

Műszaki rendszerek modellezése és optimalizálása

Adonyi Róbert, Bertók Botond, Friedler Ferenc,
Heckl István, Hegyháti Máté, Holczinger Tibor,
Imreh Csanád, Kovács Zoltán, Süle Zoltán



2014

A tananyag a TÁMOP-4.1.2.A/1-11/1-2011-0104 "A felsőfokú informatikai oktatás minőségének fejlesztése, modernizációja" c. projekt keretében a Pannon Egyetem és a Szegedi Tudományegyetem együttműködésében készült.



Tartalomjegyzék

Bevezetés	1
1. Szemléltető folyamathálózat-szintézis feladatok	3
1.1. Vegyipari gyártási feladat: Butanol, etanol és aceton előállítása gabonából, fermentációval	3
1.1.1. Hagyományos technológiák alkalmazása	4
1.1.2. További műveleti egységek: centrifuga, adszorber	8
1.1.3. További műveletek: ultraszűrés, pervaporáció	9
1.1.4. Érzékenységvizsgálat	10
1.2. Üzleti folyamatok	11
1.3. Jármű hozzárendelési példa	13
1.4. Hőcserélő hálózatok	18
1.5. Ütemezés gyártási jellegű feladatok esetén	18
1.6. Fenntarthatósági feladat	19
2. Folyamathálózatok strukturális modellje: A P-gráf módszertan alapjai	21
2.1. A P-gráf módszertan alapjai	23
2.1.1. Alapfogalmak	24
2.1.2. P-gráf	25
2.1.3. Kombinatorikusan lehetséges megoldásstruktúrák	27
2.1.4. Axiómák	27
2.1.5. Maximális struktúra	29
2.1.6. Az MSG algoritmus	30
2.1.7. Az SSG algoritmus	32
2.2. Formális jelölések és algoritmusok	32
2.2.1. Maximális struktúra generálása	35
2.2.2. Az összes megoldásstruktúra generálása	40
3. A strukturális modell kiterjesztései értékelő paraméterekkel: optimalizálás az ABB algorit-mussal	55
3.1. Általános matematikai modell	55
3.2. Gyorsított szétválasztás és korlátozás algoritmus	59
3.3. Időkorlátos PNS	64
3.3.1. Üzleti folyamatok szóhasználata	64
3.3.2. Működési idő mint erőforrás	64
3.3.3. Ismert időzítésű tevékenységek	65
3.3.4. Időben egymásra épülő tevékenységek	67
4. Fenntarthatósági mértékek	73
4.1. A fenntarthatóság fogalma	74
4.2. A fenntarthatóság mérése	74
4.3. Fenntarthatóság ellátási láncokban	75
4.4. Szakirodalmi áttekintés	75
4.5. A módszer lépéseinek rövid bemutatása	76
4.5.1. Adatgyűjtés a feladathoz kapcsolódóan	76
4.5.2. A strukturális modell átalakítása: költség nyersanyagként történő reprezentálása	76
4.5.3. P-gráf modell kibővítése a nyersanyagokra vonatkozó ökológiai lábnyommal	77

4.5.4.	P-gráf modell kibővítése a nyersanyagokra vonatkozó emergy feltételekkel . . .	77
4.5.5.	Modellezési eljárás kidolgozása fosszilis és megújuló energiahordozók felhasználásának arányára	77
4.6.	Sustainable process index	77
4.6.1.	Az SPI számolása	78
4.7.	Emergia	80
4.8.	Ökológiai lábnyom	81
4.9.	Műveleti egységek új reprezentálása	82
4.9.1.	A multi-periódusos működés jellemzése	82
4.9.2.	A több-periódusos műveleti egység modellezése	85
4.9.3.	Műveletek különböző bemenetekkel illetve költségekkel	86
4.9.4.	Több-periódusú működés a fenntartható energiatermelő ellátási láncokban . . .	86
4.10.	Költség modellezése nyersanyagként	86
4.10.1.	Eddigi módszer	86
4.10.2.	Új eljárás	87
4.11.	Az ökológiai lábnyom modellezése	88
4.12.	Szemléltető példa	92
4.13.	A szemléltető példa eredményei	95
5.	Több célfüggvény kezelése	101
5.1.	Többcélú problémák a hálózati folyamatok szintézisében	101
5.2.	Többcélú optimalizálásban használt modellek	102
5.3.	Pareto optimális megoldások generálása PNS modellek esetén	104
5.4.	Aggregált célfüggvények	106
5.5.	Korlátozott feladatok	107
5.6.	Robusztus optimalizálás	108
5.6.1.	A robusztus modell	108
5.6.2.	A fix költséges modell megoldása korlátozás és szétválasztással	109
5.6.3.	Heurisztikus algoritmusok a fix költséges modell megoldására	110
5.6.4.	A lineáris költségfüggvény robusztus modelljei	113
6.	Nevezetes feladatok megoldása PNS-el	117
6.1.	Minimális feszítőfa szintézise	117
6.2.	Legrövidebb út szintézise	117
6.3.	Maximális folyam szintézise	119
7.	Integrált szintézis feladatok	121
7.1.	Folyamat- és hőcserélő hálózat szintézise	121
7.1.1.	Hőcserélő hálózat szintézise	122
7.1.2.	Integrált folyamat- és hőcserélő hálózat szintézise	122
7.1.3.	hP-gráf	122
7.1.4.	Hőmérsékleti adatok, halmazok	123
7.1.5.	Matematikai modell	126
7.1.6.	Megoldási módszer	129
7.2.	PNS és ütemezés integrációja	129
7.2.1.	Ütemezési feladat	129
7.2.2.	S-gráf keretrendszer	130
7.2.3.	Integrált feladat	132
7.2.4.	Paraméterek	133
7.2.5.	Megoldási módszer	133
7.2.6.	Példa	136
8.	PNS szoftverek	141
8.1.	PNS Draw	141
8.2.	PNS Studio	145
9.	PNS algoritmusok implementálása	153
9.1.	Halmazok implementációja	153
9.1.1.	Objektumok sorszámozása	153

9.1.2.	Egy elem – egy bit	154
9.1.3.	Halmazműveletek implementációja	155
9.2.	LP megoldó könyvtárak	156
9.2.1.	Nyílt és zárt forráskódú megoldók	156
9.2.2.	Ritka mátrixok	156

Bevezetés

Általános tapasztalat, hogy a használt eszközeink, berendezéseink, valamint az azokat előállító gyártórendszerek, termelési folyamatok egyre bonyolultabbak. Egy-egy terméket előállító rendszer tervezése - természetes módon - szintézis jellegű tevékenység, hiszen elemi építő elemek sokaságából kell egy új minőséget képviselő egészet, a termelő rendszert létrehozni.

A szimuláció, mint módszer, nem alkalmas e tervezési tevékenység átfogó támogatására, hiszen alkalmazásakor már feltételezzük, hogy létezik a rendszer terve vagy prototípusa, vagy akár a kész megépített rendszer. Paraméterek hangolására, a működésről tapasztalatok szerzésére alkalmas eszköz, de arra, hogy a tervezés során, azaz a szintézis fázisában, figyelembe vegye a termék funkcionalitására tett elvárásokat, a környezetre gyakorolt hatásokat, költséghatékonyságra, megbízhatóságra tett elvárásokat nem alkalmas. Ezért szükséges olyan módszerek kidolgozása, amely megfelel ezeknek az elvárásoknak. Ebben a könyvben egy ilyen módszer, a P-gráf módszertan leírása található. A könyv bemutatja azt az eszköztárat, illetve annak az alkalmazását, amellyel a tervezési folyamat során a fent említett szempontok érvényesíthetőek.

A "rendszer" ember alkotta fogalom, a tudományok egyik alapvető fogalma, amelyet szigorú formális értelemben nem definiálunk. Bizonyos objektumok, jelenségek, folyamatok részekre, alkotó elemekre bonthatóságát jelenti, illetve bizonyos építőelemek között fennálló kapcsolatok révén kialakult új entitás leírására használjuk. Ebben az értelemben a rendszert nem képzeljük el felbonthatatlannak - lehetnek alrendszerei -, sem elszigeteltnek, önmagában lévőnek, mert környezet veszi körül. A rendszernek, építő elemeinek és azok között fennálló kapcsolatoknak számos jellemzője lehet. A P-gráf módszertanban és természetes módon ebben a könyvben a gyártórendszerek, a termelési folyamatok továbbá az egyéb vizsgálni tervezett rendszerek általános, absztrakt modelljének a folyamathálózatokat tekintjük. Ez egy olyan matematikai objektum, nevezetesen egy gráf, amelynek a csúcsaihoz és éleihez olyan attribútumokat rendelünk, amelyekkel a modellezni kívánt rendszereink rendelkeznek és a vizsgálat szempontjából fontos (releváns). A P-gráf (Process graph) bizonyos - a könyvben részletezett - szabályosságokkal rendelkező folyamathálózat, amely már az építő elemekhez rendelt attribútumok nélkül is jól ábrázolja a rendszer struktúráját.

A P-gráf módszertan egyik - más módszertanoktól eltérő - szembeötlő tulajdonsága, hogy a strukturális modellezést követően a megoldandó, jellemzően optimalizáló matematikai modell, megfelelő algoritmusokkal generálható. Az ember természetes elvárása hogy az általa alkotott dolgok a lehető legjobbak legyenek, azaz a megadott szempontrendszer szerint optimálisak legyenek. A strukturális modellből előállított matematikai programozási feladatok megoldására a szakirodalomban számos módszert találunk, egyes feladatosztályok jelenleg is aktívan kutatott területek. A P-gráf módszertan is tartalmaz olyan eljárásokat e feladatok megoldására, amelyek kihasználják a P-gráf modell specialitásait és így hatékonyabbak az általános megoldó módszereknél.

1. FEJEZET

Szemléltető folyamathálózat-szintézis feladatok

1.1. Vegyipari gyártási feladat: Butanol, etanol és aceton előállítása gabonából, fermentációval

A fermentáció (erjesztés) biokémiai folyamat, amikor a kiindulási szerves anyagból enzim hatására más anyagok keletkeznek. A folyamat jellemzően vizes közegben, bonyolult összetételű elegyben megy végbe, ami számottevő mennyiségben tartalmaz rostokat és más szilárd anyagokat. A keletkező komponensek koncentrációja általában kicsi. Ezért a termékek kinyerése, tisztítása jelentős költséggel jár.

Példánkban megfelelő mikroorganizmusok segítségével, enzimkatalitikus úton butanolt állítunk elő, miközben kisebb mennyiségben etanol és aceton is keletkezik. Alapanyagként valamilyen gabonát, például kukoricát használhatunk fel az erjesztés során. Magával a fermentációs folyamattal nem foglalkozunk, a termékeket tartalmazó fermentációs elegyet a vizsgált folyamat szempontjából nyersanyagnak tekintjük. Célunk a termékek költségoptimális kinyerése az elegyből.

Hagyományos technológiákat figyelembe véve alapvetően két út kínálkozik a termékek kinyerésére:

- Az egyik esetben gáz sztripper (kigőzölő) segítségével távolítjuk el az elegyből az acetont, butanolt, etanol (ABE) komponenseket, melynek során az ABE mellett jelentős mennyiségű víz is távozik (ABEW) a sztripperből a gázárammal. Az ABEW tartalmú gázáramok kondenzátoron vezetjük keresztül, ahol az ABEW cseppfolyóssá válik, így elválasztható a gáztól, amit visszavezetünk a sztripperbe. (Példánkban a kondenzátort a gáz sztripper egység részének tekintjük.) Az ABEW komponensekre történő szétválasztására a magas víztartalom miatt azeotróp desztillációt alkalmazunk.
- Az ABE kinyerése a fermentációs elegyből megvalósítható folyadék-folyadék extrakcióval is. Az extraháló oldószert sztripper segítségével választjuk el az ABE-től, majd az extraktorba visszavezetjük. A kinyert ABE nem tartalmaz jelentős mennyiségű vizet, ezért a komponensekre történő szétválasztásához desztillációs oszlopok különféle konfigurációit alkalmazhatjuk.

További lehetőség adszorber egységek alkalmazása. Az adszorpció folyadékok és gázok szilárd felületen történő megkötése. A megkötött komponensek deszorpcióval eltávolíthatók a felületről (nyomás csökkentésével és/vagy hőmérséklet emelésével). Az adszorpció és deszorpció lépéseit egymásután végrehajtva a szétválasztás megvalósítható, a folyamatos működéshez viszont a folyamat regeneratív jellege miatt több egység szükséges. Megkötő anyagként gyakran műgyanta alapú speciális töltetet alkalmaznak.

A vizsgált folyamat esetében adszorbert alkalmazhatunk a gáz sztripper után, az ABEW-ből az ABE kinyerésére. A másik lehetőség, hogy a fermentációs elegyet centrifuga segítségével szétválasztjuk ABEW-ra és a szilárd összetevőkre, majd az így kinyert ABEW elegyet adszorpcióval szétválasztjuk ABE-ra és vízre. Az ABE további feldolgozása desztillációval történik.

A fermentációs elegyből az ABE kinyerése lehetséges úgy is, hogy először ultraszűréssel eltávolítjuk a szilárd alkotórészeket az elegyből, majd a szűrletből (ABEW) pervaporációval kinyerjük a ABE elegyet.

A pervaporáció a membránszűrés egy speciális fajtája, a szűrlet a membrán túloldalán gőz halmazállapotúvá válik. A membrán érzékeny az eltömődésre, ezért van szükség az ultraszűrésre a pervaporáció előtt.

A pervaporáció jól alkalmazható azeotrópos elegyekből relatíve illékony komponensek eltávolítására. A pervaporációt követően az ABE komponensei a már említett desztillációs eljárással választható szét.

1.1.1. Hagományos technológiák alkalmazása

1.1.1.1. Feladat specifikáció

Materials					
Name	Type	Quantity type	Price	Min. flow	Max. flow
s00	raw material	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s01	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s02	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s05	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s13	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s06	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s07	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s15	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s11	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s08	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s16	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s03	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s31	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s32	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s36	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s37	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s38	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s39	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s44	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s45	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s46	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s49	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s50	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s33	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s34	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s35	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s40	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s43	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s47	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s48	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s17	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr
s09	product	mass	0,00 €/t	6 350,29 t/yr	1 000 000 000,00 t/yr
s20	product	mass	0,00 €/t	1 814,37 t/yr	1 000 000 000,00 t/yr
s19	product	mass	0,00 €/t	23 586,80 t/yr	1 000 000 000,00 t/yr

1.1. táblázat. Anyagok

FEJEZET 1. SZEMLÉLTETŐ FOLYAMATHÁLÓZAT-SZINTÉZIS FELADATOK

Name	Working hours [h/yr]	Payout period [yr]	Lower bound	Capacity Upper bound	Operating units		Investment cost		Overall cost	
					Operating cost		Fixed charge [€]		Fixed charge [€/yr]	
					Fixed charge [€/yr]	Prop. constant [€/yr]	Fixed charge [€]	Prop. constant [€]	Fixed charge [€/yr]	Prop. constant [€/yr]
G1	2 000	3,00	0,00	1000 000 000,00	871 000,00	0,00	2 180 000,00	0,00	1 597 666,67	0,00
D1	2 000	3,00	0,00	1000 000 000,00	873 000,00	0,00	2 088 000,00	0,00	1 569 000,00	0,00
A1	2 000	3,00	0,00	1000 000 000,00	218 800 000,00	0,00	49 653 000,00	0,00	235 351 000,00	0,00
D3	2 000	3,00	0,00	1000 000 000,00	864 000,00	0,00	1 831 000,00	0,00	1 474 333,33	0,00
D2	2 000	3,00	0,00	1000 000 000,00	891 000,00	0,00	2 294 000,00	0,00	1 655 666,67	0,00
A2	2 000	3,00	0,00	1000 000 000,00	87 520 000,00	0,00	19 861 000,00	0,00	94 140 333,33	0,00
E1	2 000	3,00	0,00	1000 000 000,00	5 231 000,00	0,00	1 189 000,00	0,00	5 627 333,33	0,00
S1	2 000	3,00	0,00	1000 000 000,00	864 000,00	0,00	1 914 000,00	0,00	1 502 000,00	0,00
D5	2 000	3,00	0,00	1000 000 000,00	902 000,00	0,00	2 392 000,00	0,00	1 699 333,33	0,00
D6	2 000	3,00	0,00	1000 000 000,00	873 000,00	0,00	2 058 000,00	0,00	1 559 000,00	0,00
D7	2 000	3,00	0,00	1000 000 000,00	852 000,00	0,00	1 684 000,00	0,00	1 413 333,33	0,00
D8	2 000	3,00	0,00	1000 000 000,00	892 000,00	0,00	2 275 000,00	0,00	1 650 333,33	0,00
D9	2 000	3,00	0,00	1000 000 000,00	2 666 000,00	0,00	6 684 000,00	0,00	4 894 000,00	0,00
D10	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D11	2 000	3,00	0,00	1000 000 000,00	2 513 000,00	0,00	6 301 000,00	0,00	4 613 333,33	0,00
D12	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D13	2 000	3,00	0,00	1000 000 000,00	2 425 000,00	0,00	6 079 000,00	0,00	4 451 333,33	0,00
D14	2 000	3,00	0,00	1000 000 000,00	2 558 000,00	0,00	6 412 000,00	0,00	4 695 333,33	0,00
D15	2 000	3,00	0,00	1000 000 000,00	2 737 000,00	0,00	6 862 000,00	0,00	5 024 333,33	0,00
D16	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D17	2 000	3,00	0,00	1000 000 000,00	2 691 000,00	0,00	6 746 000,00	0,00	4 939 666,67	0,00
D18	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D19	2 000	3,00	0,00	1000 000 000,00	3 218 000,00	0,00	8 068 000,00	0,00	5 907 333,33	0,00
D20	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D21	2 000	3,00	0,00	1000 000 000,00	1 246 000,00	0,00	3 124 000,00	0,00	2 287 333,33	0,00
D22	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D25	2 000	3,00	0,00	1000 000 000,00	1 679 000,00	0,00	4 210 000,00	0,00	3 082 333,33	0,00
D26	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D27	2 000	3,00	0,00	1000 000 000,00	1 658 000 000,00	0,00	4 156 000,00	0,00	3 043 333,33	0,00
D28	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00
D29	2 000	3,00	0,00	1000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00

