztás módszere a feladat megoldására.

A Branch and Bound algoritmus egyik fő összetevője a szétválasztási függvény, mely $S'(\mathbf{M})$ részhalmazait partíciókra osztja. Mivel itt $S'(\mathbf{M})$ csak implicit módon van definiálva, viszont $S'(\mathbf{M}) \subseteq \Omega_{\mathbf{M}}^{\max}$, amiből kifolyólag az $\Omega_{\mathbf{M}}^{\max}$ partíciói az $S'(\mathbf{M})$ -ben is partíciókat hoznak létre, ezért $S'(\mathbf{M})$ helyett az őt befoglaló $\Omega_{\mathbf{M}}^{\max}$ halmazt és annak részhalmazait fogjuk partícionálni.

Definiáljuk az $\omega : \Omega_{\mathbf{M}} \setminus \Omega_{\mathbf{M}}^{\max} \longrightarrow \varphi'(\Omega_{\mathbf{M}}^{\max})$ függvényt a következőképpen. Minden $\delta[m] \in \Omega_{\mathbf{M}} \setminus \Omega_{\mathbf{M}}^{\max}$ -re legyen

$$\omega(\delta[m]) = \{\delta : \delta \in \Omega_{\mathbf{M}}^{\max} \text{ és } \delta[m] \leq \delta\}.$$

Tulajdonképpen $\omega(\delta[m])$ megadja a $\delta[m]$ maximális konzisztens kiterjesztéseit. A 2.6 részben láttuk, hogy minden nem maximális konzisztens döntési leképezésnek létezik maximális konzisztens döntési leképezésre való kiterjesztése. Az is nyilvánvaló, hogy $\omega(\delta[\emptyset]) = \Omega_{\mathbf{M}}^{\max}$. Az $\omega(\delta[m]) \cap S'(\mathbf{M})$ halmaz elemeit *(reguláris) lehetséges megoldás kiterjesztések*nek fogjuk nevezni.

A [25]-ban igazolást nyert a következő állítás.

2.7.1. Lemma. *Ha $\delta[m], \delta'[m'] \in \Omega_{\mathbf{M}} \setminus \Omega_{\mathbf{M}}^{\max}$, és $\exists X \in m \cap m' : \delta[X] \neq \delta'[X]$, akkor $\omega(\delta[m]) \cap \omega(\delta'[m']) = \emptyset$.*

Legyen $\delta[m] \in \Omega_{\mathbf{M}} \setminus \Omega_{\mathbf{M}}^{\max}$. Akkor a 2.6.1. lemma értelmében $m \subset M \setminus R$, ami azt jelenti, hogy létezik $X \in M \setminus (R \cup m)$ anyag. Mivel (M, O) maximális struktúra, ezért $\Delta(X) \neq \emptyset$, így $\Delta(X)$ minden $Q_i, i = 1, \dots, 2^{|\Delta(X)|}$ részhalmazára definiálhatjuk a

$$\delta_i[m \cup \{X\}] = \delta[m] \cup \{(X, Q_i)\}$$

döntési leképezést.

Az általánosság megszorítása nélkül feltételezhetjük, hogy a $\delta_i[m \cup \{X\}]$, $i = 1, \dots, 2^{|\Delta(X)|}$ halmaz konzisztens döntési leképezései a $\delta_t[m \cup \{X\}]$, $t = 1, \dots, k$ döntési leképezések. Akkor a [25] alapján tudjuk, hogy

2.7.2. Lemma. *Az $\omega(\delta_t[m \cup \{X\}])$, $t = 1, \dots, k$ halmazok egy nem feltétlen nem triviális partíciói az $\omega(\delta[m])$ halmaznak.*

Akkor a B&B algoritmus ϖ szétválasztási függvénye a következő lesz:

$$\varpi(\delta[m]) = \{\omega(\delta_t[m \cup \{X\}]), t = 1, \dots, k\}.$$

A Branch and Bound algoritmus másik lényeges összetevője a korlátozás. Jelen esetben ez egy $g : \Omega_{\mathbf{M}} \rightarrow \mathbf{R}$ függvény, mely alsó korlátokat határoz meg a $w(\delta')$, $\delta' \in S'(\mathbf{M}) \cap \omega(\delta[m])$, $\delta[m] \in \Omega_{\mathbf{M}} \setminus \Omega_{\mathbf{M}}^{\max}$ értékekre, $g(\delta) = w(\delta)$ ha $\delta[m] \in S'(\mathbf{M})$, és $g(\delta) = \infty$ ha $\delta[m] \in \Omega_{\mathbf{M}}^{\max} \setminus S'(\mathbf{M})$.

A korlátozás és szétválasztás módszere tulajdonképpen az összes lehetséges megoldást tartalmazó leszámplálási fa olyan intelligens bejárása, mely a szétválasztó és korlátozó függvényeinek köszönhetően a fának csak egy részét generálja és járja be azért, hogy minél hamarabb eljusson az optimális megoldáshoz. Ez úgy lehetséges, hogy azokat a csúcspontokat nem fogja tovább partícionálni, melyekről a korlátozó függvény segítségével biztosan meg tudja állapítani, hogy nem tartalmazhatnak optimális megoldást. Az ilyen csúcspontokat **felderített**, **lezárt**, vagy **halott** csúcspontoknak szoktuk nevezni, míg a többi csúcspontok alkotják az **élő** csúcspontokat. Ezek után megadhatjuk a PNS-5 problémát megoldó B&B algoritmust.

Branch and Bound Algoritmus ([25])

Inicializálás

- Legyen $m = \emptyset$, $L = \{\omega(\delta[\emptyset])\}$, $z = \infty$, $s = \emptyset$, $r = 0$. Számoljuk ki $g(\delta[\emptyset])$ -t.

Iteráció (*r.* iteráció)

1. Befejezés tesztelés

Ha $L = \emptyset$, akkor VÉGE: z tartalmazza az optimumot, s pedig az optimális megoldást.

Egyébként, ha $r > 0$, akkor térjünk a 2. lépésre, ha pedig $r = 0$, akkor térjünk a 3. lépésre.

2. *Levéلكiválasztás*

Válasszunk egy $\omega(\delta[m])$ elemet az L -ből, melyre a $\frac{g(\delta[m])}{|m|}$ minimális. Ha több ilyen van, akkor válasszunk egyet véletlenszerűen közülük. Térjünk a 3. lépésre.

3. *Szétválasztás*

Alkossuk meg $\omega(\delta[m])$ -nek a $\varpi(\delta[m]) = \{\omega(\delta_i[m_i]), i = 1, \dots, k\}$ partícióit. Térjünk a 4. lépésre.

4. *Korlátszámítás*

Minden $i = 1, \dots, k$ -ra számoljuk ki a $g(\delta_i[m_i])$ korlátokat. Továbbá, ha $m = M \setminus R$ és $g(\delta_i) < z$, akkor legyen $z = g(\delta_i)$ és $s = \{\delta_i\}$. Térjünk az 5. lépésre.

5. *Felderítés*

Aktualizáljuk L értékét:

$$L = \{\omega(\delta'[m']) : \omega(\delta'[m']) \in (L \setminus \{\omega(\delta[m])\}) \cup \varpi(\delta[m]), g(\delta'[m']) < z\}.$$

Legyen $r = r + 1$, és térjünk a következő iterációra (1. lépés).

2.7.1. Megjegyzés. A leírt algoritmus nem teljes, hanem egy séma, mely a különböző választási lehetőségek pontosításával többé-kevésbé hatékonyan implementálható. Például a 3. lépésben nem határoztuk meg, hogy mely anyaggal fogjuk kiterjeszteni az m anyagalmazt a partíciók képzése céljából. Erre egy lehetőség lehet, hogy az $M \setminus (m \cup R)$ halmaz legkisebb indexű elemét választjuk. Hasonlóképpen, a g korlátozó függvény pontos képletét sem adtuk meg. Triviális korlátként megadhatnánk például a $\delta[m]$ által rögzített műveleti egységek súlyainak összegét. Ezek különböző implementációja különböző B&B algoritmusokhoz vezet. További ilyen lehetőségeket tartalmaz a [25] munka.

2.7.2. Teljes leszámolás

Vannak esetek, amikor nem csak az optimumot és egy hozzá tartozó optimális megoldást keresünk, hanem valamiért érdekel a többi, nem feltétlen optimális megoldás is. Ebben a részben egy olyan ([16])-ben kidolgozott eljárás módosított leírását mutatjuk be, mely a PNS-5 probléma összes lehetséges megoldását szolgáltatja.

Az általánosság megszorítása nélkül feltételezhetjük, hogy $O = \{u_1, \dots, u_k\}$ és $M = \{A_1, \dots, A_n\}$, ahol $P = \{A_1, \dots, A_s\}$ valamely k és $1 \leq s \leq n$ pozitív egészekre.

Bevezetünk egy lineáris rendezést az O -ban levő műveleti egységek tetszőleges halmazának részhalmazai között. Ennek definiálása érdekében legyen $\emptyset \neq o$ az O egy tetszőleges részhalmaza, továbbá legyenek $o_1 = \{u_{i_1}, \dots, u_{i_s}\}$, $i_1 < \dots < i_s$ és $o_2 = \{u_{j_1}, \dots, u_{j_t}\}$, $j_1 < \dots < j_t$ az o tetszőleges részhalmazai. Akkor $o_1 \models o_2$ ha az alábbi feltételek valamelyike teljesül:

- (i) $o_1 = o_2$,
- (ii) $|o_1| < |o_2|$,
- (iii) $|o_1| = |o_2|$ és $i_k < j_k$ a legkisebb olyan k indexre, melyre $1 \leq k \leq s$ és $i_k \neq j_k$.

Abban az esetben, ha $o_1 \neq o_2$, az \models helyett az \vdash jelölést használhatjuk. Könnyű belátni, hogy az \models reláció lineáris rendezés.

Egy $\delta[m] \in \Omega_M$ döntési leképezés felépítéséhez az m halmaz elemeinek rendezett, $\hat{m} = \langle A_{i_1}, \dots, A_{i_t} \rangle$ -vel jelölt sorozatait fogjuk használni. Ha egy $A \notin m$ anyaggal kiterjesztjük a $\delta[m]$ döntési leképezést, akkor a kiterjesztett döntési leképezés tartománya $\langle A_{i_1}, \dots, A_{i_t}, A \rangle$ lesz, melyet

$$\hat{m} \bigvee \{A\} = \langle A_{i_1}, \dots, A_{i_t} \rangle \bigvee \{A\}$$

-val fogunk jelölni.

Most készen állunk arra, hogy az eljárást bemutassuk.

Teljes leszámolás (Complete Enumeration, ([16]))

Inicializálás

- Legyen $m_0 = \emptyset$, $S := \emptyset$ és $\delta_0[m_0] = \emptyset$, továbbá $i = 0$.

Iteráció

1. Legyen $\delta_i[m_i]$ az $\hat{m}_i = \langle A_{j_1}, \dots, A_{j_k} \rangle$ rendezett tartománnyal rendelkező aktuális döntési leképezés. Továbbá legyen

$$W_i = (\text{mat}^{in}(\text{op}(\delta_i[m_i])) \cup P) \setminus (R \cup m_i),$$

és térjünk a 2. lépésre.

2. Ha $W_i = \emptyset$, akkor konstruáljuk meg a $\delta_i[m_i]$ döntési leképezés δ'_i lezárását, továbbá aktualizáljuk S -et: $S = S \cup \{\delta'_i\}$, és térjünk a 4. lépésre. Egyébként térjünk a 3. lépésre.
3. Válasszuk ki W_i -ből a legkisebb indexű X elemet. Vizsgáljuk meg $\Delta(X)$ nem üres részhalmazait a \models relációra vonatkozóan. Válasszuk ki a $\Delta(X)$ első olyan Q részhalmazát, melyre $\delta_i[m_i] \cup \{(X, Q)\}$ egy konzisztens döntési leképezés, feltéve, hogy van ilyen.

Legyen $\hat{m}_{i+1} = \hat{m}_i \vee \{X\}$, és

$$\delta_{i+1}[m_{i+1}] = \delta_i[m_i] \cup \{(X, Q)\},$$

Legyen $i = i + 1$, és térjünk a következő iterációra (1. lépés).

Ha $\Delta(X)$ egyetlen nem üres részhalmaza sem megfelelő konzisztens döntési leképezés, akkor térjünk a 4. lépésre.

4. Ha $\delta_i(A_{j_k}) \subset \Delta(A_{j_k})$ és van olyan nem üres $Q \subseteq \Delta(A_{j_k})$ részhalmaz, melyre $\delta_i(A_{j_k}) \neq Q$, $\delta_i(A_{j_k}) \leq Q$, továbbá az

$$\{(A_{j_1}, \delta_i(A_{j_1}))\} \cup \dots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}}))\} \cup \{(A_{j_k}, Q)\}$$

döntési leképezés konzisztens, akkor térjünk az 5. lépésre.

Egyébként térjünk a 6. lépésre.

5. Válasszuk a $\Delta(A_{j_k})$ részhalmazai közül a 4. lépésben leírt feltételt teljesítő \models szerinti első részhalmazát és jelöljük ezt Q' -vel. Legyen:

$$\hat{m}_{i+1} = \langle A_{j_1}, \dots, A_{j_k} \rangle, \text{ és}$$

$$\delta_{i+1}[m_{i+1}] = \{(A_{j_1}, \delta_i(A_{j_1}))\} \cup \dots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}}))\} \cup \{(A_{j_k}, Q')\}.$$

Legyen $i = i + 1$, és térjünk a következő iterációra (1. lépés).

6. Legyen $k = k - 1$. Ha $k = 0$, akkor VÉGE. Egyébként térjünk a 4. lépésre.

3. fejezet

A döntési leképezések száma

A fejezet a [3], [4] és [5] cikkek eredményeit tartalmazza. Ezek olyan közös publikációk, melyek eredményeit szerzőtársaimmal oszthatatlanoknak tekintjük.

A konzisztencia szükséges ahhoz, hogy eljussunk a lehetséges megoldásokig. Azonban több is igaz: minden konzisztens döntési leképezés azonosítható a fentiek szerint hozzárendelt műveleti egységek részhalmazával, hiszen, ha $m \subseteq M \setminus R$ adott, akkor azon műveleti egységek közül választhatunk, amelyek kimenetében szerepel legalább egy anyag m -ből. Viszont ha egy műveleti egységet valamelyik anyaghoz hozzárendeltük, akkor éppen a konzisztencia miatt ezt már minden kimeneti anyagához hozzá kell rendelnünk. Ez az egyszerű észrevétel tehát azt jelenti, hogy az eddigieknél jobban kezelhetjük a konzisztens döntési leképezéseket és megszámlálhatjuk őket.

3.1. Általános eset

3.1.1. Tétel. ([3]) Minden $\emptyset \neq m \subseteq M \setminus R$ -re, az m -en definiálható konzisztens döntési leképezések száma $2^{|\cup_{X \in m} \Delta(X)|}$.

Bizonyítás

Jelöljük $\tau(m)$ -el az m anyaghalmaz felett definiálható konzisztens döntési leképezések számát. Eljárásunkban $|m|$ szerinti indukciót fogunk alkalmazni.

Ha $|m| = 1$, akkor $X \rightarrow Q$ egy konzisztens döntési leképezés bármely $Q \subseteq \Delta(X)$ -re, ahol X az m egyetlen elemét jelöli. A konzisztens döntési leképezések száma ebben az esetben $2^{|\Delta(X)|}$.

Most legyen $1 \leq i < |M \setminus R|$ egy tetszőleges egész szám és feltételezzük, hogy az állítás igaz minden olyan $m \subseteq M \setminus R$ -re, melyre $|m| = i$. Vegyünk egy tetszőleges $i + 1$ elemű $m' \subseteq M \setminus R$ halmazt. Az általánosság megszorítása nélkül feltételezhetjük, hogy $m' = \{X_1, \dots, X_i, X_{i+1}\}$. Legyen $W = \Delta(X_{i+1}) \setminus (\cup\{\Delta(X_t) : t = 1, \dots, i\})$. W alapján két esetet különböztetünk meg.

1. ESET. $W = \emptyset$. Akkor $\bigcup_{X \in m'} \Delta(X) = \bigcup_{X \in m} \Delta(X)$.

Igazolnunk kell hát, hogy $\tau(m') = \tau(m)$. A konzisztencia definíciójából következik, hogy minden $\delta[m']$ konzisztens döntési leképezésnek az $\{X_1, \dots, X_i\}$ halmazra való szűkítése is konzisztens döntési leképezés. Másfelől, ha azonos halmazon definiált két konzisztens döntési leképezés különböző, akkor a kiterjesztéseik is különbözőek lesznek. Elegendő tehát azt igazolni, hogy bármely $\delta[\{X_1, \dots, X_i\}]$ döntési leképezésnek egyetlen kiterjesztése van az $\{X_1, \dots, X_i, X_{i+1}\}$ halmazra.

Először megkonstruáljuk a $\delta[\{X_1, \dots, X_i\}]$ egy kiterjesztését az $\{X_1, \dots, X_i, X_{i+1}\}$ halmazra. Legyen

$$\delta'(X_{i+1}) = \{(\alpha, \beta) : (\alpha, \beta) \in \Delta(X_{i+1}) \ \& \ \exists j \in \{1, \dots, i\} : (\alpha, \beta) \in \delta(X_j)\},$$

és

$$\delta'(X_t) = \delta(X_t), \quad \forall t \in \{1, \dots, i\}.$$

Mindenek előtt igazolnunk kell a $\delta'[\{X_1, \dots, X_i, X_{i+1}\}]$ konzisztenciáját, nevezetesen azt, hogy

$$(1) \quad \delta'(X_t) \cap \Delta(X_{i+1}) \subseteq \delta'(X_{i+1}),$$

és

$$(2) \quad \delta'(X_{i+1}) \cap \Delta(X_t) \subseteq \delta'(X_t)$$

teljesülnek bármely $X_t \in \{X_1, \dots, X_i\}$ esetén.

Az (1) érvényessége a δ' definíciójából következik. A (2) igazolására vegyünk egy $(\alpha, \beta) \in \delta'(X_{i+1}) \cap \Delta(X_t)$ tetszőleges műveleti egységet valamely $t \in \{1, \dots, i\}$ -re. Mivel $(\alpha, \beta) \in \delta'(X_{i+1})$, ezért $\exists j \in \{1, \dots, i\} : (\alpha, \beta) \in \delta(X_j) \cap \Delta(X_{i+1})$. Akkor $(\alpha, \beta) \in \delta(X_j) \cap \Delta(X_t)$. Másfelől, mivel $j, t \in \{1, \dots, i\}$ ezért δ konzisztenciájából következik, hogy $\delta(X_j) \cap \Delta(X_t) \subseteq \delta(X_t) = \delta'(X_t)$. Következésképpen $(\alpha, \beta) \in \delta'(X_t)$, ami azt jelenti, hogy a (2) feltétel is teljesül.

Most legyen $\delta^*[\{X_1, \dots, X_i, X_{i+1}\}]$ a $\delta[\{X_1, \dots, X_i\}]$ egy kiterjesztése. Meg fogjuk mutatni, hogy $\delta'(X_t) = \delta^*(X_t)$, $\forall t \in \{1, \dots, i+1\}$. Ha $1 \leq t \leq i$, akkor az egyenlőség nyilvánvalóan teljesül. Azt kell igazolnunk tehát, hogy $\delta'(X_{i+1}) \subseteq \delta^*(X_{i+1})$ és $\delta'(X_{i+1}) \supseteq \delta^*(X_{i+1})$. Ennek érdekében legyen $(\alpha, \beta) \in \delta'(X_{i+1})$ egy tetszőleges műveleti egység. A δ' definíciója alapján $(\alpha, \beta) \in \delta(X_j) \cap \Delta(X_{i+1})$ valamely $X_j \in \{X_1, \dots, X_i\}$ -re. De $\delta(X_j) = \delta^*(X_j)$ és δ^* konzisztens döntési leképezés, következésképpen $(\alpha, \beta) \in \delta^*(X_j) \cap \Delta(X_{i+1}) \subseteq \delta^*(X_{i+1})$. Fordítva, legyen $(\alpha, \beta) \in \delta^*(X_{i+1})$. Mivel $W = \emptyset$, ezért $\exists j \in \{1, \dots, i\} : (\alpha, \beta) \in \Delta(X_j)$, és ily módon $(\alpha, \beta) \in \delta^*(X_{i+1}) \cap \Delta(X_j)$. De δ^* konzisztenciája miatt $\delta^*(X_{i+1}) \cap \Delta(X_j) \subseteq \delta^*(X_j) = \delta(X_j)$, következésképpen $(\alpha, \beta) \in \Delta(X_{i+1}) \cap \delta(X_j)$, de akkor δ' definíciója miatt $(\alpha, \beta) \in \delta'(X_{i+1})$.

2. ESET. $W \neq \emptyset$. Az 1. ESET.-ben tett észrevételek alapján elegendő megmutatni, hogy a $\delta[\{X_1, \dots, X_i\}]$ konzisztens döntési leképezésnek $2^{|W|}$ kiterjesztése van a $\{X_1, \dots, X_i, X_{i+1}\}$ halmazra. Ennek érdekében legyen

$$T = \{(\alpha, \beta) : (\alpha, \beta) \in \Delta(X_{i+1}) \ \& \ \exists t \in \{1, \dots, i\} : (\alpha, \beta) \in \delta(X_t)\}.$$

T és W definícióiból nyilvánvaló, hogy $T \cap W = \emptyset$. Meg fogjuk mutatni,

hogy a

$$\delta'(X) = \begin{cases} \delta(X), & \text{ha } X \in \{X_1, \dots, X_i\} \\ T \cup Q, & \text{ha } X = X_{i+1} \end{cases}$$

döntési leképezés minden $Q \subseteq W$ -re konzisztens. A $\delta[\{X_1, \dots, X_i\}]$ konzisztenciája miatt elegendő igazolnunk a

$$(3) \quad \delta'(X_j) \cap \Delta(X_{i+1}) \subseteq \delta'(X_{i+1}),$$

és

$$(4) \quad \delta'(X_{i+1}) \cap \Delta(X_j) \subseteq \delta'(X_j),$$

tartalmazásokat minden $j = 1, \dots, i$ -re.

Ezért legyen $j \in \{1, \dots, i\}$ egy tetszőleges index és $(\alpha, \beta) \in \delta'(X_j) \cap \Delta(X_{i+1})$. Akkor $(\alpha, \beta) \in T$, és így $(\alpha, \beta) \in \delta'(X_{i+1})$, amiből következik (3). Most legyen $(\alpha, \beta) \in \delta'(X_{i+1}) \cap \Delta(X_j)$. Akkor $(\alpha, \beta) \in (T \cup Q) \cap \Delta(X_j) = T \cap \Delta(X_j)$. Az $(\alpha, \beta) \in T$ tartalmazásból következik, hogy $(\alpha, \beta) \in \delta(X_t)$ valamely $t \in \{1, \dots, i\}$ -re, következésképpen $(\alpha, \beta) \in \delta(X_t) \cap \Delta(X_j)$. Mivel δ konzisztens, ezért $\delta(X_t) \cap \Delta(X_j) \subseteq \delta(X_j) = \delta'(X_j)$, ami a (4) teljesülését jelenti.

Mivel W lehetséges Q részhalmazainak száma $2^{|W|}$, ezért a fenti módszerrel a $\delta[\{X_1, \dots, X_i\}]$ döntési leképezésnek $2^{|W|}$ különböző kiterjesztését kapjuk. Meg kell még mutatnunk, hogy a fenti döntési leképezésnek nincsenek további konzisztens kiterjesztései $\{X_1, \dots, X_i, X_{i+1}\}$ -re.

Legyen ehhez $\delta^*[\{X_1, \dots, X_i, X_{i+1}\}]$ egy tetszőleges kiterjesztése δ -nak, és $(\alpha, \beta) \in T$. Akkor $(\alpha, \beta) \in \delta(X_t) \cap \Delta(X_{i+1}) = \delta^*(X_t) \cap \Delta(X_{i+1})$ valamely $t \in \{1, \dots, i\}$ -re. Mivel δ^* konzisztens, ezért $\delta^*(X_t) \cap \Delta(X_{i+1}) \subseteq \delta^*(X_{i+1})$, és így $(\alpha, \beta) \in \delta^*(X_{i+1})$. Következésképpen $T \subseteq \delta^*(X_{i+1})$. Most legyen $(\alpha, \beta) \in \delta^*(X_{i+1}) \setminus T$. Ha $(\alpha, \beta) \notin W$, akkor $(\alpha, \beta) \in \Delta(X_t)$ valamely $t \in \{1, \dots, i\}$ -re, és akkor δ^* konzisztenciája miatt $(\alpha, \beta) \in \delta^*(X_t) = \delta(X_t)$, ami azt jelenti, hogy $(\alpha, \beta) \in T$, de ez ellentmondás. Tehát $(\alpha, \beta) \in W$, így $\delta^*(X_{i+1}) \subseteq T \cup W$. Ez azt jelenti, hogy δ^* -nak meg kell egyeznie δ valamely előzőekben már felsorolt kiterjesztésével.

Az indukciós feltevést használva tehát azt kapjuk, hogy

$$\tau(\{X_1, \dots, X_{i+1}\}) = 2^{|\cup\{\Delta(X_t):t=1,\dots,i\}|} 2^{|W|} = 2^{|\cup\{\Delta(X_t):t=1,\dots,i+1\}|},$$

amivel igazoltuk az állítást. \square

Megjegyezzük, hogy m -en összesen $2^{\sum_{x \in m} |\Delta(x)|}$ döntési leképezés definiálható. Tételünkéből következik, hogy $m = M \setminus R$ -re, vagyis a maximális konzisztens döntési leképezésekre $\tau(\Omega_{\mathbf{M}}^{\max}) = 2^{|O|}$ adódik. Ez azt mutatja, hogy szoros kapcsolat áll fenn az O részhalmazai és a maximális konzisztens döntési leképezések között. Könnyen belátható, hogy a $\gamma(\delta) = op(\delta)$ egy bijektív leképezés $\Omega_{\mathbf{M}}^{\max}$ és $\varphi(O)$ között.

A 2.6.1. definícióban meghatároztunk egy ρ függvényt a lehetséges megoldás struktúrák és a maximális konzisztens döntési leképezések között. Világos, hogy ρ az $S(\mathbf{M})$ -et kölcsönösen egyértelműen beleképezi $\Omega_{\mathbf{M}}^{\max}$ -be. Ez viszont azt is jelenti, hogy a $2^{|O|}$ egy triviális felső korlát $S(\mathbf{M})$ -re. Természetesen ez nagyon durva becslés. Azonban ha akár már csak (A2)-t figyelembe vesszük, sokkal jobb becslés adható $|S(\mathbf{M})|$ -re. A [3] munka alapján ezt fogjuk megvizsgálni a továbbiakban.

Legyen $(m, o) \in S(\mathbf{M})$ egy tetszőleges lehetséges megoldás struktúra és $\rho(m, o) = \delta$. Ha $X \in mat^{in}(op(\delta))$, akkor létezik $u = (\alpha, \beta) \in op(\delta)$ úgy, hogy $X \in \alpha$. A δ definíciója szerint $u \in o$, és így $X \in m$. (A2) alapján $X \in mat^{out}(op(\delta)) \cup R$, tehát felírhatjuk, hogy:

$$(\mathcal{A}'2) \quad mat^{in}(op(\delta)) \subseteq mat^{out}(op(\delta)) \cup R.$$

Az $(\mathcal{A}'2)$ -nek megfelelő maximális konzisztens döntési leképezések száma nyilván nem kevesebb, mint a lehetséges megoldás struktúrák száma, így felülről becslünk, ha az előbbit meghatározzuk. Ennek érdekében legyen (M, O) egy PNS probléma folyamat gráfja, $M = \{X_1, \dots, X_k\}$ és $O = \{u_1, \dots, u_n\}$. Legyen továbbá

$$O(X_j) = \{u : u = (\alpha, \beta) \in O \ \& \ X_j \in \alpha\}$$

minden $X_j \in M$ -re. Tetszőleges $j \in \{1, \dots, k\}$ -ra legyen

$$A_j = \{\delta : \delta \in \Omega_{\mathbf{M}}^{\max} \ \& \ X_j \in mat^{in}(op(\delta)) \setminus (mat^{out}(op(\delta)) \cup R)\}.$$

A_j azon maximális döntési leképezéseket tartalmazza, melyek X_j miatt nem elégítik ki az $(\mathcal{A}'2)$ -t. Minden $\emptyset \neq I \subseteq \{1, \dots, k\}$ -ra vezessük be az $A_I = \bigcap_{i \in I} A_i$ jelölést, továbbá legyen $A_\emptyset = \Omega_{\mathbf{M}}^{\max}$. Világos, hogy $I = \{i_1, \dots, i_l\}$ esetén

$$A_I = \{\delta : \delta \in \Omega_{\mathbf{M}}^{\max} \ \& \ \{X_{i_1}, \dots, X_{i_l}\} \subseteq mat^{in}(op(\delta)) \setminus (mat^{out}(op(\delta)) \cup R)\}$$

azon döntési leképezéseket jelenti, amelyek esetén $(\mathcal{A}'2)$ éppen az I -beli indexű anyagok miatt sérül. Így a Szitaformulát alkalmazva, az $(\mathcal{A}'2)$ -nek megfelelő maximális konzisztens döntési leképezések számára

$$|\Omega_{\mathbf{M}}^{\max} \setminus (A_1 \cup A_2 \cup \dots \cup A_k)| = \sum_{I \subseteq \{1, \dots, k\}} (-1)^{|I|} \cdot |A_I|$$

adódik.

3.1.1. Megjegyzés. Figyeljük meg, hogy a kapott korlát független az előállítandó anyagok halmazától, azaz érvényes bármely $P \subseteq M \setminus R$ -re.

3.1.1. Példa. Az általános esetre vonatkozó korlátszámítás szemléltetésére legyen $M = \{X_1, \dots, X_{12}\}$, $O = \{u_1, \dots, u_7\}$, $P = \{X_8\}$ és $R = \{X_{10}, X_{11}, X_{12}\}$, ahol a műveleti egységek bemeneti és kimeneti anyagait a 3.1. táblázat tartalmazza.

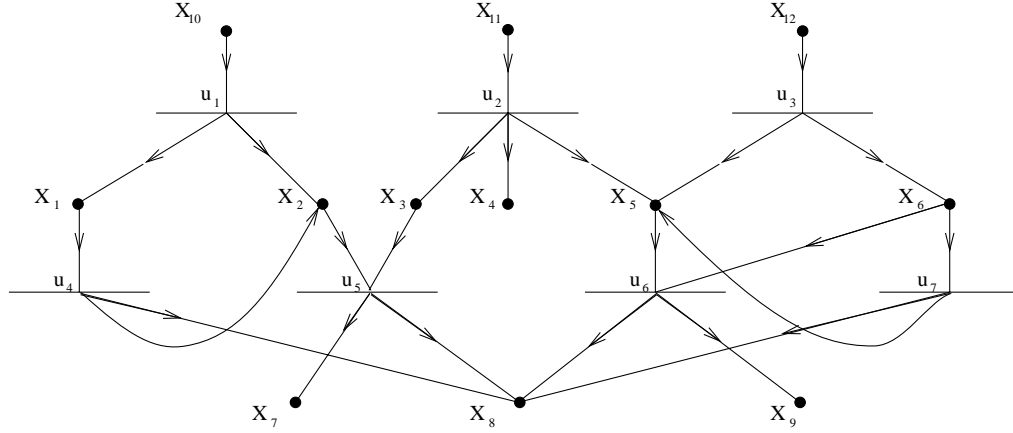
Műveleti egységek		
	bemenetek	kimenetek
u_1	X_{10}	X_1, X_2
u_2	X_{11}	X_3, X_4, X_5
u_3	X_{12}	X_5, X_6
u_4	X_1	X_2, X_8
u_5	X_2, X_3	X_7, X_8
u_6	X_5, X_6	X_8, X_9
u_7	X_6	X_5, X_8

3.1. táblázat.

A megfelelő folyamat gráfot a 3.1. ábra szemlélteti.

A maximális konzisztens döntési leképezések és O részhalmazai közötti kapcsolatot felhasználva tudjuk, hogy A_1 akkor és csakis akkor tartalmazza a δ -t, ha $op(\delta)$ kielégíti az $u_1 \notin op(\delta)$ és $u_4 \in op(\delta)$ tulajdonságokat. Ezen maximális döntési leképezések száma 2^5 , tehát $|A_1| = 2^5$. Hasonlóan kapjuk, hogy $|A_2| = 2^4$, $|A_3| = 2^5$, $|A_5| = 2^3$, $|A_6| = 3 \cdot 2^4$ és $|A_j| = 0$ a többi j indexre. Következésképpen

$$\sum_{I \subseteq \{1, \dots, k\} \text{ \& } |I|=1} |A_I| = 136.$$



3.1. ábra.

A kételemű részhalmazokat vizsgálva, $A_{\{1,2\}}$ akkor és csakis akkor tartalmazza δ -t, ha $u_1, u_4 \notin \text{op}(\delta)$ és $u_4, u_5 \in \text{op}(\delta)$, ezért $A_{\{1,2\}} = \emptyset$. Hasonlóan $A_{\{1,3\}} = 2^3$, mivel $A_{\{1,3\}}$ akkor és csakis akkor tartalmazza δ -t, ha $u_1, u_2 \notin \text{op}(\delta)$ és $u_4, u_5 \in \text{op}(\delta)$. Kiszámolva és összegezve a részhalmazokra a megfelelő értékeket azt kapjuk, hogy:

$$\sum_{I \subseteq \{1, \dots, k\} \text{ \& } |I|=2} |A_I| = 60.$$

Folytatva az eljárást, a három elemű részhalmazokra a 12 értéket kapjuk. Végül azt tapasztaljuk, hogy $|I| > 3$ -ra $|A_I| = 0$. Következésképpen a keresett érték:

$$2^7 - 136 + 60 - 12 = 40.$$

Megjegyezzük, hogy ebben a példában $|\Omega_{\mathbf{M}}^{\max}| = 128$ és $|S(\mathbf{M})| = 19$.

A Szitaformulában $|A_I|$ meghatározására van szükség, ami általában, tetszőleges folyamat gráf esetén, rendkívül bonyolult: $|A_I|$ az $\{X_{i_1}, \dots, X_{i_t}\}$ azon $\alpha_{j_1}, \dots, \alpha_{j_s}$ fedőrendszerének számával egyenlő, amelyekre léteznek az $(\alpha_{j_t}, \beta_{j_t}) \in O$, $t = 1, \dots, s$ műveleti egységek úgy, hogy $\{X_{i_1}, \dots, X_{i_t}\} \cap \beta_{j_t} = \emptyset$, $t = 1, \dots, s$ -re.

3.2. Speciális esetek

Bizonyos sajátos esetekben $|A_I|$ meghatározása természetesen egyszerűsödhet. A továbbiakban azt a speciális esetet fogjuk megvizsgálni, amikor egyetlen input anyaggal működő, ún. szeparátor típusú műveleti egységeink vannak, melyekre tehát $|\alpha| = 1$, bármely $u = (\alpha, \beta) \in O$ műveleti egységre. Legyen ismételtlen $I = \{i_1, \dots, i_l\}$ és

$$O^*(X_{i_j}) = O(X_{i_j}) \setminus (\cup_{i \in I} \Delta(X_i)).$$

$O^*(X_{i_j})$ azon műveleti egységek halmaza, melyeknek bemeneti anyaguk az X_{i_j} , de nem termelnek egyetlen anyagot sem az $\{X_t : t \in I\}$ halmazból. Akkor $|A_I|$ -re a következő képletet kapjuk ([3]):

$$|A_I| = \left(\prod_{t=1}^l \left(2^{|O^*(X_{i_t})|} - 1 \right) \right) \cdot 2^{|\mathcal{O} \setminus (\cup_{i \in I} \Delta(X_i)) \setminus (\cup_{i \in I} O(X_i))|}.$$

A továbbiakban két speciális szeparátor típusú műveleti egységeket tartalmazó PNS problémaosztály esetén a lehetséges megoldás struktúrák számára explicit módon kiszámolható képleteket fogunk adni.

Mindkét esetben legyen $M = \{X_1, \dots, X_k\}$ az anyagok halmaza és $O = \{u_1, \dots, u_k\}$ a műveleti egységek halmaza. Figyeljül meg tehát, hogy az anyagok és műveleti egységek száma egyenlő.

Az első, az ún. Egyenes modellben

$$u_1 = (\alpha_1, \beta_1), \text{ ahol } \alpha_1 = X_1 \text{ és } \beta_1 = X_2,$$

$$u_k = (\alpha_k, \beta_k), \text{ ahol } \alpha_k = X_k \text{ és } \beta_k = X_{k-1},$$

és általában:

$$u_i = (\alpha_i, \beta_i), \text{ ahol } \alpha_i = X_i \text{ és } \beta_i = \{X_{i-1}, X_{i+1}\}, (2 \leq i \leq k-1).$$

Elképzelhetjük sorban egymás mellett a műveleti egységeket úgy, hogy mindegyik egyetlen bemeneti anyaggal rendelkezik és a két szomszédjának a bemeneti anyagait gyártja. Megjegyzendő, hogy a valóságban egy

anyaggal több, egymással összetartozó anyagot vagy információt is modellezhetünk, amit a műveleti egységek feldolgoznak és továbbítanak, jelen esetben a szomszédaiknak.

Az egyenes két végén csak egyetlen szomszéd van. Ha teljesebb szimmetriát akarunk, akkor a második, a Lánc modellünket is tekinthetjük, ahol

$$\beta_1 = \{X_2, X_k\} \text{ és } \beta_k = \{X_{k-1}, X_1\}.$$

Mindkét modellünkben lehetséges $|A_I|$ kiszámítása, azonban számos kombinatorikai probléma adódik, amelyeket meg kell oldanunk. A műveleti egységek részhalmazában gondolkodva, nyilván az A_I olyan $S_I \subseteq O$ műveleti egységek halmazának megfelelő maximális konzisztens döntési leképezéseket tartalmaz, melyekre $u_{i_s} \in S_I$, de az u_{i_s} egyik szomszédja sincs S_I -ben ($1 \leq s \leq j$). (Az Egyenes modellben a két szélső műveleti egységnek csak egy szomszédja van, a Lánc modellben mindegyik műveleti egységnek egységesen két szomszédja van.) Adott I -re legyen tehát $S_I = \{u_{i_1}, \dots, u_{i_j}\}$, $N_I(i)$ az u_i szomszédai indexeinek halmaza, és jelölje

$$F_I = \{i' : i' \neq i_s \text{ és } i' \notin N(i_s), 1 \leq s \leq j, 1 \leq i' \leq k\}$$

a "szabad" műveleti egységek halmazát, tehát azokét, amelyek működése nem érinti $(\mathcal{A}'2)$ teljesülését. Egy konkrét I -re $|A_I| = 2^{|F_I|}$. Sajnos $|F_I|$ nem csak pusztán j -től függ, hanem az I struktúrájától is. Az u_{i_1}, \dots, u_{i_j} nem szomszédos műveleti egységek az Egyenes vagy a Lánc modellben. A leszámolás szempontjából nagyon fontos, hogy hány olyan intervallumra vágják fel a műveleti egységeket, amely egyelemű, és ezek hogyan helyezkednek el.

3.2.1. Az Egyenes modell

Tegyük fel, hogy az $S_I = \{u_{i_1}, \dots, u_{i_j}\}$, műveleti egységei r darab intervallumot határoznak meg I -ben, ahol egy intervallum hosszúságán az I által meghatározott műveleti egység halmaz két eleme közötti műveleti egységek számát értjük. Három esetet különböztethetünk meg az I típusa alapján:

- a) $i_1 = 1$ és $i_j = k$,
 b) $i_1 = 1$ és $i_j < k$ vagy $i_1 > 1$ és $i_j = k$,
 b) $i_1 > 1$ és $i_j < k$.

Könnyű belátni hogy ezen eseteknek megfelelően F_I elemszáma rendre $|F_I| = k - 3j + r + 2$, $|F_I| = k - 3j + r + 1$, illetve $|F_I| = k - 3j + r$.

Kérdés, hogy hány olyan I halmaz van, mely a fenti eseteknek megfelel. Jelölje $L_{\square}(r, j, k)$, $L_{\square}(r, j, k)$, illetve $L_{\square}(r, j, k)$ a fenti eseteknek megfelelő I -k számát. Az r darab egyetlen műveleti egységet tartalmazó intervallumot $\binom{j-1}{r}$ módon lehet kiválasztani a $(j-1)$ -ből, míg a maradék műveleti egységeket a hosszabb intervallumokba rendre $\binom{k-2j}{j-r-2}$, $\binom{k-2j}{j-r-1}$, $\binom{k-2j}{j-r}$ -féleképpen lehet szétosztani a három esetben. Következésképpen

$$\begin{aligned} L_{\square}(r, j, k) &= \binom{j-1}{r} \cdot \binom{k-2j}{j-r-2}, \\ L_{\square}(r, j, k) &= \binom{j-1}{r} \cdot \binom{k-2j}{j-r-1}, \text{ és} \\ L_{\square}(r, j, k) &= \binom{j-1}{r} \cdot \binom{k-2j}{j-r}. \end{aligned}$$

Figyelembe véve a paraméterek határait, a Szitaformula az Egyenes modell esetében azt adja, hogy ([4]):

$$\begin{aligned} L^{(1)} = 2^k + \sum_{1 \leq j \leq \frac{k+1}{2}} (-1)^j \cdot & \left[\sum_{\substack{0 \leq r \leq j-1 \\ k-3j+r+2 \geq 0}} \binom{j-1}{r} \cdot \binom{k-2j}{j-r-2} \cdot 2^{k-3j+r+2} + \right. \\ & + \sum_{\substack{0 \leq r \leq j-1 \\ k-3j+r+1 \geq 0}} \binom{j-1}{r} \cdot \binom{k-2j}{j-r-1} \cdot 2^{k-3j+r+1} + \\ & \left. + \sum_{\substack{0 \leq r \leq j-1 \\ k-3j+r \geq 0}} \binom{j-1}{r} \cdot \binom{k-2j}{j-r} \cdot 2^{k-3j+r} \right]. \end{aligned}$$

3.2.2. A Lánc modell

Jelen esetben is az Egyenes modell jelöléseit fogjuk használni. Most a szimmetriának köszönhetően $|F_I|$ -t könnyebb kiszámolni, nem szükséges eseteket megkülönböztetni, hanem minden I -re $|F_I| = 2^{k-3j+r}$, az I halmazok számának meghatározása azonban éppen a forgásszimmetria miatt nehezebb. A műveleti egységek intervallum struktúráinak száma:

$$\binom{j}{r} \cdot \binom{k-2j-1}{j-r-1},$$

míg az $\{u_1; \text{struktúra}\}$ párok száma:

$$k \cdot \binom{j}{r} \cdot \binom{k-2j-1}{j-r-1},$$

de így j -szer számoltuk az I -ket, ugyanis az u_1 -gyel j -féleképpen vághatjuk el a láncot, bármely intervallumban, így a párok száma j -szer több. A Szi-taformulával megkapjuk a keresett számot a Lánc modell esetében is ([4]):

$$C^{(1)} = 2^k + \sum_{1 \leq j < \frac{k}{2}} (-1)^j \cdot \sum_{\substack{0 \leq r \leq j-1 \\ k-3j+r \geq 0}} \frac{k}{j} \cdot \binom{j}{r} \cdot \binom{k-2j-1}{j-r-1} \cdot 2^{k-3j+r} + e_k,$$

ahol

$$e_k = \begin{cases} (-1)^{\frac{k}{2}} \cdot 2 & , \text{ ha } k \text{ páros,} \\ 0 & , \text{ ha } k \text{ páratlan.} \end{cases}$$

Az e_k „hibatag” kezeli a maximális j esetét külön, ilyenkor csak két I van, a páratlan indexek, vagy a páros indexek.

A fentiek alaján elmondhatjuk, hogy az $L^{(1)}$ és $C^{(1)}$ formulák a tekintett két modellben az $|S(\mathbf{M})|$ -et felülről korlátozzák.

3.2.3. Azonosságok

Általános esetben a felmerülő bonyolult kombinatorikai és gráfelméleti problémák következtében nem várható, hogy a Szitaformula kiszámolható korlátot adjon a lehetséges megoldás struktúrák számára. A tárgyalt speciális struktúrák esetén azonban olyan képleteket kaptunk, melyek lehetővé teszik a felső korlátok közvetlen kiszámolását.

Először az Egyenes modellben, egy $U \subseteq O$, $U = op(\delta)$ akkor teljesíti az $(\mathcal{A}'2)$ feltételt, ha minden $u_i \in U$ műveleti egységnek van szomszédja U -ban. Legyen $O = \{u_1, \dots, u_k\}$ és $U = \{u_{i_1}, \dots, u_{i_t}\}$. Az U -beli műveleti egységek q darab 1 hosszú intervallumot határoznak meg. A fentiekhez hasonló módon megszámlálhatjuk az U és $O \setminus U$ partícióit. Akkor ([4]):

$$L^{(2)} = 1 + \sum_{2 \leq t \leq k} \sum_{1 \leq q \leq \min\{\frac{k}{2}, k-t+1\}} \binom{t-q-1}{q-1} \cdot \binom{k-t+1}{q}.$$

Az U részhalmazainak közvetlen megszámlálása a Lánc modellben sokkal bonyolultabb, ebben az esetben a szimmetria nem segít. A Láncban az u_1 helyzete szerint 3 esetet különböztetünk meg:

- 1) $u_1 \in U$ és bal oldalán van egy másik U -beli elem, de jobb oldalán nincsen;
- 2) $u_1 \in U$ és az i ($i-1$) jobboldali szomszédja U -beli elem, ahol ($i > 1$);
- 3) $u_1 \notin U$ és az i ($i-1$) jobboldali szomszédja nem eleme U -nak, ahol ($i \geq 1$).

Ezek az összes lehetséges különböző esetek. Leszögezett (k, t, q, i) paraméterekre az eseteknek megfelelően az U részhalmazainak száma:

$$\binom{t-q-1}{q-1} \cdot \binom{k-t-1}{q-1},$$

$$\binom{t-i-q+1}{q-1} \cdot \binom{k-t-1}{q-1}, \text{ illetve}$$

$$\binom{t-q-1}{q-1} \cdot \binom{k-t-i}{q-1}.$$

Így a következő képletet kapjuk ([4]):

$$\begin{aligned} C^{(2)} = & 1 + \sum_{2 \leq t \leq k} \left[\sum_{1 \leq q \leq \min\{\frac{t}{2}; k-t\}} \binom{t-q-1}{q-1} \cdot \binom{k-t-1}{q-1} + \right. \\ & + \sum_{2 \leq i \leq t} \sum_{1 \leq q \leq \frac{t-i}{2}+1} \binom{t-i-q+1}{q-1} \cdot \binom{k-t-1}{q-1} + \\ & \left. + \sum_{1 \leq i \leq k-t} \sum_{1 \leq q \leq \min\{\frac{t}{2}; k-t-i+1\}} \binom{t-q-1}{q-1} \cdot \binom{k-t-i}{q-1} \right] + 1, \end{aligned}$$

melyben az első 1-es $t = 0$, az utolsó 1-es pedig a $t = k-t$ esetet jelentik.

Összesítve, két szép kombinatorikus azonosságot is kapunk ([4]):

$$L^{(1)} = L^{(2)} \quad \text{és} \quad C^{(1)} = C^{(2)}.$$

3.2.4. Szemléltetés

Az eredmény hatékonyságának szemléltetésére az alábbi táblázatokban leírjuk az $|S(\mathbf{M})|$ -re kapott korlátokat. Míg az első táblázat a különböző méretű, Egyenes és Lánc struktúrájú feladatokra kiszámolt korlátokat tartalmazza, a második táblázat a korlátok és a maximális döntési leképezések számának arányait szemlélteti a különböző méretű feladatokra.

k :	$L^{(1)}$:	$C^{(1)}$:
3	4	5
4	7	10
5	12	17
6	21	29
7	37	51
8	65	90
9	114	158
10	200	277
11	351	486
12	616	853
13	1081	1497
14	1897	2627
15	3329	4610
16	5842	8090
17	10252	14197
18	17991	24914
19	31572	43721
20	55405	76725
21	97229	134643
22	170625	236282
23	299426	414646
24	525456	727653
25	922111	1276942
26	1618192	2240877
27	2839729	3932465
28	4983377	6900995
29	8745217	12110402
30	15346786	21252274

k :	$\frac{L^{(1)}}{2^k}$:	$\frac{C^{(1)}}{2^k}$:
3	5.0000000000E-01	6.2500000000E-01
4	4.3750000000E-01	6.2500000000E-01
5	3.7500000000E-01	5.3125000000E-01
6	3.2812500000E-01	4.5312500000E-01
7	2.8906250000E-01	3.9843750000E-01
8	2.5390625000E-01	3.5156250000E-01
9	2.2265625000E-01	3.0859375000E-01
10	1.9531250000E-01	2.7050781250E-01
11	1.7138671875E-01	2.3730468750E-01
12	1.5039062500E-01	2.0825195313E-01
13	1.3195800781E-01	1.8273925781E-01
14	1.1578369141E-01	1.6033935547E-01
15	1.0159301758E-01	1.4068603516E-01
16	8.9141845705E-02	1.2344360352E-01
17	7.8216552731E-02	1.0831451416E-01
18	6.8630218504E-02	9.5039367684E-02
19	6.0218811040E-02	8.3391189581E-02
20	5.2838325502E-02	7.3170661930E-02
21	4.6362400052E-02	6.4202785490E-02
22	4.0680170050E-02	5.6334018708E-02
24	3.1319618239E-02	4.3371498586E-02
26	2.4112939849E-02	3.3391669387E-02
28	1.8564525990E-02	2.5708209726E-02
30	1.4292808268E-02	1.9792722558E-02
40	3.8662157089E-03	5.3539468199E-03
60	2.8289357042E-04	3.9175187251E-04
80	2.0700221556E-05	2.8661325825E-05
100	1.5277899843E-06	2.0515052235E-06

3.3. Következtetések

Sajnos a korlát kiszámítása általános esetben nem könnyű feladat. A fenti táblázatokból viszont látszik, hogy elég nagy k -ra az $(\mathcal{A}'2)$ segítségével megfogalmazott korlátok és a maximális döntési leképezések számának arányai 0 fele tartanak, ami mutatja a korlátok élességét, legalábbis a fenti speciális esetekben.

4. fejezet

Összevonásos redukció

A 2.3. alfejezetben leírtunk egy polinomiális idejű algoritmust a maximális struktúra meghatározására, mely csökkentette a feladat méretét azáltal, hogy szükségtelen műveleti egységeket törölt a modelltől. Ebben a fejezetben egy új módszert fogunk kidolgozni, mely a maximális struktúra által meghatározott feladat méretét tovább fogja csökkenteni oly módon, hogy bizonyos műveleti egységek összevonásával csökkenni fog a műveleti egységek száma. A fejezetben bemutatandó módszerek a ([20]) cikk eredményeit tartalmazzák, amely bár közös munka, de jelen dolgozat szerzőjének meghatározó szerepe volt az eredmények elérésében.