1.2. táblázat. Műveleti egységek

FEJEZET 1. SZEMLÉLTETŐ FOLYAMATHÁLÓZAT-SZINTÉZIS FELADATOK

Material rates of operating units	Operating units									
	G1	D1	A1	D3	D2	A2	E1	S1	D5	D6
s00 [t/yr]	-1 646 540,00						-1 646 540,00			
s01 [t/yr]	1 578 500,00									
s02 [t/yr]			31 751,50			31 751,50				
s05 [t/yr]	68 038,90	-68 038,90			-68 038,90					
s13 [t/yr]		58 059,80	-58 059,80							
s06 [t/yr]				-12 700,60	12 700,60					
s07 [t/yr]					55 338,30	-55 338,30				
s15 [t/yr]									25 401,20	-25 401,20
s11 [t/yr]							-1 770 830,00	1 770 830,00		
s08 [t/yr]								31 751,50	-31 751,50	
s16 [t/yr]							1 802 580,00	-1 802 580,00		
s03 [t/yr]							1 614 790,00			
s17 [t/yr]										
s09 [t/yr]		9 979,03		9 979,03					6 350,29	
s20 [t/yr]			2 721,55	2 721,55						1 814,37
s19 [t/yr]			23 586,80			23 586,80				23 586,80

(a)

Material rates of operating units	Operating units											
	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18
s11 [t/yr]												
s08 [t/yr]	-31 751,50		-31 751,50		-31 751,50		-31 751,50	-31 751,50	-31 751,50		-31 751,50	
s31 [t/yr]			0,91	-0,91								
s32 [t/yr]					0,91	-0,91						
s36 [t/yr]											-0,91	0,91
s33 [t/yr]									0,91	0,91		
s34 [t/yr]									-0,91	-0,91		
s35 [t/yr]											0,91	-0,91
s17 [t/yr]	8 164,66	-8 164,66										
s09 [t/yr]		6 350,29	6 350,29		3 175,15	3 175,15	6 350,29	6 350,29	6 350,29		6 350,29	
s20 [t/yr]		1 814,37		1 814,37		1 814,37	1 814,37	1 814,37		1 814,37		1 814,37
s19 [t/yr]	23 586,80		11 793,40	11 793,40	23 586,80		23 586,80	23 586,80	23 586,80		23 586,80	

(b)

Material rates of operating units	Operating units								
	D19	D20	D21	D22	D25	D26	D27	D28	D29
s00 [t/yr]									
s08 [t/yr]	-31 751,50		-31 751,50		-31 751,50		-31 751,50		
s36 [t/yr]									
s37 [t/yr]	0,91	-0,91							
s38 [t/yr]	-0,91	0,91							
s39 [t/yr]			0,91	0,91					
s44 [t/yr]					-0,91	0,91			
s45 [t/yr]					-0,91	0,91			
s46 [t/yr]					0,91	-0,91			
s49 [t/yr]							0,91		-0,91
s50 [t/yr]							-0,91		0,91
s35 [t/yr]									
s40 [t/yr]			0,91	-0,91					
s43 [t/yr]					-0,91	0,91			
s47 [t/yr]							-0,91	0,91	
s48 [t/yr]							0,91	-0,91	
s17 [t/yr]									
s09 [t/yr]		6 350,29	6 350,29			6 350,29		6 350,29	
s20 [t/yr]		1 814,37		1 814,37		1 814,37		907,19	907,19
s19 [t/yr]	23 586,80			23 586,80		23 586,80			23 586,80

(c)

1.3. táblázat. Az egység méretű műveleti egységek be- és kimeneti anyagáramai

1.1.1.2. Optimalis megoldás

Materials							
Name	Type	Quantity type	Price	Min. flow	Max. flow	Flow	Cost [€/yr]
s00	raw material	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	-1 646 540,00 t/yr	0,00
s09	product	mass	0,00 €/t	6 350,29 t/yr	1000 000 000,00 t/yr	6 350,29 t/yr	0,00
s20	product	mass	0,00 €/t	1 814,37 t/yr	1000 000 000,00 t/yr	1 814,37 t/yr	0,00
s19	product	mass	0,00 €/t	23 586,80 t/yr	1000 000 000,00 t/yr	23 586,80 t/yr	0,00
s11	intermediate	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	0,00 t/yr	0,00
s16	intermediate	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	0,00 t/yr	0,00
s03	intermediate	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	1 614 790 t/yr	0,00
s08	intermediate	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	0,00 t/yr	0,00
s39	intermediate	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	0,00 t/yr	0,00
s40	intermediate	mass	0,00 €/t	0,00 t/yr	1000 000 000,00 t/yr	0,00 t/yr	0,00
Total cost of materials [€/yr]							0,00

1.4. táblázat. Anyagok

Operating units												
Name	Size factor	Working hours [h/yr]	Payout period [yr]	Lower bound	Capacity Upper bound	Operating cost		Investment cost		Overall cost		Cost [€/yr]
						Fixed charge [€/yr]	Prop. constant [€/yr]	Fixed charge [€]	Prop. constant [€]	Fixed charge [€]	Prop. constant [€/yr]	
E1	1,00	2 000	3,00	0,00	1 000 000 000,00	5 321 000,00	0,00	1 189 000,00	0,00	5 627 333,33	0,00	5 627 333,33
S1	1,00	2 000	3,00	0,00	1 000 000 000,00	864 000,00	0,00	1 914 000,00	0,00	1 502 000,00	0,00	1 502 000,00
D21	1,00	2 000	3,00	0,00	1 000 000 000,00	1 246 000,00	0,00	3 124 000,00	0,00	2 287 333,33	0,00	2 287 333,33
D22	1,00	2 000	3,00	0,00	1 000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Total cost of operating units [€/yr]												9 416 666,67

1.5. táblázat. Műveleti egységek

Solution summary information			
Total cost of materials	0,00	€/yr	
Total cost of operation units	9 416 666,67	€/yr	
Overall cost of solution	9 416 666,67	€/yr	

(a)

Material rates and flows of operating units		E1	S1	D21	D22	Overall material flows
		Operating unit size factors				
		1,00	1,00	1,00	1,00	
s00 [t/yr]	rate flow	-1 646 540,00 -1 646 540,00				-1 646 540,00
s09 [t/yr]	rate flow			6 350,29 6 350,29		6 350,29
s20 [t/yr]	rate flow				1 814,37 1 814,37	1 814,37
s19 [t/yr]	rate flow				23 586,80 23 586,80	23 586,80
s11 [t/yr]	rate flow	-1 770 830,00 -1 770 830,00	1 770 830,00 1 770 830,00			0,00
s16 [t/yr]	rate flow	1 802 580,00 1 802 580,00	-1 802 580,00 -1 802 580,00			0,00
s03 [t/yr]	rate flow	1 614 790,00 1 614 790,00				1 614 790,00
s08 [t/yr]	rate flow		31 751,50 31 751,50	-31 751,50 -31 751,50		0,00
s39 [t/yr]	rate flow			0,91 0,91	-0,91 -0,91	0,00
s40 [t/yr]	rate flow			0,91 0,91	-0,91 -0,91	0,00

(b)

1.6. táblázat. Műveleti egységek anyagáramai

1.1.2. További műveleti egységek: centrifuga, adszorber

1.1.2.1. Feladat specifikáció

Az Anyagok megegyeznek az 1.1-es táblázattal.

Az 1.2. táblázatban ismertetett műveleti egységek az adszorberrel egészülnek ki:

Operating units										
Name	Working hours [h/yr]	Payout period [yr]	Capacity		Operating cost		Investment cost		Overall cost	
			Lower bound	Upper bound	Fixed charge [€/yr]	Prop. constant [€/yr]	Fixed charge [€]	Prop. constant [€]	Fixed charge [€/yr]	Prop. constant [€/yr]
B3B4	2 000	3,00	0,00	1 000 000 000,00	132 000,00	0,00	3 806 000,00	0,00	1 400 666,67	0,00

1.7. táblázat. Műveleti egységek

Az 1.3-as táblázat az adszorber be- és kimeneti áramával egészül ki:

Material rates of operating units	Operating units B3B4
s05 [t/yr]	-68 038,90
s08 [t/yr]	31 751,50

1.8. táblázat. Az egység méretű műveleti egységek be- és kimeneti anyagáramai

1.1.2.2. Optimális megoldás

Materials								
Name	Type	Quantity type	Price	Min. flow	Max. flow	Flow	Cost [€/yr]	
s00	raw material	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	-1 646 540,00 t/yr	0,00	0,00
s09	product	mass	0,00 €/t	6 350,29 t/yr	1 000 000 000,00 t/yr	6 350,29 t/yr	0,00	0,00
s20	product	mass	0,00 €/t	1 814,37 t/yr	1 000 000 000,00 t/yr	1 814,37 t/yr	0,00	0,00
s19	product	mass	0,00 €/t	23 586,80 t/yr	1 000 000 000,00 t/yr	23 586,80 t/yr	0,00	0,00
s05	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00	0,00
s01	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	1 578 500,00 t/yr	0,00	0,00
s08	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00	0,00
s39	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00	0,00
s40	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00	0,00
Total cost of materials [€/yr]								0,00

1.9. táblázat. Anyagok

Operating units												
Name	Size factor	Working hours [h/yr]	Payout period [yr]	Capacity		Operating cost		Investment cost		Overall cost		Cost [€/yr]
				Lower bound	Upper bound	Fixed charge [€/yr]	Prop. constant [€/yr]	Fixed charge [€]	Prop. constant [€]	Fixed charge [€/yr]	Prop. constant [€/yr]	
G1	1,00	2 000	3,00	0,00	1 000 000 000,00	871 000,00	0,00	2 180 000,00	0,00	1 597 666,67	0,00	1 597 666,67
D21	1,00	2 000	3,00	0,00	1 000 000 000,00	1 246 000,00	0,00	3 124 000,00	0,00	2 287 333,33	0,00	2 287 333,33
B3B4	1,00	2 000	3,00	0,00	1 000 000 000,00	132 000,00	0,00	3 806 000,00	0,00	1 400 666,67	0,00	1 400 666,67
D22	1,00	2 000	3,00	0,00	1 000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Total cost of operating units [€/yr]												5 285 666,67

1.10. táblázat. Műveleti egységek

Solution summary information

Total cost of materials	0,00	€/yr
Total cost of operation units	5 285 666,67	€/yr
Overall cost of solution	5 285 666,67	€/yr

(a)

Material rates and flows of operating units		G1	D21	B3B4	D22	Overall material flows
		Operating unit size factors				
		1,00	1,00	1,00	1,00	
s00 [t/yr]	rate flow	-1 646 540,00 -1 646 540,00				-1 646 540,00
s09 [t/yr]	rate flow		6 350,29 6 350,29			6 350,29
s20 [t/yr]	rate flow				1 814,37 1 814,37	1 814,37
s19 [t/yr]	rate flow				23 586,80 23 586,80	23 586,80
s05 [t/yr]	rate flow	68 038,90 68 038,90		-68 038,90 -68 038,90		0,00
s01 [t/yr]	rate flow	1 578 500,00 1 578 500,00				1 578 500,00
s08 [t/yr]	rate flow		-31 751,50 -31 751,50	31 751,50 31 751,50		0,00
s39 [t/yr]	rate flow		0,91 0,91		-0,91 -0,91	0,00
s40 [t/yr]	rate flow		0,91 0,91		-0,91 -0,91	0,00

(b)

1.11. táblázat. Műveleti egységek anyagáramai

1.1.3. További műveletek: ultraszűrés, pervaporáció

1.1.3.1. Feladat specifikáció

Az Anyagok megegyeznek az 1.1-es táblázattal.

Az 1.7-es táblázathoz képest az ultraszűrés és pervaporáció műveletek összevont egységével egészül ki a táblázat:

Name	Working hours [h/yr]	Payout period [yr]	Capacity Lower bound	Capacity Upper bound	Operating cost		Investment cost		Overall cost	
					Fixed charge [€/yr]	Prop. constant [€/yr]	Fixed charge [€]	Prop. constant [€]	Fixed charge [€/yr]	Prop. constant [€/yr]
UIP1	2 000	3,00	0,00	1 000 000 000,00	520 000,00	0,00	88 006 000,00	0,00	29 855 333,33	0,00

1.12. táblázat. Műveleti egységek

Az 1.8-as táblázathoz képest az ultraszűrés és pervaporáció műveletek összevont egységével egészül ki a táblázat:

Material rates of operating units	Operating units UIP1
s00 [t/yr]	-1 646 540,00
s08 [t/yr]	31 751,50

1.13. táblázat. Az egység méretű műveleti egységek be- és kimeneti anyagáramai

1.1.3.2. Optimális megoldás

A pervaporáció művelet bevonása az alkalmazható technológiák körébe nem változtatta meg az optimális megoldásstruktúrát, továbbra is az 1.1.2.2-es részben ismertett megoldás az optimális. A következőkben a pervaporáció beruházási költségének csökkentésével megvizsgáljuk, hogy milyen paraméterek mellett válik versenyképessé a technológia.

1.1.4. Érzékenységvizsgálat

Azt találtuk, hogy több mint 90%-kal kell a pervaporáció beruházási költségét csökkenteni ahhoz, hogy alkalmazása költségoptimális legyen.

1.1.4.1. Optimális megoldás

Materials							
Name	Type	Quantity type	Price	Min. flow	Max. flow	Flow	Cost [€/yr]
s00	raw material	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	-1 646 540,00 t/yr	0,00
s09	product	mass	0,00 €/t	6 350,29 t/yr	1 000 000 000,00 t/yr	6 350,29 t/yr	0,00
s20	product	mass	0,00 €/t	1 814,37 t/yr	1 000 000 000,00 t/yr	1 814,37 t/yr	0,00
s19	product	mass	0,00 €/t	23 586,80 t/yr	1 000 000 000,00 t/yr	23 586,80 t/yr	0,00
s08	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00
s39	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00
s50	intermediate	mass	0,00 €/t	0,00 t/yr	1 000 000 000,00 t/yr	0,00 t/yr	0,00
Total cost of materials [€/yr]							0,00

1.14. táblázat. Anyagok

Operating units												
Name	Size factor	Working hours [h/yr]	Payout period [yr]	Capacity		Operating cost		Investment cost		Overall cost		Cost [€/yr]
				Lower bound	Upper bound	Fixed charge [€/yr]	Prop. constant [€/yr]	Fixed charge [€]	Prop. constant [€]	Fixed charge [€]	Prop. constant [€/yr]	
D21	1,00	2 000	3,00	0,00	1 000 000 000,00	1 246 000,00	0,00	3 124 000,00	0,00	2 287 333,33	0,00	2 287 333,33
UIP1	1,00	2 000	3,00	0,00	1 000 000 000,00	520 000,00	0,00	3 555 794,00	0,00	1 705 264,67	0,00	1 705 264,67
D22	1,00	2 000	3,00	0,00	1 000 000 000,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Total cost of operating units [€/yr]												3 992 598,00

1.15. táblázat. Műveleti egységek

Solution summary information

Total cost of materials	0,00	€/yr
Total cost of operation units	3 992 598,00	€/yr
Overall cost of solution	3 992 598,00	€/yr

(a)

Material rates and flows of operating units		D21	U1P1	D22	Overall material flows
		Operating unit size factors			
		1,00	1,00	1,00	
s00 [t/yr]	rate flow		-1646 540,00 -1646 540,00		-1 646 540,00
s09 [t/yr]	rate flow	6 350,29 6 350,29			6 350,29
s20 [t/yr]	rate flow			1 814,37 1 814,37	1 814,37
s19 [t/yr]	rate flow			23 586,80 23 586,80	23 586,80
s08 [t/yr]	rate flow	-31 751,50 -31 751,50	31 751,50 31 751,50		0,00
s39 [t/yr]	rate flow	0,91 0,91		-0,91 -0,91	0,00
s40 [t/yr]	rate flow	0,91 0,91		-0,91 -0,91	0,00

(b)

1.16. táblázat. Műveleti egységek anyagáramai

1.2. Üzleti folyamatok

Egy az olajiparból származó ellátási-lánc feladatot mutatunk be a következőkben. A megoldandó feladatban egy olajipari vállalat három telephelye szerepel, amelyek Pécs, Dombóvár és Kaposvár. A cél hogy biztosítva legyen a megkövetelt biodízel mennyiség Kaposváron a másik kettő telephelyről minimális összköltséggel és kockázattal.

Megszabott mennyiségű biodízel és komponensei állnak rendelkezésre Dombóváron és Pécsen. A pécsi telephely rendelkezik egy limitált keverési kapacitással a rendelkezésre álló komponensekből történő biodízel előállítására, vagyis ott a négy fő komponens - amelyek a K2 komponens (HDS gázolaj), a K4 komponens (kerozin), a K7 komponens (szulfur nélküli gázolaj) és a K8 komponens (FAME biokomponens) - feldolgozható és abból biodízel állítható elő. A biodízel mindegyik telephelyen tartályokba tölthető, amely tartályok teherautókkal szállíthatók Kaposvárra, ahol aztán azok lefejtethetők. Mindegyik telephelyen van egy felső korlátja az ott levő erőforrásoknak.

A szemléltető feladat kombinatorikus komponenseit egy hármassal (P, R, O) adhatjuk meg:

$$P = \{\text{Biodízel lefejtve a Kaposvári telephelyen}\}$$

$$R = \{\text{Biodízel a dombóvári telephelyen}, \text{Biodízel a pécsi telephelyen}, \text{K2 komponens}, \text{K4 komponens}, \text{K7 komponens}, \text{K8 komponens}\}$$

$$\begin{aligned} O = & \{B, U1, U2, U3, T1, T2, D\} = \\ = & \left\{ \left(\{K2 \text{ komponens}, K4 \text{ komponens}, K7 \text{ komponens}, K8 \text{ komponens}\}, \right. \right. \\ & \left. \left. \{Biodízel előállítva Pécsen\} \right), \right. \\ & \left. \left(\{Biodízel a dombóvári telephelyen\}, \{Biodízel tartályba töltve Dombóváron\} \right), \right\} \end{aligned}$$

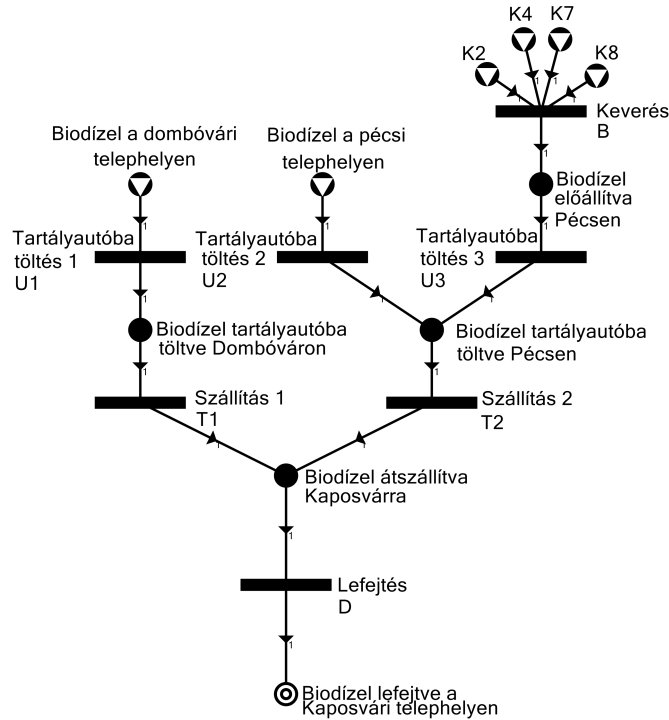
$$\left(\left\{ \text{Biodízel a pécsi telephelyen} \right\}, \left\{ \text{Biodízel tartályba töltve Pécsen} \right\} \right), \\
\left(\left\{ \text{Biodízel előállítva Pécsen} \right\}, \left\{ \text{Biodízel tartályba töltve Pécsen} \right\} \right), \\
\left(\left\{ \text{Biodízel tartályba töltve Dombóváron} \right\}, \left\{ \text{Biodízel átszállítva Kaposvárra} \right\} \right), \\
\left(\left\{ \text{Biodízel tartályba töltve Pécsen} \right\}, \left\{ \text{Biodízel átszállítva Kaposvárra} \right\} \right), \\
\left(\left\{ \text{Biodízel átszállítva Kaposvárra} \right\}, \left\{ \text{Biodízel lefejtve a Kaposvári telephelyen} \right\} \right) \Bigg\}$$

A szemléltető példa tevékenységeit és azok, az előzőekben felsorolt előfeltételeit és következményeit, illetve a megbízhatóságukat az az 1.17-es táblázat foglalja össze.