4.1. Az összevonás megvalósítása

4.1.1. Definíció. *Két $u_1 = (\alpha_1, \beta_1)$ és $u_2 = (\alpha_2, \beta_2)$ műveleti egység **összevonásán** azok helyettesítését értjük egy új, $u = (\alpha_1 \cup \alpha_2, \beta_1 \cup \beta_2)$ műveleti egységgel.*

Nyilvánvaló, hogy két vagy több műveleti egység összevonásának következtében egy más PNS probléma strukturális modelljét kapjuk. Természetesen, ha ezt az új modellt az eredeti feladat megoldására szeretnénk használni, akkor szoros összefüggéseket kell találnunk az eredeti és az új

strukturális modellek lehetséges megoldásai között. Egy ennek megfelelő új modell megtalálásában az alapvető kérdés nyilván az, hogy mely műveleti egységek vonhatók össze.

4.1.2. Definíció. Legyen $\mathbf{M} = (P, R, O)$ egy PNS probléma redukált strukturális modellje. Akkor az $u_1, u_2 \in O$ műveleti egységeket **összevonhatók**-nak nevezzük, ha bármely lehetséges megoldás struktúrában egyszerre szerepelnek vagy egyikük sem szerepel benne. Formálisan, u_1 és u_2 összevonhatók, ha $u_1 \in (m, o) \iff u_2 \in (m, o), \forall (m, o) \in S(\mathbf{M})$.

Könnyen belátható, hogy ez a reláció reflexív, szimmetrikus és tranzitív, tehát ekvivalencia reláció az O halmazon, melyet \equiv -val fogunk jelölni. Tetszőleges $u \in O$ -ra jelölje $C(u)$ az u műveleti egység ekvivalencia osztályát. Definiáljuk az $\mathbf{M}/\equiv = (P, R, O^*)$ strukturális modellt úgy, hogy

$$O^* = \{(\cup\{\alpha_t : u_t = (\alpha_t, \beta_t) \in C(u)\}, \cup\{\beta_t : u_t = (\alpha_t, \beta_t) \in C(u)\}) : u \in O\}$$

Szemléletesen az \mathbf{M}/\equiv jelentését a következőképpen képzelhetjük el. Minden ekvivalencia osztály esetén, az ekvivalencia osztályhoz tartozó műveleti egységeket egyetlen új műveleti egységbe vonjuk össze, mely az eredetieket fogja helyettesíteni \mathbf{M}/\equiv -ben. Nyilvánvalóan \mathbf{M}/\equiv egy PNS probléma strukturális modellje, melynek maximális struktúrája (M, O^*) .

Definiáljuk a $\Psi : M \cup O \longrightarrow M \cup O^*$ leképezést a következőképpen:

$$\begin{cases} \Psi(X) = X & , \text{ ha } X \in M, \\ \Psi(u_s) = (\cup\{\alpha_t : u_t \in C(u)\}, \cup\{\beta_t : u_t \in C(u)\}) & , \text{ ha } u_s \in C(u), \\ \Psi(m) = \{\Psi(X) : X \in m\} & , \text{ ha } m \subseteq M, \text{ és} \\ \Psi(o) = \{\Psi(u) : u \in o\} & , \text{ ha } o \subseteq O \end{cases}$$

Az utóbbi kiterjesztéseknek megfelelően az (M, O) maximális struktúra (m, o) részfolyamat gráfjának képe a Ψ mellett $(\Psi(m), \Psi(o))$ lesz. Az egyszerűség kedvéért ezt a leképezést is Ψ -vel fogjuk jelölni. Akkor a szoros kapcsolatot az $S(\mathbf{M})$ és $S(\mathbf{M}/\equiv)$ lehetséges megoldás struktúrái között az alábbi tétel mondja ki.

4.1.1. Tétel. ([20]) $A \Psi : S(\mathbf{M}) \longrightarrow S(\mathbf{M}/\equiv)$ egy bijektív leképezés.

Bizonyítás Legyen $(m, o) \in S(\mathbf{M})$ egy lehetséges megoldás struktúra. Először is megmutatjuk, hogy $(\Psi(m), \Psi(o))$ az \mathbf{M}/\equiv egy lehetséges megoldás struktúrája. Nyilvánvalóan $(\Psi(m), \Psi(o))$ az (M, O^*) egy folyamat gráfja. Ezért elegendő bizonyítani, hogy $(\Psi(m), \Psi(o))$ kielégíti a 2.1. alfejezetben definiált $(\mathcal{A}1) - (\mathcal{A}4)$ feltételeket. Mivel $P \subseteq m = \Psi(m)$, ezért az $(\mathcal{A}1)$ feltétel teljesül. Az $(\mathcal{A}2)$ teljesülését is könnyen beláthatjuk, ha észrevesszük, hogy Ψ megőrzi a nyersanyagokat. Az $(\mathcal{A}3)$ bizonyításához legyen $u \in \Psi(o)$ egy tetszőleges műveleti egység. Akkor létezik legalább egy $u_j \in o$, melyre $\Psi(u_j) = u$. Másfelől $(m, o) \in S(\mathbf{M})$, és így $(\mathcal{A}3)$ alapján létezik $[u_j, Y_n]$ út az (m, o) -ban $Y_n \in P$ -vel. Most ha ezen út csomópontjainak Ψ melletti képét tekintjük, akkor egy $[u, Y'_n]$ utat kapunk $(\Psi(m), \Psi(o))$ -ben, melyre $Y'_n \in P$, amiből következik $(\mathcal{A}3)$ teljesülése. Végül $(\mathcal{A}4)$ bizonyítása érdekében vegyünk egy tetszőleges $X \in \Psi(m)$ anyagot. Akkor $X \in m$ és $(\mathcal{A}4)$ alapján létezik olyan $u_j = (\alpha_j, \beta_j)$ műveleti egység, melyre $X \in \alpha_j \cup \beta_j$. Legyen $\Psi(u_j) = (\alpha, \beta)$. A $\Psi(u_j)$ definíciója alapján $X \in \alpha \cup \beta$, ami igazolja $(\mathcal{A}4)$ -et.

A bijektivitás igazolásához először megmutatjuk, hogy Ψ injectív leképezés. Ebből a célból legyen $(m, o) \neq (m', o') \in S(\mathbf{M})$. Ha $m \neq m'$, akkor $\Psi(m) \neq \Psi(m')$, és így a Ψ melletti képek különbözőek. Ellenkező esetben $o \neq o'$. Tegyük fel, hogy $(\Psi(m), \Psi(o)) = (\Psi(m'), \Psi(o'))$. Mivel $o \neq o'$, az általánosság megszorítása nélkül feltehetjük, hogy létezik $u' \in o'$ melyre $u' \notin o$. Legyen $\Psi(u') = u$. Mivel $(\Psi(m), \Psi(o)) = (\Psi(m'), \Psi(o'))$, ezért létezik $\bar{u} \in o$ melyre $\Psi(\bar{u}) = u$. Akkor a Ψ definíciója alapján $\bar{u} \equiv u'$, és így az ekvivalencia reláció definíciójából $u' \in o$, ami ellentmondás. Következésképpen Ψ egy inektív leképezés.

Végül Ψ szürjektivitását kell megmutatnunk, vagyis azt, hogy az $S(\mathbf{M})$ halmazt a teljes $S(\mathbf{M}/\equiv)$ halmazra képezi le. Ebből a célból legyen (m^*, o^*) az $S(\mathbf{M}/\equiv)$ egy tetszőleges lehetséges megoldás struktúrája. Legyen $m = m^*$ és $o = \{u_j : u_j \in O \ \& \ \Psi(u_j) \in o^*\}$. Nyilvánvalóan $\Psi(m, o) = (\Psi(m), \Psi(o)) = (m^*, o^*)$, ezért azt kell bizonyítanunk, hogy (m, o) az \mathbf{M} egy lehetséges megoldás struktúrája. Könnyen belátható, hogy (m, o) az (M, O) részfolyamat gráfja. Az alábbiakban meg fogjuk mutatni az $(\mathcal{A}1) - (\mathcal{A}4)$ feltételek teljesülését is (m, o) -ra.

Mivel $(m^*, o^*) \in S(\mathbf{M}/\equiv)$, ezért az $(\mathcal{A}1)$ feltétel alapján $P \subseteq m^*$. Másfelől $m = m^*$, amiből következik az $(\mathcal{A}1)$ teljesülése (m, o) -ra.

Mivel egy csúcs előző csúcsa (m^*, o^*) -ban egy (m, o) -beli csúcs előző csúcsának Ψ melletti képe, és (m^*, o^*) teljesíti az $(\mathcal{A}2)$ feltételt, ezért az $(\mathcal{A}2)$ feltételt (m, o) -nak is teljesítenie kell.

Most tegyük fel, hogy $(\mathcal{A}3)$ nem teljesül (m, o) -ra. Jelöljük o_1 -el o azon műveleti egységeinek halmazát, melyekre nincsen út (m, o) -ben egy céltermékig, vagyis legyen

$$o_1 = \{u_j : u_j \in o \text{ \& nincsen olyan } [u_j, Y] \text{ út } (m, o)\text{-ban, melyre } Y \in P\}.$$

Feltételezésünk szerint $o_1 \neq \emptyset$. Most legyen (m', o') egy folyamat gráf, melyre $o' = o \setminus o_1$ és $m' = \text{mat}(o')$. Megmutatjuk, hogy (m', o') az \mathbf{M} egy lehetséges megoldás struktúrája.

Mivel $(m^*, o^*) \in S(\mathbf{M}/\equiv)$, $(\mathcal{A}1)$ alapján bármely $X \in P$ -re létezik olyan u műveleti egység, mely X -et közvetlenül gyártja. De akkor létezik egy X -et közvetlenül gyártó u' műveleti egység is o -ban, melyre nyilván $u' \notin o_1$. Következésképpen $u' \in o'$, ami azt mutatja, hogy $P \subseteq m'$, vagyis (m', o') teljesíti az $(\mathcal{A}1)$ feltételt.

$(\mathcal{A}2)$ igazolásához legyen $X \in m'$ egy tetszőleges anyag. Ha $X \in R$, mivel (m^*, o^*) lehetséges megoldás struktúra, ezért X egy forrás (m^*, o^*) -ben. De Ψ definíciója alapján, ha X -nek (m, o) -ban lenne őse, akkor (m^*, o^*) -ban is kellene legyen, ami ellentmondás. Tehát X forrás (m, o) -ban, de $(m', o') \subseteq (m, o)$, ezért X forrás (m', o') -ben. Fordítva, legyen X most forrás (m', o') -ben. Megmutatjuk, hogy akkor X forrás (m, o) -ban is. Ellenkező esetben X kimenete lenne egy $u_1 \in o_1$ -beli műveleti egységnek. Mivel X forrás (m', o') -ben és így X bemenete egy o' -beli műveleti egységnek, ezért létezne egy $[u_1, Y]$ út (m, o) -ban, ami ellentmondana az o_1 definíciójának. Tehát X forrás (m, o) -ban. De akkor X forrás (m^*, o^*) -ban is, viszont (m^*, o^*) lehetséges megoldás struktúra, ezért teljesíti az $(\mathcal{A}2)$ feltételt, amiből azt kapjuk, hogy $X \in R$. Következésképpen (m', o') az $(\mathcal{A}2)$ feltételt is teljesíti.

Az $(\mathcal{A}3)$ és $(\mathcal{A}4)$ feltételek teljesülése következik o_1 és (m', o') definícióiból, ezért azt kapjuk, hogy (m', o') az \mathbf{M} egy lehetséges megoldás struktúrája.

Könnyen belátható, hogy $\Psi(m', o') = (m^*, o^*) = \Psi(m, o)$, amiből Ψ injektivitása miatt azt kapjuk, hogy $(m', o') = (m, o)$. Ez azt jelenti, hogy

$o_1 = \emptyset$, ami ellentmondás. Következésképpen az (m, o) teljesíti az (A3) feltételt.

Igazolnunk kell még, hogy (m, o) az (A4) feltételt is teljesíti. Ha $X \in m$ egy tetszőleges anyag, akkor $X \in m^*$, és mivel (m^*, o^*) teljesíti az (A3)-et, ezért létezik olyan $u = (\alpha, \beta) \in o^*$ műveleti egység, melyre $X \in \alpha \cup \beta$. Ez azt jelenti, hogy létezik olyan $u_j = (\alpha_j, \beta_j) \in o$ műveleti egység, melyre $\Psi(u_j) = u$ és $X \in \alpha_j \cup \beta_j$, ellenkező esetben ugyanis $X \notin \alpha \cup \beta$ lenne, ami ellentmondás. Következésképpen (m, o) teljesíti az (A4) feltételt.

Összesítve, megmutattuk Ψ injektivitását és szürjektivitását, vagyis igazoltuk, hogy Ψ egy bijektív leképezés. \square

Az nyilvánvaló, hogy általában az \mathbf{M}/ \equiv modellben kevesebb műveleti egység található, mint az \mathbf{M} -ben. Mivel a két modell lehetséges megoldás struktúrái kölcsönösen megfeleltethetők egymásnak, ezért szeretnénk az \mathbf{M}/ \equiv modellben egy PNS- β -mal ekvivalens feladatot felírni, amitől azt reméljük, hogy valamivel egyszerűbben megoldható lesz, mint az eredeti feladat.

Definiálunk tehát az \mathbf{M}/ \equiv modellben egy \bar{w} súlyfüggvényt a következőképpen:

$$\bar{w}(u) = \sum_{\substack{u_t \in C(u') \\ \Psi(u')=u}} w(u_t), \text{ minden } u \in O^*\text{-ra}$$

Mivel az ekvivalens műveleti egységek Ψ melletti képei azonosak, ezért \bar{w} jól definiált leképezés.

Így a PNS következő modellje:

$$(PNS-6) \quad \min \left\{ \sum_{u \in o} \bar{w}(u) : (m, o) \in S(\mathbf{M}/ \equiv) \right\}.$$

Kiterjesztjük a súlyfüggvényeket a lehetséges megoldás struktúrákra a következőképpen. Legyen

$$w(m, o) = \sum \{w(u) : u \in o\}, \text{ minden } (m, o) \in S(\mathbf{M})\text{-re, és}$$

$$\bar{w}(m^*, o^*) = \sum \{\bar{w}(u) : u \in o^*\}, \text{ minden } (m^*, o^*) \in S(\mathbf{M}/ \equiv)\text{-re.}$$

Akkor bármely $(m, o) \in S(\mathbf{M})$ lehetséges megoldás struktúrára teljesül az, hogy $w(m, o) = \bar{w}(\Psi(m, o))$. Így a 4.1.1. tétel alapján az alábbi tétel teljesülése is nyilvánvaló.

4.1.2. Tétel. ([20]) *A PNS-3 feladat tetszőleges optimális megoldásának Ψ melletti képe optimális megoldása a PNS-6 feladatnak és fordítva, PNS-6 bármely optimális megoldásának Ψ melletti őse optimális megoldása PNS-3-nak.*

Ahhoz, hogy a PNS probléma adott példányán az összevonásos redukciót elvégezhessük, először meg kell határozni az ekvivalencia relációt.

Legyen $\mathbf{M} = (P, R, O)$ egy PNS probléma strukturális modellje, melyre $S(\mathbf{M}) \neq \emptyset$, továbbá legyen $u_j \in O$ egy tetszőleges műveleti egység. Felépíthetünk egy új PNS strukturális modellt a következőképpen:

$$\mathbf{M}(u_j) = (P, R, O \setminus \{u_j\}).$$

Jelöljük az $\mathbf{M}(u_j)$ maximális struktúráját (M_j, O_j) -vel, feltéve hogy létezik, ellenkező esetben $M_j = O_j = \emptyset$. Akkor a következő állítás érvényes.

4.1.3. Tétel. ([20]) *Bármely $u_i, u_j \in O$ műveleti egységekre $u_i \equiv u_j$ akkor és csak akkor, ha $u_i \in O \setminus O_j$ és $u_j \in O \setminus O_i$ egyidejűleg teljesülnek vagy egyidejűleg nem teljesülnek.*

Bizonyítás Tegyük fel, hogy $u_i \in O \setminus O_j$ és $u_j \in O \setminus O_i$ tetszőleges $u_i \neq u_j \in O$ műveleti egységekre, és legyen (m, o) egy tetszőleges lehetséges megoldás struktúra. Három esetet különböztetünk meg.

1. ESET. (m, o) nem tartalmazza u_i -t. Akkor (m, o) részhalmaza (M_i, O_i) -nek és így feltevésünk értelmében (m, o) nem tartalmazza u_j -t.

2. ESET. (m, o) nem tartalmazza u_j -t. Akkor (m, o) részhalmaza (M_j, O_j) -nek és így feltevésünk értelmében (m, o) nem tartalmazza u_i -t.

3. ESET. (m, o) egyaránt tartalmazza u_i -t és u_j -t is.

Mivel további esetek nincsenek, így bebizonyítottuk $u_i \equiv u_j$ -t.

A feltétel szükségességének bizonyításához legyenek $u_i \neq u_j \in O$ tetszőleges műveleti egységek, melyekre $u_i \equiv u_j$. Az $\mathbf{M}(u_j)$ strukturális modell (M_j, O_j) maximális struktúrája azon lehetséges megoldás struktúrák egyesítése, melyek nem tartalmazzák u_j -t, feltéve, hogy léteznek egyáltalán ilyen lehetséges megoldás struktúrák. Mivel $u_i \equiv u_j$, így ezen megoldás struktúrák egyike sem tartalmazhatja u_i -t, tehát ezek egyesítése sem fogja tartalmazni u_i -t, vagyis $u_i \in O \setminus O_j$. Hasonló indoklással kaphatjuk meg azt is, hogy $u_j \in O \setminus O_i$. Ha minden lehetséges megoldás struktúra tartalmazza u_j -t, azaz $O_j = \emptyset$, akkor az $u_i \equiv u_j$ ekvivalenciából következik, hogy minden lehetséges megoldás struktúra tartalmazza u_i -t is, és így $O_i = \emptyset$ és a megfelelő tartalmazások nyilvánvalóan teljesülnek. \square

Az 4.1.3. tételből kapjuk az alábbi eredményt:

4.1.1. Következmény. $O_j = O \setminus \{u_j\}$ akkor és csak akkor, ha u_j nem összevonható egyetlen másik műveleti egységgel sem.

Legyen $O = \{u_1, \dots, u_n\}$. A 4.1.3. tételből és az MSG maximális struktúra generáló algoritmusból kapjuk az alábbi eljárást az ekvivalencia reláció meghatározására. Az eljárás során alapértelmezésnek tekintjük, hogy más előírás hiányában egy lépés elvégzése után az algoritmus a leírási sorrendben következő lépésére tér.

Összevonásos ekvivalencia relációt meghatározó eljárás (ER)

1. Legyen $i := 1$, $k := 1$, $N = \{1, \dots, n\}$.
2. Határozzuk meg az $\mathbf{M}(u_i)$ maximális struktúráját az MSG maximális struktúra generáló algoritmussal.
3. Ha $O_i = O \setminus \{u_i\}$, akkor legyen $V_k = \{u_i\}$, $N = N \setminus \{i\}$, és $k = k + 1$.
4. Ha $i \neq n$, akkor $i = i + 1$ és térjünk a 2. lépésre.
5. Ha $N = \emptyset$, akkor VÉGE.

Egyébként jelölje i az N legkisebb elemét és legyen

$$J = \{t : t \in N, u_t \in O \setminus O_i\}, \text{ illetve } V = \emptyset.$$

6. Ha $J = \emptyset$, akkor legyen $N = N \setminus \{i\}$, $V_k = V \cup \{u_i\}$, $k = k + 1$, és térjünk az 5. lépésre.
7. Válasszunk egy j elemet J -ből. Legyen $J = J \setminus \{j\}$. Ha $u_i \in O \setminus O_j$, akkor legyen $V = V \cup \{u_j\}$, $N = N \setminus \{j\}$, és térjünk a 6. lépésre.

Az 1. lépés egyszer, a 2., 3., 4. lépések n -szer hajtódnak végre. Ugyanakkor tudjuk, hogy az MSG maximális struktúra generáló algoritmus polinomiális, $O(n^2)$ időben megvalósítható. Az 5. lépésre vagy a 4. lépésről egyszer, vagy a 6. lépésről az N elemszámának csökkentésével tudunk jutni, mindaddig amíg $N \neq \emptyset$. Ebből az következik, hogy az 5. lépés legfeljebb $(|N| + 1)$ -szer hajtódhat végre. A 6. és 7. lépésekre csak az 5. lépés végrehajtása következtében lehet jutni. Az 5. lépés minden végrehajtásához - a J halmaz elemszámának a 7. lépés minden végrehajtásakor történő csökkentése következtében - a 6. lépés egyetlen, a 7. lépés legfeljebb $|J|$ végrehajtása tartozik. Mivel $J \subseteq N$, ezért $|J| \leq |N| = n$. Összességében tehát azt kapjuk, hogy az összevonásos ekvivalencia relációt meghatározó ER eljárás is legrosszabb esetben polinomiális $O(n^3)$ idő alatt végrehajtható.

Az eljárás helyessége az előző tételekből következik, futásának eredményeképpen megkapjuk az \equiv reláció V_1, \dots, V_k ekvivalencia osztályait.

Mindezek után az összevonásos redukcióval kapcsolatban a következő kérdések merülnek fel.

- (1) *Vajon az összevonásos redukció csak egy szép elméleti eredmény, vagy a gyakorlati problémák esetén ténylegesen csökkenti a feladatok méretét?*
- (2) *Képes-e a méretcsökkenés ellensúlyozni az összevonás következtében keletkező műveleti egységek nagyobb bonyolultságát az ismert PNS megoldó algoritmusok futási idejének szempontjából?*

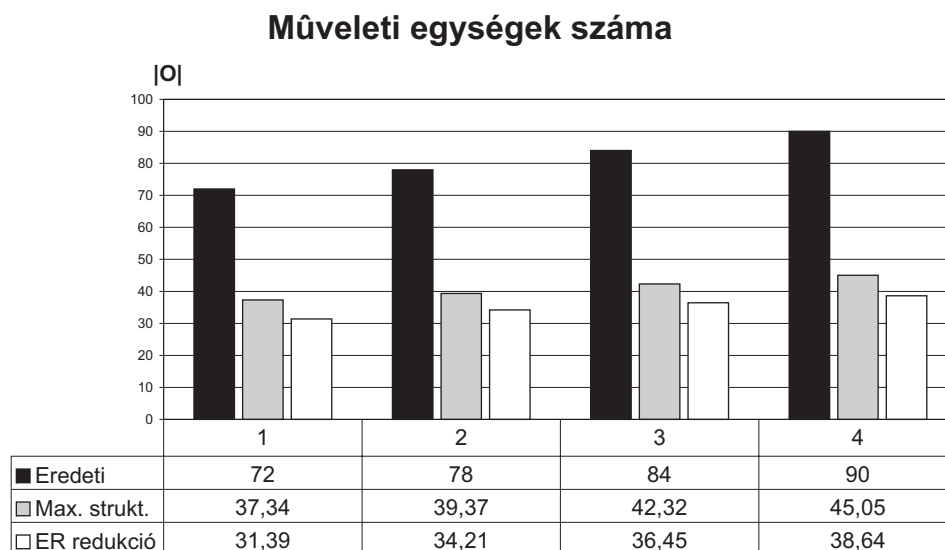
Mindkét kérdést empirikusan vizsgáltuk. A számítási tapasztalatokat és az eredményeket a következő alfejezetben ismertetjük.

4.2. Empirikus analízis

Az első empirikus analízis célja a méretcsökkenés becslése volt. Pontosabban azt vizsgáltuk, hogy az összevonás általában mekkora méretcsökkenést eredményez. Ennek érdekében készítettünk 1000 véletlenül generált PNS problémát (ezek generálására vonatkozóan ld. [25]), és minden problémára meghatároztuk annak maximális struktúráját, majd végrehajtottuk az összevonásos redukciót.

A 4.1. ábra a műveleti egységek átlagos számát mutatja az eredeti problémákban, a maximális struktúrákban, illetve az összevonásos redukció elvégzése után.

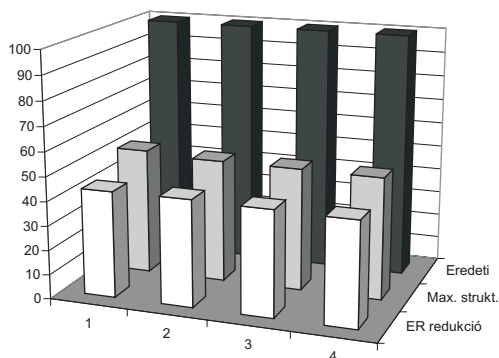
Az 4.2. ábra ugyanezeket az információkat százalékos arányban szemlélteti.



4.1. ábra.

Az empirikus analízisek azt mutatják, hogy az összevonásos redukció általában kb. 7%-os méretcsökkenést eredményez. Nyilvánvaló ugyanakkor, hogy a méretcsökkenés következtében a probléma szerkezete bonyolultabbá válik, nevezetesen a műveleti egységeknek több bemeneti és kimeneti

Műveleti egységek számának százalékos aránya



	1	2	3	4
ER redukció	43,59722222	43,85897436	43,39285714	42,93333333
Max. strukt.	51,86111111	50,47435897	50,38095238	50,05555556
Eredeti	100	100	100	100

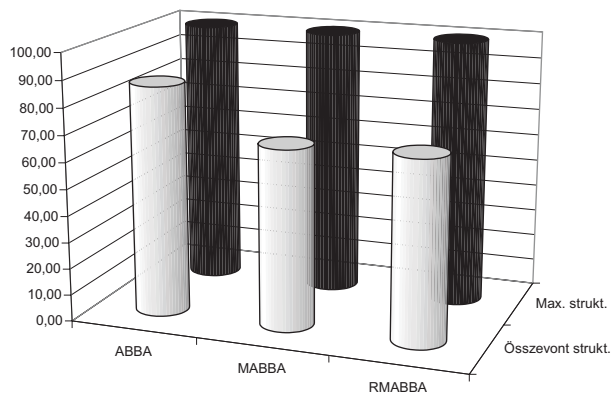
4.2. ábra.

anyagaik lesznek. Ezért érdekesnek látszott az ismert megoldó algoritmusok viselkedésének tanulmányozása is az összevonás után keletkezett feladatokon. Ennek érdekében három algoritmust vizsgáltunk meg: az ABBA ([13, 14]) gyorsított korlátozási és szétválasztási algoritmust (Accelerated Branch-and-Bound Algorithm), az MABBA ([24]) módosított gyorsított korlátozási és szétválasztási eljárást (Modified Accelerated Branch-and-Bound Procedure), és az RMABBA ([25]) finomított módosított gyorsított korlátozási és szétválasztási eljárást. Generáltunk 1000 véletlen PNS problémát, mindegyiknek meghatároztuk a maximális struktúráját, majd elvégeztük az összevonásos redukciót. Végül mindhárom algoritmussal megoldottuk az eredeti maximális struktúrával rendelkező, illetve az összevonás után keletkezett feladatokat. A 4.3. ábra mutatja a futási idők százalékos arányainak átlagait.

4.3. Következtetések

Az empirikus analízisek azt mutatták, hogy az összevonásos redukció megfelelő eszköz a feladat méretének csökkentésére, továbbá az összevonás után

Futási idők százalékos aránya



	ABBA	MABBA	RMABBA
□ Összevont strukt.	86,59	67,75	68,86
■ Max. strukt.	100	100	100

4.3. ábra.

keletkező műveleti egységek fokozottabb bonyolultsága nem feltétlen okoz hosszabb futási időt: a kisebb méretű PNS problémák kevesebb futási időt igényeltek még akkor is, ha a műveleti egységeik bonyolultabbak lettek.

5. fejezet

Előrettekintő B&B algoritmus

A jelen fejezet a szerző saját eredményein alapszik, melyek a [21]-ben nyertek publikálást.

A 2.3. és 4.1. alfejezetekben bemutatunk két technikát (maximális struktúra meghatározása, illetve összevonásos ekvivalencián alapuló redukció), melyek segítségével a probléma méretét csökkenteni tudjuk. A célunk azonban végső sorban mégiscsak az, hogy a PNS feladatot megoldjuk.

Láttuk a 2.5. alfejezetben, hogy a PNS probléma NP teljes ([2, 17, 27]), ami indokoltá teszi a korlátozás és szétválasztás jellegű megoldó algoritmusok használatát. Eddig három ilyen algoritmus volt ismeretes: az ABBA ([14]), MABBA ([24]), illetve RMABBA ([24]). Ezen algoritmusok empirikus analíziseiből ([25]) látszott, hogy az RMABBA-ban használt élesebb korlátozó függvény kiszámítása túlságosan költséges, ezért használata általában nem indokolt. A másik két algoritmus esetén az empirikus vizsgálatok azt mutatták, hogy az MABBA lényegesen jobban dolgozott, mint az ABBA ([25]).

A következőkben bemutatásra kerülő új, LABBA (*Look Ahead B & B Algorithm = Előrettekintő korlátozási és szétválasztási algoritmus*) -nak nevezett eljárás két szempontból is különbözni fog az előzőektől. Először is, felhasználja az előző fejezetben bemutatott összevonás ötletét. Ennek segítségével az előző eljárások korlátozó függvényei javíthatók, a B&B (Branch

and Bound) fa mérete pedig csökkenthető az ekvivalens műveleti egységek egyidejű rögzítésével. A másik szempont, hogy az LABBA figyelembe veszi a különböző választási lehetőségek további hatásait, míg az ABBA és MABBA olyan anyagkiválasztási szabályokon alapszanak, melyek ezt nem tették meg. A különböző anyagkiválasztások következményeinek vizsgálata a B&B fa méretének további csökkenését eredményezi.

5.1. Az algoritmus bemutatása

Nyilvánvalóan, amikor van egy $\delta[m]$ döntési leképezésünk, melyet szeretnénk kiterjeszteni oly módon, hogy az m halmazt bővítjük egy Y elemmel, akkor az Y elemet sokféleképpen kiválaszthatjuk. Az Y kiválasztásának módját **anyagkiválasztási szabálynak** nevezzük, mely nagyon fontos, hiszen meghatározza a B&B algoritmus levél kiválasztási stratégiáját.

A legegyszerűbb anyagkiválasztási szabály az, amikor az $M \setminus (m \cup R)$ halmazból választunk egy anyagot. Ebben az esetben megtörténhet, hogy olyan Y anyagot választunk, melyre $\Delta(Y)$ azon rendelkezésre álló műveleti egységei, melyek Y -t közvetlenül gyártják, egyértelműen meghatározottak. Ez az észrevétel egy másik anyagválasztási módhoz, a ([14])-ban bevezetett **bemenő anyag kiválasztási szabályhoz** vezet. Ebben az esetben az alábbi halmazból választunk:

$$(\text{mat}^{in}(\text{op}(\delta[m]) \cup P) \setminus (m \cup R)).$$

Ez az anyagválasztási mód képezi alapját a ([14])-ben leírt ABBA módszer levél kiválasztási szabályának. Figyeljük meg, hogy egy ilyen anyagválasztási stratégia mellett elképzelhető, hogy olyan Y -t választunk, mely a $\delta[m]$ által már rögzített műveleti egységeknek kimenete. Ebben az esetben megtörténhet, hogy a konzisztencia miatt Y -hoz ugyanazokat a műveleti egységeket kell rendelnünk, melyek a $\delta[m]$ által meghatározott részgráfban Y -t gyártják. Ez a lépés nyilván szükségtelen. Ez a probléma az MABBA algoritmusban meg lett oldva, mert az anyag kiválasztási szabály módosítva lett oly módon, hogy Y -t az alábbi halmazból választjuk:

$$(\text{mat}^{in}(\text{op}(\delta[m])) \cup P) \setminus (\text{mat}^{out}(\delta[m]) \cup R).$$

Az összevonásos ekvivalencia felhasználásával és az anyagválasztás hatásainak vizsgálatával azonban tovább javítható az anyagkiválasztási szabály hatékonysága. Az összevonásos ekvivalencia alapján, amikor egy döntési leképezésben az anyagokhoz hozzárendeljük az őket gyártó műveleti egységeket, akkor bármely műveleti egységgel együtt rögzíthetjük a vele ekvivalens többi műveleti egységet is. Továbbá megtehetjük azt is, hogy a kiterjesztés következő anyagát úgy választjuk ki, hogy az általa bevont műveleti egységek minél több pillanatnyilag szükséges anyagot termeljenek. Ezek eredményeképpen a B&B fa kisebb lesz, az eljárás pedig hatékonyabb.

Az alábbiakban bemutatásra kerülő LABBA algoritmusában fel fogjuk használni ezeket az ötleteket.

Mindenekelőtt bevezetünk néhány speciális döntési leképezést. Ennek érdekében rendeljük hozzá minden $\delta[m] \in \Omega_{\mathbf{M}}$ konzisztens döntési leképezéshez egy $O_{\delta[m]} \subseteq O$ műveleti egység halmazt, melyre

$$O_{\delta[m]} = \text{op}(\delta[m]) \cup \left(\bigcup \{C(u) : u \in \text{op}(\delta[m])\} \right).$$

Így $O_{\delta[m]}$ tartalmazni fogja a $\delta[m]$ által közvetlenül, és az összevonásos ekvivalencia reláció által közvetve rögzített műveleti egységeket.

5.1.1. Definíció. Legyen $\delta[m] \in \Omega_{\mathbf{M}}$, melyre $|m| < |M \setminus R|$; továbbá legyen Y egy anyag, melyre

$$Y \in (\text{mat}^{in}(O_{\delta[m]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta[m]}) \cup R),$$

feltéve, hogy az utóbbi nem üres halmaz. Jelölje K_1, \dots, K_r a $\Delta(Y)$ -nek az \equiv reláció $\Delta(Y)$ -re való szűkítése szerinti ekvivalencia osztályait. Minden $J \subseteq \{K_1, \dots, K_r\}$ nem üres részhalmazra legyen $K_J = \cup \{K_t : K_t \in J\}$. Akkor a

$$\delta_t[m \cup \{Y\}] = \delta[m] \cup \{(Y, K_J)\}, \quad J \subseteq \wp'(\{K_1, \dots, K_r\})$$

alakú konzisztens döntési leképezéseket a $\delta[m]$ Y szerinti **irreguláris kiterjesztésének** nevezzük, ha

$$O_{\delta(A)} \cap \Delta(B) \subseteq \delta(B), \quad \forall A, B \in m \cup \{Y\}.$$

Nyilvánvalóan minden irreguláris kiterjesztés egy kiterjesztés is. Tekintsük az irreguláris kiterjesztés reflexív és tranzitív lezártját az $\Omega_{\mathbf{M}}$ halmazon. Nyilvánvalóan a kapott reláció reflexív, tranzitív és antiszimmetrikus, azaz részben rendezés, melyet \preceq -vel fogunk jelölni. Továbbá, ha $\delta[m] \preceq \delta'[m']$ és $\delta[m] \neq \delta'[m']$, akkor ezt $\delta[m] \prec \delta'[m']$ -vel jelöljük.

Legyen δ_0 az a döntési leképezés, melyre $\delta_0[\emptyset] = \emptyset$.

5.1.2. Definíció. *Legyen*

$$\Sigma_{\mathbf{M}} = \{\delta[m] : \delta[m] \in \Omega_{\mathbf{M}} \ \& \ \delta_0[\emptyset] \preceq \delta[m]\}.$$

A $\Sigma_{\mathbf{M}}$ elemeket **irreguláris döntési leképezéseknek** fogjuk hívni.

Egy döntési leképezés tehát akkor és csakis akkor irreguláris, ha egyenlő $\delta_0[\emptyset]$ -el, vagy megkapható abból irreguláris döntési leképezések sorozatának elemeként. A fenti definíciókból közvetlenül kapjuk az alábbi lemmát.

5.1.1. Lemma. *Ha $\delta[m] \in \Sigma_{\mathbf{M}}$, akkor $\delta(X) \neq \emptyset, \forall X \in m$.*

5.1.3. Definíció. *Legyen $\delta[m] \in \Sigma_{\mathbf{M}}$, melyre*

$$(\text{mat}^{\text{in}}(O_{\delta[m]}) \cup P) \setminus (\text{mat}^{\text{out}}(O_{\delta[m]}) \cup R) = \emptyset.$$

Definiálunk egy δ' döntési leképezést a következőképpen:

$$\delta'(X) = \{(\alpha, \beta) : (\alpha, \beta) \in O_{\delta[m]} \ \& \ X \in \beta\}, \ \forall X \in M \setminus R.$$

*Akkor δ' -t $\delta[m]$ **irreguláris lezárásának** nevezzük és $\text{icl}(\delta[\mathbf{m}])$ -el jelöljük.*

5.1.1. Megjegyzés. Egy reguláris döntési leképezés irreguláris lezárása egyértelműen meghatározott maximális döntési leképezés. Továbbá, ha $m = M \setminus R$, akkor $\text{icl}(\delta[\mathbf{m}]) = \delta[m]$.

A bevezetett irreguláris döntési leképezések segítségével definiáljuk a $\Sigma_{\mathbf{M}}$ halmaz alábbi részhalmazát, amely fontos lesz az eljárás leírásában. Legyen

$$S^*(\mathbf{M}) = \{\text{icl}(\delta[m]) : \delta[m] \in \Sigma_{\mathbf{M}} \& (\text{mat}^{in}(O_{\delta[m]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta[m]}) \cup R) = \emptyset\}.$$

$S^*(\mathbf{M})$ -re érvényes a következő állítás.

5.1.2. Lemma. ([21]) *Legyen $\mathbf{M} = (P, R, O)$ egy PNS probléma strukturális modellje. Akkor $S^*(\mathbf{M}) \subseteq S'(\mathbf{M})$.*

Bizonyítás

Legyen $\delta \in S^*(\mathbf{M})$ egy tetszőleges döntési leképezés, továbbá legyen $o = \text{op}(\delta)$ és $m = \text{mat}(o)$. Nyilvánvalóan (m, o) egy folyamat gráf és $(m, o) \subseteq (M, O)$. Megmutatjuk, hogy $(m, o) \in S(\mathbf{M})$ és $\rho((m, o)) = \delta$.

Mivel $\delta \in S^*(\mathbf{M})$, ezért létezik olyan $\delta_0[m_0] \prec \delta_1[m_1] \prec \dots \prec \delta_k[m_k]$ irreguláris döntési leképezés sorozat, melyre $m_0 = \emptyset$, $\text{icl}(\delta_k[m_k]) = \delta$, $o = O_{\delta_k[m_k]}$, és $(\text{mat}^{in}(O_{\delta_k[m_k]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta_k[m_k]}) \cup R) = \emptyset$. Az utóbbi egyenlőségből következik, hogy $P \subseteq \text{mat}^{out}(O_{\delta_k[m_k]}) = \text{mat}^{out}(o)$.

Most legyen $\bar{o} = \{u : u \in o \& \exists [u, Y_n] \text{ út, melyre } Y_n \in P\}$, és (\bar{m}, \bar{o}) egy folyamat gráf, melyre $\bar{m} = \text{mat}(\bar{o})$. Először igazoljuk, hogy (\bar{m}, \bar{o}) \mathbf{M} egy lehetséges megoldás struktúrája.

Mivel $P \subseteq \text{mat}^{out}(o)$ és $\bar{m} = \text{mat}(\bar{o})$, ezért $P \subseteq \bar{m}$, tehát (\bar{m}, \bar{o}) -ra az (A1) feltétel teljesül.

Az (A2) feltétel bizonyításához legyen $X \in \bar{m}$ egy tetszőleges anyag. Ha $X \in R$, mivel $\bar{o} \subseteq O$, és O nem tartalmaz közvetlenül az X nyersanyagot gyártó műveleti egységeket, ezért nem létezik (Y, X) él (\bar{m}, \bar{o}) -ban. Ha $X \notin R$, akkor $\bar{m} = \text{mat}(\bar{o})$ -ból következik, hogy létezik legalább egy olyan műveleti egység \bar{o} -ban, melynek X közvetlen bemenete vagy kimenete. Így két esetet különböztethetünk meg. Ha létezik olyan műveleti egység \bar{o} -ban, mely közvetlenül gyártja X -et, akkor létezik (Y, X) él (\bar{m}, \bar{o}) -ben,

következésképpen (\bar{m}, \bar{o}) teljesíti az $(\mathcal{A}2)$ feltételt. Ellenkező esetben X -et nem gyártja egyetlen \bar{o} -beli műveleti egység sem, de bemenete \bar{o} legalább egy műveleti egységének. Akkor a \bar{o} definíciója alapján igaz, hogy ez o -ra is érvényes. Ez azt jelenti, hogy $X \in \text{mat}^{in}(op(\delta)) = \text{mat}^{in}(O_{\delta_k[m_k]})$ és $X \notin \text{mat}^{out}(op(\delta)) = \text{mat}^{out}(O_{\delta_k[m_k]})$ ami ellentmond a $\delta \in S^*(\mathbf{M})$ feltevésünknek. Következésképpen ez az eset nem lehetséges, és így (\bar{m}, \bar{o}) teljesíti az $(\mathcal{A}2)$ feltételt.

Az $(\mathcal{A}3)$ feltétel teljesülése (\bar{m}, \bar{o}) -ra az \bar{o} definíciójából következik. Végül, mivel $\bar{m} = \text{mat}(\bar{o})$, ezért (\bar{m}, \bar{o}) az $(\mathcal{A}4)$ feltételt is teljesíti, tehát $(\bar{m}, \bar{o}) \mathbf{M}$ egy lehetséges megoldás struktúrája.

Következő lépésként igazoljuk, hogy $o = \bar{o}$. A \bar{o} definíciója alapján $\bar{o} \subseteq o$. Másfelől i szerinti indukcióval bizonyítjuk, hogy $O_{\delta_i[m_i]} \subseteq \bar{o}$. Ha $i = 1$ és $u \in O_{\delta_1[m_1]}$, akkor létezik olyan $v \in o$ műveleti egység, melyre $u \equiv v$ és v közvetlenül gyárt végterméket. Akkor az \bar{o} definíciójából $v \in \bar{o}$, továbbá mivel $u \equiv v$ és (\bar{m}, \bar{o}) egy lehetséges megoldás struktúra, ezért u és v egyidőben elemei az (\bar{m}, \bar{o}) -nek, és így $u \in \bar{o}$. Most legyen $1 \leq i < k$ és feltételezzük, hogy az állítás igaz $O_{\delta_i[m_i]}$ -ra. Legyen $u \in O_{\delta_{i+1}[m_{i+1}]}$. Akkor két lehetőség van. Ha $u \in O_{\delta_i[m_i]}$, akkor az indukciós feltevés szerint $u \in \bar{o}$. Ellenkező esetben létezik olyan $v \in o$ műveleti egység, melyre $u \equiv v$ és v gyárt legalább egy bemeneti anyagot a $O_{\delta_i[m_i]}$ halmaz műveleti egységei számára. Akkor, \bar{o} definíciója alapján $v \in \bar{o}$. Másfelől, mivel $u \equiv v$ és (\bar{m}, \bar{o}) egy lehetséges megoldás struktúra, teljesül az $u \in \bar{o}$ tartalmazás. Következésképpen $o = \bar{o}$, és így az m és \bar{m} definíciói alapján $(m, o) = (\bar{m}, \bar{o})$.

A bizonyítás befejezéseként vegyük észre, hogy a ρ definíciójából következik a $\rho((m, o)) = \delta$ egyenlőség. \square

5.1.3. Lemma. ([21]) *Legyen $\mathbf{M} = (P, R, O)$ egy PNS probléma strukturális modellje. Ha (m, o) a PNS-3 probléma egy optimális megoldása, akkor $\rho((m, o)) \in S^*(\mathbf{M})$.*

Bizonyítás Legyen $(m, o) \in S(\mathbf{M})$ a PNS-3 probléma egy optimális megoldása, melyre $\rho((m, o)) = \delta$. Igazoljuk, hogy $\delta \in S^*(\mathbf{M})$. Vegyük észre, hogy a bizonyításhoz elegendő megkonstruálni egy $\delta_t[m_t]$, $t = 0, 1, \dots, k$ irreguláris döntési leképezés sorozatot úgy, hogy $\delta_t[m_t] \prec \delta_{t+1}[m_{t+1}]$, $t = 0, 1, \dots, k-1$, és $\text{icl}(\delta_k[m_k]) = \delta$. Ehhez tegyük fel, hogy van egy $\delta_t[m_t]$, $t = 0, 1, \dots, i$,

sorozatunk, melyre $\delta_t[m_t] \prec \delta_{t+1}[m_{t+1}]$, $t = 0, 1, \dots, i-1$, és minden $t = 0, 1, \dots, i$ -re és $X \in m_t$ -re $\delta(X) = \delta_t(X)$. Nyilvánvalóan $m_0 = \emptyset$ és $\delta_0[\emptyset]$ egy megfelelő sorozat $i = 0$ -ra. Most két esetet fogunk megkülönböztetni.

$$1. \text{ ESET. } (mat^{in}(O_{\delta_i[m_i]}) \cup P) \setminus (mat^{out}(O_{\delta_i[m_i]}) \cup R) \neq \emptyset.$$

Legyen Y egy tetszőlegesen rögzített elem a fenti halmazból. Akkor $\Delta(Y) \cap O_{\delta_i[m_i]} = \emptyset$, ugyanis ellenkező esetben létezne egy $u = (\alpha, \beta) \in O_{\delta_i[m_i]}$ műveleti egység $Y \in \beta$ -val, ami ellentmondana Y választásának. Jelöljük most $\delta(Y)$ -t Q -val, továbbá legyen $\equiv_{\Delta(Y)}$ az \equiv összevonásos ekvivalencia reláció $\Delta(Y)$ -ra való szűkítése. Igazoljuk, hogy Q a $\Delta(Y)$ egyes $\equiv_{\Delta(Y)}$ szerinti ekvivalencia osztályainak egyesítése. Ennek érdekében legyen $u, v \in \Delta(Y)$, $u \in Q$ és $u \equiv v$. Akkor $u \in o$, és mivel (m, o) egy lehetséges megoldás struktúra és $u \equiv v$, ezért $v \in o$. Így δ definíciója alapján $v \in Q = \delta(Y)$. Következésképpen Q egyesítése a $\Delta(Y)$ bizonyos ekvivalencia osztályainak. Most legyen $\delta_{i+1}[m_{i+1}] = \delta_i[m_i] \cup \{(Y, Q)\}$. Nyilvánvalóan $\delta(X) = \delta_{i+1}(X)$, $\forall X \in m_{i+1}$, és így $\delta_{i+1}[m_{i+1}]$ egy irreguláris kiterjesztése a $\delta_i[m_i]$ -nak, mely azonos a δ -val az m_{i+1} halmazon. Ily módon a

$$(mat^{in}(O_{\delta_i[m_i]}) \cup P) \setminus (mat^{out}(O_{\delta_i[m_i]}) \cup R) \neq \emptyset$$

esetben létezik a sorozatnak következő tagja.

$$2. \text{ ESET. } (mat^{in}(O_{\delta_i[m_i]}) \cup P) \setminus (mat^{out}(O_{\delta_i[m_i]}) \cup R) = \emptyset.$$

Megmutatjuk, hogy $\text{icl}(\delta_k[m_k]) = \delta$ ahol $k = i$. Legyen ezúttal $o' = \text{op}(\text{icl}(\delta_k[m_k]))$. Akkor $o' = O_{\delta_k[m_k]}$. Másfelől minden $u \in O_{\delta_k[m_k]}$ műveleti egységre létezik olyan $X \in m_k$ anyag és v műveleti egység (mely nem feltétlen különbözik u -tól) úgy, hogy $v \in \delta_k(X)$ és $u \equiv v$. Mivel $\delta_k(Y) = \delta(Y)$, $\forall Y \in m_k$, ezért $v \in \delta(X)$, és így $v \in o$. Most az $u \equiv v$ -ből azt kapjuk, hogy $u \in o$. Következésképpen $o' \subseteq o$.

Az $o' = o$ egyenlőség igazolásához tegyük fel, hogy $o' \subset o$. Mivel $\text{icl}(\delta_k[m_k]) \in S^*(\mathbf{M})$, ezért a 2.6.2. és 5.1.2. lemmákból következik, hogy létezik olyan $(m', o') \in S(\mathbf{M})$ amire $\rho((m', o')) = \text{icl}(\delta_k[m_k])$. Akkor, mivel $o' \subset o$ és a súlyok mind pozitívak, azt kapjuk, hogy

$$\sum_{u \in o'} w(u) < \sum_{u \in o} w(u)$$

ami ellentmondás, mivel (m, o) a PNS-3 feladat optimális megoldása. Következésképpen $o = o'$, és így $\text{icl}(\delta_k[m_k]) = \delta$, amivel igazoltuk a lemmát.

□

A 5.1.2. és 5.1.3. lemmák alapján azt kapjuk, hogy a PNS-5 probléma helyett megoldhatjuk az alábbi feladatot:

$$(PNS-7) \quad \min \left\{ \sum_{u \in \text{op}(\delta)} w(u) : \delta \in S^*(\mathbf{M}) \right\}.$$

A PNS-7 feladat megoldására egy új B&B eljárást fogunk kidolgozni.

A szétválasztási szabály megadásához definiáljuk a $\vartheta(\delta[m])$ függvényt minden $\delta[m] \in \Sigma_{\mathbf{M}}$ irreguláris döntési leképezésre a következőképpen:

$$\vartheta(\delta[m]) = \{ \delta' : \delta' \in S^*(\mathbf{M}) \ \& \ (\exists \bar{\delta}[m] \in \Sigma_{\mathbf{M}}) (\delta[m] \preceq \bar{\delta}[m] \ \& \ \text{icl}(\bar{\delta}[m]) = \delta') \}.$$

Figyeljük meg, hogy $\vartheta(\delta_0[\emptyset]) = S^*(\mathbf{M})$. Egyébként $\vartheta(\delta[m])$ -et a következőképpen tudjuk szemléltetni. $\vartheta(\delta[m])$ a $\delta[m]$ azon irreguláris kiterjesztéseit tartalmazza, melyek a PNS-7 feladatnak lehetséges megoldásai; ezek mindegyike megkapható egy irreguláris kiterjesztés sorozattal és a irreguláris lezárás művelettel. Ezen szemléltetés alapján a $\vartheta(\delta[m])$ elemeit $\delta[m]$ **irreguláris lehetséges megoldás kiterjesztéseinek** hívjuk.