Tevékenységek	Előfeltételek	Következmények	Magbízhatóság
Keverés (B)	K2 komponens, K4 komponens, K7 komponens, K8 komponens	Biodízel előállítva Pécsen	$r_B = 98\%$
Tartályautóba töltés 1 (U1)	Biodízel a dombóvári telephelyen	Biodízel tartályautóba töltve Dombóváron	$r_{U1} = 98\%$
Tartályautóba töltés 2 (U2)	Biodízel a pécsi telephelyen	Biodízel tartályautóba töltve Pécsen	$r_{U2} = 98\%$
Tartályautóba töltés 3 (U3)	Biodízel előállítva Pécsen	Biodízel tartályautóba töltve Pécsen	$r_{U3} = 98\%$
Szállítás 1 (T1)	Biodízel tartályautóba töltve Dombóváron	Biodízel átszállítva Kaposvárra	$r_{T1} = 98\%$
Szállítás 2 (T2)	Biodízel tartályautóba töltve Pécsen	Biodízel átszállítva Kaposvárra	$r_{T2} = 98\%$
Lefejtés (D)	Biodízel átszállítva Kaposvárra	Biodízel lefejtve a Kaposvári telephelyen	$r_D = 98\%$

1.17. táblázat. A tevékenységek előfeltételei és következményei

Feladatunk azon megoldásstruktúra meghatározása, amely az egyes tevékenységek megbízhatóságának és költségének ismeretében megadja az modellezett üzleti folyamat megbízhatóságát, de egyúttal a költségeket is nyomon követhetjük.



1.1. ábra. A szemléltető példa maximális struktúrájának P-gráf reprezentációja

1.3. Jármű hozzárendelési példa

A jármű hozzárendelési probléma (Vehicle Assignment Problem, VAP) egy specializált hozzárendelési probléma, melyben különböző elvégzendő szállítási feladatok végrehajtása céljából rendelünk hozzájuk járműveket. A problémában az elvégzendő szállítások, a rendelkezésre álló járművek, valamint a környezet (távolságok, helyek, stb.), és minden egyéb, a feladat során szükséges tulajdonság (jármű kapacitás, maximális sebesség, fogyasztás, stb.) előre definiált, továbbá az is, hogy mely jármű mely feladatok elvégzésére képes. A feladat egy olyan hozzárendelés megvalósítása, melyben az összes szükséges feladatot elvégezzük, és az valamely előre meghatározott szempontból optimális. A feladatnak többféle megoldási módja is megtalálható a szakirodalomban, a heurisztikus módszereken keresztül a bizonyítottan optimális eredményt szolgáltatóig.

Tekintsünk az elvégzendő szállítási feladatokra úgy, mint végcélokra. A erőforrások pedig legyenek a kiindulási helyükön állomásozó használható járművek. A köztes entitások jelöljék a szállítások során érinthető helyeket, esetleg átpakolási, vagy pihenőhelyeket, amiket érintenie kell, vagy érinthet az adott jármű minden esetben. A tevékenységek ezen analógia alapján az egyes helyek közti lehetséges mozgásokat fogják reprezentálni. A tevékenységek magukba foglalják az egyes járművek használatba hozatalát, a járművek különböző helyek közti mozgatását és a járművek célba juttatását egy olyan helyről, amelyből vezet út a kívánt célhoz.

A feladat tehát három teljesítendő szállítást, azaz végcélt tartalmaz ($P1_Almádi$, $P2_Székesfehérvár$, $P3_Tihany$), és ehhez három különböző jármű áll rendelkezésre ($R1$, $R2$, $R3$), ahogy az a feladat maximális struktúráján látható, amelyet az 1.2 ábra szemléltet. Az $R1$ -es teherautó Balatonkenesén állomásozik és méretéből és típusából adódóan Veszprémből Székesfehérvárra tud árut szállítani. Az $R2$ -es teherautó Székesfehérváron állomásozik és árut szállíthat Veszprémből Tihanyra, illetve Tihanyból Balatonalmádiba. Az $R3$ -as teherautó Tihanyban állomásozik és árut szállíthat Veszprémből Tihanyra, Veszprémből Székesfehérvárra, illetve Tihanyból Balatonalmádiba. A feladat kombinatorikus komponensei egy hármassal $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ adhatóak meg, ahol

$$\mathcal{R} = \{P1_Almádi, P2_Székesfehérvár, P3_Tihany\}$$

$$\mathcal{P} = \{R1, R2, R3, EUR, CO_2\}$$

$$\begin{aligned}
\mathcal{O} = & \left\{ r3_Balatonkenese-Vp, r3_Vp-Szf, r2_Szf-Vp, r2_Szf-Tihany, r2_Vp-Tihany, \right. \\
& r2_Tihany-Almádi, r2_Almádi-Vp, r1_Tihany-Almádi, r1_Tihany-Vp, r1_Almádi-Vp_2, \\
& r1_Vp-Szf, r1_Vp-Tihany, r1_Tihany-Vp_2, r1_Szf-Vp, r1_Szf-Tihany, r1_Tihany-Tihany2, \\
& Travel_R3, Travel_R2, Travel_R1, R1_Tihany-Vp_3, R1_Almádi-Vp, R1_start_Tihany2 \Big\} = \\
= & \left\{ \left(\{R3, Km_R3\}, \{R3_Vp\} \right), \left(\{R3_Vp, Km_R3\}, \{P2_Székesfehérvár\} \right), \right. \\
& \left(\{Km_R2, R2\}, \{R2_Vp\} \right), \left(\{Km_R2, R2\}, \{R2_Tihany\} \right), \\
& \left(\{Km_R2, R2_Vp\}, \{P3_Tihany, R2_Tihany\} \right), \\
& \left(\{Km_R2, R2_Tihany\}, \{P1_Almádi, R2_Almádi\} \right), \\
& \left(\{Km_R2, R2_Almádi\}, \{R2_Vp\} \right), \left(\{Km_R1, R1_Tihany2\}, \{P1_Almádi, R1_Almádi\} \right), \\
& \left(\{Km_R1, R1\}, \{R1_Vp_2\} \right), \left(\{Km_R1, R1_Almádi\}, \{R1_Vp_1\} \right), \\
& \left(\{Km_R1, R1_Vp_1\}, \{P2_Székesfehérvár, R1_Szf\} \right), \\
& \left(\{Km_R1, R1_Vp_2\}, \{P3_Tihany, R1_Tihany\} \right), \\
& \left(\{Km_R1, R1_Tihany\}, \{R1_Vp_1\} \right), \left(\{Km_R1, R1_Szf\}, \{R1_Vp_2\} \right), \\
& \left(\{Km_R1, R1_Szf\}, \{R1_Tihany2\} \right), \left(\{Km_R1, R1_Tihany\}, \{R1_Tihany2\} \right), \\
& \left(\{EUR, CO_2\}, \{Km_R3\} \right), \left(\{EUR, CO_2\}, \{Km_R2\} \right) \\
& \left(\{EUR, CO_2\}, \{Km_R1\} \right), \left(\{Km_R1, R1\}, \{R1_Vp_1\} \right) \\
& \left. \left(\{Km_R1, R1_Almádi\}, \{R1_Vp_2\} \right), \left(\{Km_R1, R1\}, \{R1_Tihany2\} \right) \right\}
\end{aligned}$$

Entitások	Típus	Legkorábbi rendelkezésre állás (Lt) [min]	Határidő (tb) [min]
$R1$	erőforrás	20	0
$R2$	erőforrás	0	0
$R3$	erőforrás	0	0
EUR	erőforrás	0	0
CO_2	erőforrás	0	0
$P1_Almádi$	végcél	0	180
$P2_Székesfehérvár$	végcél	0	160
$P3_Tihany$	végcél	0	130

1.18. táblázat. Az entitások paraméterei

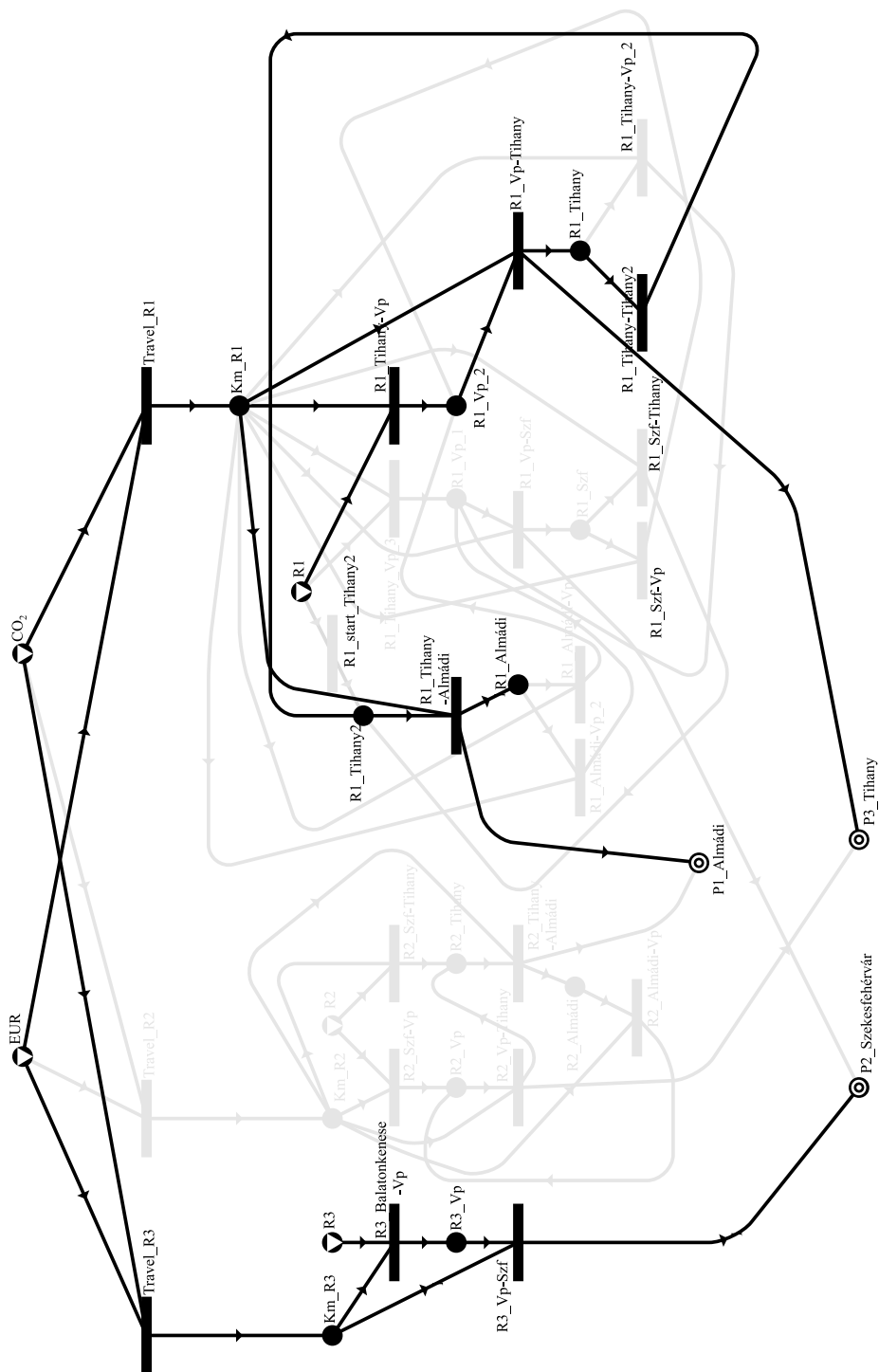
A feladat entitásainak idő paramétereit a 1.18. táblázat, a tevékenységeinek idő paramétereit pedig a 1.19. táblázat foglalja magába. Ahogy az a 1.18. táblázatban látható, három végcél tartalmaz a feladat a $P1_Almádi$ -t, amit 180 perc alatt, a $P1_Székesfehérvár$ -t, amit 160 perc alatt és a $P1_Tihany$ -t, amit 130 perc alatt kell teljesíteni.



Tevékenységek	Fix idő (t_f) [min]	Arányos idő (t_p) [min]
<i>R3_Balatonkenese-Vp</i>	60	0
<i>R3_Vp-Szf</i>	80	0
<i>R2_Szf-Vp</i>	80	0
<i>R2_Szf-Tihany</i>	140	0
<i>R2_Vp-Tihany</i>	60	0
<i>R2_Tihany-Almádi</i>	40	0
<i>R2_Almádi-Vp</i>	50	0
<i>R1_Tihany-Almádi</i>	40	0
<i>R1_Tihany-Vp</i>	60	0
<i>R1_Almádi-Vp_2</i>	50	0
<i>R1_Vp-Szf</i>	80	0
<i>R1_Vp-Tihany</i>	60	0
<i>R1_Tihany-Vp_2</i>	60	0
<i>R1_Szf-Vp</i>	80	0
<i>R1_Szf-Tihany</i>	140	0
<i>R1_Tihany-Tihany2</i>	0	0
<i>Travel_R3</i>	0	0
<i>Travel_R2</i>	0	0
<i>Travel_R1</i>	0	0
<i>R1_Tihany-Vp_3</i>	60	0
<i>R1_Almádi-Vp</i>	50	0
<i>R1_start_Tihany2</i>	0	0

1.19. táblázat. A tevékenységek előfeltételei, következményei és idő paraméterei

A később (a 3.3-as fejezetben) tárgyalt időkorlát relaxációt felhasználva a feladat megoldható, amelynek optimális megoldás struktúrája az 1.3 ábrán látható. Azaz az *R1*-es és az *R3*-as teherautókkal teljesíthető mind a három szállítási feladat a megadott határidőkön belül. Az *R1*-es autó Balatonkeneséről indulva 160 perc alatt ér Székesfehérvárra (*P1_Székesfehérvár*) a Veszprémből szállított áruval, azaz pont teljesíti a megadott határidőt. Az *R2*-es autó Tihanyból kiindulva először Veszprémből Tihanyba (*P3_Tihany*) szállít árut az indulástól számított 130 percen belül, majd pedig Tihanyból Balatonalmádiba (*P2_Balatonalmádi*) az indulástól számított 160 percen belül, azaz mindegyik szállítás teljesíti a kitűzött határidőket.



13. ábra. Az időkorlátokkal kibővített jármű hozzárendelési feladat megoldás struktúrája

1.4. Hőcserélő hálózatok

A hőcserélő hálózatok szintézise (Heat Exchanger Network Synthesis, HENS) során a hőcserélő berendezések egy olyan hálózatának (Heat Exchanger Network, HEN) a tervezése a cél, mely optimálisan kielégíti a termelési rendszerben fellépő hűtési és fűtési igényeket. A hűtési és fűtési igények kielégíthetők a rendszerbe kívülről behozott energiával (pl. hűtőtorony vagy forró gőz). A meglévő fűtési és hűtési igényeket, mint hőelvonási lehetőségeket és hőforrásokat kiaknázva, a felhasznált külső energia mennyisége és így a termelési rendszer költsége is csökkenthető. Az optimális HEN meghatározásához figyelembe kell venni a hőcserélő egységek költségeit. A HENS feladatok általában nagy méretű, összetett, kombinatorikus jellegű feladatok. A hőcserélő hálózat magas költsége miatt az optimális HEN megtalálása fontos feladat.

A hőcserélő hálózat egy olyan a termelő folyamatba integrált folytonos alrendszer, melyben egyes folyamatok hőmérsékletét növelni, másokét pedig csökkenteni kell. Az előbbieket hidegáramoknak (kezdeti hőmérsékletük alacsonyabb, mint a végső), az utóbbiakat melegáramoknak nevezzük. Ezeket az áramokat közösen hőáramoknak nevezzük. Ha egy hőáram fajhője (fajlagos hőkapacitása) c , akkor m tömegű anyag hőmérsékletének ΔT -vel való megváltoztatásához $\Delta Q = cm\Delta T$ energiára van szükség. Folytonos rendszer esetén a képletben a tömeg helyett a folyam nagyságát kell venni, az energia helyett pedig az időegység alatt befektetett energia nagyságát kapjuk.

A hőcsere működését a termodinamika első és második törvénye írja le. A termodinamika első törvénye az energia megmaradását mondja ki, azaz egy zárt rendszerben jelen levő energia összege nem változik. A termodinamika második törvénye azt mondja, hogy a hőáramlás iránya a melegebb helyről a hidegebb hely felé mutat. A hőcserélő hálózatok tervezésénél a második törvény alapján a cél a hőáramok többszöri felhasználása egészen addig, míg a törvény ezt megengedi. Melegenergia szolgáltatónak (hot utility) azokat a gyártórendszeren kívüli eszközöket nevezzük, melyek hőt juttatnak a rendszerbe a megfelelő hőigények kielégítésére. A hidegenergia szolgáltató (cold utility) a gyártási rendszerben jelen levő, fel nem használható hőt vonja ki. Melegenergia szolgáltatóként használhatunk gőzt, melyet bojleremből, vagy megcsapolt turbinákból nyerhetünk. Hidegenergia szolgáltatóként vizet vagy hideg levegőt használhatunk.

A melegáramokat és a melegenergia szolgáltatókat hőforrásoknak nevezzük, a hidegáramokat és hidegenergia szolgáltatókat pedig hőnyelőknek hívjuk. Egy hőforrás és hőnyelő között a hőátadás hőcserélő berendezésen keresztül történik. A termodinamika második főtétele szerint akkor lehetséges hőcsere, ha a hőforrás melegebb, mint a hőnyelő, azonban a gyakorlatban gazdaságossági szempontokból megköveteljük, hogy egy minimális, MT (vagy ΔT_{min}) hőmérséklet különbség legyen a hőforrás és a hőnyelő között.