A bevezetett ϑ függvényre vonatkozóan vegyük észre, hogy ha $\delta[m]$ és $\delta'[m']$ két olyan irreguláris döntési leképezés, melyre $\delta(X) \neq \delta'(X)$ valamely $X \in m \cap m'$ -re, akkor

$$\vartheta(\delta[m]) \cap \vartheta(\delta'[m']) = \emptyset.$$

A fentiek értelmében bármely $\delta[m] \in \Sigma_{\mathbf{M}}$ -ra, ha $\delta[m]$ -nek van legalább egy nem reguláris kiterjesztése, akkor definiálhatjuk a $\vartheta(\delta[m])$ nem feltétlen nem triviális partícióját. Ugyanis könnyen belátható, hogy ha $\delta_t[m \cup \{X\}] = \delta[m] \cup \{(X, K_{j_t})\}$, $t = 1, 2, \dots, k$, irreguláris kiterjesztései $\delta[m]$ -nek, akkor a $\vartheta(\delta_t[m \cup \{X\}])$, $t = 1, 2, \dots, k$, halmazok a $\vartheta(\delta[m])$ egy (nem feltétlen nem triviális) partícióját alkotják. Az ilyen partíciók képezik az eljárásunk szétválasztási szabályát. Pontosabban az aktuális B&B fa minden levele $\vartheta(\delta[m])$ alakú lesz, a $\vartheta(\delta_t[m \cup \{X\}])$, $t = 1, 2, \dots, k$ halmazok pedig az adott levél leszármazottai lesznek.

A korlátozás és szétválasztás eljárás másik fő összetevője a korlátozó függvény, mely a B&B fa minden leveléhez hozzárendeli az annak megfelelő lehetséges megoldások célfüggvényértékeinek egy alsó korlátját. Eljárásunkban egy levél $\vartheta(\delta[m])$ alakú lesz, melyhez a $\delta[m]$ irreguláris lehetséges megoldás kiterjesztései tartoznak. Mivel a műveleti egységeknek pozitív súlyuk van, ezért $\delta[m]$ bármely kiterjesztésének súlya nem kisebb, mint az $O_{\delta[m]}$ műveleti egységek súlyainak összege. Ennek megfelelően a g^* korlátozó függvényt a következőképpen definiáljuk:

$$g^*(\delta[m]) = \sum_{u \in O_{\delta[m]}} w(u).$$

A korlátozás és szétválasztás implementációjának elősegítése érdekében a következő jelöléseket, szabályokat és feltételezéseket fogjuk használni.

(a) Az eljárás során L fogja jelölni az élő levelek halmazát, z^* fogja tartalmazni a célfüggvény aktuális pillanatig elért legjobb értékét, z pedig a megfelelő legjobb lehetséges megoldást.

(b) A levél kiválasztási szabály a következő: a $\vartheta(\delta[m])$ levelet akkor választjuk ki L -ből, ha

$$g^*(\delta[m])/|m|$$

értéke minimális, feltéve, hogy $m \neq \emptyset$. Ha több ilyen levél van, akkor ezek közül tetszőlegesen választunk egyet.

(c) Az anyagkiválasztás meghatározásához annak hatásait a következőképpen vizsgáljuk. Figyeljük meg, hogy az aktuális

$$T = (\text{mat}^{in}(O_{\delta[m]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta[m]}) \cup R)$$

halmaz a szükséges, de még nem termelt anyagokat tartalmazza. Akkor egy olyan X -et fogunk választani, melyre a $|(\text{mat}^{out}(\Delta(X)) \cap T)|$ értéke maximális.

(d) Az eljáráshoz hozzátartozik a lehetséges megoldások felismerése: akkor találtunk lehetséges megoldást, ha teljesül az alábbi egyenlőség:

$$(\text{mat}^{in}(O_{\delta[m]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta[m]}) \cup R) = \emptyset.$$

(e) A szétválasztási lépésben egy részprobléma akkor van felderítve, ha $\delta[m]$ -nek a választott X anyagra nézve nincs irreguláris kiterjesztése.

Előretekintő B&B algoritmus (LABBA, Look Ahead B&B Algorithm) ([21])

Inicializálás

- Határozzuk meg az összevonásos ekvivalencia relációt.
- Legyen $L := \{\vartheta(\delta_0[\emptyset])\}$, $z^* := \infty$, $s := \emptyset$, és $r := 0$.
Határozzuk meg $g^*(\delta_0[\emptyset])$ -t.

Iteráció (r. iteráció)

1. *Befejezés*

Ha $L = \emptyset$, akkor VÉGE: az s tartalmazza az optimális megoldást és z^* tartalmazza az optimum értéket. Egyébként térjünk a 2. lépésre.

2. *Levéلكiválasztás*

Ha L egyelemű, akkor válasszuk ki az egyetlen elemét. Egyébként válasszunk egy olyan $\vartheta(\delta[m])$ levelet L -ből, melyre a $g^*(\delta[m])/|m|$ érték minimális; ha több ilyen érték van, akkor válasszunk egyet tetszőlegesen közülük.

3. *Megoldástesztelés*

Ha $T = (\text{mat}^{in}(O_{\delta[m]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta[m]}) \cup R) \neq \emptyset$, akkor térjünk a 4. lépésre.

Egyébként alkossuk meg a $\delta[m]$ irreguláris lezárását, jelölje ezt δ' , továbbá ha $w(\delta') < z^*$, akkor aktualizáljuk a z^* és s értékeket: $z^* := w(\delta')$ és $s := \{\delta'\}$; ellenkező esetben z^* és s értékei nem változnak.

Legyen $\Phi := \emptyset$ és térjünk a 6. lépésre.

4. *Szétválasztás*

Válasszunk egy $X \in T$ anyagot, melyre $|\text{mat}^{\text{out}}(\Delta(X)) \cap T|$ maximális, és alkossuk meg a $\delta[m]$ X szerinti irreguláris kiterjesztéseit.

Ha nem létezik $\delta[m]$ -nek ilyen kiterjesztése, akkor legyen $L := L \setminus \{\vartheta(\delta[m])\}$ és térjünk az 1. lépésre.

Egyébként legyenek $\delta_i[m_i]$, $i = 1, 2, \dots, k$ a $\delta[m]$ X szerinti irreguláris kiterjesztései. Akkor legyen $\Phi = \{\vartheta(\delta_i[m_i]) : 1 \leq i \leq k\}$, és térjünk az 5. lépésre.

5. *Korlátozás*

Számoljuk ki a $g^*(\delta_i[m_i])$ értékeket $i = 1, 2, \dots, k$ -ra, és térjünk a 6. lépésre.

6. *Felderítés*

Definiáljuk újra az L halmazt a következőképpen:

$$L := \{\vartheta(\bar{\delta}[\bar{m}]) : \vartheta(\bar{\delta}[\bar{m}]) \in (L \setminus \{\vartheta(\delta[m])\}) \cup \Phi, g^*(\bar{\delta}[\bar{m}]) < z^*\}.$$

Legyen $r := r + 1$ és kezdjük egy új iterációt (térjünk az 1. lépésre).

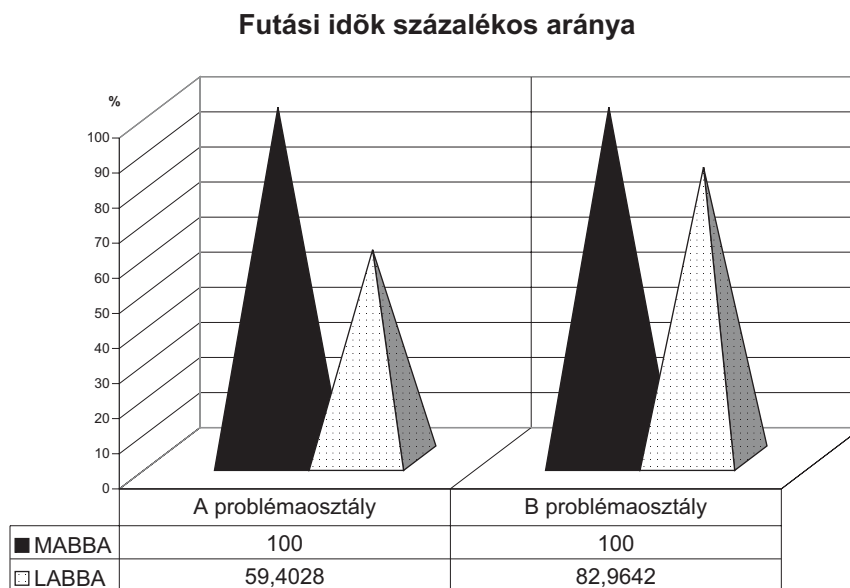
Az algoritmus hatékonyságára vonatkozóan empirikus analíziseket végeztünk, melyek eredményeit a következő részben ismertetjük.

5.2. Empirikus analízis

5.2.1. A vizsgálatok ismertetése

Az ([25])-ban ismertetett empirikus analízisek alapján az MABBA eljárás tűnt a legjobbnak az addig kidolgozottak közül, ezért a mi algoritmusunkat ezzel hasonlítottuk össze. Ennek érdekében kétféle PNS problémaosztályt vizsgáltunk. (Az osztályok és generálásuk részletesebb leírása a ([25])-ban található.) Mindkét osztály esetén 1000 darab 100 anyagot tartalmazó PNS problémát generáltunk véletlenszerűen, melyeket mindkét eljárással (MABBA és LABBA) megoldottunk ([21]).

A 5.1. ábra a futási idők átlagának százalékos arányát, míg a 5.2. ábra az iterációk számának százalékos arányát mutatja.



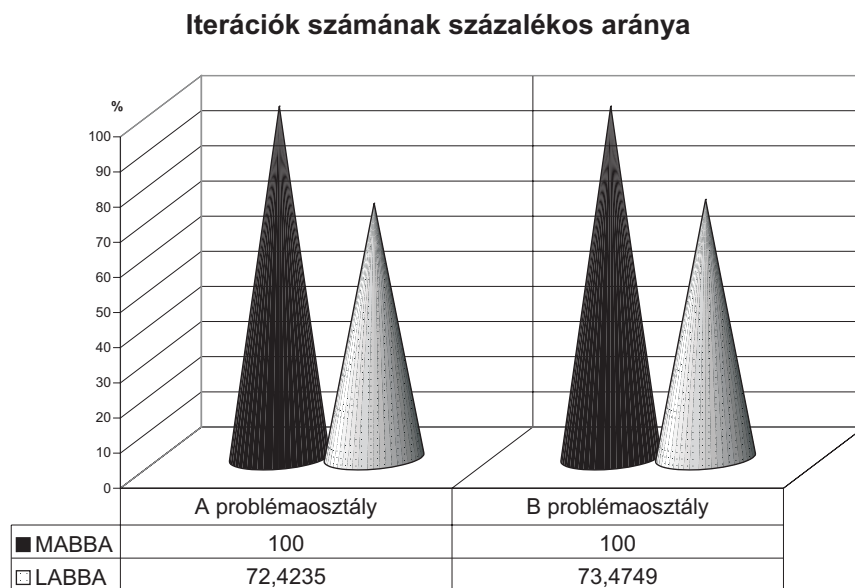
5.1. ábra.

5.2.2. Konklúziók

Az empirikus vizsgálatokra vonatkozóan hangsúlyoznunk kell, hogy azok következtetései csak a vizsgált feladatosztályokra jelenthetők ki bizonyossággal. Megjegyezzük továbbá, hogy a vizsgált osztályok egyike (A feladatosztály (lsd. [25])) több gyakorlati alkalmazásra épül.

Az empirikus analízisünk azt mutatja, hogy úgy futási idő, mint iterációs szám szempontjából az LABBA lényegesen jobbnak bizonyult a régebbi MABBA algoritmusnál. A hatékonyság javulása annak köszönhető, hogy az LABBA a többi eljárástól az alábbiakban különbözik:

(1) *a lehetséges megoldások halmaza a PNS-7-ben kisebb, mint az előző megfogalmazásokban,*



5.2. ábra.

(2) az ekvivalens műveleti egységek súlyát figyelembe vevő g^* korlátozó függvény élesebb az előző korlátozó függvényeknél,

(3) az anyagkiválasztás következményeinek tanulmányozása a B&B fa mélységének csökkenését eredményezi.

Ezen szempontok alapján várható, hogy az LABBA más feladatosztályok esetén is elődeinél hatékonyabban oldja meg a feladatot.

6. fejezet

Parciális leszámplálási eljárás

Ez a fejezet a szerző saját eredményeit tartalmazza, melyek a [23]-ben lettek publikálva.

A 2.7.2 részben bemutatunk egy [16]-ben kidolgozott eljárást, mely a PNS-5 probléma összes lehetséges megoldását felsorolja. Lehetnek azonban olyan esetek is, amikor csak az optimális megoldások érdekelnek, de azokat mind fel szeretnénk sorolni. A feladat megoldható a 2.7.2 részben leírt teljes leszámplálással is, például úgy, hogy először megkeressük az optimumot, majd utána a teljes leszámplálásnál ugyan az összes lehetséges megoldást végigjárjuk, de csak az optimális megoldásokat tartjuk meg. Ez a megoldás azonban egyáltalán nem hatékony, hiszen sok felesleges lehetséges, de nem optimális megoldást vizsgál meg. Ezért kidolgoztunk egy ennél hatékonyabb eljárást, mely ugyan még mindig nem csak az optimális megoldásokat találja meg, viszont az általa felsorolt megoldás halmaz, mely tartalmazza az összes optimális megoldást, az empirikus vizsgálatokból ítélve, lényegesen kisebb, mint az összes lehetséges megoldások halmaza, hiszen az algoritmus a teljes leszámplálásnál sokkal hatékonyabbnak bizonyult.

A fejezet első részében ezt a parciális leszámplálásnak nevezett eljárást fogjuk bemutatni, míg a fejezet második részében tárgyaljuk a teljes és parciális leszámplálási algoritmusok összehasonlítására végzett empirikus vizsgálatokon kapott eredményeket.

6.1. A parciális leszámolás bemutatása

Az alábbi eljárás nem sorolja fel az összes lehetséges megoldást, viszont felsorolja az összes optimális megoldást. Előnye, hogy kevesebb döntési leképezés vizsgálatát igényli, mint a teljes leszámolás, és ennél fogva hatékonyabban dolgozik.

Parciális leszámolás (Partial Enumeration, [23])

Inicializálás

- Határozzuk meg az összevonásos ekvivalenciát. Legyen o_0 azon műveleti egységek halmaza, melyeknek minden lehetséges megoldás struktúrában szerepelniük kell. Legyen $m_0 = \emptyset$ és $i = 0$.

Iteráció

1. Legyen $\delta_i[m_i]$ az $\hat{m}_i = \langle A_{j_1}, \dots, A_{j_k} \rangle$ tartománnyal rendelkező aktuális irreguláris döntési leképezés. Továbbá legyen

$$T_i = (\text{mat}^{in}(O_{\delta_i[m_i]}) \cup P) \setminus (\text{mat}^{out}(O_{\delta_i[m_i]}) \cup R),$$

és térjünk a 2. lépésre.

2. Ha $T_i = \emptyset$, akkor alkossuk meg a $\delta_i[m_i]$ irreguláris lezárását, melyet jelöljünk δ'_i -val. Aktualizáljuk S értékét: $S = S \cup \{\delta'_i\}$, és térjünk a 4. lépésre. Egyébként térjünk a 3. lépésre.
3. Válasszunk egy legkisebb indexű X anyagot T_i -ből, melyre

$$|\text{mat}^{out}(\Delta(X)) \cap T_i|$$

maximális. Vizsgáljuk meg a $\delta_i[m_i]$ döntési leképezés $\Delta(X) \setminus o_0$ megfelelő részhalmazaira való irreguláris kiterjesztéseit a \models lineáris rendezés figyelembe vételével.

Válasszuk az első olyan $K_J \subseteq \Delta(X) \setminus o_0$ részhalmazt, melyre $\delta_i[m_i] \cup \{(X, K'_J)\}$ a $\delta_i[m_i]$ egy irreguláris kiterjesztése, ahol

$$K'_J = K_J \cup (\Delta(X) \cap o_0).$$

feltételezve, hogy létezik ilyen K'_J . Legyen $\hat{m}_{i+1} = \hat{m}_i \vee \{X\}$ és

$$\delta_{i+1}[m_{i+1}] = \delta_i[m_i] \cup \{(X, K'_J)\}.$$

Legyen $i = i + 1$, és térjünk a következő iterációs lépésre.

Ha $\Delta(X)$ egyetlen megfelelő részhalmaza sem lehet a $\delta_i[m_i]$ irreguláris kiterjesztése, akkor térjünk a 4. lépésre.

4. Ha $\delta_i(A_{j_k}) \subset \Delta(A_{j_k})$ és van olyan $K_J \subseteq \Delta(A_{j_k}) \setminus o_0$ részhalmaz, melyre $\delta_i(A_{j_k}) \neq K'_J$, $\delta_i(A_{j_k}) \leq K'_J$, ahol

$$K'_J = K_j \cup (\Delta(X) \cap o_0)$$

és a

$$\{(A_{j_1}, \delta_i(A_{j_1}))\} \cup \dots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}}))\} \cup \{(A_{j_k}, K'_J)\}$$

döntési leképezés a $\{(A_{j_1}, \delta_i(A_{j_1}))\} \cup \dots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}}))\}$ -nak egy irreguláris kiterjesztése, akkor térjünk a 5. lépésre. Egyébként térjünk a 6. lépésre.

5. Válasszuk a $\Delta(A_{j_k}) \setminus o_0 \models$ szerinti első 4. pontbeli feltételt kielégítő részhalmazát és jelöljük K_J -vel. Legyen

$$\hat{m}_{i+1} = \langle A_{j_1}, \dots, A_{j_k} \rangle, \text{ és}$$

$$\delta_{i+1}[m_{i+1}] = \{(A_{j_1}, \delta_i(A_{j_1}))\} \cup \dots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}}))\} \cup \{(A_{j_k}, K'_J)\},$$

ahol

$$K'_J = K_j \cup (\Delta(X) \cap o_0).$$

Legyen $i := i + 1$, és térjünk a következő iterációs lépésre.

6. Legyen $k = k - 1$. Ha $k = 0$, akkor VÉGE. Egyébként térjünk a 4. lépésre.

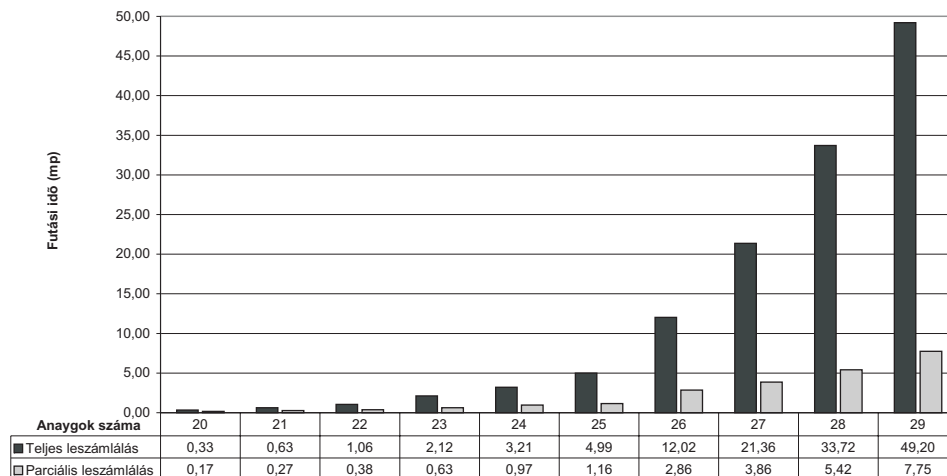
Az, hogy az eljárás a megfelelő lehetséges megoldásokat határozza meg, következik abból, hogy S pontosan az $S^*(\mathbf{M})$ elemeit fogja tartalmazni és így a 5.1.3. lemma alapján ezek tartalmazni fogják a PNS-3 összes optimális megoldását.

6.2. Empirikus analízis

Az, hogy a parciális leszámlálás a teljes leszámlálásnál hatékonyabban oldja meg az optimális megoldások felsorolásának feladatát, a fenti elméleti eredmények fényében várható volt. Ennek mértékét elméleti módszerekkel azonban meglehetősen nehéz feladat megadni, ezért ismételten az empirikus vizsgálatokhoz folyamodtunk. Minden $n = 20, 21, \dots, 29$ -re generáltunk 1000 darab n anyagot tartalmazó maximális struktúrájú véletlen PNS problémát a [25] cikknek megfelelően. Minden problémát megoldottunk a teljes illetve parciális leszámlási algoritmussal, és minden megoldás esetén mértük a futási időt és meghatároztuk az iterációs lépések számát. Végül átlagoltuk az eredményeket, melyeket a 6.1. illetve a 6.2. ábrákon láthatunk.

Az 5.1.2. lemmából tudjuk, hogy $S^*(\mathbf{M}) \subseteq S'(\mathbf{M})$. Szerettük volna azonban megbecsülni az eltérés mértékét is, ezért minden feladatra a két halmaz számosságának arányát is meghatároztuk. A 6.3. ábra a kapott $|S^*(\mathbf{M})|/|S'(\mathbf{M})|$ arányok átlagait szemlélteti a különböző méretű feladatok esetén.

Futási idők átlagai



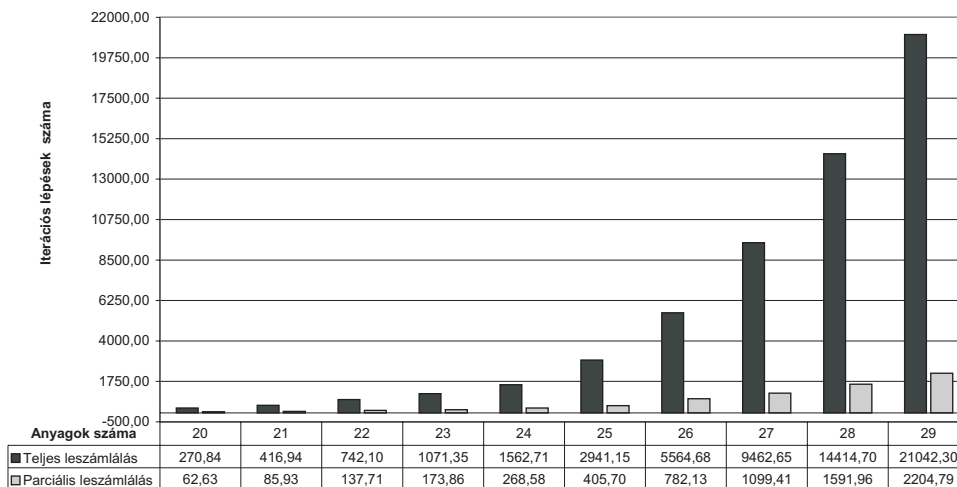
6.1. ábra.

Végül kíváncsiak voltunk arra is, hogy a parciális leszámolás mennyivel lesz hatékonyabb a teljes leszámolásnál egy tényleges gyakorlati probléma megoldásában, ezért az összehasonlítást a [16]-ben leírt Folpet (N-(trichloromethylthio)phthalamide) folyamatra vonatkozóan is elvégeztük és a következő eredményeket kaptuk:

	Teljes leszámolás	Parciális leszámolás
Iterációs lépések száma	15573	2305
Futási idő (mp)	36	9
$S'(\mathbf{M})$ és $S^*(\mathbf{M})$	3465	1575

Ugyan az arányok szempontjából lényegtelen, de a teljesség kedvéért megjegyezzük, hogy az empirikus vizsgálatok egy AMD-K6(tm)-2/333 CPU-val rendelkező PC-n voltak elvégezve.

Az iterációs lépések számainak átlagai



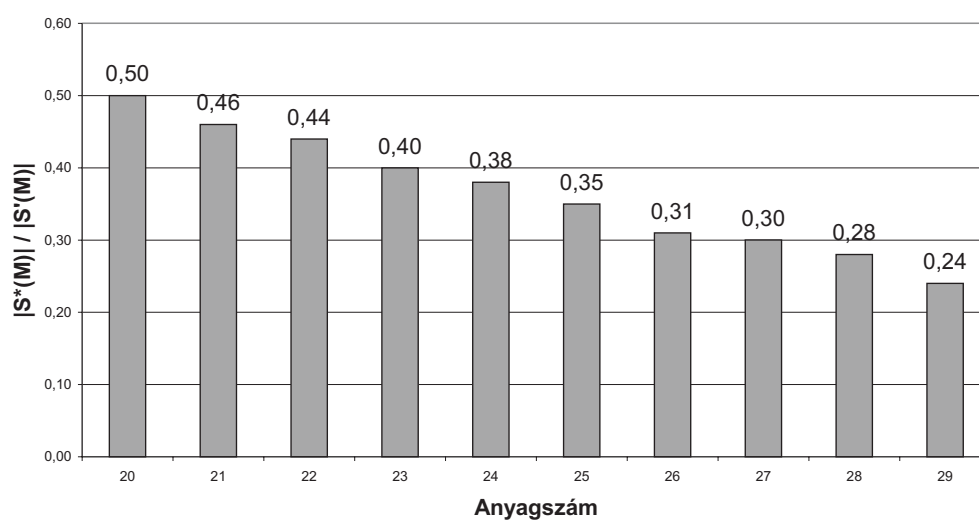
6.2. ábra.

6.2.1. Konklúziók

Az empirikus analízisek azt mutatták, hogy úgy a futási idő, mint az iterációs lépések számában az új parciális leszámítás jobbnak bizonyult a korábbi teljes leszámításnál. Hasonló eredményeket kaptunk a valós ipari alkalmazás esetén is. Mindez persze nem jelenti azt, hogy a teljes leszámítás a továbbiakban szükségtelen lenne, hiszen a hátránya előnye is egyben, hogy az összes lehetséges megoldást meg tudja adni, ellentétben a parciális leszámítással, ha viszont csak az optimális megoldásokat keressük, akkor azok felsorolására az új parciális leszámítás sokkal alkalmasabbnak bizonyult.

Az eljárások által meghatározott lehetséges megoldáshalmazokkal kapcsolatban viszont nyitott kérdés maradt az, hogy

a parciális leszámítás milyen feltételek mellett képes csak az optimális megoldásokat, illetve az összes lehetséges megoldást felsorolni.

Az $|S^*(M)| / |S'(M)|$ arányok átlagai

6.3. ábra.

7. fejezet

Automataelméleti megközelítés

Ebben a fejezetben a PNS probléma egy eddigiektől eltérő jellegű megoldását fogjuk bemutatni. A [28] cikkben a PNS egy módosított változata lett kidolgozva, melynek megoldására egy automataelméleti megközelítésen alapuló eljárást került kidolgozásra. Az eljárás alapötlete az, hogy a módosított PNS probléma adott példányához hozzárendelhető egy automata, melyre teljesül az, hogy az eredeti feladat megoldása visszavezethető az automata átmeneti gráfjában egy legrövidebb út megtalálására. A további kutatások során ([22]) kiderült, hogy ez az algoritmus tovább javítható oly módon, hogy a legrövidebb út keresésével egyidőben az átmeneti gráfnak csak egy részét generáljuk.

7.1. A módosított PNS probléma

A 2.1.3 definícióban az [18] és [15] alapján megadtunk $(\mathcal{A}1)$ - $(\mathcal{A}4)$ feltételeket, melyeket egy folyamat részgráfnak ki kell elégítenie ahhoz, hogy lehetséges megoldás struktúrának tekintsük. Ezek a feltételek azonban még mindig megengedik olyan lehetséges megoldás struktúrák létezését, melyek a valóságban nem tudnak végrehajtódni. A végrehajthatóság biztosítása céljából a [28]-ban az $(\mathcal{A}1)$ - $(\mathcal{A}4)$ feltételek ki lettek egészítve egy újabb feltétellel és az így kapott feladatot **módosított PNS problémának** nevezték el. Ennek a megoldására a [29]-ban leírt ötletet fogjuk továbbfejlesz-

teni úgy, hogy definiálunk a [28]-ban leírt módon egy automatát, melyről a [28] alapján tudjuk, hogy a módosított PNS probléma optimális megoldásának megtalálása ekvivalens az automata átmeneti grájában egy legrövidebb út megtalálásával, és ily módon feladatunk az átmeneti gráfban egy ilyen legrövidebb út minél hatékonyabb megtalálása lesz.

Mindenekelőtt azonban definiálnunk kell a módosított PNS problémát, ami azt jelenti, hogy meg kell határoznunk a végrehajthatósági feltételt. Ehhez először is egy eljárást fogunk írni, amit folyamat gráf **színező eljárásának** fogunk nevezni. Ennek érdekében legyen (\bar{M}, \bar{O}) egy folyamat gráf és R egy anyaghalmaz. Azt mondjuk, hogy (\bar{M}, \bar{O}) az R által **színezhető**, ha az (\bar{M}, \bar{O}) minden csúcspontja beszínezhető az alábbi eljárással.

Színező eljárás ([28])

1. Színezzük be $\bar{M} \cap R$ minden anyagát.
2. Mindaddig, amíg van olyan műveleti egység, melynek minden bemenete színezve van, válasszunk egy ilyen műveleti egységet és színezzük be annak kimeneti anyagait. Ha nincs ilyen műveleti egység, akkor VÉGE.

Megjegyzendő, hogy a színezhetőség tulajdonképpen a végrehajthatóságot jelenti. Ha egy folyamat gráf színezhető, akkor minden műveleti egységéhez hozzárendelhető az az időpont, amikor a kimeneti anyagait beszíneztük. Megfelelő időegység választásával, adott műveleti egység színezési időpontját a műveleti egység folyamatban való ütemezési időpontjának is tekinthetjük. Egy ilyen ütemezett folyamat nyilvánvalóan elvégezhető, ami nem jelenti azt, hogy az így kapott ütemezés optimális is lenne. Fordítva, ha egy folyamat végrehajtható, akkor az abban résztvevő műveleti egységeknek létezik egy ütemezése, és ebben a sorrendben a folyamat műveleti egységei be is színezhetőek.

Most már definiálhatjuk a végrehajthatósági feltételt:

$$(A5) \quad (\bar{M}, \bar{O}) \text{ az } R \text{ által színezhető.}$$

Egy PNS probléma azon lehetséges megoldás struktúráit, amelyek az (A1) - (A4) feltétel mellett még az (A5) feltételt is teljesítik, **módosított**

lehetséges megoldás struktúráknak nevezzük. Ha $\mathbf{M} = (P, R, O)$ egy PNS probléma strukturális modellje, akkor jelöljük $\bar{S}(\mathbf{M})$ -el a módosított lehetséges megoldás struktúrák halmazát. Most megadhatjuk a megoldás struktúrákhoz rendelt optimalizációs problémát:

$$(PNS-8) \quad \min \left\{ \sum_{u \in \bar{O}} w(u) : (\bar{M}, \bar{O}) \in \bar{S}(\mathbf{M}) \right\}.$$

A továbbiakban **módosított PNS problémán** a $(PNS-8)$ problémát értjük. A [28]-ban tárgyaltak szerint a $(PNS-8)$ is egy NP nehéz feladat, ami az eredeti PNS-2 probléma NP nehézségének [2]-ben leírt igazolásához hasonlóan bizonyítható.

Nyilvánvalóan $\bar{S}(\mathbf{M})$ egy véges, nem üres halmaz, így a $PNS-8$ -nak van optimális megoldása. Másfelől hatékonyan eldönthető, hogy $\bar{S}(\mathbf{M}) = \emptyset$ teljesül-e a következőképpen. A 2.3 alfejezetben bemutatunk egy algoritmust, mely polinomiális időben eldönti az ” $S(\mathbf{M}) = \emptyset$?” kérdést és - amennyiben létezik - meghatározza az (\bar{M}, \bar{O}) maximális struktúrát. Ha (\bar{M}, \bar{O}) létezik, akkor rátérhetünk annak színezésére, mely meghatározza abban azt az $(\widehat{M}, \widehat{O})$ folyamat részgráfot, mely azon műveleti egységeket tartalmazza, melyek kimenete színezett. Könnyen bizonyítható, hogy $S(\mathbf{M}) \neq \emptyset$ akkor és csak akkor, ha $P \subseteq \widehat{M}$. Ebben az esetben törölve $(\widehat{M}, \widehat{O})$ -ből azokat a műveleti egységeket, melyek nem teljesítik az $(\mathcal{A}3)$ feltételt, megkapjuk (M, O) azon legnagyobb folyamat részgráfját, mely teljesíti az $(\mathcal{A}1)$ - $(\mathcal{A}5)$ feltételeket. Az eldönthetőség alapján a továbbiakban csak módosított lehetséges megoldásokkal rendelkező módosított PNS problémákkal fogunk foglalkozni.

Megjegyezzük továbbá, hogy amennyiben az $\mathbf{M} = (O, P, R)$ strukturális modell (M, O) folyamatgráfja körmentes, úgy az eredeti és a módosított PNS problémák egybeesnek. Körmentes PNS problémák megoldásával kapcsolatosan bővebben a [7] és [31] cikkekben olvashatunk.

7.2. A módosított PNS probléma megoldása

7.2.1. Automataelméleti alapfogalmak

Legyen X egy tetszőleges nem üres halmaz. Jelölje X^* az X feletti véges szavak halmazát, λ az üres (0 karakterből álló) szót, és legyen $X^+ = X^* \setminus \{\lambda\}$. Tetszőleges $p, q \in X^*$ szavakra azt mondjuk, hogy p **részszava** q -nak, és ezt $p \propto q$ -vel jelöljük, ha p megkapható q -ból néhány karakter törlésével.

7.2.1. Definíció. *Automata* alatt egy olyan $\mathbf{A} = (A, X)$ párost értünk, melyben A az **állapotok** véges, nem üres halmaza, X a **bemeneti jelek** véges, nem üres halmaza, és minden $x \in X$ egy egyoperandusú $x^{\mathbf{A}}$ műveletet határoz meg A -n. Minden $a \in A$, $x \in X$ -re $ax^{\mathbf{A}}$ úgy értelmezhető, mint az az állapot, melybe \mathbf{A} kerül az x bemeneti jel feldolgozása következtében. Tetszőleges $p \in X^*$ szóra $ap^{\mathbf{A}}$ induktív módon definiálható a következőképpen:

$$(1) a\lambda^{\mathbf{A}} = a,$$

$$(2) ap^{\mathbf{A}} = (av^{\mathbf{A}})x^{\mathbf{A}}, \text{ ha } p = vx, v \in X^* \text{ és } x \in X.$$

7.2.2. Definíció. Egy $\mathbf{A} = (A, X)$ automata **átmeneti gráfján** egy olyan $\mathcal{G}_{\mathbf{A}} = (A, E)$ gráfot értünk, melyben minden $a, b \in A$ állapot párra $(a, b) \in E$ akkor és csak akkor, ha létezik olyan $x \in X$ bemeneti jel, melyre $ax^{\mathbf{A}} = b$. Az átmeneti gráf éleit szokás szerint a megfelelő bemeneti jellel címkézzük.

7.2.3. Definíció. Egy **felismerő** egy olyan $\mathcal{A} = (\mathbf{A}, a_0, F)$ rendszer, mely egy $\mathbf{A} = (A, X)$ automatából, egy $a_0 \in A$ **kezdőállapotból**, és egy $F \subseteq A$ **végállapotok halmazából** áll. Az \mathcal{A} által **felismert** vagy **elfogadott nyelvet** a következő halmaz definiálja:

$$L(\mathcal{A}) = \{p : p \in X^* \text{ and } a_0 p^{\mathbf{A}} \in F\}.$$

Azt is szoktuk mondani, hogy $L(\mathcal{A})$ felismerhető (elfogadott) az \mathbf{A} automatával.

7.2.2. A probléma megoldása

A PNS-8 probléma bármely strukturális modelljéhez hozzá fogunk rendelni egy automatát úgy, hogy a probléma módosított lehetséges megoldás strukturái leírhatók legyenek az automata olyan bemeneti szavaival, melyeket elfogad egy, az automata alapján felépített, felismerő. Akkor az automata átmeneti gráfját a műveleti egységek súlyaival címkézve, a súlyozott gráfban a kezdőállapotból a végállapotok halmazába vezető bármely legrövidebb út a PNS strukturális modelljének egy optimális megoldását határozza meg.

Az automata megépítéséhez legyen $\mathbf{M} = (O, P, R)$ egy PNS probléma strukturális modellje és w a súlyfüggvény.

A $\mathbf{B} = (B, O')$ automatát a következőképpen definiáljuk. Legyen $B = B' \cup \{\diamond\}$, melyre $B' = \varphi'(M)$ és $\diamond \notin B'$, és legyen

$$O' = \{u : u = (C, D) \in O \text{ és } R \cap D = \emptyset\}.$$

Az automata egy állapota megfelel egy adott pillanatban rendelkezésre álló anyagok halmazának. Az \diamond állapot a sikertelen átmenetek jelölésére szolgál. Az átmeneteket a következőképpen definiáljuk. Minden $Q \in B'$ és $u = (C, D) \in O'$ -re legyen

$$Qu^{\mathbf{B}} = \begin{cases} Q \cup D & \text{ha } C \subseteq Q \\ \diamond & \text{egyébként,} \end{cases}$$

továbbá

$$\diamond u^{\mathbf{B}} = \diamond.$$

7.2.1. Megjegyzés. Ha $Q \in \mathbf{B}$, p egy O' feletti szó, és $u \in O'$ előfordul p -ben, akkor $Q(pu)^{\mathbf{B}} = Qp^{\mathbf{B}}$.

A $\mathcal{G}_{\mathbf{B}}$ gráf súlyozását a következőképpen definiáljuk. Ha (Q, Q') egy él \mathcal{G} -ben, melynek címkéi u_{j_1}, \dots, u_{j_t} , akkor az él súlya

$$w' = \min\{w(u_{j_1}), \dots, w(u_{j_t})\}$$

lesz, továbbá egyetlen olyan u_{j_l} , $1 \leq l \leq t$, címkét tartunk meg, melyre $w' = w(u_{j_l})$, a többi címkéket töröljük. Jelöljük az így kapott súlyozott, címkézett gráfot $(\mathcal{G}_{\mathbf{B}}, w)$ -vel.

Legyen most $\mathcal{B} = (\mathbf{B}, R, F)$ egy olyan felismerő, melyre $F = \{Q : Q \in B' \text{ és } P \subseteq Q\}$. Akkor teljesül az alábbi állítás.

7.2.1. Lemma. ([28]) *Minden $p = u_{i_1} \dots u_{i_k} \in L(\mathcal{B})$ szóra, ha $[R, Rp^{\mathbf{B}}]$ egy R -ből a $(\mathcal{G}_{\mathbf{B}}, w)$ egy vágállapotába vezető legrövidebb út, akkor u_{i_1}, \dots, u_{i_k} páronként különbözők, továbbá (\bar{M}, \bar{O}) a PNS-8-nak egy optimális megoldása, ahol $\bar{O} = \{u_{i_1}, \dots, u_{i_k}\}$ és $\bar{M} = \text{mat}(\bar{O})$.*

A fentiek alapján, egy optimális megoldás meghatározásához a következő lépéseket kell megtennünk:

1. Építsük meg a \mathbf{B} automata átmeneti gráfját és határozzuk meg a végállapot halmazt.
2. Végezzük el az átmeneti gráf súlyozását és újracímkézését az előzőekben leírtak szerint oly módon, hogy végül minden él egyetlen címkével rendelkezzen.
3. Határozzunk meg egy R -ből F -be vezető legrövidebb utat.
4. A kapott legrövidebb út segítségével határozzuk meg a PNS-8 egy optimális megoldását.

7.2.2. Megjegyzés. Figyeljük meg, hogy az eljárás általában nem igényli a teljes átmeneti gráf megépítését, hanem elegendő az R -ből generált részgráf elkészítése.

Mivel az állapotok \bar{M} részhalmazai, a fenti megjegyzés alapján felépíthetünk egy eljárást, mely az átmeneti gráfnak csak egy részgráfját generálja, és ezzel egyidőben egy megfelelő legrövidebb utat is meghatároz a gráfban.

7.3. A PAT algoritmus

A továbbiakban egy, a szerző saját munkájának eredményeként létrejött és a [22]-ben publikált eljárást fogunk bemutatni, mely a fenti és néhány további észrevétel alapján az előző algoritmusnál hatékonyabban fogja megtalálni az optimális megoldást.

7.3.1. Megjegyzés. Adott állapotban általában a sikeres átmenetet biztosító bemeneti jelek az O' halmaznak csupán egy valódi részhalmazát képezik.

7.3.2. Megjegyzés. Legyenek Q és Q' állapotok, melyek legkisebb távolsága R -től $w = w([R, Q])$ illetve $w' = w([R, Q'])$. Ha $Q' \subseteq Q$ és $w' \geq w$, akkor Q' és a Q' -be bemenő és kimenő élek törölhetők az átmeneti gráfból.

Most definiáljunk egy újabb \ll relációt a műveleti egységek O' halmazán.

7.3.1. Definíció. Minden $u, v \in O'$ -re legyen

$$v \ll u \text{ ha } u = v \text{ vagy } \text{mat}^{\text{out}}(v) \cap \text{mat}^{\text{in}}(u) \neq \emptyset.$$

Ez a reláció reflexív és tranzitív. Jelölje \ll^* a \ll tranzitív lezárását.

7.3.2. Definíció. Azt mondjuk, hogy két $u, v \in O'$ műveleti egység **egymást kölcsönösen eléri**, ha $u \ll^* v$ és $v \ll^* u$.

Könnyen belátható, hogy a kölcsönös elérhetőség reflexív, tranzitív és szimmetrikus, azaz ekvivalencia reláció O' -n, melyet \bowtie -nel fogunk jelölni.

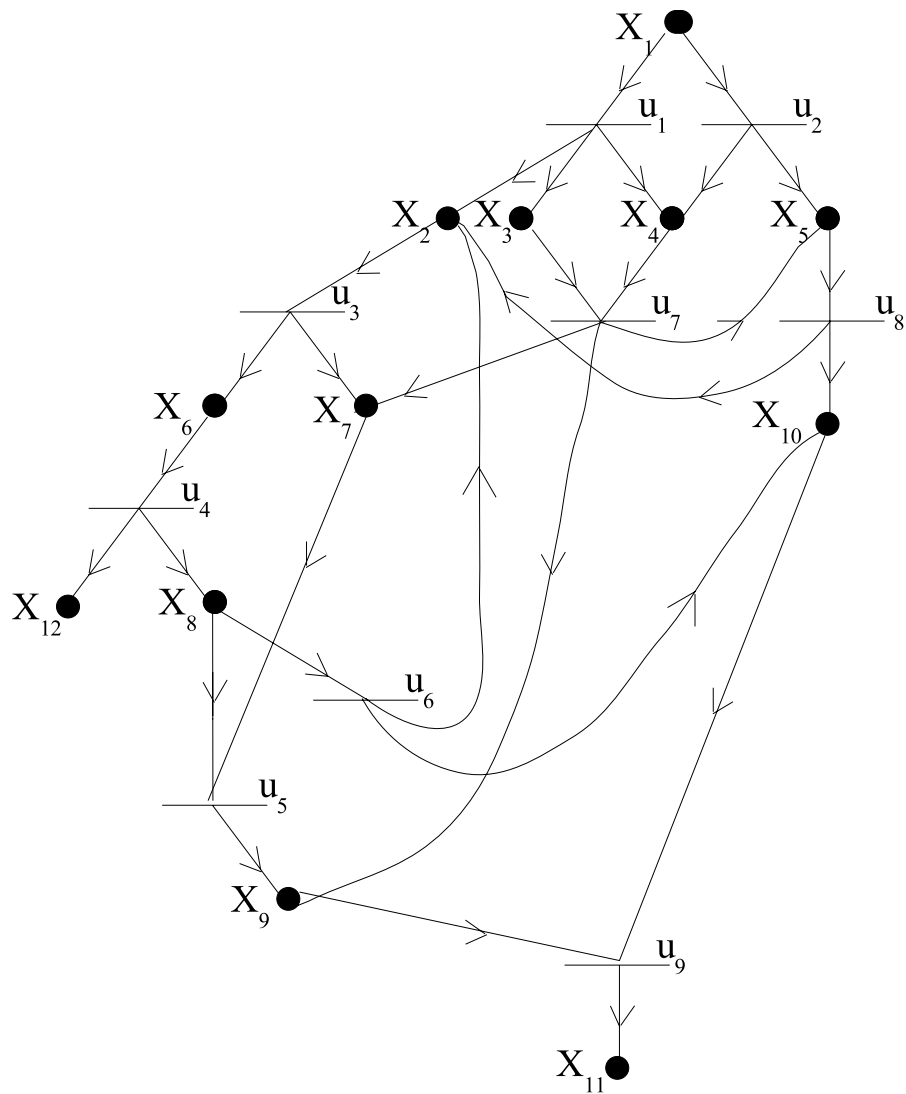
7.3.3. Megjegyzés. Felmerül a kérdés, hogy mennyire hatékonyan valósítható meg a \bowtie reláció meghatározása. Legyen (M, O) a PNS probléma folyamat gráfja, ahol $O = \{u_1, \dots, u_n\}$ és $M = \text{mat}(O)$. Akkor (M, O) -hoz rendeljünk hozzá egy másik (Θ, Λ) gráfot a következőképpen. Legyen $\Theta = O$ és bármely $1 \leq i \neq j \leq n$ -re $(u_i, u_j) \in \Lambda \iff u_i \ll u_j$. Ezt lineáris időben meg tudjuk tenni. Ugyanakkor az is nyilvánvaló, hogy a \bowtie reláció meghatározása (M, O) -ban ekvivalens az erősen összefüggő komponensek meghatározásával (Θ, Λ) -ban. Mivel az utóbbi polinomiális $(O(n^2))$ időben elvégezhető ([9]), a megoldás \bowtie relációra való konvertálása pedig triviális, ezért elmondhatjuk, hogy a \bowtie reláció hatékonyan meghatározható. A PNS megoldása során természetesen nem szükséges a (Θ, Λ) gráf tényleges megépítése, az erősen összefüggő komponenseket meghatározó mélységi bejárás alapuló algoritmus könnyen módosítható úgy, hogy közvetlenül az (M, O) gráfban határozza meg a \bowtie relációt.

Most definiálhatunk egy hasonló, \lll relációt az O' / \bowtie -on.

7.3.3. Definíció. Bármely $C, C' \in O' / \bowtie$ -ra $C \lll C'$ ha $C = C'$ vagy léteznek $u \in C$ és $v \in C'$ műveleti egységek úgy, hogy $u \ll^* v$.

7.3.1. Példa. Legyen $\mathbf{M} = (O, P, R)$ az 7.1. ábrán látható strukturális modell:

- $M = \{X_1, X_2, \dots, X_{12}\}$,
- $R = \{X_1\}$,
- $P = \{X_{11}\}$, és
- $O = \{u_1, u_2, \dots, u_9\}$, ahol



7.1. ábra.

- $u_1 = (\{X_1\}, \{X_2, X_3, X_4\})$,
- $u_2 = (\{X_1\}, \{X_4, X_5\})$,
- $u_3 = (\{X_2\}, \{X_6, X_7\})$,
- $u_4 = (\{X_6\}, \{X_8, X_{12}\})$,
- $u_5 = (\{X_7, X_8\}, \{X_9\})$,
- $u_6 = (\{X_8\}, \{X_2, X_{10}\})$,

- $u_7 = (\{X_3, X_4\}, \{X_5, X_7, X_9\})$,
- $u_8 = (\{X_5\}, \{X_2, X_{10}\})$,
- $u_9 = (\{X_9, X_{10}\}, \{X_{11}\})$.

Akkor $u_2 \ll u_7 \ll u_8 \ll u_3 \ll u_4 \ll u_6 \ll u_3 \ll u_4 \ll u_5 \ll u_9$, és a \bowtie szerinti ekvivalencia osztályok halmaza $\mathcal{C} = \{C_1, C_2, \dots, C_7\}$, ahol

- $C_1 = \{u_1\}$,
- $C_2 = \{u_2\}$,
- $C_3 = \{u_3, u_4, u_6\}$,
- $C_4 = \{u_5\}$,
- $C_5 = \{u_7\}$,
- $C_6 = \{u_8\}$,
- $C_7 = \{u_9\}$.

és $C_1 \lll C_2 \lll C_5 \lll C_6 \lll C_3 \lll C_4 \lll C_7$.

Könnyen belátható, hogy a \lll reláció reflexív, tranzitív és antiszimmetrikus, azaz részben rendezés \mathcal{C} -n. [33] alapján tudjuk, hogy minden részben rendezés kiegészíthető lineáris rendezésre. Így, az általánosság megszorítása nélkül feltételezhetjük, hogy $\mathcal{C} = \{C_1, C_2, \dots, C_h\}$ valamely $1 \leq h \leq |O'|$ -ra úgy, hogy bármely $i, j \in \{1, \dots, h\}$ -re $C_i \lll C_j$ -ből következik $i \leq j$. Definiáljuk az $\ell : (O')^* \rightarrow \{1, \dots, h\}$ függvényt a következőképpen:

$$\ell(p) = \begin{cases} 0 & , \text{ ha } p = \lambda, \\ m & , \text{ ha } p \in O', p \in C_m \\ \max\{\ell(u_t) : 1 \leq t \leq l\} & , \text{ ha } p = u_1 \dots u_l \in (O')^+. \end{cases}$$

7.3.4. Definíció. *Kiterjesztett állapot* egy $(Rp, p, w(p))$ hármast értünk, ahol p egy olyan szó, mely az automatát az R -ből egy F -beli állapotba viszi át. Azt mondjuk, hogy $(Rp, p, w(p))$ egy **optimális kiterjesztett állapot**, ha egy kiterjesztett állapot és $w(p) \leq w(p')$ bármely $(Rp', p', w(p'))$ kiterjesztett állapotra.

7.3.4. Megjegyzés. Ha $(Rp, p, w(p))$ egy optimális kiterjesztett állapot, akkor p egy legrövidebb út R -ből egy végállapotba.