Egy általános HENS problémát a bemenetek segítségével definiálhatunk, ahol a bemenetek a következők:

- A melegáramok halmaza (F^H);
- A hidegáramok halmaza (F^C);
- Az anyagáramok nagysága és hőkapacitása;
- A felhasználható hőforrások (U^H) és hőnyelők (U^C) és ezek hőmérséklete és költsége;
- A hőcserélők költsége.

1.5. Ütemezés gyártási jellegű feladatok esetén

Az ütemezési feladatot szakaszos üzemű termelő rendszerek esetén általában a gyártási folyamat struktúrájával (az úgynevezett recepttel), és a felhasználható gépek (berendezések) halmazának megadásával definiálják. A receptben szereplő munkafázisok (művelet, taszk) elvégzésére több különböző berendezés is használható. Természetesen a használt berendezéstől függ a működési ideje és a működési költség az adott munkafázisnak.

Az ütemezési feladatban mindegyik művelethez hozzá kell rendelni egy berendezést, amely majd ezt a műveletet végre fogja hajtani. A berendezések több műveletet is végrehajthatnak, azaz áthelyezhetőek, de természetesen egy időben csak egy műveletet végezhetnek. Egy berendezés áthelyezésekor berendezéstől és műveletektől függően egy váltási időt figyelembe kell venni. A váltási idő az az időtartam, amennyinek legalább el kell telnie a berendezéshez rendelt két egymás utáni művelet között. Általában a váltási idő a berendezések konfigurálására, tisztítására, vagy esetleg szállítására kell. Az összes berendezésnek meg kell határozni az optimális műveleti sorrendjét, azaz ha egy berendezés végrehajtott egy műveletet akkor ezután melyik az a következő művelet amit az adott berendezésnek el kell végeznie.

Szakaszos rendszerekben a termelés **batch**ekben (adagokban) történik. Azaz ha az előállítandó termék mennyisége több, mint amennyi a recept egyszeri végrehajtása során keletkezik, akkor a recept többszöri megismétlésével állítják elő a kívánt termékmennyiséget. A recept egyszeri végrehajtását, ütemezését jelenti egy batch-nyi termék előállítása.

Az ütemezési feladatok megoldása során valamely szempont szerinti optimális ütemezést keresünk. Példák ütemezési célokra:

- Adott mennyiségű termék legyártása minél rövidebb idő alatt.
- Adott idő alatt minél több termék legyártása.
- Termékek legyártása határidőre.
- Működési költségek minimalizálása.

1.6. Fenntarthatósági feladat

Amennyiben egy térségben több fajta megújuló energiahordozó érhető el, például fa, biogáz, stb., akkor ezek felhasználásának mikéntje komplex kombinatorikus feladat. Ennek a feladatnak a megfogalmazása és a P-gráf alapú megoldási módszer bemutatása a célunk.

Tegyük fel például, hogy egy burgenlandi régióban kukoricát termesztene, de számos erdőterület is megtalálható. A kukorica magból bio-etanol állítható elő, de takarmányként is fel lehet használni. Ekkor a megmaradt kukoricacsutkákból és szárból biogáz állítható elő, erjesztéssel. Az állatok trágyáját szintén fel lehet használni erre a célra. A gázt kétfajta kazánban lehet elégetni, az egyik csak hőt, a másik hőt és elektromosságot is termel. A hőt el lehet adni, de egy részére szükség van a kukoricacsutka és a fa szárításához. A fát közvetlenül el lehet égetni, vagy előállíthatunk belőle pelletet illetve faaprítékot. A közvetlen égetéshez több manuális munka kell az adagolás miatt, a pellet és az apríték adagolása viszont automatikus, de az előállításukhoz áram szükséges. Jelen van még a rendszerben földgáz, amellyel a lakossági és egyéb hőigényeket bármikor el lehet látni, és az elektromos hálózat, amely az áram-igényt tudja fedezni. Lehetnek még más, itt le nem írt lehetőségek, és adottak az egyes technológiákhoz kapcsolódó technikai és költségparaméterek. Ezt a feladatot fejben nem lehet optimálisan megoldani, hiszen nagyon sok az egymásra hatás, a feltétel és a lehetséges elrendezés. Szisztematikusan kell meghatározni minden lehetőséget, és azokat adott szempont szerint kell elemezni.

Olyan módszert szeretnénk megvalósítani, amely képes az optimális struktúra mellett számos más alternatíva meghatározására is, így a döntéshozók olyan feltételeket is figyelembe tudnak venni, amelyek az eredeti modellben nem voltak megfogalmazva. A módszer másik újdonsága, hogy az optimalizálás során nem csak a költséget veszi figyelembe, hanem több más fenntarthatósági mértéket is számol modell szinten. Így például az "emergency"-t vagy az ökológia lábnyomot. Ez lehetővé teszi, hogy a megoldásokból meghatározzuk a Pareto frontot, vagy, hogy ezen értékekre fogalmazzunk meg feltételeket.

Egy olyan esettanulmányt választottunk a módszer szemléltetésére, amelyben az a feladat, hogy egyszerre gyártsunk hőt és elektromosságot egy adott régióban.

A cél az, hogy teljesítsük az elektromosság- és hőtermelési követelményeket (2000 és 5000 MWh/év). Természetesen rendelkezésre állnak nem megújuló technológiák, például elektromosság vásárlása külső forrásból, illetve hőtermelés földgáz segítségével. A vizsgált körzet rendelkezik megújuló

forrásokkal, mint szilázs kukorica (összegyűjtött és erjesztett), szilázs fű, kukoricacsutka (a szárat és a levelet is beleértve) és fa. Emellett több energia konverziós technológia áll rendelkezésre, például biogáz erőmű, CHP biogáz erőmű, gázkazán, pellet gép és hagyományos kazán. Érdemes megjegyezni, hogy néhány műveleti egység, például a pellet gép, csak adott méretekben kapható. Ezekhez a méretekhez tartozó paraméterekből állítottuk elő a költségfüggvényt. Az eredeti adatok Lutzenberger és szerzőtársai 2008-as Konzeption von Modellen multifunktionaler Energiezentren für die Ost-Steiermark című művében találhatóak.

Név	Költség	Max. [ha/év]
electricity_grid	149 €/MWh	
natural_gas	0.5 €/m ³	
area_corn		300
area_corn_silage		400
area_grass_silage		1200
area_wood		600

1.20. táblázat. Nyersanyagok tulajdonságai

Név	Min. [MWh/év]
hot_utility	5000
electric_utility	2000

1.21. táblázat. Termékek tulajdonságai

2. FEJEZET

Folyamathálózatok strukturális modellje: A P-gráf módszertan alapjai

A folyamatszintézis feladata egy folyamatrendszer optimális struktúrájának, valamint a rendszert alkotó, különféle műveleteket végrehajtó funkcionális egységek optimális konfigurációinak, kapacitásainak meghatározása [5]. Szerepe kritikus az anyag- és energiafogyasztás, illetve a környezetre gyakorolt negatív hatások csökkentésében, ezáltal pedig a nyereségesség növelésében. Szakirodalmi példa támasztja alá, hogy a hatékony folyamatszintézis az energiafogyasztást akár 50, a költséget pedig 35%-al is csökkentheti [6].

Ideális esetben egy folyamat struktúráját, és a folyamatot alkotó műveletek konfigurációit egyidejűleg tudnánk megtervezni és szintetizálni, mivel a teljesítményük kihatással van egymásra. Ez azonban a gyakorlatban rendkívül nehéz, ha nem lehetetlen, köszönhetően a probléma duális, egyidejűleg folytonos és diszkrét természetének. Az utóbbi a feladat kombinatorikus komplexitáshoz vezet, ami a probléma optimális megoldásának megtalálását jelentősen megnehezíti. Emiatt a folyamatszintézisnek 3 fázisát különböztetjük meg: a makroszkopikus, mezoszkopikus és a mikroszkopikus fázist. A különböző részfeladatokat már ezek között a fázisok között osztjuk el, a végcél, azaz az optimális folyamat megtervezését és szintézisét szem előtt tartva. Ahogyan az a nevükből is következik, a 3 fázist az alapján különböztetjük meg, hogy milyen részletességgel foglalkoznak a folyamat megtervezésével. A részrendszerek, a funkcionális egységek összekapcsolása, azaz a rendszer szintézise a makroszkopikus fázisban megy végbe, ezért a folyamatszintézis szempontjából ez a fázis a legjelentősebb.

A folyamatszintézis feladatok megoldására kidolgozott módszerek két nagy csoportba osztjuk, mégpedig a heurisztikus és az algoritmikus azaz matematikai programozáson alapuló módszerekre. Léteznek úgynevezett hibrid módszerek is, amelyek heurisztikus szabályok mellett egyidejűleg támaszkodnak a matematikai programozásra is.

A heurisztikus módszerek megvalósítása általában egyszerű, még nagy feladatok esetében is, azonban természetüknél fogva csak lokálisan hatékonyak. Ennek oka, hogy az emberi tapasztalatok, melyeken a heurisztikák szabályai alapulnak, véges, és gyakran korlátozott, limitált számú megfigyelésből erednek. Következésképpen a heurisztikus módszerek önmagukban gyakran alkalmatlanok a globális, vagy közel globális optimális megoldások megtalálására.

A heurisztikus módszerekkel szemben, a meglévő, hagyományos algoritmikus módszerek, melyek főbb lépései a 2.1-es ábrán láthatóak, csak viszonylag kis, illetve mérsékelt méretű feladatok kezelésére alkalmasak. Ezek az algoritmikus módszerek csak akkor lesznek megfelelően precízek, ha az általuk használt matematikai programozási modelleket explicit módon meg tudjuk konstruálni.

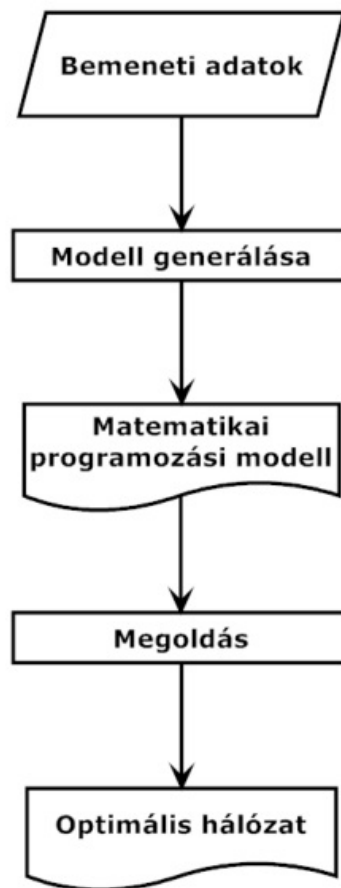
Ezen módszerek esetében a feladat definíciója mindig megadja a rendelkezésre álló nyersanyagokat, a gyártani kívánt termékeket valamint a felhasználható műveleti egységeket. A különböző kapcsolódó paraméterekre – mint költségekre, árakra illetve anyagegyensúly feltételekre – írjuk fel a matematikai programozási modelleket. Ezeket a modelleket általában félig-meddig korábbi tapasztalatokra alapozva adják meg, folytonos függvények segítségével. Mindemellett ezek a hagyomá-

nyos módszerek nem tudják közvetlenül és szisztematikusan, azaz algoritmikusan képezni a költségfüggvényt és a kapcsolódó feltételrendszert, amely megfelelően kifejezné a folyamatok hálózatstruktúráját. Nem adnak módszert arra sem, hogy algoritmikusan készítsünk el egy olyan struktúráját a folyamatnak, amely minden lehetséges hálózatot redundancia nélkül tartalmaz. Ez azért kritikus, mert a matematikai programozás egyrészt csak olyan eredményt adhat, ami része a matematikai modellnek, másrészt használhatatlanul lelassulhat, ha a modell indokolatlanul nagy. A konvencionális módszerek esetében nincs garanciánk a folyamatszintézishez használt modellek és az abból származó megoldások minőségére.

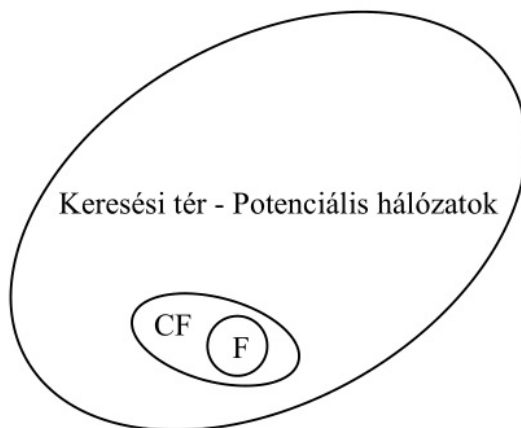
Bármely hagyományos algoritmikus módszer esetében egy vizsgált folyamat lehetséges hálózatstruktúrája „manuálisan”, folyamatos fejlesztés, javítások és módosítások eredményeként áll elő. A javítások és módosítások folyamata során egyrészt gyakori a heurisztikus és intuitív megoldások alkalmazása, másrészt pedig a kézzel felírt, minden lehetséges és nem lehetséges hálózatot tartalmazó teljes hálózat struktúrába beválasztott műveleti egységek közötti redundáns kapcsolatok megszüntetése. Mindkét megközelítésnek megvannak a maga nehézségei. A heurisztikus módszereknek a fent már említett buktatói mellett – ahogyan azt hamarosan majd látni fogjuk – a lehetséges műveleti egységek potenciális összekapcsolási lehetőségeinek nagy száma. Látható tehát, hogy heurisztikák alkalmazása még az algoritmikus módszerek esetében is gyakori.

Ahogyan azt korábban már említettük, a folyamatszintézis komplexitását a probléma kettős, egyidejűleg folytonos és diszkrét természete okozza. Az utóbbinak köszönhetően a probléma komplexitása exponenciálisan növekszik a rendelkezésre álló műveleti egységek számával, amit jelöljön n , mivel az optimális hálózatot 2^{n-1} lehetséges alternatíva közül kellene meghatározni mindaddig, amíg nem rendelkezünk olyan tudással a hálózat struktúrájáról, ami alapján helytelen alternatívák eltávolításával ezt a számot csökkenteni lehetne. A 2^{n-1} már viszonylag kis n esetén is hatalmas szám is lehet: $n = 35$ esetén 34.36×10^9 , míg $n = 36$ esetén már 68.72×10^9 . Ez a probléma exponenciális jellegét is jól illusztrálja, az n -t eggyel növelve az alternatívák száma lényegében megduplázódott. Olyan robusztus döntéstámogatási rendszerekre van tehát szükség a folyamatszintézis feladatok kombinatorikus komplexitásának megfelelő kezelésére, amelyek matematikailag szigorúak, lehetőleg axiomatikusak, és számítógépekre hatékonyan implementálhatóak.

Mindez a kombinatorika egy jól kidolgozott ágának, a gráfelmélet eredményeinek felhasználásával valósítható meg. Eredményül egy gráfelméletben gyökerező, algoritmikus módszert kapunk, amelyet ebben a fejezetben részletesebben is bemutatunk. A módszer a gráfok egy speciális osztályán alapszik, amely segítségével a folyamatok struktúrája egyértelműen reprezentálható, így lehetőség nyílik a gyakorlatban is megvalósítható folyamatokra jellemző kombinatorikus tulajdonságok kihasználására. Matematikailag az ilyen kombinatorikus tulajdonságok axiómaként fogalmazhatóak meg, amelyek három algoritmus alapjául szolgálnak [1, 3, 4]. A 2.2-es ábrán látható, hogyan csökkenti a módszer a keresési teret az optimális megoldás keresése során. A keresési tér, amely az összes lehetséges hálózatot tartalmazza, leszűkül azokra a hálózatokra, amelyek kielégítik az axiómákat, azaz az úgynevezett kombinatorikusan lehetséges hálózatokra, amit az ábrán CF jelöl. A két halmaz mérete között több nagyságrendbeli eltérés is lehet, amely a megoldáshoz igényelt számítási kapacitás drasztikus csökkenéséhez vezethet. A módszerrel megoldott gyakorlati példák során előfordult olyan eset is, amikor ez az arány egy az egymilliárdhoz volt. Itt érdemes talán megjegyezni, hogy a folyamathálózatok merőben eltérőek és lényegesen komplexebbek sok egyéb típusú (például tele-



2.1. ábra. A hagyományos algoritmikus módszerek főbb lépései



2.2. ábra. A keresési tér csökkentése

fon vagy autópálya) hálózaton az érintett műveleti egységek nagy számának, illetve a rajtuk keresztüláramló anyagok változatosságának tekintetében (például egy telefonhálózaton csak fotonok és elektronok haladnak át). Emellett a probléma duális, folytonos és kombinatorikus (egész) természete miatt egy folyamatszintézis feladat matematikai programozási modelljéül vagy egy MILP, vagy egy MINLP szolgál. Ezen dualitás mértéke eltérő a különböző folyamatszintézis feladatok, mint például az RNS (reaction-network synthesis, azaz reakcióhálózat-szintézis), SNS (separation-network synthesis, azaz szétválasztási hálózatok szintézise), HENS (heat-exchanger-network synthesis, azaz hőcserélő hálózatok szintézise) vagy PNS (process-network synthesis, azaz folyamathálózat-szintézis) esetében. Természeténél fogva a legutóbbi a leginkább kombinatorikus.

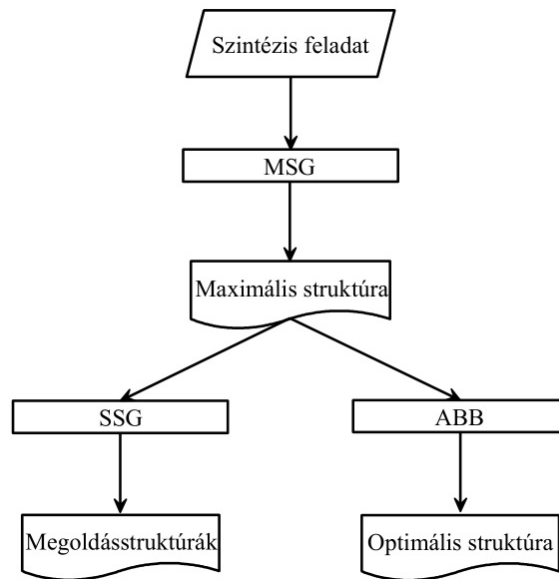
Ennek a gráfelméleti alapokon nyugvó, algoritmikus módszernek egy korai [1, 4], sikeres alkalmazása egy régebbi már létező Folpet folyamat újraszintetizálása, amely N-(trichloromethylthio) phtalimide előállítására szolgál. Ez egy gyomirtó, amelyet egészen az 1980-as évek végéig gyártottak, jelenleg azonban már nem alkalmazható a toxicitása miatt. Az újraszintetizálás a folyamat strukturális optimalizálását vizsgálta. A vizsgálat eredményeként kiderült, hogy a folyamatban résztvevő műveleti egységek közül 5 redundáns. A fennmaradó 35 műveleti egységből a módszer PC-re (PC/AT) implementált verziója kevesebb, mint egy perc alatt előállította a szigorú szuperstrukturát, 3456 kombinatorikusan lehetséges folyamathálózatot eredményezve.

Egy másik esetben azt vizsgálták a módszer felhasználásával, hogyan lehet egy meglévő mezőgazdasági komplexumot optimálisan átalakítani [7]. A komplexum 10 nagyobb üzemből és a hozzájuk kapcsolódó melléképületekből állt, amit a modellben 20 műveleti egység reprezentált. A feladathoz tartozó optimális hálózat meghatározása kevesebb, mint 5 másodpercet vett igénybe egy alacsony teljesítményű (Pentium III, 500 MHz) asztali számítógépen.

Az első esetben a 35 műveleti egységnek köszönhetően lehetséges $(2^{35} - 1)$ alternatíva közül kell megtalálni az optimálisat, ami 34539738647, a második esetben pedig a 20 műveleti egység miatt $(2^{20} - 1) = 1048575$. A lehetséges hálózatok nagy száma miatt a hagyományos algoritmikus módszerekkel az ilyen nagyságrendű feladatok már nem kezelhetők megfelelően. Az alábbiakban bemutatjuk a már említett, gráfelméleti alapokon nyugvó, algoritmikus módszert, amivel még az ilyen komplexitású gyártási folyamatok is hatékonyan kezelhetők és modellezhetők folyamatszintézis feladatként.

2.1. A P-gráf módszertan alapjai

A P-gráf módszertant az 1990-es évek elején dolgozták ki komplex vegyipari termelőrendszerek modellezésére és optimalizálására. A nevét egy irányított páros gráfról, a P-gráfról kapta, amely segítségével lehetőség nyílik a lehetséges megoldásstrukturák kombinatorikus tulajdonságainak kihasználására, ezáltal pedig a nagymértékű feladatok optimumának meghatározására is. A módszertan főbb lépései a 2.3-as ábrán láthatóak. A módszertanról jó áttekintést ad a következő könyv: [12]



2.3. ábra. A P-gráf módszertan főbb lépései

2.1.1. Alapfogalmak

Jelölje $\mathcal{M}$ azoknak az anyagoknak a halmazát, amelyek a modellezendő gyártási folyamatban definiálva vannak. $\mathcal{M}$ egy véges, nemüres halmaz. Az, hogy $\mathcal{M}$ elemeit milyen részletességgel adjuk meg, mindig az adott feladattól, illetve a modellezéstől függ. Például

$$\mathcal{M}_1 := \{A, B, C, D, E, F\}$$

és

$$\mathcal{M}_2 := \{(x_1, \emptyset, \emptyset), (\emptyset, x_2, \emptyset), (\emptyset, \emptyset, x_3), (x_1, x_2, \emptyset), (\emptyset, x_2, x_3), (x_1, x_2, x_3)\}$$

egyaránt megfelelő lehet, attól függően, hogy csak maguk az anyagok, vagy azok összetétele is lényeges.

Az $\mathcal{M}$ anyagaihoz kapcsolódó szintézis feladatot egy $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ hármassal adjuk meg, ahol $\mathcal{P}$ jelöli a termékek, $\mathcal{R}$ a nyersanyagok, $\mathcal{O}$ pedig a rendelkezésre álló, gyártás során felhasználható műveleti egységek halmazát. A $\mathcal{P}$ halmaz elemei azok az anyagok, amelyeket a folyamat során valamilyen szempont szerint optimálisan szeretnénk előállítani, az $\mathcal{R}$ halmaz elemei pedig azok az anyagok, amelyek kiinduláskor, a gyártás megkezdésekor rendelkezésünkre állnak. Matematikailag az $\mathcal{M}$, $\mathcal{P}$ és $\mathcal{R}$ halmazok közötti reláció a következőképpen fejezhető ki:

$$\mathcal{P} \subset \mathcal{M}, \mathcal{R} \subset \mathcal{M} \text{ és } \mathcal{P} \cap \mathcal{M} = \emptyset.$$

Az $\mathcal{O}$ halmaz elemei, azaz a műveleti egységek végzik az egyes anyagok közötti átalakításokat. Ennek megfelelően az $\mathcal{O}$ és az $\mathcal{M}$ halmaz kapcsolata:

$$\mathcal{O} \subseteq \wp(\mathcal{M}) \times \wp(\mathcal{M}).$$

Egy műveleti egységet matematikailag egy rendezett párral modellezünk, azaz $(\alpha, \beta) \in \mathcal{O}$, ahol α jelöli az egység bemenetei anyagainak, β pedig a kimenetei anyagainak a halmazát. A fentiek illusztrálására, azaz egy gyártási folyamat megadására tekintsük az alábbi hipotetikus példát:

$$\mathcal{M} := \{A, B, C, D, E, F, G, H, I, J, K\}$$

$$\mathcal{P} := \{A\}$$

$$\mathcal{R} := \{D, F, H, I\},$$

illetve

$$\mathcal{O} := \left\{ (\{B\}, \{A, E\}), (\{C\}, \{A, J\}), (\{D, E\}, \{B\}), (\{E, F\}, \{B\}), (\{F, G\}, \{C, K\}), \right. \\ \left. (\{H\}, \{E\}), (\{I, J\}, \{G\}) \right\} = \{O1, O2, O3, O4, O5, O6, O7\}$$

A fenti, 10 anyagot és 7 műveleti egységet tartalmazó példában az A anyagot szeretnénk gyártani a kiinduláskor rendelkezésre álló D, F, H, I anyagokból. Azokat az anyagokat, amelyek se nem nyersanyagok, se nem végtermékek, mint a fenti példában B, C, E, G, J nem szoktuk külön halmazként feltüntetni a feladat megadásakor.

Azokat az anyagokat, amelyek bizonyos műveleti egységek kimeneti, illetve bizonyos műveleti egységek bemeneti anyaghalmazának is elemei, köztes anyagoknak nevezzük. Tehát köztes anyagok azok az anyagok, amelyek kiinduláskor nem állnak rendelkezésünkre, azokat a gyártás során műveleti egységek állítják elő, illetve a termék előállításához fel is használják őket. A fenti B, C, E, G, J is ilyen, a G anyagot például az $O7$ műveleti egység gyártja és $O5$ használja fel.

Azokat az anyagokat, amelyeket a gyártás során műveleti egységek állítanak elő, de más műveleti egységek nem használják fel és nem elemei a $\mathcal{P}$ halmaznak sem, melléktermékeknek nevezzük. A fenti példában ilyen melléktermék a K anyag, amelyet $O5$ gyárt, de nem használja fel egyetlen más műveleti egység sem, továbbá nem eleme a termékek halmazának sem.

2.1.2. P-gráf

Annak érdekében, hogy a lehetséges, gyakorlatban is megvalósítható megoldások kombinatorikus tulajdonságait ki tudjuk használni az opcionális megoldás keresése során, a folyamat egyértelmű strukturális reprezentációja szükséges. A gyakorlatban általában használt hagyományos gráfok erre alkalmatlanok, az egyértelmű reprezentációhoz egy speciális, irányított páros gráf, az úgynevezett P-gráf (Process-graph, P-graph) szükséges. Egy gráf akkor páros, ha a csúcshalmaza particionálható két diszjunkt halmazra úgy, hogy az azonos halmazban lévő csúcsok közül egymással semelyik kettő nem szomszédos. A folyamat struktúrájának reprezentálása során is ezt a tulajdonságot használjuk ki, mivel alapvetően két anyag, illetve műveleti egység típusú csúcsot különböztetünk meg és két azonos típusú csúcs nem kapcsolódhat egymáshoz közvetlenül.

A P-gráf matematikai definíciója a következő:

Legyen m és o két véges halmaz, amelyekre

$$o \subseteq \wp(m) \times \wp(m).$$

Ekkor a P-gráf egy olyan (m, o) pár, ahol a gráf csúcsai a

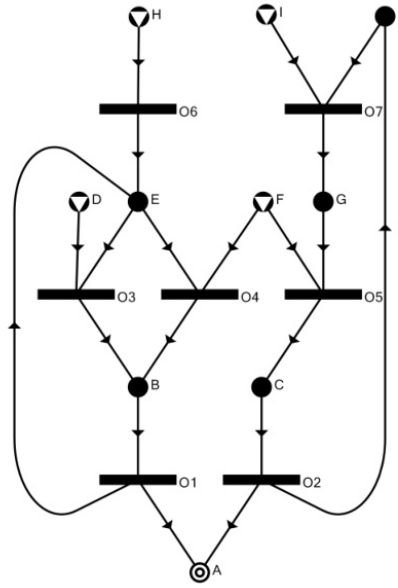
$$V = m \cup o$$

halmaz elemei. Az m halmaz elemei az anyag típusú csúcsok, az o halmaz elemei pedig a műveleti egység típusú csúcsok. A gráf élei az

$$A = A_1 \cup A_2$$

halmaz elemei, ahol

$$A_1 := \left\{ (x, y) \mid y = (\alpha, \beta) \in o \text{ és } x \in \alpha \right\}$$



2.4. ábra. A hálózatszintézis-feladathoz tartozó P-gráf

és

$$A_2 := \{(x, y) | y = (\alpha, \beta) \in o \text{ és } x \in \beta\}$$

A fenti formális definícióban x jelöli az anyag típusú csúcsokat, y a műveleti egység típusú csúcsokat, α jelöli azon anyag típusú csúcsok halmazát, amelyekből mutat irányított él a műveleti egység típusú csúcsokba, β pedig azon anyag típusú csúcsok halmazát, amelyekbe mutat irányított él a műveleti egység típusú csúcsokból. Más szavakkal mondva, az A_1 élhalmaz minden eleme anyag típusú csúcsokból műveleti egység típusú csúcsokba mutat, míg az A_2 élhalmaz minden eleme műveleti egység típusú csúcsból mutat anyag típusú csúcsba.

Két P-gráf, (m_1, o_1) és (m_2, o_2) unióját és metszetét, melyek szintén P-gráfok, a következőképpen definiáljuk:

$$(m_1, o_1) \cup (m_2, o_2) = (m_1 \cup m_2, o_1 \cup o_2)$$

$$(m_1, o_1) \cap (m_2, o_2) = (m_1 \cap m_2, o_1 \cap o_2)$$

Továbbá az m_1, o_1 gráf az m_2, o_2 gráf részgráfja, azaz

$$(m_1, o_1) \subseteq (m_2, o_2)$$

ha

$$m_1 \subseteq m_2 \text{ és } o_1 \subseteq o_2.$$

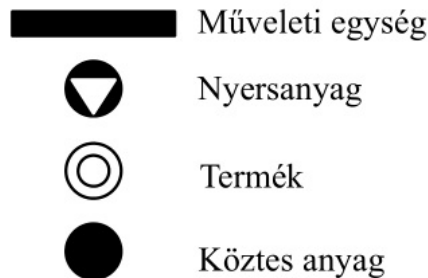
Egy folyamat struktúrája definiálható P-gráfként. Legyen $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ egy szintézis feladat, legyen m az $\mathcal{M}$ anyaghalmaz részhalmaza, o pedig az $\mathcal{O}$ műveleti egység részhalmaza, továbbá tegyük fel, hogy $o \subseteq \wp(m) \times \wp(m)$ teljesül. Ekkor a rendszer struktúrája az m anyaghalmaz, és az o műveleti egység halmaz felhasználásával az (m, o) P-gráffal formálisan definiálható. A definícióból közvetlenül következnek az alábbi tulajdonságok:

1. Az anyagokat a gráf anyag típusú, a műveleti egységeket a műveleti egység típusú csúcsai reprezentálják.
2. Egy anyag és egy műveleti egység típusú csúcs között akkor és csak akkor megy él, ha a

megfelelő anyag és a műveleti egység kapcsolata része a reprezentálandó folyamatnak.

3. Az élek irányai megegyeznek a folyamat előrehaladásának irányával.

Az anyag típusú csúcsokat körökkel, a műveleti egységek típusúakat vízszintes téglalapokkal ábrázoljuk. A fenti példához tartozó P-gráf a 2.4-es ábrán látható. A különböző anyag típusú csúcsok reprezentáció, mint nyersanyag, termék, vagy melléktermék a 2.5-ös ábrán láthatóak.



2.5. ábra. Szimbólumok

2.1.3. Kombinatorikusan lehetséges megoldásstruktúrák

A P-gráf segítségével nem csak a rendszer szintaktikája, hanem a szemantikája is kifejezhető. A gyakorlatban egy valós folyamat struktúrája nem írható le egy tetszőleges P-gráffal: egy valós folyamat megfelelő reprezentálásához minden P-gráfnak teljesítenie kell bizonyos kombinatorikus tulajdonságokat. Ezeket a tulajdonságokat alapulva különböző axiómákat fogalmazhatunk meg.

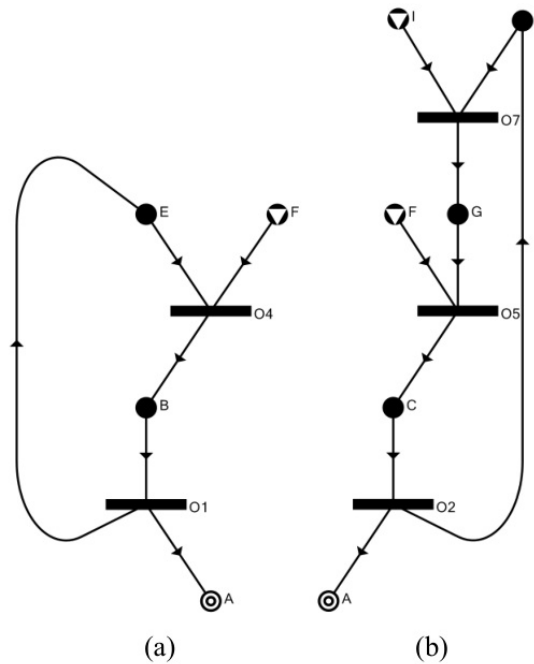
Vannak bizonyos kombinatorikus tulajdonságok, amelyeket egy megoldásban szereplő műveleti egységeknek és nyersanyagoknak mindig teljesíteniük kell. Például ha egy struktúrában nincs kapcsolat a termékek és a nyersanyagok között, akkor a struktúra alkalmatlan gyakorlati folyamatok reprezentálására. Kulcsfontosságú tehát azoknak az általános kombinatorikus tulajdonságoknak a megtalálása, amelyekkel egy folyamat struktúrájának rendelkeznie kell. Az így összegyűjtött tulajdonságokkal minden, a szintézis feladat lehetséges megoldásaihoz tartozó struktúrának rendelkeznie kell. Azok, és csakis azok a struktúrák lehetnek lehetséges megoldásstruktúrák, amelyek ezek a tulajdonságokkal rendelkeznek; semmilyen egyéb struktúrát vagy feltételt nem kell figyelembe venni az optimális megoldás keresése során.

2.1.4. Axiómák

Az alábbi axiómák [2] fogalmazzák meg azokat a tulajdonságokat, amelyekkel egy valós, gyakorlatban megvalósítható folyamatot reprezentáló struktúrának rendelkeznie kell:

- (S1) Minden legyártandó termék, azaz $\mathcal{P}$ minden eleme szerepel a struktúrában.
- (S2) Egy a struktúrában szereplő anyag akkor és csak akkor nyersanyag, ha egyetlen a struktúrában szereplő műveleti egység sem állítja elő.
- (S3) Minden a struktúrában szereplő műveleti egység a szintézis feladatban definiált.
- (S4) Minden a struktúrában szereplő műveleti egységből vezet út legalább egy legyártandó termékhez.
- (S5) Ha egy x anyag része a struktúrának, akkor létezik a struktúrában olyan műveleti egység, amely x anyagot felhasználja vagy előállítja.

Önmagában mindegyik axióma triviálisnak tűnik, azonban azokat egyszerre alkalmazva kiszűrhetők a kombinatorikusan nem megfelelő, gyakorlati alkalmazások modellezésére használhatatlan há-



2.6. ábra. Két kombinatorikusan lehetséges megoldásstruktúra

lózatok. Ha egy szintézis feladathoz tartozó P-gráf kielégíti ezeket az axiómákat, akkor a P-gráfot a probléma egy megoldásstruktúrájának mondjuk. Tekintsük az alábbi szintézis feladatot a megoldásstruktúrák koncepciójának az illusztrálására:

$$\begin{aligned}\mathcal{M} &:= \{A, B, C, D, E, F, G, H, I\} \\ \mathcal{P} &:= \{A\} \\ \mathcal{R} &:= \{D, F, H\} \\ \mathcal{O} &:= \left\{ (\{C\}, \{A, I\}), (\{B\}, \{A, E\}), (\{D, E\}, \{B\}), (\{E, F\}, \{B\}), (\{F, G\}, \{C\}), \right. \\ &\quad \left. (\{H, I\}, \{G\}) \right\}\end{aligned}$$

A feladathoz tartozó két különböző megoldásstruktúra látható a 2.6-os ábrán. Érdekes megjegyezni, hogy ha egy műveleti egység típusú csúcs része egy megoldásstruktúrát reprezentáló P-gráfnak, akkor a műveleti egység bemeneti és kimenetei anyaghalmazainak összes eleme is a gráfhoz tartozik. Az is említést érdemel, hogy egy megoldásstruktúra nem tartalmazza feltétlenül az $\mathcal{M}$ anyaghalmaz összes elemét, illetve az $\mathcal{R}$ nyersanyaghalmazból sem kell feltétlenül mindent felhasználnia a gyártás során.

Mivel a gyártandó termék, A , mindkét struktúrában szerepel, az első axióma mindkét esetben teljesül. A második axióma az első struktúrában az F , a második struktúrában az F és G csúcsok miatt teljesül, mivel nyersanyagként egyedül ezekbe a csúcsokba nem vezet él műveleti egységekből, azaz ezeket az anyagokat nem gyártja semmi. Az első struktúra két, a második pedig három műveleti egységet tartalmaz, amelyek mindegyike definiálva van a szintézisfeladatban, így a harmas axióma is teljesül. A négyes axiómának megfelelően, mindkét struktúrában minden műveleti egység típusú csúcsból vezet út a leggyártandó termékhez. Az ötös axióma is teljesül, mivel mindkét struktúrában minden egyes anyag típusú csúcs legalább egy műveleti egységnek a kimenete, vagy a bemenete. Ez a két struktúra tehát kielégíti az összes axiómát, szemben a 2.7-es ábrán látható struktúrával, amelyre az egyes, kettes, négyes és ötös axióma sem teljesül.