7.3.1. Lemma. ([22]) *Legyen $s = (Rp, p, w(p))$ egy optimális kiterjesztett állapot $p = u_1 \dots u_l \in (O')^*$ -al. Bármely k -ra, melyre $1 \leq k < l$ és $\ell(u_1) \leq \dots \leq \ell(u_k)$, létezik egy $\tau : \{1, \dots, k+1\} \rightarrow \{1, \dots, k+1\}$ permutáció úgy, hogy $\ell(u_{\tau(1)}) \leq \dots \leq \ell(u_{\tau(k)}) \leq \ell(u_{\tau(k+1)})$, és ha $p' = u_{\tau(1)} \dots u_{\tau(k+1)} u_{k+2} \dots u_l$, akkor $s' = (Rp', p', w(p'))$ úgyszintén optimális kiterjesztett állapot.*

Bizonyítás Ha $\ell(u_{k+1}) \geq \ell(u_k)$, akkor az identikus leképezés egy megfelelő permutáció. Tegyük fel, hogy $\ell(u_{k+1}) < \ell(u_k)$. Akkor létezik egy r , $1 \leq r \leq k$ egész szám, melyre $\ell(u_1) \leq \dots \leq \ell(u_{r-1}) \leq \ell(u_{k+1}) < \ell(u_r) \leq \dots \leq \ell(u_k)$. A \mathcal{C} rendezése miatt ez azt jelenti, hogy bármely $r \leq t \leq k$ -ra $\text{mat}^{\text{out}}(u_t) \cap \text{mat}^{\text{in}}(u_{k+1}) = \emptyset$. Ebből következik, hogy $\text{mat}^{\text{in}}(u_{k+1}) \subseteq Ru_1 \dots u_{r-1}$, továbbá $Ru_1 \dots u_l = Ru_1 \dots u_{r-1} u_{k+1} u_r \dots u_k u_{k+2} \dots u_l$. Ez azt adja, hogy a

$$\tau(t) = \begin{cases} t & , \text{ ha } 1 \leq t < r, \\ k+1 & , \text{ ha } t = r, \\ t-1 & , \text{ ha } r+1 \leq t \leq k+1. \end{cases}$$

permutációval teljesül az állítás. \square

7.3.2. Lemma. ([22]) *Legyen $s = (Rp, p, w(p))$ egy optimális kiterjesztett állapot $p = u_1 \dots u_l \in (O')^*$ -gal. Minden olyan k -ra, melyre $1 \leq k \leq l$, létezik olyan $\tau : \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ permutáció, melyre $\ell(u_{\tau(1)}) \leq \dots \leq \ell(u_{\tau(k)})$ és ha $p' = u_{\tau(1)} \dots u_{\tau(k)} u_{k+1} \dots u_l$, akkor $s' = (Rp', p', w(p'))$ egy optimális kiterjesztett állapot.*

Bizonyítás A 7.3.1. lemma alapján az állítás k szerinti indukcióval könnyen bizonyítható. \square

7.3.3. Lemma. ([22]) *Bármely $s = (Rp, p, w(p))$ optimális kiterjesztett állapotra, melyre $p = u_1 \dots u_l$, létezik egy $\tau : \{1, \dots, l\} \rightarrow \{1, \dots, l\}$ permutáció úgy, hogy $\ell(u_{\tau(1)}) \leq \dots \leq \ell(u_{\tau(l)})$ és $p' = u_{\tau(1)} \dots u_{\tau(l)}$ -re $s' = (Rp', p', w(p'))$ egy optimális kiterjesztett állapot.*

Bizonyítás Az állítás a 7.3.2. lemmából $k = l$ -re azonnal következik. \square

7.3.5. Megjegyzés. A 7.2.1. és 7.3.3. lemmák mutatják egy olyan $s^* = (Rp, p, w(p))$ optimális kiterjesztett állapot létezését, melyben $p = u_1 \dots u_l$, az u_t , $t = 1, \dots, l$ jelek páronként különbözőek, továbbá $\ell(u_1) \leq \dots \leq \ell(u_l)$ is teljesül.

Az 7.3.5. megjegyzés alapján elegendő az előbbi lemmák által meghatározott típusú kiterjesztett állapotokat generálni és vizsgálni, ami azt jelenti, hogy kevesebb állapotot tartalmazó átmeneti gráfban keressük a legrövidebb utat, ez pedig hatékonyabbá teszi az eljárást. A következő algoritmus ezt az ötletet is felhasználja.

PAT algoritmus a PNS-8 optimális megoldásainak meghatározására ([22])

Inicializálás.

- $i := 0$, $L_0 := \{(R, \lambda, 0)\}$.

Iteráció

1. $M_i = \{(Rp, p, w(p)) \in L_i : w(p) \leq w(q), \forall (Rq, q, w(q)) \in L_i\}$.
 $S_i = \{(Rp, p, w(p)) \in M_i : P \subseteq Rp\}$.
 Ha $S_i \neq \emptyset$, akkor VÉGE; az S_i elemei optimális kiterjesztett állapotok.
2. Válasszunk egy tetszőleges $(Rt, t, w(t)) \in M_i$ elemet, és legyen $t = u_1 \dots u_n$.
3. Legyen $i = i + 1$, $L_i = L_{i-1}$.
4. $L_i = L_i \setminus \{(Rt, t, w(t))\}$.
5. Legyen $V_i = \{v \in O' \setminus \{u_1, \dots, u_n\} : mat^{in}(v) \subseteq Rt \text{ és } \ell(v) \geq \ell(t)\}$.
 Ha $V_i = \emptyset$, akkor térjünk az 1. lépésre.
 Egyébként legyen $V_i = \{v_1, \dots, v_m\}$.

6. Minden $j = 1, \dots, m$ értékre rendre hajtsuk végre az alábbi lépéseket:

- $A(v_j) := \{(Rq, q, w(q)) \in L_i : Rq \supseteq Rtv_j \text{ és } w(q) \leq w(tv_j) \text{ és } (w(q) < w(tv_j) \text{ vagy } \ell(q) \leq \ell(tv_j))\}$,
- $D(v_j) := \{(Rq, q, w(q)) \in L_i : Rq \subseteq Rtv_j \text{ és } w(q) \geq w(tv_j) \text{ és } (w(q) > w(tv_j) \text{ vagy } \ell(q) \geq \ell(tv_j))\}$,
- ha $A(v_j) = \emptyset$, akkor legyen

$$L_i := (L_i \setminus D(v_j)) \cup \{(Rtv_j, tv_j, w(tv_j))\}.$$

7. Térjünk az 1. lépésre.

7.3.6. Megjegyzés.

- (a) Bármely $i \geq 0$ egészre, ha létezik L_i és $(Rp, p, w(p)) \in L_i$, ahol $p = u_1 \dots u_l$, akkor u_1, \dots, u_l páronként különbözőek és $\ell(u_1) \leq \dots \leq \ell(u_l)$.
- (b) Bármely $i \geq 1$ egészre, ha létezik L_i és $(Rp, p, w(p)) \in L_i$, ahol $p = u_1 \dots u_l$, akkor minden olyan j -re, melyre $0 \leq j < l$ és $p' = u_1 \dots u_j$, igaz, hogy $(Rp', p', w(p')) \notin L_i$.

Legyen $\mathcal{S}(L_i)$ az L_i -ből kapható alábbi hármasok halmaza:

$$\mathcal{S}(L_i) = \{(Rpq, pq, w(pq)) : (Rp, p, w(p)) \in L_i \text{ és } q \in (O')^+ \text{ és}$$

a pq -ban előforduló jelek páronként különböznek egymástól\}.

7.3.7. Megjegyzés. Vegyük észre, hogy a 7.3.6. megjegyzés (b) pontja alapján bármely $i \geq 0$ -ra, $L_i \cap \mathcal{S}(L_i) = \emptyset$.

7.3.4. Lemma. ([22]) *Bármely $i \geq 0$ egészre, melyre létezik L_i , van olyan $s_i^* = (Rp_i, p_i, w(p_i))$ optimális kiterjesztett állapot $p_i = u_1^{(i)} \dots u_{m_i}^{(i)}$ -vel, melyre*

- (i) az $u_1^{(i)}, \dots, u_{m_i}^{(i)}$ jelek páronként különbözőek,

- (ii) $s_i^* \in L_i \cup \mathcal{S}(L_i)$,
- (iii) létezik olyan $0 \leq k_i \leq m_i$ és $p'_i = u_1^{(i)} \dots u_{k_i}^{(i)}$, melyre
- $$(Rp'_i, p'_i, w(p'_i)) \in L_i,$$
- (iv) $\ell(u_1^{(i)}) \leq \dots \leq \ell(u_{m_i}^{(i)})$.

Bizonyítás Az állítást i szerinti indukcióval igazoljuk. Ha $i = 0$, akkor a 7.3.5. megjegyzésből következik annak teljesülése $k_0 = 0$, $(R\lambda, \lambda, 0) \in L_0$, és $s_0^* = s^* \in \mathcal{S}(L_0)$ -al. Most legyen $i \geq 0$ tetszőleges egész érték, feltételezzük az állítás érvényességét i -re, és igazoljuk $(i + 1)$ -re. Az indukciós feltevés alapján létezik olyan $s_i^* = (Rp_i, p_i, w(p_i))$ optimális kiterjesztett állapot $p_i = u_1^{(i)} \dots u_{m_i}^{(i)}$ -vel, mely teljesíti az (i), (ii), (iii), és (iv) feltételeket. Ha a 1. lépésben megállunk, akkor nem létezik L_{i+1} és az állítás teljesül. Most tegyük fel, hogy az algoritmus végrehajtása nem ér véget a 1. lépésben. Legyen $(Rt, t, w(t))$ a 2. lépésben választott hármast. Nyilván $(Rt, t, w(t)) \neq s_i^*$, mivel $(Rt, t, w(t)) = s_i^*$ -vel az eljárás véget ért volna. Ha $V_{i+1} = \emptyset$, akkor $(Rt, t, w(t))$ különbözik $(Rp'_i, p'_i, w(p'_i))$ -től, ugyanis ellenkező esetben $(Rt, t, w(t)) = (Rp'_i, p'_i, w(p'_i))$ -ből és $(Rt, t, w(t)) \neq s_i^*$ -ből azt kapnánk, hogy létezik $u_{k_i+1}^{(i)} \in V_{i+1}$, ami ellentmondás. Következésképpen, ha $V_{i+1} = \emptyset$, akkor a 4. lépésben L_{i+1} -ből töröljük $(Rt, t, w(t))$ -t, de $(Rp'_i, p'_i, w(p'_i)) \in L_{i+1}$ marad, ezért $s_{i+1}^* = s_i^*$ egy olyan optimális kiterjesztett állapot, mely teljesíti az elvárt tulajdonságokat. Most tegyük fel, hogy $V_{i+1} \neq \emptyset$.

Az $(i + 1)$ -dik lépésben legyenek L'_{i+1} és L''_{i+1} közvetlenül a 3. illetve 6. lépések utáni aktuális L_{i+1} halmazok. Jelöljük továbbá $L_{i+1}^{(j)}$ -vel az aktuális L_{i+1} halmazt közvetlenül a 6. lépés j -re való végrehajtása előtt.

Az egyszerűség kedvéért a továbbiakban jelölje $\bar{s} = (Rp, p, w(p))$ mindig a rendelkezésre álló aktuális optimális kiterjesztett állapotot, melyben $p = u_1 \dots u_m$ és legyen az ennek megfelelő $(Rp', p', w(p')) \in L'_{i+1}$, ahol $p' = u_1 \dots u_k$ valamely $0 \leq k \leq m$ -re. Ha $(Rp', p', w(p')) \in L''_{i+1}$, akkor $s_{i+1}^* = \bar{s}$ egy megfelelő optimális kiterjesztett állapot. Most tegyük fel, hogy $(Rp', p', w(p')) \notin L''_{i+1}$, azaz, hogy $(Rp', p', w(p'))$ -t töröltük L'_{i+1} -ből. Két esetet különböztetünk meg.

1. ESET. $(Rp', p', w(p')) \notin M_i$ vagy $(Rp', p', w(p')) \in M_i$ de a 2. lépésben nem ezt a hármast választottuk. Akkor $(Rp', p', w(p'))$ -t

csak a 6. lépésben törölhetjük, így $(Rp', p', w(p')) \in L_{i+1}^{(1)}$, tehát $j = 1$ esetben az állítás teljesül. Minden olyan esetben, amikor a 6. lépésben valamely j -re egy $(Rp', p', w(p'))$ -t törölünk az aktuális $L_{i+1}^{(j)}$ -ből, teljesülnie kell $(Rp', p', w(p')) \in D(v_j)$ -nek $v_j \in V_{i+1}$ -re. Ez akkor történik, amikor

1.1. ESET $Rp' \subseteq Rtv_j$, $w(p') > w(tv_j)$ vagy

1.2. ESET $Rp' \subseteq Rtv_j$, $w(p') \geq w(tv_j)$, $\ell(p') \geq \ell(tv_j)$,

ahol a 2. lépésben $(Rt, t, w(t))$ -t választottuk. Az 1.1. esetben, mivel $Rp' \subseteq Rtv_j$, ezért $(Rtv_j u_{k+1} \dots u_m, tv_j u_{k+1} \dots u_m, w(tv_j u_{k+1} \dots u_m))$ olyan kiterjesztett állapot lenne, melyre $w(p) > w(tv_j u_{k+1} \dots u_m)$, ami ellentmondana \bar{s} optimalitásának. Tehát az 1.1. eset lehetetlen. Az 1.2. esetben jelölje \hat{q} az $u_{k+1} \dots u_m$ azon részsavát, melyet úgy kapunk, hogy töröljük belőle a tv_j -ben előforduló jeleket. Akkor $(Rtv_j, tv_j, w(tv_j)) \in L_{i+1}^{(j+1)}$ és könnyű belátni, hogy $s' = (Rtv_j \hat{q}, tv_j \hat{q}, w(tv_j \hat{q}))$ az állítás által kért feltételeket teljesítő optimális kiterjesztett állapot.

2. ESET. $(Rp', p', w(p')) \in M_i$ és ezt a hármast választjuk ki a 2. lépésben. Akkor $(Rp', p', w(p')) \notin L_{i+1}^{(1)}$ mivel ez a hármast a 4. lépésben törlésre került, továbbá $u_{k+1} \in V_{i+1}$. Ha $A(u_{k+1}) \neq \emptyset$, akkor

2.1. ESET. $Rp' u_{k+1} \subseteq Rq$, $w(p' u_{k+1}) > w(q)$ vagy

2.2. ESET. $Rp' u_{k+1} \subseteq Rq$, $w(p' u_{k+1}) \geq w(q)$, $\ell(q) \leq \ell(p' u_{k+1})$

teljesül valamely $(Rq, q, w(q)) \in L_{i+1}$ -re. Vegyük észre, hogy \bar{s} optimalitása miatt a 2.1. eset lehetetlen. A 2.2. esetben jelölje \hat{q} az $u_{k+2} \dots u_m$ azon részsavát, melyet úgy kapunk, hogy töröljük belőle a q -ban előforduló jeleket, és legyen $q' = q\hat{q}$. Akkor könnyű belátni, hogy $(Rq', q', w(q'))$ egy kívánt tulajdonságokkal rendelkező optimális kiterjesztett állapot. Ha $A(u_{k+1}) = \emptyset$, akkor az $(Rp' u_{k+1}, p' u_{k+1}, w(p' u_{k+1}))$ hármast hozzáadjuk az aktuális L_{i+1} -hez és így $s_{i+1}^* = \bar{s}$ az állítás feltételeit kielégítő optimális kiterjesztett állapot lesz. Ha valamely $v_j \neq u_{k+1}$, $v_j \in V_{i+1}$ -re az aktuális $(Rp', p', w(p'))$ -t töröljük L_{i+1} -ből, akkor ennek $D(v_j)$ -ben kell lennie és akkor az 1. esetben leírtakhoz hasonlóan meg tudunk határozni egy, az állítás feltételeit kielégítő, optimális kiterjesztett állapotot. Ezzel igazoltuk az állítást. \square

7.3.1. Következmény. *Bármely $i \geq 0$ -ra, ha létezik L_i , akkor $L_i \neq \emptyset$.*

7.3.5. Lemma. *Bármely $i \geq 0$ -ra, ha $S_i = \emptyset$ akkor*

$$|L_i \cup \mathcal{S}(L_i)| > |L_{i+1} \cup \mathcal{S}(L_{i+1})|.$$

Bizonyítás Az állítást i szerinti indukcióval fogjuk igazolni. Ha $i = 0$, akkor $L_0 \in (L_0 \cup \mathcal{S}(L_0)) \setminus (L_1 \cup \mathcal{S}(L_1))$ és az állítás nyilvánvalóan teljesül. Most tegyük fel, hogy az állítás teljesül $i \geq 0$ -ra és igazoljuk $(i + 1)$ -re. Legyen $S_i = \emptyset$ és $(Rt, t, w(t))$ a 2. lépésben kiválasztott hármast. Akkor az 7.3.6. megjegyzés alapján $(Rt, t, w(t)) \in L_i \setminus (L_{i+1} \cup \mathcal{S}(L_{i+1}))$. Másfelől $L_{i+1} \cup \mathcal{S}(L_{i+1}) \subseteq L_i \cup \mathcal{S}(L_i)$ nyilvánvalóan igaz, ami azt jelenti, hogy az állítás bármely i -re teljesül. \square

7.3.1. Tétel. ([22]) *A PAT algoritmus véges számú lépés után véget ér és egy optimális kiterjesztett állapotot határoz meg.*

Bizonyítás Először igazoljuk, hogy az algoritmus véges számú lépés után véget ér. A 7.3.1. következmény alapján az eljárás az $S_i \neq \emptyset$ esetben fog véget érni. Mivel $L_0 \cup \mathcal{S}(L_0)$ egy véges halmaz, ezért a 7.3.5. lemma és a 7.3.1. következmény alapján legtöbb $|L_0 \cup \mathcal{S}(L_0)|$ lépés után $L_i \cup \mathcal{S}(L_i) = \emptyset$ kellene legyen, de mivel ha létezik L_i , akkor $L_i \neq \emptyset$, ezért L_i akkor már nem létezhet, azaz az algoritmus valahol meg kellett álljon.

Igazolnunk kell még, hogy ha $S_i \neq \emptyset$, akkor $s \in S_i$ egy optimális kiterjesztett állapot. Legyen $s = (Rp, p, w(p))$ az S_i tetszőleges eleme. S_i definíciója alapján s egy kiterjesztett állapot. Az 7.3.4. lemmából következik olyan $s_i^* = (Rq, q, w(q))$ optimális kiterjesztett állapot létezése, melyre $q = u_1 \dots u_l$ úgy, hogy $u_1 \dots u_k \in L_i$ valamely $0 \leq k \leq l$ -re. Akkor $w(u_1 \dots u_k) \leq w(q)$. Másfelől M_i definíciója alapján $w(p) \leq w(u_1 \dots u_k)$. Következésképpen $w(p) \leq w(q)$, ami azt jelenti, hogy s egy optimális kiterjesztett állapot. \square

8. fejezet

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani dr. Imreh Balázs egyetemi docens PhD témavezetőmnek, aki befogadott a PNS kutató csoportba, folyamatosan irányította munkámat, továbbá számos szakmai és emberi segítségnyújtással, tanáccsal és biztatással meghatározóan hozzájárult nem csak szakmai fejlődésemhez.

Köszönettel tartozom továbbá dr. Imreh Csanád, dr. Kovács Zoltán és Blázsik Zoltán kollégáimnak, akik hasznos szakmai tanácsokkal segítettek munkámat, illetve dr. Ferucio Laurențiu Țiplea volt diplomamunka vezetőmnek, dr. Csendes Tibornak, illetve Székely Ildikónak, akiknek köszönhetően tudomást és további hasznos információkat szereztem a JATE PhD programjában külföldiként való részvételi lehetőségeimről.

9. fejezet

A disszertáció összefoglalása

A kémiában, biológiában, ipari alkalmazásokban gyakran felmerül olyan probléma, amelyben bizonyos „nyersanyagok” és „műveleti egységek” rendelkezésünkre állnak és előírt anyagokat kívánunk előállítani az adott műveleti egységek összekapcsolásával. A probléma egy lehetséges modellezése a „Process Network Synthesis” (PNS), melyben minden műveleti egység az anyagok egy részhalmazát inputként igényli és anyagok egy másik részhalmazát állítja elő. A gráfelméleti megközelítésben egy anyagtól irányított él vezet azokhoz a műveleti egységekhez, amelyek azt input anyagként felhasználják, illetve egy műveleti egységet irányított éllel kötünk össze azokkal az anyagokkal, amelyeket output anyagként termel. Így egy kétrészes (anyagok; műveleti egységek) irányított gráfot kapunk, a folyamat gráfját. Egy ilyen hálózatban az előírt anyagok legyártása általában többféleképpen, a rendelkezésre álló gépek különböző részhalmazáival is megvalósítható. Statikusan képzelve a termelési folyamatot, a műveleti egységek egy részrendszerének működésével a kívánt anyagokat bizonyos alapvető feltételek teljesülése esetén kaphatjuk meg. Ily módon a lehetséges megoldásoknak rendelkezniük kell bizonyos strukturális tulajdonságokkal ([15]), ami miatt ezeket megoldás struktúráknak is szokás nevezni. Ezek között kitüntetett szerepe van a maximális struktúrának, mely a lehetséges megoldás struktúrák uniója. A maximális struktúra meghatározása azért hasznos, mert ily módon a hálózatból töröljük azokat a műveleti egységeket, melyek úgysem szerepelnének egyetlen lehetséges megoldásban sem, és ily módon csökkentjük a probléma méretét. Mivel a maximális struktúra generálására polinomiális idejű algoritmus ad-

ható ([11]), ezért célszerűnek tűnik a PNS problémák megoldását ezzel kezdeni. A legtöbb esetben azonban minket nem a legtöbb műveleti egységet igénybe vevő, hanem ellenkezőleg, valamilyen szempontból leggazdaságosabb megoldások érdekelnek, tehát nem elégedhetünk meg a maximális struktúra meghatározásával.

A dolgozatban tárgyalt PNS modell csak strukturális szempontból tekinti a problémát, mivel annak leírása anyagmennyiségekre vonatkozó előírásokat nem tartalmaz. A rendelkezésre álló műveleti egységek viszont rendelkeznek bizonyos költséggel és az anyagelőállítás során használt műveleti egységek összköltségét szeretnénk optimalizálni: keressük a műveleti egységek azon legkisebb összköltségű részhalmazát, mely a rendelkezésre álló nyersanyagokból képes előállítani a kívánt végtermékeket. Mivel a minimum meghatározása még ebben a legegyszerűbb strukturális esetben is a halmazlefedési problémával ekvivalens ([2, 17, 27]), így a halmazlefedési probléma NP teljességéből ([32, 1]) következően a strukturális PNS-probléma is sajnos NP-teljes. Nem várható tehát hatékony megoldás rá. Ezért indokolt exponenciális idjű algoritmusok és azok különböző heurisztikákkal kombinált, korlátozás és szétválasztás módszerére alapuló változatainak kidolgozása ([13, 24, 21, 23]).

A Branch-and-Bound jellegű módszerek felépítésében fontos szerepet játszik az úgynevezett döntési leképezés fogalma ([12]), mely lényegében meghatározza adott anyagra az őt gyártó műveleti egységek halmazát. Gyakorlati szempontból nyilván nagyon komoly előny, hogy csak bizonyos úgynevezett "konzisztens" döntési leképezéseket kell figyelembe venni, ami abból az észrevételből származik, hogy egy műveleti egység, ha működik, nem teheti meg, hogy bizonyos kimeneti anyagait gyártja, másokat pedig nem. Ezen belül további szűkítést eredményez az az észrevétel, hogy nem működhet olyan műveleti egység, amelyik valamelyik input anyagát egyetlen működő műveleti egységtől sem nyeri. A konzisztens döntési leképezések és a lehetséges megoldás struktúrák közötti kapcsolatot felhasználva, a konzisztens döntési leképezések megszámlálásával, a szitaformula segítségével felső korlát adható a lehetséges megoldás struktúrák számára ([3]). Mivel a korlát tényleges kiszámítása a probléma struktúrájától függ és általában, tetszőleges folyamat gráf esetén, meglehetősen bonyolult, ezért megvizsgáltunk két speciális PNS problémaosztályt is, melyre ténylegesen kiszámítható képleteket tudtunk adni, miközben szép kombinatorikus azonosságokat is kaptunk ([4, 5]).

A továbbiakban észrevettük, hogy bizonyos műveleti egységek, nevezetesen azok, amelyek egyszerre vannak jelen vagy egyikük sem szerepel a lehetséges megoldásokban, együtt kezelhetők. Ebből kiindulva definiáltuk az összevonás műveletét, mely a maximális struktúrához képest is általában kb. 7%-os további méretcsökkenést eredményez ([20]). Ugyanakkor az összevonás következtében megjelenő új műveleti egységek az eltávolítottaknál több bemeneti és kimeneti anyaghalmazzal rendelkeznek (bár a be- és kimeneti anyagok száma összességében megmarad), ezért felmerül a kérdés, hogy az összevonás ötlete vajon egyáltalán használható-e a feladat hatékonyabb megoldására. A döntési leképezések mélyebbre ható tanulmányozásának következtében kapott néhány további észrevételt is felhasználva, kidolgoztunk egy új, *Előretékintő B & B (ER)* nevezetű, korlátozás és szétválasztás típusú eljárást, mely az összevonás ötletét implicit módon alkalmazva, vizsgálatainkban az eddig ismert legjobb MABBA eljárásnál lényegesen hatékonyabban oldotta meg a feladatot ([21]).

A gyakorlatban előállhatnak olyan esetek, amikor nem csak egy optimális megoldás, hanem több vagy az összes, lehetséges vagy optimális megoldás is érdekel. A [16]-ban kidolgozásra került egy eljárás, mely egy PNS probléma összes lehetséges megoldását felsorolja. Lehetnek azonban olyan esetek is, amikor csak az optimális megoldások érdekelnek, de azokat mind fel szeretnénk sorolni. A feladat megoldható a [16]-ban megadott teljes leszámllálással is, például úgy, hogy először megkeressük az optimumot, majd utána a teljes leszámllálásnál ugyan az összes lehetséges megoldást végigjárjuk, de csak az optimális megoldásokat tartjuk meg, ez a megoldás azonban egyáltalán nem hatékony, hiszen sok felesleges lehetséges, de nem optimális megoldást vizsgál meg. Ezért kidolgoztunk egy ennél hatékonyabb eljárást, mely ugyan még mindig nem csak az optimális megoldásokat találja meg, viszont az általa felsorolt megoldás halmaza, mely tartalmazza az összes optimális megoldást, a bemutatott empirikus vizsgálatokból ítélve, lényegesen kisebb, mint az összes lehetséges megoldások halmaza, hiszen az algoritmus a teljes leszámllálásnál sokkal hatékonyabbnak bizonyult ([23]). Nyitott kérdés maradt azonban, hogy a parciális leszámllálás milyen feltételek mellett képes csak az optimális megoldásokat végigjárni.