Legyen $S(\mathcal{P}, \mathcal{R}, \mathcal{O})$ az a halmaz, amely a $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ szintézisfeladat összes megoldásstruktúráját tartalmazza. Ekkor a megoldásstruktúrákra vonatkozóan az alábbi tételt fogalmazhatjuk meg:

2.1. Tétel. *A megoldásstruktúrák zártak az unióra, azaz két megoldásstruktúra uniója is megoldás-*

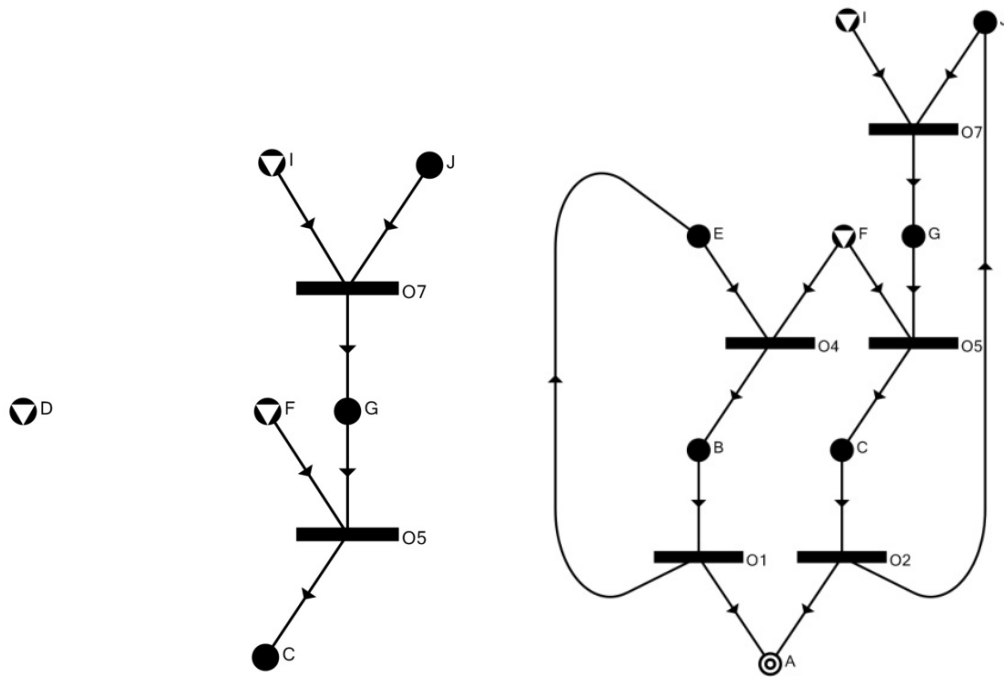
struktúra. Formálisan: ha

$$\sigma_1 \in S(\mathcal{P}, \mathcal{R}, \mathcal{O}) \text{ és } \sigma_2 \in S(\mathcal{P}, \mathcal{R}, \mathcal{O})$$

akkor

$$(\sigma_1 \cup \sigma_2) \in S(\mathcal{P}, \mathcal{R}, \mathcal{O}).$$

Az előző példa két megoldásstruktúrájának unióját ábrázolja a 2.8-as ábra. Mint az látható, a megoldásstruktúrák uniója maga is egy megoldásstruktúra.



2.7. ábra. Ez a struktúra nem elégíti ki az axiómákat

2.8. ábra. A 2.6-os ábrán látható struktúrák uniója

2.1.5. Maximális struktúra

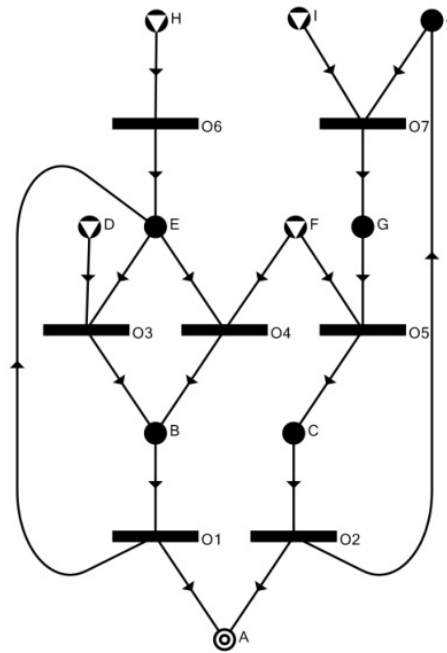
Mivel a megoldásstruktúrák halmaza véges és zárt az unióra, a halmaznak lesz egy eleme, $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O})$, amely az összes megoldásstruktúra uniója, azaz

$$\mu(\mathcal{P}, \mathcal{R}, \mathcal{O}) = \bigcup_{\sigma \in S(\mathcal{P}, \mathcal{R}, \mathcal{O})} \sigma$$

feltéve, ha a megoldásstruktúrák halmaza nem üres, azaz $S(\mathcal{P}, \mathcal{R}, \mathcal{O}) \neq \emptyset$. Ekkor a halmaz $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O})$ szintézisfeladat maximális struktúrájának nevezzük. A szakirodalomban a maximális struktúrát gyakran nevezik szuperstruktúrának is. A maximális struktúra minden csúcsa és éle része legalább egy megoldásstruktúrának, illetve minden megoldásstruktúra P-gráfja a maximális struktúrát reprezentáló P-gráfnak. A fent bevezetett példához tartozó maximális struktúra a 2.9-es ábrán látható.

2.2. Tétel. A maximális struktúra maga is megoldásstruktúra, azaz $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O}) \in S(\mathcal{P}, \mathcal{R}, \mathcal{O})$.

A maximális struktúra a feladat összes lehetséges megoldását tartalmazza, így az optimálisat is. Tulajdonképpen a maximális struktúra meghatározásával lecsökkentjük a keresési teret, mivel az optimá-



2.9. ábra. A szintézisfeladathoz tartozó maximális struktúra

lis megoldás meghatározása során elegendő csupán a kombinatorikus lehetséges megoldásstruktúrákat megvizsgálnunk. A korábban már említett, 35 műveleti egységes ipari példa esetében ez a csökkenés több, mint 99,99%-os, mivel 34,539,738,647 helyett elegendő csupán 3465 lehetséges alternatívát megvizsgálni.

2.1.6. Az MSG algoritmus

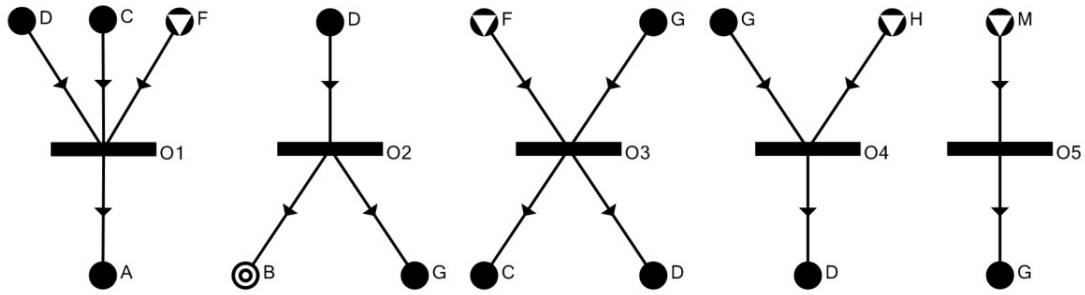
A szintézis feladathoz tartozó maximális struktúrát a P-gráf módszertan segítségével határozzuk meg. Az algoritmus a fent már ismertetett axiómákon alapszik, és számítógépes implementációja az alábbi négy főbb lépésből áll.

Az első lépésben definiáljuk magát a $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ szintézis-feladatot az $\mathcal{M}$ anyaghalmaz, a $\mathcal{P}$ termék-halmaz, az $\mathcal{R}$ nyersanyag-halmaz, illetve az $\mathcal{O}$, a lehetséges műveleti egységek halmazának megadásával. Az $\mathcal{M}$ halmaznak nem csak a köztes anyagokat, de a termékeket és a nyersanyagokat, azaz $\mathcal{P}$ és $\mathcal{R}$ elemeit is tartalmaznia kell.

A második lépésben meghatározzuk egy kezdeti struktúrát, vagy kiindulási hálózatot úgy, hogy összekapcsoljuk a műveleti egységeket a hozzájuk tartozó közös anyag típusú csúcsok mentén.

A harmadik lépésben eltávolítjuk a hálózatból azokat az anyagokat és műveleti egységeket, amelyek nem lehetnek részei a maximális struktúrának, mert megsértik az axiómák valamelyikét. Ez az algoritmus redukciós szakasza. Kikerülnek azok a műveleti egységek, amelyek nyersanyagokat gyártanak, illetve azok az anyagok, amelyek nem számítanak nyersanyagoknak, de mégsem állítja elő őket semmi. Ez iteratívan történik, mivel előfordulhat, hogy bizonyos anyagok és műveleti egységek eltávolítása újabb anyagok és műveleti egységek eltávolítását vonja maga után.

A végső fázis az algoritmus építő szakasza. Ebben a fázisban lépésről lépésre építjük fel a hálózatot a termékektől kiindulva először azokat a műveleti egységeket, amelyek ezek bemeneteit gyártják, egészen addig, amíg el nem érünk a nyersanyagokig. Ha szemléletesen szeretnénk fogalmazni, akkor a lebontás felülről lefelé, a nyersanyagoktól a termékek felé haladva, az építés pedig alulról felfelé, azaz a termékektől a nyersanyagok felé haladva történik. A szuperstruktúrát persze nem elég önmagában meghatározni, hanem lehetőleg minél hatékonyabban szeretnénk ezt megtenni. Az MSG ennek is eleget tesz, mivel a futási ideje polinomiális. Az MSG algoritmus működését az



2.10. ábra. A példához tartozó műveleti egységek

alábbi hipotetikus példán szemléltetjük:

$$\mathcal{M} = \{A, B, C, D, E, F, G, H, L, M\}$$

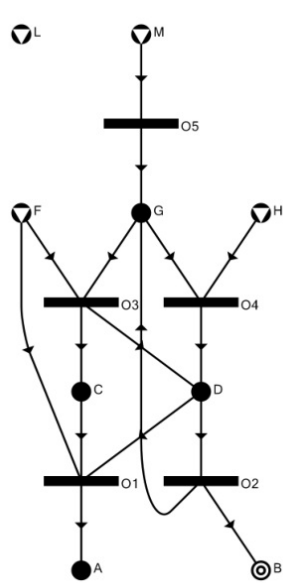
$$\mathcal{P} = \{B\}$$

$$\mathcal{R} = \{F, H, L, M\}$$

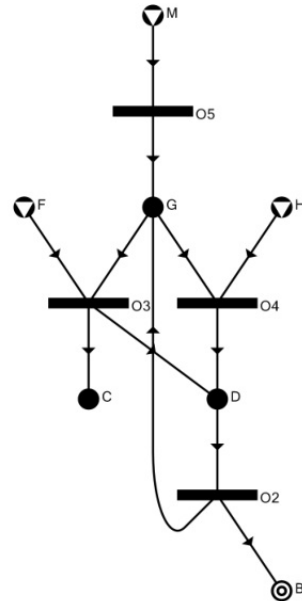
$$\mathcal{O} = \left\{ (\{C, D, F\}, \{A\}), (\{D\}, \{B, G\}), (\{F, G\}, \{C, D\}), (\{G, H\}, \{D\}), (\{M\}, \{G\}) \right\} = \\ = \{O1, O2, O3, O4, O5\}$$

A gyártás során a B terméket szeretnénk előállítani a rendelkezésre álló F, H, L, M nyersanyagokból. A definíciónak megfelelően $\mathcal{P}$ és $\mathcal{R}$ elemei szerepelnek az $\mathcal{M}$ halmazban is. A példában szereplő 5 műveleti egység látható a 2.10-es ábrán.

A kiindulási hálózatot a műveleti egységek közös kimeneteinek és bemeneteinek összekapcsolásával kapjuk. Ebben a példában $O3$ -at és $O1$ -et C -n keresztül $O3$ -at és $O4$ -et $O1$ -hez és $O2$ -hez D -n keresztül, végül $O2$ -t és $O5$ -öt $O3$ -hoz és $O4$ -hez G -n keresztül kapcsoljuk össze. Az eredményül kapott kiindulási hálózat a 2.11-es ábrán látható.



2.11. ábra. A szintézisfeladathoz tartozó kiindulási struktúra



2.12. ábra. A szintézisfeladathoz tartozó maximális struktúra

A redukciós szakaszban az L nyersanyagot kivesszük a struktúrából, mert sérti az ötös axiómát, illet-

ve az $O1$ műveleti egységet is, mert sérti a négyes axiómát. Az $O1$ műveleti egység eltávolításából következik, hogy az A anyag sem marad része a struktúrának, mivel nem lesz, ami előállítja. A többi anyag és műveleti egység nem sért egyetlen axiómát sem, így részei maradnak a struktúrának.

Az eredményül kapott struktúra a 2.12-es ábrán látható, amely megfelel a maximális struktúrának is, amit az algoritmus építő szakaszának végrehajtása után kapunk.

2.1.7. Az SSG algoritmus

Az MSG eredményeként kapott maximális struktúra tartalmazza az összes kombinatorikusan lehetséges megoldásstruktúrát, amelyek alkalmasak annak a gyártási folyamatnak a modellezésére, amely során a megadott termékeket állítjuk elő a rendelkezésre álló nyersanyagokból. A maximális struktúra tartalmazza többek között a megadott célfüggvény, általában a költség szerinti optimális megoldást is. Nincs azonban általánosan elfogadott szabály azzal kapcsolatban, melyik megoldás az optimális, például nem biztos, hogy az a megoldásstruktúra állítja elő a terméket minimális költséggel, amely a legkevesebb műveleti egységet tartalmazza, így ebben a fázisban még mindenképpen a megoldásstruktúrák további vizsgálata szükséges.

A további vizsgálathoz ad segítséget az SSG (Solution Structure Generation) algoritmus, amely minden kombinatorikusan lehetséges struktúrát pontosan egyszer generál. Az algoritmusnak többféle számítógépes megvalósítása is létezik.

Az egyik megvalósítás a döntési leképezéseken alapszik, az alábbiakban ennek a vázlatát ismertetjük. A döntési leképezések során arról döntünk, hogy mely anyagot mely műveleti egységgel vagy egységekkel gyártunk, azaz mely műveleti egységeket vonjuk be egy adott megoldásstruktúrába. Ebből következőleg a döntési leképezések során arról is döntünk, mely műveleti egységeket zárjuk ki az adott struktúrából. A döntések során ügyelnünk kell a konzisztenciára is, ha egy műveleti egységről már döntöttünk egy anyagra vonatkozóan, hogy nem kerül be a struktúrába, akkor egy másik anyagra vonatkozó döntés során már nem választhatjuk be. Egy műveleti egységnek, ha szerepel a struktúrában, minden kimeneti anyagát elő kell állítania; egy inkonzisztens döntés azt eredményezné, hogy bizonyos anyagokat előállít, bizonyos anyagokat pedig nem. Ez a gyakorlatban nem megengedett, például sok vegyipari anyag előállításakor keletkeznek káros melléktermékek, amelyek a gyártás velejárói. Az SSG algoritmus döntési leképezésen alapuló megvalósítása rekurzívan hívja magát. Az algoritmus működését az alábbi hipotetikus példán illusztráljuk:

$$\begin{aligned}\mathcal{M} &= \{A, B, C, D, E, F, G\} \\ \mathcal{P} &= \{A\} \\ \mathcal{R} &= \{E, F\} \\ \mathcal{O} &= \left\{ (\{C\}, \{A, G\}), (\{D\}, \{A, B\}), (\{E, G\}, \{C\}), (\{F, C\}, \{D\}) \right\} = \{O1, O2, O3, O4\}\end{aligned}$$

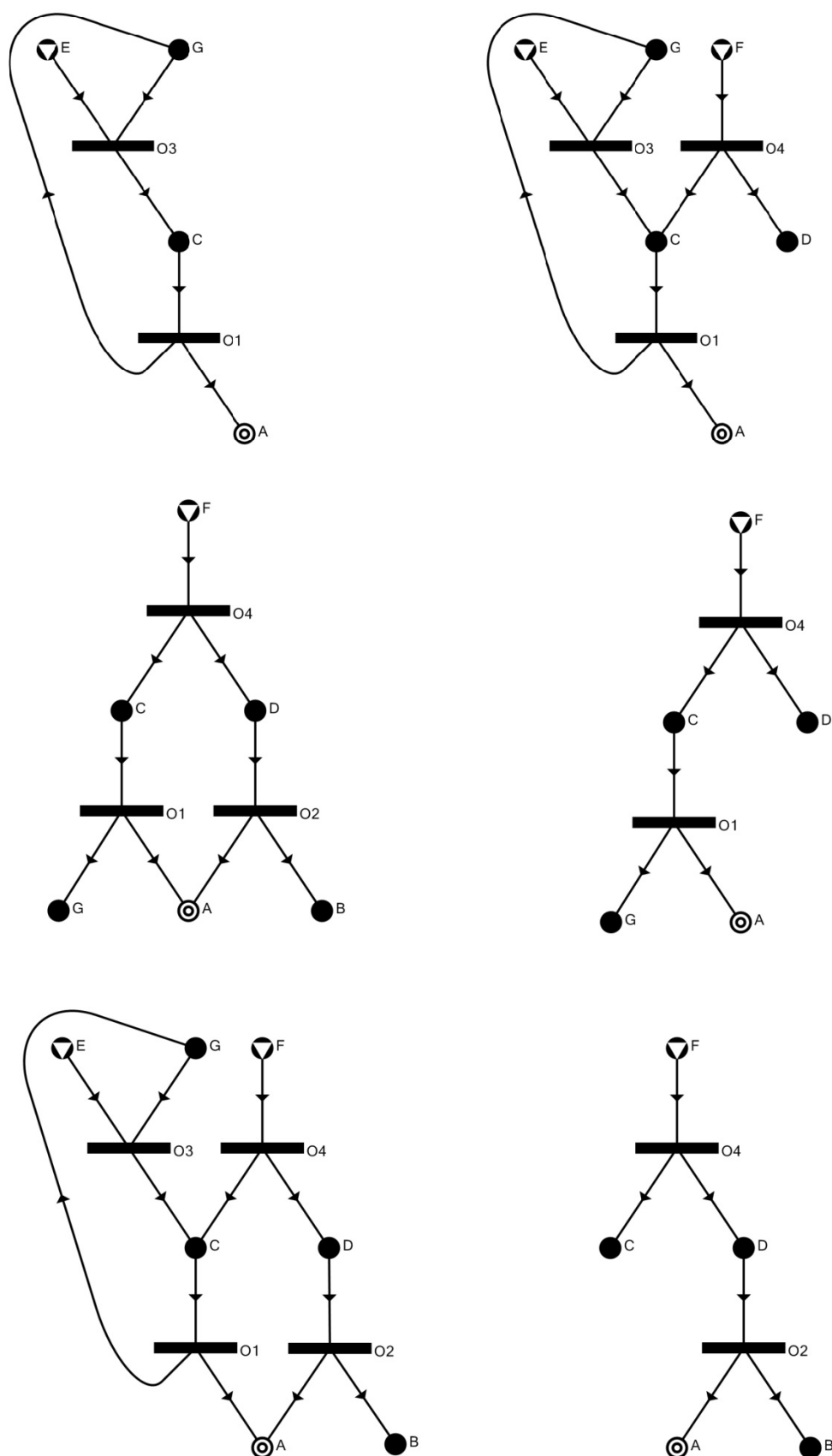
Az SSG a megadott szintézisfeladat összes kombinatorikusan lehetséges megoldásstruktúráját, köztük a maximális struktúrát, amely maga is megoldásstruktúra, pontosan egyszer generálja. A generált megoldásstruktúrák a 2.13-as ábrán láthatóak.

2.2. Formális jelölések és algoritmusok

Ahhoz, hogy matematikai értelemben szigorúan definiálni tudjuk a felsorolt algoritmusokat be kell vezetnünk néhány jelölést.

- $\psi^-(o)$ jelölje az o halmazban lévő műveleti egységek bemeneti anyagainak a halmazát:

$$\psi^-(o) = \bigcup_{(\alpha, \beta) \in o} \alpha;$$



2.13. ábra. Az SSG által generált megoldás-struktúrák

- $\psi^+(o)$ jelölje az o halmazban lévő műveleti egységek kimeneti anyagainak a halmazát:

$$\psi^+(o) = \bigcup_{(\alpha, \beta) \in o} \beta;$$

- $\psi(o)$ jelölje az o halmazban lévő műveleti egységek be- és kimeneti anyagainak a halmazát:

$$\psi(o) = \psi^-(o) \cup \psi^+(o);$$

- $\varphi^-(m)$ jelölje az m halmazban lévő anyagokat előállító berendezések halmazát:

$$\varphi^-(m) = \{(\alpha, \beta) \in o : \beta \cap m \neq \emptyset\};$$

- $\varphi^+(m)$ jelölje az m halmazban lévő anyagokat felhasználó berendezések halmazát:

$$\varphi^+(m) = \{(\alpha, \beta) \in o : \alpha \cap m \neq \emptyset\};$$

- $\varphi(m)$ jelölje az m halmazban lévő anyagokat előállító és felhasználó berendezések halmazát:

$$\varphi(m) = \varphi^-(m) \cup \varphi^+(m).$$

Az egyszerűség kedvéért feltesszük, hogy a fenti függvényeket nem csak halmazokra, hanem egyetlen anyagra vagy műveleti egységre is meg lehet hívni. Ilyenkor az anyagot vagy a műveleti egységet egy egyelemű halmaznak tekintjük. Például $\psi^-(o_i)$ az o_i műveleti egység bemeneti anyagainak a halmazát adja vissza, azaz gyakorlatilag egy $\psi^-(\{o_i\})$ függvényhívást jelent.

Ahhoz, hogy egy $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ szintézisfeladathoz tartozó (m, o) P-gráf egy valós folyamatot írjon le öt kombinatorikus tulajdonságot kell teljesítenie, amelyeket már korábban megadtunk. Most az öt axióma formális jelölését is definiáljuk.

- (S1) Minden legyártandó termék szerepel a struktúrában:

$$\mathcal{P} \subseteq m;$$

- (S2) Egy a struktúrában szereplő anyag akkor és csak akkor nyersanyag, ha egyetlen a struktúrában szereplő műveleti egység sem állítja elő:

$$m \setminus \psi^-(o) = m \cap \mathcal{R};$$

- (S3) Minden a struktúrában szereplő műveleti egység a szintézis feladatban definiált:

$$o \in \mathcal{O};$$

- (S4) Minden a struktúrában szereplő műveleti egységből vezet út legalább egy legyártandó termékhez:

$$\forall o_i \in o \text{ esetén } \exists \text{path}[o_i, m_j], \text{ ahol } m_j \in \mathcal{P};$$

- (S5) Ha egy anyag része a struktúrának, akkor létezik a struktúrában olyan műveleti egység, amely az anyagot felhasználja vagy előállítja:

$$m \subseteq \psi(o).$$

Mint már írtuk, ezek az axiómák könnyen elfogadhatóak, de azért értelmezzük őket külön-külön.

- Az (S1) axióma azt mondja ki, hogy a P-gráf csak akkor írja le megfelelően a rendszert, ha az összes kötelezően előállítandó terméket tartalmazza. Természetesen, ha egy struktúrában nincs benne az összes termék, akkor a feladatnak nincs olyan megoldása, amelyben mindegyik előállításra kerülne, azaz nem megvalósítható a rendszer.
- Az (S2) axióma szerint nyersanyagot nem állíthatunk elő és amit nem állítunk elő az nyersanyag. Azaz anyagok nem keletkeznek előállítás nélkül, csak ha meg tudjuk venni őket (nyersanyagok), valamint olyan anyagot, amit elő tudunk állítani, azt nem vesszük meg. Ez utóbbi feltétel olyan esetekben, amikor nem tudunk megfelelő mennyiségű anyagot előállítani, akkor problémát okozhat, de kezelhető ez az eset egy új nyersanyagcsomópont felvételével, amiből egy „vásárlás” műveleti egységgel elő tudjuk állítani a gyártásban részt vevő anyagot.
- Az (S3) axióma azt fejezi ki, hogy nem alkalmazhatunk a tervezés során olyan műveleti egységeket, amelyek a feladatban nem definiáltak. Ez azt is magában foglalja, hogy minden műveleti egységnek egyetlen bemeneti vagy kimeneti anyaga sem hagyható el és újat sem rakhatunk hozzá, mivel az már egy másik műveleti egység lenne.
- Az (S4) axióma szerint minden, egy megoldásstruktúrában részt vevő műveleti egységnek valamilyen hasznos munkát kell végeznie, azaz a termék előállításának folyamatában részt kell vennie. Ha egy műveleti egységből nem vezet út termékhez, akkor azt ki lehet hagyni a struktúrából anélkül, hogy a termék előállítása csorbát szenvedne, és ezzel csökkenteni lehet a rendszer költségét.
- Az (S5) axióma szerint nincsenek a megoldásban izolált anyagcsomópontok. Ezek az anyagok valójában nem növelik a rendszer költségét és a megoldhatóságot sem befolyásolják, viszont redundáns megoldásokat eredményeznek.

Azokat a struktúrákat, amelyek mind az öt axiómát teljesítik kombinatorikusan lehetséges megoldásstruktúrának (vagy röviden megoldásstruktúrának) nevezzük. Továbbá semmilyen más struktúrát nem tekintünk megoldásstruktúrának.

2.2.1. Maximális struktúra generálása

A szintézis feladathoz tartozó maximális struktúrát az MSG (Maximal Structure Generation) algoritmus segítségével lehet meghatározni. Első lépésként definiálni kell a szintézisfeladatot az $\mathcal{M}$, $\mathcal{P}$, $\mathcal{R}$ és $\mathcal{O}$ halmazok megadásával.

Az MSG algoritmus pszeudokódja a 2.14-es ábrán látható. Az algoritmus működésének bemutatásához tekintsük a következő példát, amely az előzőekben megadott módon definiált az $\mathcal{M}$, $\mathcal{P}$, $\mathcal{R}$ és $\mathcal{O}$ halmazokkal. Az ezen adatok alapján felépített kiindulási hálózat a 2.15-ös ábrán látható.

$$\begin{aligned}\mathcal{M} &= \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, Q, T, U, V\} \\ \mathcal{P} &= \{B\} \\ \mathcal{R} &= \{F, H, M, T\} \\ \mathcal{O} &= \left\{ O1 = (\{C, D, F\}, \{A\}), O2 = (\{D\}, \{B, G\}), O3 = (\{E\}, \{B, U\}), O4 = \right. \\ &= (\{F, G\}, \{C, D\}), O5 = (\{G, H\}, \{D\}), O6 = (\{H, I\}, \{E\}), O7 = (\{J, K\}, \{E\}), O8 = \\ &= (\{M\}, \{G\}), O9 = (\{N, Q\}, \{H\}), O10 = (\{T, U\}, \{I\}), O11 = (\{V\}, \{J\}) \left. \right\}\end{aligned}$$

Nézzük először a redukciós rész működését. Itt első lépésben kivesszük a műveleti egységek közül azokat, amelyek nyersanyagot állítanak elő ($\mathcal{O} := \mathcal{O} \setminus \varphi^-(\mathcal{R})$), mivel ezek megsértik az (S2) axiómát. A példánkban az $O9$ műveleti egység előállítja a H nyersanyagot, ezért ezt a műveleti egységet nem vesszük figyelembe. Az eredményül kapott P-gráf a 2.16-os ábrán látható, ahol a nyersanyagokat piros színnel is megjelöltük.

```

begin
  # redukciós rész
   $\mathcal{O} := \mathcal{O} \setminus \varphi^-(\mathcal{R});$ 
   $\mathcal{M} := \psi(\mathcal{O});$ 
   $r := \psi^-(\mathcal{O}) \setminus (\psi^+(\mathcal{O}) \cup \mathcal{R});$ 
  while  $r \neq \emptyset$  do
    legyen  $x \in r;$ 
     $\mathcal{M} := \mathcal{M} \setminus \{x\};$ 
     $o := \varphi^+(\{x\});$ 
     $\mathcal{O} := \mathcal{O} \setminus o;$ 
     $r := (r \cup (\psi^+(o) \setminus \psi^+(\mathcal{O}))) \setminus \{x\};$ 
  if  $\mathcal{P} \cap \mathcal{M} \neq \mathcal{P}$  then
    stop;

  # kompozíciós rész
   $p := \mathcal{P}; m := \emptyset; o := \emptyset;$ 
  while  $p \neq \emptyset$  do
    legyen  $x \in p;$ 
     $m := m \cup \{x\};$ 
     $o_x := \varphi^-(\{x\});$ 
     $o := o \cup o_x;$ 
     $p := (p \cup \psi^-(o_x)) \setminus (\mathcal{R} \cup m);$ 
     $m := \psi(o);$ 
end

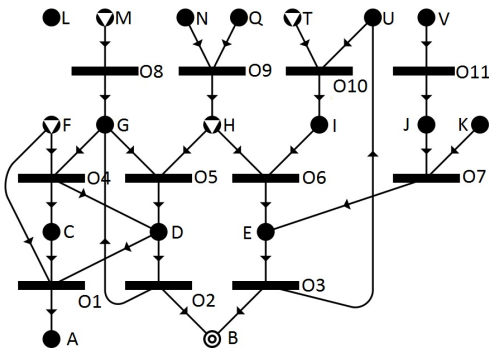
```

2.14. ábra. MSG algoritmus

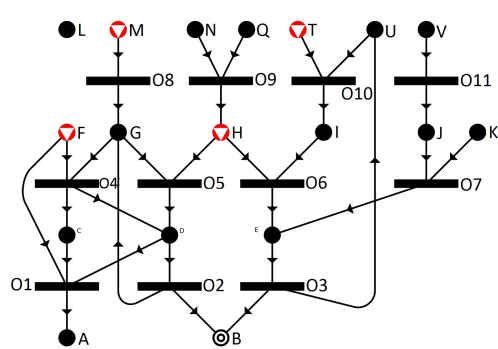
A redukciós rész következő lépésében azok az anyagok kerülnek eltávolításra, amelyek nem bemenetei és nem kimenetei egyetlen műveleti egységnek sem ($\mathcal{M} := \psi(\mathcal{O})$), azaz megsértik az (S5) axiómát. Ezek az L , N és Q anyagok a példában. Az eredményül kapott P-gráf a 2.17-es ábrán látható.

Az algoritmus ezek után egy r halmazon dolgozik, amely azokat az anyagokat tartalmazza, amelyeket valamely műveleti egység felhasznál, de nem állítja elő semmi, és nem nyersanyag ($r := \psi^-(\mathcal{O}) \setminus (\psi^+(\mathcal{O}) \cup \mathcal{R})$), azaz ezek megsértik az (S4) feltételt. A halmaz az algoritmus futása során folyamatosan változik, mivel egy műveleti egység kizárása új ilyen anyagokat eredményezhet. A továbbiakban az ábrákon piros színnel jelöljük azokat az anyagcsomópontokat, amelyek az r halmazban vannak. Az első iteráció során a K és a V anyag tartozik a halmazba (2.18-as ábra).

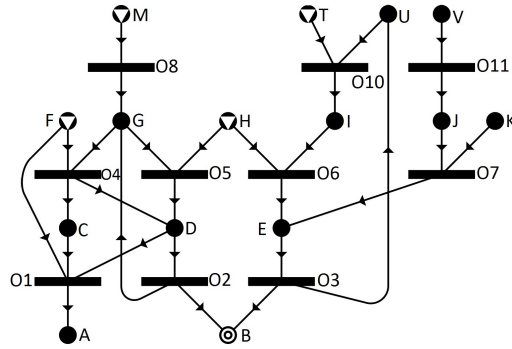
Az iteráció során először távolítsuk el a hálózattól a K anyagcsomópontot ($\mathcal{M} := \mathcal{M} \setminus \{x\}$), és az őt felhasználó $O7$ műveleti egységet ($o := \varphi^+(\{x\}); \mathcal{O} := \mathcal{O} \setminus o$). Az $O7$ műveleti egység eltávolítása nem változtatja meg az r halmazt, mivel az általa előállítható anyagot (E) másik műveleti egység is elő tudja állítani ($O6$), de a K anyagcsomópontot kivesszük a halmazból ($r := (r \cup (\psi^+(o) \setminus \psi^+(\mathcal{O}))) \setminus \{x\}$).



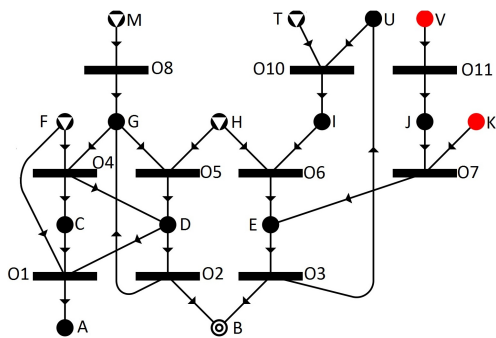
2.15. ábra. MSG példa: kiinduló struktúra



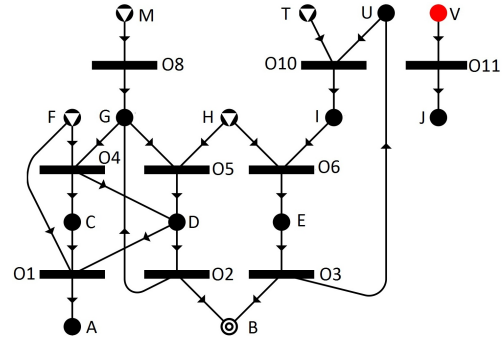
2.16. ábra. MSG példa redukciós rész: első lépés után



2.17. ábra. MSG példa redukciós rész: második lépés után



2.18. ábra. MSG példa redukciós rész: (S4) axiómát megsértő anyagsomópontok



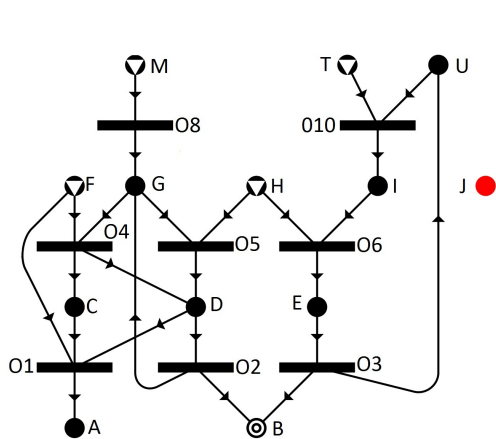
2.19. ábra. MSG példa redukciós rész: első iteráció után

Az első iteráció utáni hálózat a 2.19-es ábrán látható.

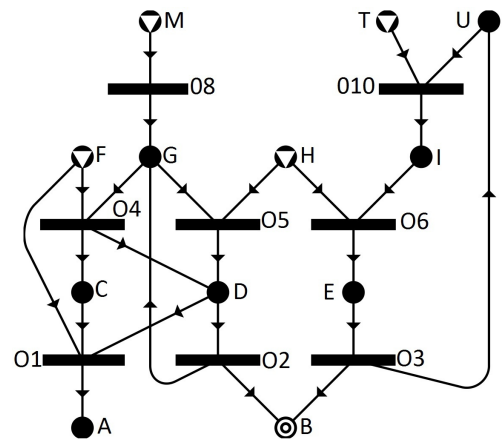
Az iteráció második lépésében távolítsuk el a hálózatból a V anyagsomópontot és az őt felhasználó $O11$ műveleti egységet. Az r halmazból eltávolítjuk a V anyagsomópontot, de bekerül helyette az $O11$ műveleti egység által előállított J anyag, mivel azt másik műveleti egység nem képes gyártani. A második iteráció utáni hálózat a 2.20-as ábrán látható.

Az iteráció harmadik lépésében távolítsuk el a hálózatból a J anyagsomópontot. Mivel J -t nem használja fel a hálózat egyik műveleti egysége sem, ezért ebben a lépésben műveleti egységet nem távolítunk el, de az r halmazból kivesszük J -t. Az iteráció utáni hálózat a 2.21-es ábrán látható.