A dolgozat befejező részében a PNS egy teljesen újszerű, automataelméleti megközelítését vizsgáltuk meg. A [28] cikk alapján láttuk, hogy a lehetséges megoldások [18] és [15] munkákban meghatározott feltételei nem biztosítják a végrehajthatóságot, ezért a [28]-ben kidolgozásra került egy

úgynevezett színező eljárás, mely meghatározza a végrehajtható folyamatokat, melyeket módosított lehetséges megoldásoknak nevezünk. Ugyancsak a [28]-ban meg lett adva egy eljárás a módosított lehetséges optimális megoldás meghatározására, melynek alapötlete az, hogy a módosított PNS probléma adott példányához hozzárendelhető egy automata, melyre teljesül az, hogy az eredeti feladat megoldása visszavezethető az automata átmeneti grájában egy legrövidebb út megtalálására. A disszertáció ezen részének tulajdonképpeni célja ezen eljárás továbbfejlesztése volt. Definiáltunk egy ekvivalencia relációt a műveleti egységek halmazán, egy részben rendezést az ekvivalencia osztályokon, melyeknek segítségével, néhány további észrevétel felhasználásával, egy nyilvánvalóan hatékonyabb eljárást sikerült kidolgoznunk az optimális módosított lehetséges megoldás meghatározására, melynek az a lényege, hogy a legrövidebb út keresésével egyidőben az átmeneti gráfnak csak egy részét generáljuk ([22]).

Összesítve, a PNS nem az egyetlen, de egy lehetséges és hasznosnak bizonyult modellje a hálózati folyamatoknak, mely lehetővé tette strukturális összefüggések feltárását és a folyamatok bonyolultságához képest valamivel hatékonyabb megoldások megtalálását.

10. fejezet

Summary of the doctoral thesis

10.1. Synopsis

In a manufacturing system, materials of different properties are consumed through various mechanical, physical and chemical transformations to result in desired products. Devices in which these transformations are carried out are called operating units, e.g., a lathe or a chemical reactor. Hence, a manufacturing system can be considered as a network of operating units which is called process network. The importance of process network synthesis (PNS) arises from the fact that such networks are ubiquitous in the chemical and allied industries. A process design problem in general, and flowsheeting in particular mean to construct a manufacturing system. A design problem is defined from a structural point of view by the raw materials, the desired products, and the available operating units, which determine the structure of the problem as a process graph containing the corresponding interconnections among the operating units. Thus, the appropriate process networks can be described by some subgraphs of the process graph belonging to the design problem under consideration. Naturally, the cost minimization of a process network is indeed essential. For this purpose, several papers have appeared for solving PNS problems by global optimization methods (cf. [10] and [19]) and by combinatorial approach based on the feasible graphs of processes (see, e.g., [18], [15], [11]). However its solution is difficult in general. It has been pointed out that the PNS-problems are NP-complete (see [2], [17] and [27]).

In general, there are three basic approaches to attack NP-hard problems. The first approach is to develop exponential time algorithms for solving the problem. In case of PNS problem, some exponential time algorithms based on the Branch and Bound technique were developed and studied in [16], [12], [13], [14], [24], [25], [20], [21], and [23]. This approach is studied also in this work. Another approach is to investigate specially structured instances for which interesting structural properties and bounds (e.g. for number of feasible solutions, see [3], [4], [5]) can be determined, which are called *well-solvable classes* and their instances can be solved efficiently. Some well-solvable classes were presented in [30], [6], [6], [31]. The third approach is to establish fast (polynomial time) algorithms which do not guarantee an optimal solution in general, but always result in a feasible solution which is close to the optimal solution in some sense. Such algorithms, called *heuristic algorithms* or *heuristics*, are important for several reasons. The feasible solutions found by such algorithms can be used in exponential time algorithms, furthermore, there is often not enough time to find an optimal solution or the size of the problem is too large to use an exponential algorithm. In these cases, heuristic algorithms can be useful again. The first heuristic algorithm for this problem was presented in [8].

10.2. The structural model of PNS

The foundations of PNS and the background of the combinatorial model studied here can be found in [15], [18]. In the combinatorial approach, the structure of a process can be described by the process graph (see [15]) defined as follows.

Let M be a finite nonempty set, the set of the materials. Furthermore, let $\emptyset \neq O \subseteq \varphi'(M) \times \varphi'(M)$ with $M \cap O = \emptyset$, where $\varphi'(M)$ denotes the set of all nonempty subsets of M . The elements of O are called *operating units*, and for an operating unit, $u = (\alpha, \beta) \in O$, α and β are called the *input-set* and *output-set* of the operating unit, respectively. The pair (M, O) is defined to be a *process graph* or *P-graph* in short. The set of vertices of this directed graph is $M \cup O$, and the set of arcs is $A = A_1 \cup A_2$, where $A_1 = \{(X, Y) : Y = (\alpha, \beta) \in O \text{ and } X \in \alpha\}$ and $A_2 = \{(Y, X) : Y = (\alpha, \beta) \in O \text{ and } X \in \beta\}$.

Let the process graphs (m, o) and (M, O) be given. (m, o) is defined to be a *subgraph* of (M, O) , if $m \subseteq M$, $o \subseteq O$ and $o \subseteq \varphi'(m) \times \varphi'(m)$.

By a *structural model* of PNS, we mean the triplet, $\mathbf{M} = (P, R, O)$, where $P \subseteq M$ and $O \subseteq \varphi'(M) \times \varphi'(M)$ are finite nonempty sets representing the set of desired products and that of available operating units, respectively, $R \subseteq M$ is a finite set representing the set of raw materials and $P \cap R = \emptyset$.

Then, the process graph (M, O) , where $M = \cup\{\alpha \cup \beta : (\alpha, \beta) \in O\}$, presents the interconnections among the operating units of O . Furthermore, every feasible process network, producing the given set P of products from the given set R of raw materials using operating units from O , corresponds to a subgraph of (M, O) . Investigating the corresponding subgraphs of (M, O) , therefore, we can determine the feasible process networks. If we do not consider further constraints such as material balance, then the subgraphs of (M, O) which can be assigned to the feasible process networks have common combinatorial properties. They are studied in [15] and their description is given by the following definition.

A subgraph (m, o) of (M, O) is called a *solution-structure* of (P, R, O) if the following conditions are satisfied:

$$(A1) \ P \subseteq m,$$

$$(A2) \ \forall X \in m, X \in R \Leftrightarrow \text{no } (Y, X) \text{ arc in the process graph } (m, o),$$

$$(A3) \ \forall Y_0 \in o, \exists \text{ path } [Y_0, Y_n] \text{ in } (m, o) \text{ with } Y_n \in P,$$

$$(A4) \ \forall X \in m, \exists(\alpha, \beta) \in o \text{ such that } X \in \alpha \cup \beta.$$

The set of the solution-structures of $\mathbf{M} = (P, R, O)$ will be denoted by $S(P, R, O)$ or $S(\mathbf{M})$.

Let us consider PNS problems in which each operating unit has a weight. We are to find a feasible process network with the minimal weight where by weight of a process network we mean the sum of the weights of the operating units belonging to the process network under consideration. Each feasible process network in such a class of PNS problems is determined uniquely from the corresponding solution-structure and vice versa. Thus,

the problem can be formalized as follows. Let a structural model of PNS problem $\mathbf{M} = (P, R, O)$ be given. Moreover, let w be a positive real-valued function defined on O , the *weight function*. The basic model is then

$$(PNS-2) \quad \min \left\{ \sum_{u \in o} w(u) : (m, o) \in S(P, R, O) \right\}.$$

In this work by PNS problem we always mean PNS problem with weight, and the solution-structures are also called *feasible solutions*. It is known (see [2, 17, 27]) that this PNS problem is NP-complete.

Now, let $o \subseteq O$ and $m \subseteq M$ be arbitrary sets. Let us define the following functions on the sets o and m :

$$mat^{in}(o) = \bigcup_{(\alpha, \beta) \in o} \alpha, \quad mat^{out}(o) = \bigcup_{(\alpha, \beta) \in o} \beta,$$

$$mat(o) = mat^{in}(o) \cup mat^{out}(o),$$

and

$$\Delta(m) = \{u : u \in O \ \& \ m \cap mat^{out}(u) \neq \emptyset\}.$$

Let m be a subset of $M \setminus R$; furthermore, let $\delta(X)$ be a subset of $\Delta(X)$ for each $X \in m$. Mapping δ from set m into the set of subsets of O , $\delta[m] = \{(X, \delta(X)) : X \in m\}$, is called a *decision-mapping belonging to \mathbf{M}* ; $\delta[m]$ is said to be *consistent* when $\delta(X) \cap \Delta(Y) \subseteq \delta(Y)$ is valid for all $X, Y \in m$, and the set of all consistent decision-mappings of \mathbf{M} is denoted by $\Omega_{\mathbf{M}}$. A decision-mapping can be visualised as a sequence of decisions, each of which is concerned with a single material involved in the process being synthesized; it identifies the set of operating units to be considered for producing directly the material of interest. If $\delta_1[m_1]$ and $\delta_2[m_2]$ are arbitrary consistent decision-mappings, then, $\delta_2[m_2]$ is called an *extension* of $\delta_1[m_1]$ if $m_1 \subseteq m_2$ and $\delta_1(X) = \delta_2(X)$ for all $X \in m_1$. Relation extension is a partial ordering on $\Omega_{\mathbf{M}}$. Let us denote the set of all maximal elements of this partially ordered set by $\Omega_{\mathbf{M}}^{\max}$ and its elements are called *maximal consistent decision-mappings*.

We use a coloring of process graphs. For this purpose, let (\bar{M}, \bar{O}) be a process graph and R a set of materials. It is said that (\bar{M}, \bar{O}) is *colorable* by R if every material vertex of (\bar{M}, \bar{O}) can be colored by the following procedure.

Coloring Procedure ([28])

Step 1. Color every material in $\bar{M} \cap R$.

Step 2. If there is an operating unit whose all input materials have already colored, then color its every output material. Terminate otherwise.

A *modified PNS problem* is a PNS problem satisfying (A1) - (A4) and

$$(A5) \quad (\bar{M}, \bar{O}) \text{ is colorable by } R.$$

The weighted modified PNS problem is then:

$$(PNS-8) \quad \min \left\{ \sum_{u \in \bar{O}} w(u) : (\bar{M}, \bar{O}) \in \bar{S}(\mathbf{M}) \right\}.$$

where $\bar{S}(\mathbf{M})$ is the set of feasible solutions of the modified PNS problem.

10.3. Results of the thesis

In the Chapter 2 we define the mathematical model of the PNS problem, furthermore, we recall the most fundamental definitions and results for this problem. In Chapter 3, based on the works [3, 4, 5], the number of the consistent decision-mappings is counted, and an upper bound is presented for the number of the feasible solutions of a PNS problem. Since the expression of this bound depends on the structure of the process, the determination of its value is much easier if we restrict ourselves to special classes of PNS problems. This is why we study also two special classes and determine the bounds for them. In the Chapter 4, based on the work [20], a new method called *merging reduction* is introduced which is based on the merging of operating units. The mergeable operating units are determined by an equivalence relation on the set of the operating units, and all of the operating units included in an equivalence class are merged into one new operating unit. This reduction has the following property: an optimal solution of the original problem can be derived from an optimal solution of the reduced problem

and conversely. Presentation of this reduction is equipped with an empirical analysis on randomly generated problems which shows the measure of the size decrease.

Since the combinatorial version of the Process Network Synthesis (PNS) problem is NP-complete, it is important to establish effective B&B procedures for its solution. Until now three such algorithms have been established: the Accelerated B&B Algorithm (ABBA for short) [14], a modified version of the ABBA (MABBA in short) [24], and the modified ABBA equipped with a stronger bounding function [24]. The empirical analysis presented in [25] shows that the best procedure among them is MABBA. In Chapter 5, based on the work [21], a new B&B procedure is presented which is based on the merging reduction of PNS problems and on a new branching rule. Our new procedure called *Look Ahead B&B Algorithm* (LABBA in short) differs in two aspects from the previous ones. It uses the new reduction idea obtained in [20] which is based on the observation that the operating units of a PNS problem can be classified such that for each feasible solution, the equivalent operating units simultaneously either appear or do not appear in the feasible solution considered. Using this result, the bounding functions of the earlier procedures can be improved and the size of the B&B tree can be decreased by fixing the equivalent operating units simultaneously. Another aspect is that the LABBA studies the effects of the different possible branchings while in the ABBA and MABBA, the branching rule is based on such a material selection which is independent of its effect. By examining the effects of the different material selections, the size of the B&B tree can be also decreased. Finally, the power of this procedure is justified by an empirical analysis.

Sometimes, it is not sufficient to know only one optimal solution, one may need all of the feasible solutions or all of the optimal solutions. For obtaining all of the feasible solutions, a complete enumeration procedure and possible applications are also described in [16]. In Chapter 6, based on the work [23], a new and relatively fast procedure is presented. It does not generate all of the feasible solutions, but it provides such a subset of feasible solutions which contains all of the optimal solutions of problem considered. The power of this enumeration procedure is illustrated by an empirical analysis.

In [28], a procedure is given for solving modified Process Network Synthesis problems. The idea of this procedure is that for every instance of the modified problem, an automaton can be defined such that an optimal solution can be found by performing a shortest path method in the weighted transition graph of this automaton. In the Chapter 7, we recall this algorithm and improve its efficiency by defining an equivalence relation on operating units, a partial ordering on the equivalence classes, and thus, generating only a part of the transition graph and computing a required shortest path simultaneously. The results of this chapter are based on the work [22].

Tárgymutató

$\langle \dots \rangle$, 34	$\delta_i[m \cup \{Y\}]$, 65
$\mathcal{A}1, \mathcal{A}2, \mathcal{A}3, \mathcal{A}4$, 10	δ , 28
$\mathcal{A}5$, 84	$\delta[m]$, 26
$C(u)$, 53	ℓ , 92
F_I , 45	λ , 86
\mathcal{G}_A , 86	\equiv , 53
(\mathcal{G}_B, w) , 88	\hat{m} , 34
$L(\mathcal{A})$, 87	\leq , 27
\mathcal{M} , 13	\ll , 89
$\mathbf{M}(u_j)$, 57	\lll , 90
\mathbf{M}/\equiv , 53	\models , 34
N_I , 45	$\mu(\mathbf{M})$, 12
$O(X)$, 41	ω , 31
$O^*(X)$, 44	\prec , 66
$O_{\delta[m]}$, 65	\preceq , 66
$S'(\mathbf{M})$, 30	\propto , 86
$S(\mathbf{M})$, 11	ρ , 28
$S^*(\mathbf{M})$, 67	\sim , 13
S_I , 45	\ll^* , 89
$\mathcal{S}(L_i)$, 95	\triangleleft , 14
$\overline{S}(\mathbf{M})$, 85	φ , 8
X^* , 86	φ' , 8
X^+ , 86	ϖ , 32
$[X_1, X_n]$, 9	$\vartheta(\delta[m])$, 70
Δ , 26	\vdash , 34
\bowtie , 90	g , 32
$\Omega_{\mathbf{M}}$, 27	$\text{icl}(\delta[m])$, 66
$\Omega_{\mathbf{M}}^{\max}$, 27	$\text{mat}(o)$, 13
Ψ , 53	mat^{in} , 13
$\Sigma_{\mathbf{M}}$, 66	mat^{out} , 13
\bigvee , 34	$\text{op}(\delta)$, 28

- w , 19
 \bar{w} , 56
 x^A , 86
- ABBA, 61
 anyag, 8
 anyagkiválasztási szabály, 64
 automata, 86
- állapot, 86
 átmeneti gráf, 86
- B&B, 64
 bemenő anyag kiválasztási szabály, 64
 bemeneti jelek, 86
- céltermék, 9
- döntési leképezés, 26
 degenerált maximális struktúra, 12
- egymást kölcsönösen elérő
 (műveleti egységek), 89
 ekvivalencia
 (strukturális modellek), 13
 előállítandó anyagok, 9
 elfogadott nyelv, 86
 ER, 58
 erősen összefüggő komponensek, 90
- élő csúcspont, 32
- felhasználható nyersanyagok, 9
 felderített csúcspont, 32
 felismerő, 86
 felismerhető, 87
 felismert nyelv, 86
 folyamat gráf, 8
 forrás, 9
- gyártás, 8
- halott csúcspont, 32
- irreguláris kiterjesztés, 65
 irreguláris lehetséges megoldás kiterjesztés, 70
 irreguláris lezárás, 66
- kölcsönösen (egymást) elérő
 (műveleti egységek), 89
 kezdőállapot, 86
 kiterjesztés
 (döntési leképezések), 27
 kiterjesztett állapot, 92
 konzisztens, 26
- LABBA, 72
 lehetséges megoldás, 30
 lehetséges megoldás kiterjesztés, 31
 lehetséges megoldás struktúra, 10
 lezárás (döntési leképezések), 28
 lezárt csúcspont, 32
- módosított lehetséges megoldás
 struktúra, 85
 módosított PNS probléma, 83, 85
 műveleti egység, 8
 MABBA, 61
 maximális struktúra, 12
 maximális struktúra generáló algoritmus, 16
- MSG, 16
- NP teljesség, 26
 nyersanyagok, 9
- optimális kiterjesztett állapot, 92
- összevonásos ekvivalencia relációt
 meghatározó eljárás, 58
 összevonás (műveleti egységek), 52
 összevonható műveleti
 egységek, 53

- parciális leszámlálás, 76
 - PNS-1*, 19
 - PNS-2*, 19
 - PNS-3*, 21
 - PNS-4*, 24
 - PNS-5*, 30
 - PNS-6*, 56
 - PNS-7*, 70
 - PNS-8*, 85

- részgráf, 9
- részszó, 86
- redukált strukturális modell, 15
- reguláris döntési leképezés, 26
- reguláris kiterjesztés (döntési leképezések), 27
- reguláris lehetséges megoldás kiterjesztés, 31
- reguláris lezárás (döntési leképezések), 28
- RMABBA, 61

- strukturális modell, 9
- színező eljárás (folyamat gráf), 84
- színezhető folyamat gráf, 84
- szeparátor típusú műveleti egység, 44

- teljes leszámlálás, 34

- út, 9

- végállapotok halmaza, 86
- végtermék, 9

Irodalomjegyzék

- [1] Aho A.V., J. E. Hopcroft, J. D. Ullman, The Design and Analysis of Computer Algorithms, *Addison-Wesley, Reading Mass*, 1974.
- [2] Blázsik, Z., B. Imreh, A note on connection between PNS and set covering problems, *Acta Cybernetica*, **12**, 1996, 309-312.
- [3] Blázsik, Z., Cs. Holló, B. Imreh, On Decision-Mappings Related to Process Network Synthesis Problem, *Acta Cybernetica* **13**, 1998, 319-328.
- [4] Blázsik, Z., Cs. Holló, B. Imreh, Explicit bound for the number of feasible solutions of special PNS-problem classes, *P.U.M.A*, **9**, 1998, 17-27.
- [5] Blázsik, Z., Cs. Holló, B. Imreh, Kiszámolható korlátok speciális PNS-problémaosztályok lehetséges megoldásai számára, *Új utak a magyar operációkutatásban, szerk. Komlósi, S., Szántai T., Dialóg Campus Kiadó, Budapest-Pécs*, 1999, 182-194.
- [6] Blázsik, Z., Cs. Holló, B. Imreh, Cs. Imreh, Z. Kovács, On Bottleneck and k-sum version of the Process Network Synthesis Problem, *Novi Sad Journal Of Mathematics*, **3**, 2000, 11-19.
- [7] Blázsik, Z., Cs. Holló, B. Imreh, Cs. Imreh, Z. Kovács, On a well-solvable class of the PNS problem, *Novi Sad Journal Of Mathematics*, **3**, 2000, 21-30.
- [8] Blázsik, Z., Cs. Holló, Cs. Imreh, Z. Kovács, Heuristics for the Process Network Synthesis Problem, *New Trends in Equilibrium Systems, Mátraháza Optimization Days*, Kluwer Academic Publishers, 2000, 1-16.

- [9] C. Croitoru, *Tehnici de bază în Optimizarea Combinatorie*, Ed. Universității Al. I. Cuza, Iași, 1992
- [10] Floudas, C. A., I. E. Grossmann, Algorithmic Approaches to Process Synthesis: Logic and Global Optimization, AiChE Symposium Series No. 304, **91** (Eds: L. T. Biegler and M. F. Doherty), (1995), 198-221. .
- [11] Friedler, F., K. Tarján, Y. W. Huang, and L. T. Fan, Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for maximal structure generation, *Computer chem. Engng.* **17**, 1993, 924-942.
- [12] Friedler, F., J. B. Varga, and L. T. Fan, Decision-Mappings: A Tool for Consistent and Complete Decisions in Process Synthesis, *Chem. Eng. Sci.*, **50** (11), 1995, 1755-1768.
- [13] Friedler, F., J.B. Varga, E. Fehér, and L.T. Fan, Combinatorially Accelerated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis, International Conference on State of the Art in Global Optimization: Computational Methods and Applications, Princeton, 1995.
- [14] Friedler, F., J. B. Varga, E. Fehér, L. T. Fan, Combinatorially Accelerated Branch-and -Bound Method for Solving the MIP Model of Process Network Synthesis, *Nonconvex Optimization and its Applications*, (eds.: C. A. Floudas and P. M. Pardalos), Kluwer Academic Publishers, Norwell, MA, U.S.A., 1996, 609-626.
- [15] Friedler, F., K. Tarján, Y. W. Huang, L. T. Fan, Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems, *Chem. Eng. Sci.*, **47** (8), 1992, 1973-1988.
- [16] Friedler, F., K. Tarján, Y. W. Huang, L. T. Fan, Combinatorial Algorithms for Process Synthesis, *Computer chem. Engng.*, **16**, 1992, 313-320.
- [17] Friedler, F., J. Fulop, B. Imreh, On the reformulation of some classes of PNS-problems as set covering problems, *Acta Cybernetica*, **13**, 1998, 329-337.
- [18] Friedler, F., L. T. Fan, B. Imreh, Process Network Synthesis: Problem Definition, *Networks*, **28**, 1998, 119-124.

- [19] Grossmann, I. E., V. T. Voudouris, O. Ghattas, Mixed-Integer Linear Programming Reformulations for Some Nonlinear Discrete Design Optimization Problems, In: Recent Advances in Global Optimization (Eds: C. A. Floudas and P. M. Pardalos) Princeton University Press, New Jersey, 1992.
- [20] Holló, Cs., Z. Blázsik, Cs. Imreh, Z. Kovács, On a Merging Reduction of the Process Network Synthesis Problem, *Acta Cybernetica* **14**, 1999, 251-261.
- [21] Holló, Cs., A Look Ahead Branch-and-Bound Procedure for Solving PNS Problems, *P.U.M.A.*, **2**, 2000, 265-279.
- [22] Holló, Cs., A Procedure Based on Automaton Theory Approach for Solving Modified PNS Problems, *P.U.M.A.*, **1-2**, 2002, 159-169.
- [23] Holló, Cs., A Partial Enumeration Algorithm for Solving PNS Problems, *Mathematical and Computer Modelling*, **7-9**, 2003, 855-864.
- [24] Imreh, B., F. Friedler, L. T. Fan, An Algorithm for Improving the Bounding Procedure in Solving Process Network Synthesis by a Branch-and-Bound Method, *Developments in Global Optimization*, ed. I. M. Bonze, T. Csendes, R. Horst, P. M. Pardalos, Kluwer Academic Publisher, Dordrecht, Boston, London, 1996, 301-348.
- [25] Imreh, B., G. Magyar, Empirical Analysis of Some Procedures for Solving Process Network Synthesis Problem, *Journal of Computing and Information Technology*, **6**, 1998, 373-382.
- [26] Imreh, B., *Kombinatorikus optimalizálás*, Novadat, 2000.
- [27] Imreh, B., J. Fülöp, F. Friedler, A note on the Equivalence of the Process Network Synthesis and Set Covering problems, *Acta Cybernetica* **14**, 2000, 497-502.
- [28] Imreh, B., Automaton Theory Approach for Solving PNS Problems, *Acta Cybernetica*, **15**, 2002, 327-338.
- [29] Imreh, B., Z. Kovács, A note on separation-networks and automata, *Publicationes Mathematicae*, submitted for publication.

- [30] Imreh, Cs., Jól megoldható PNS osztályokról, *Új utak a magyar operációkutatásban, szerk. Komlósi, S., Szántai T., Dialóg Campus Kiadó, Budapest-Pécs, 1999, 168-181.*
- [31] Imreh, Cs., A new well-solvable class of PNS problems, *Computing*, **66**, 2001, 289-296.
- [32] Karp R. M., Reducibility among Combinatorial Problems in Complexity of Computer Computations, *R. E. Miller and T. W. Thatcher, eds., Plenum Press, New York, 1972.*
- [33] Szpilrajn, E., Sur l'extension de l'ordre partiel, *Fund. Math.* **16** (1930), 386-389.
- [34] Volpert, A. I., Differential equations on graphs, *Mat. sb.*, **88**, 1972, 578-588.