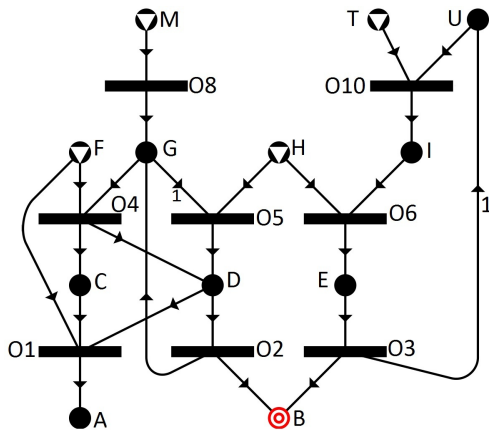
Mivel az r halmaz kiürült, ezért az iterációknak vége van. A redukciós rész utolsó lépéseként meg kell vizsgálni, hogy az összes termék továbbra is része-e a hálózatnak (**if** $\mathcal{P} \cap \mathcal{M} \neq \mathcal{P}$ **then stop**), azaz



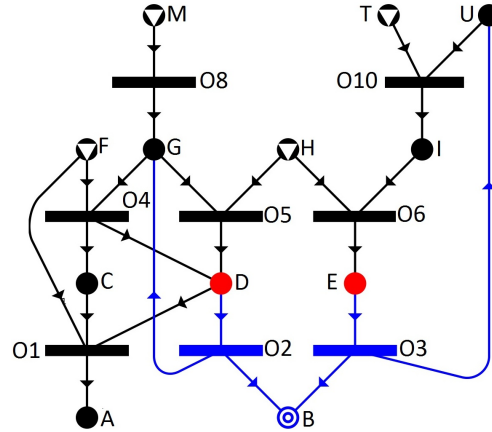
2.20. ábra. MSG példa redukciós rész: második iteráció után



2.21. ábra. MSG példa: redukciós rész eredménye



2.22. ábra. MSG példa kompozíciós rész: kiinduló struktúra



2.23. ábra. MSG példa kompozíciós rész: az első iteráció után

teljesíti-e az (S1) axiómát. Mivel ez teljesül, ezért továbbléphetünk a kompozíciós részre, melynek bemenete a redukciós rész eredménye lesz.

A kompozíciós részben a termékek felől haladunk a nyersanyagok felé, a struktúrába bevesszük azokat a műveleti egységeket, amelyekből elérhetőek a termékek. A kompozíciós részben három halmazt használunk a maximális struktúra generálásához.

- A p halmaz azokat az anyagcsomópontokat tartalmazza, amelyeket elő kell állítani. Ez a kiindulásnál megegyezik a termékek halmazával ($p := \mathcal{P}$). A lépéseket bemutató ábrákon a p halmaz elemeit piros színnel jelöljük.
- Az m halmaz azokat az anyagcsomópontokat tartalmazza, amelyeket már megvizsgáltunk és előállítottunk. Természetesen ez a halmaz az algoritmus indulásakor üres ($m := \emptyset$). Ezeket a csomópontokat az ábrákon kék színnel jelöljük.
- Az o halmaz azokat a műveleti egységeket tartalmazza, amelyeket már beválasztottunk a maximális struktúrába. Induláskor ez a halmaz is üres ($o := \emptyset$).

Mint már korábban leírtuk, a p halmaz megegyezik a termékek halmazával. A kompozíciós rész kiinduló struktúrája a 2.22-es ábrán látható, ahol a B termék pirossal meg van jelölve, mint a p halmaz eleme.

A kompozíció egy iterációjában a p halmazból kiválasztunk egy x anyagot és

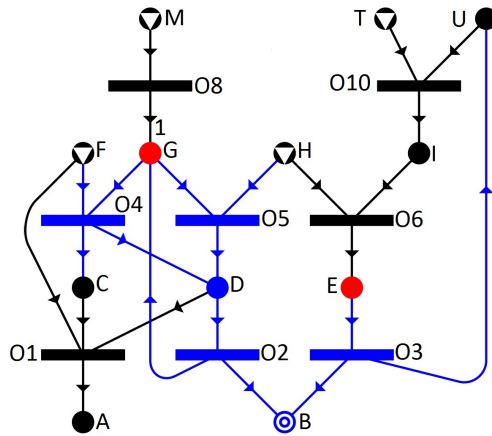
- x -et berakjuk az m halmazba ($m := m \cup \{x\}$),
- berakjuk az x -et előállítani képes műveleti egységeket ($o_x := \varphi^{-}(\{x\})$) az o halmazba ($o := o \cup o_x$),
- az újonnan bevett műveleti egységek bemeneti anyagai közül azokat, amelyek nem nyersanyagok és még nem vizsgáltuk őket berakjuk az előállítandó anyagok közé ($p := (p \cup \psi^{-}(o_x)) \setminus (\mathcal{R} \cup m)$).

Az első iterációban B anyagot tudjuk kiválasztani. Ezt az anyagot az $O2$ és $O3$ műveleti egység képes előállítani, azaz ezeket bevesszük a struktúrába. Az $O2$ és $O3$ műveleti egységek bemeneti anyagai D és E anyagok, amelyek nem nyersanyagok, és még nem vizsgáltuk őket, ezért berakjuk őket a p halmazba, ahonnan eltávolítjuk a B anyagot. Az iteráció utáni hálózat a 2.23-as ábrán látható.

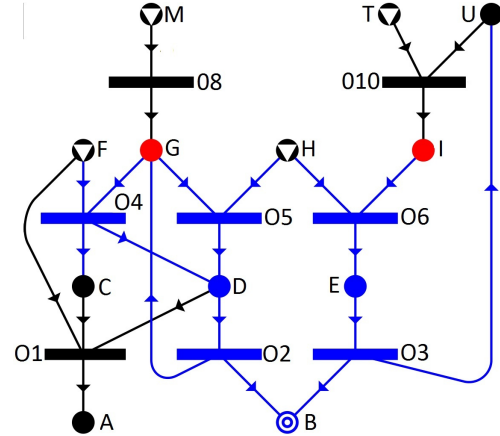
A második iterációban a D anyagot vizsgáljuk, ami alapján az $O4$ és $O5$ műveleti egységek kerülnek bele a struktúrába. Az $O4$ és $O5$ műveleti egységek bemeneti anyagai az F , G és H anyagok, amelyek közül F és H nyersanyag, így az nem kerül bele az előállítandó anyagok halmazába. A második iteráció utáni hálózat a 2.24-es ábrán látható.

A harmadik iterációban az E anyagot vizsgáljuk, ami alapján az $O6$ műveleti egység kerül be a struktúrába. Az $O6$ műveleti egység bemeneti anyagai H és I anyagok, amelyek közül H nyersanyag,

így az nem kerül bele az előállítandó anyagok halmazába. A harmadik iteráció utáni hálózat a 2.25-ös ábrán látható.



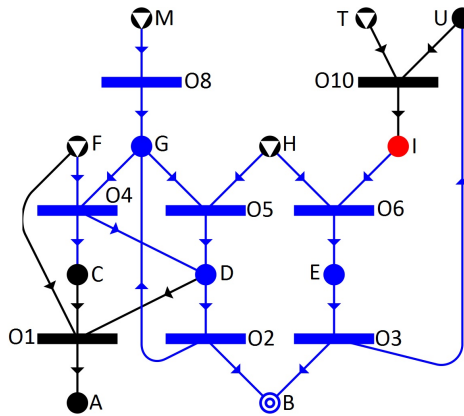
2.24. ábra. MSG példa kompozíciós rész: a második iteráció után



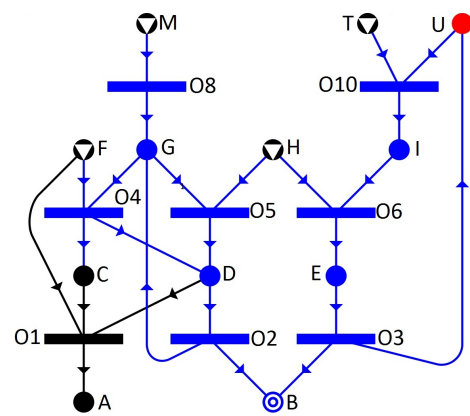
2.25. ábra. MSG példa kompozíciós rész: a harmadik iteráció után

Negyedik iterációban a G anyag kiválasztása történik meg. A G anyagot az $O2$ és $O8$ műveleti egységek tudják előállítani, ezért ezeket hozzáadjuk a struktúrához. Megjegyezzük, hogy az $O2$ műveleti egység már szerepel a struktúrában, ezért ennek a hozzáadása nem változtatja meg az o halmazt. Az $O2$ és $O8$ műveleti egységek bemeneti anyagai a D és M anyagok. Az M anyag nyersanyag, ezért nem adjuk hozzá a p halmazhoz, valamint a D anyagot a korábbiakban már vizsgáltuk, ezért ez sem kerül bele a halmazba. A negyedik iteráció utáni hálózat a 2.26-os ábrán látható.

A következő iterációban az I anyag kerül kiválasztásra, melyet az $O10$ műveleti egység képes előállítani. Az $O10$ műveleti egységet hozzáadjuk a struktúrához, majd megvizsgáljuk a bemeneteit. Az $O10$ műveleti egység működéséhez a T és U anyagokra van szükség. Mivel T nyersanyag, azért ezzel nem foglalkozunk, de U anyagot hozzáadjuk a még megvizsgálandó anyagokat tartalmazó p halmazhoz. Az ötödik iteráció utáni hálózat a 2.27-es ábrán látható.



2.26. ábra. MSG példa kompozíciós rész: a negyedik iteráció után

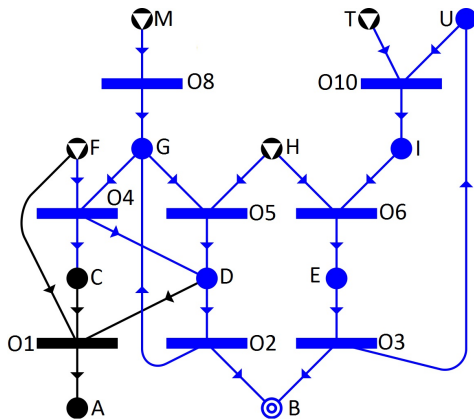


2.27. ábra. MSG példa kompozíciós rész: az ötödik iteráció után

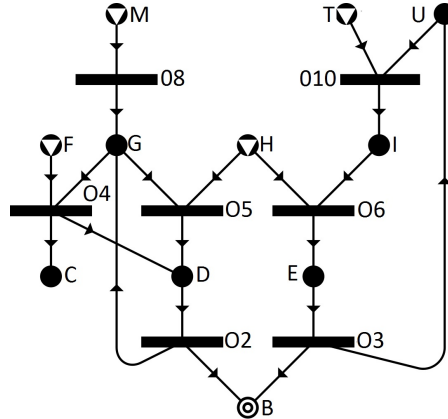
A hatodik iterációban a p halmazban kizárólag az U anyag van, ezért ennek a vizsgálatára kerül sor. Az U anyagot csak az $O3$ műveleti egység tudja előállítani, ezért hozzáadjuk a struktúrához. Az $O3$ műveleti egységnek egyetlen bemeneti anyaga van, az E anyag. Mivel az E anyagot a harmadik iterációban már vizsgáltuk, ezért nem kerül bele a p halmazba. A hatodik iteráció utáni hálózat a 2.28-as ábrán látható.

A hatodik iteráció után a p halmaz üres, azaz az összes olyan anyagot megvizsgáltunk, amely a B termékből elérhető a nyilakon visszafelé haladva. A 2.28-as ábrán látható, hogy az $O1$ műveleti

egység nem került bele a struktúrába, mivel az megsérti az (S4) axiómát, azaz nem vezet belőle út termékhez.



2.28. ábra. MSG példa kompozíciós rész: a hatodik iteráció után



2.29. ábra. MSG példa: maximális struktúra

A kompozíciós résznek egy utolsó sora van ($m := \psi(o)$), amely arra hivatott, hogy a műveleti egységekhez kapcsolódó összes anyag szerepeljen a maximális struktúrában. Erre azért van szükség, mivel az m halmazban csak azok az anyagok vannak benne, amelyeket a kompozíciós részben megvizsgáltunk. Azaz nincsenek benne a szükséges nyersanyagok és azok az anyagok, amelyek egyik műveleti egységnek sem bemenetei (melléktermékek).

Az algoritmus kimenete egy (m, o) pár, ahol m a maximális struktúrában szereplő anyagok, o pedig az itt szereplő műveleti egységek halmaza. A példa maximális struktúrája a 2.29-es ábrán látható.

2.2.2. Az összes megoldásstruktúra generálása

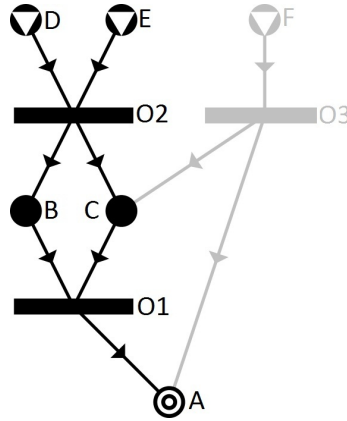
Az algoritmust, amely elő képes állítani az összes megoldásstruktúrát SSG (Solution Structure Generation) algoritmusnak nevezzük. Az algoritmusnak többféle számítógépes megvalósítása is létezik, mi a továbbiakban a döntési leképezésen alapuló megoldást fogjuk ismertetni.

Legyen m anyaghalmaz az $\mathcal{M}$ egy részhalmaza, valamint minden $x \in m$ anyagra legyen $\delta(x) \subseteq \varphi^-(x)$ és $\Delta(x) = \varphi^-(x)$. Ekkor $\delta[m] = \{(x, \delta(x)) | x \in m\}$ egy **döntési leképezés**, valamint $\Delta[m] = \{(x, \Delta(x)) | x \in m\}$ a **teljes döntési leképezés** az m halmaz felett. A döntési leképezés **komplementere** az m halmaz felett a $\bar{\delta}[m] = \{(x, y) | x \in m, y = \varphi^-(x) \setminus \delta(x)\}$ leképezés.

Szavakkal leírva a teljes döntési leképezés olyan párok halmaza, ahol a pár első eleme egy anyag-csomópont, a második eleme pedig az őt előállítani képes összes műveleti egység halmaza. Egy döntési leképezés esetén a pár második elemében olyan műveleti egységek vannak, amelyekről az optimalizálás során úgy döntöttünk, hogy elő fogják állítani az aktuális anyagot. A döntési leképezés komplementere esetén a párok második eleme egy olyan halmaz, amelyben lévő műveleti egységekről úgy döntöttünk, hogy nem állítják elő az anyagot.

Vigyázni kell arra, hogy a δ (és a Δ) függvények mást jelentenek ha utánuk kerek vagy ha szögletes zárójel van. Kerek zárójel esetén a paraméter egy anyag és a leképezés eredménye egy műveleti egység halmaza. Szögletes zárójel esetén a paraméter egy anyaghalmaz, a leképezés eredménye pedig az előzőekben leírt párok halmaza.

A 2.30-as ábrán látható példán keresztül bemutatjuk a jelölések használatát. A döntések során az $O1$ és az $O2$ műveleti egységet beválasztottuk a struktúrába az $O3$ műveleti egységet pedig kizártuk, ezért jelöltük ez utóbbit szürkével az ábrán. Az ábráról leolvashatóak a Δ és a δ függvények. $\Delta(A) = \{O1, O3\}$, $\Delta(B) = \{O2\}$, $\Delta(C) = \{O2, O3\}$, $\Delta(D) = \Delta(E) = \Delta(F) = \emptyset$, valamint $\delta(A) = \{O1\}$, $\delta(B) = \{O2\}$, $\delta(C) = \{O2\}$, $\delta(D) = \delta(E) = \delta(F) = \emptyset$. A döntési leképezést több különböző anyaghalmazra is definiálni lehet, a példában a teljes döntési leképezést a teljes anyaghalmazra



2.30. ábra. Döntési leképezés példa

definiáljuk, a döntési leképezést pedig az $\{A, B, C\}$ halmazra.

$$\begin{aligned}\Delta[\{A, B, C, D, E, F\}] &= \\ &= \{(A, \Delta(A)), (B, \Delta(B)), (C, \Delta(C)), (D, \Delta(D)), (E, \Delta(E)), (F, \Delta(F))\} = \\ &= \{(A, \{O1, O3\}), (B, \{O2\}), (C, \{O2, O3\}), (D, \emptyset), (E, \emptyset), (F, \emptyset)\}.\end{aligned}$$

$$\delta(\{A, B, C\}) = \{(A, \delta(A)), (B, \delta(B)), (C, \delta(C))\} = \{(A, \{O1\}), (B, \{O2\}), (C, \{O2\})\}.$$

A bevezetett matematikai jelölések segítségével azt is el tudjuk dönteni, hogy egy döntési leképezés konzisztens-e. Egy $\delta[m]$ döntési leképezés konzisztens, ha $|m| \leq 1$ vagy $(\delta(x) \cap \delta(y)) \cup (\bar{\delta}(x) \cap \bar{\delta}(y)) = \Delta(x) \cap \Delta(y)$ bármely $x, y \in m$ esetén. Más szavakkal egy döntési leképezés konzisztens, ha legfeljebb egy anyagról döntöttünk, vagy ha bármely két anyag (x és y) esetén bármely, mind a két anyagot előállítani képes műveleti egységről $(\Delta(x) \cap \Delta(y))$ mind a két anyagnál egyformán döntöttünk a bevitelükről $(\delta(x) \cap \delta(y))$ vagy kizárásukról $(\bar{\delta}(x) \cap \bar{\delta}(y))$.

A 2.30-as ábrán egy konzisztens döntési leképezés látható. Ehhez a példához inkonzisztens döntési leképezést úgy lehetne készíteni, ha a B anyagnál úgy döntünk, hogy be vesszük az $O2$ műveleti egységet, a C anyagnál pedig kizárjuk, vagy ha az A anyagnál kizárt $O3$ műveleti egységről a C anyagnál a beviteléről döntünk.

Az SSG algoritmusnál a fentiekben bevezetett döntési leképezést nem csak a konzisztencia vizsgálatához fogjuk használni, hanem ez fogja nekünk leírni a megoldásstruktúrát is. Az eszköz képes erre is, hiszen tartalmazza az összes döntést a műveleti egységekről, amelyek viszont egyértelműen definiálják az aktuális struktúrát. Maga az algoritmus (2.31-es ábra) egy rekurzív eljárást tartalmaz, amely három paraméterében rekurzív. Az algoritmusban a változók értelmezése a következő:

- p - azoknak az anyagoknak a halmaza, amelyek előállításáról még dönteni kell.
- m - azoknak az anyagoknak a halmaza, amelyekről már született döntés.
- $\delta[m]$ - döntési leképezés tartalmazza az eddigi döntéseket.
- x - az anyag, amelyről éppen döntünk.
- C - a lehetséges döntések halmaza az x anyag előállításáról.
- c - az aktuális döntés, azokat a műveleti egységeket tartalmazza, amelyeket beveszünk a struktúrába.

```

begin
  if  $\mathcal{P} = \emptyset$  then
    stop;
    SSG( $\mathcal{P}, \emptyset, \emptyset$ );
  end

procedure SSG( $p, m, \delta[m]$ )
begin
  if  $p = \emptyset$  then
    kiír  $\delta[m]$ ;
    return;
  legyen  $x \in p$ ;
   $C := \wp(\Delta(x)) \setminus \{\emptyset\}$ ;
  for all  $c \in C$  do
    if  $\forall y \in m, c \cap \bar{\delta}(y) = \emptyset$  and  $(\Delta(x) \setminus c) \cap \delta(y) = \emptyset$  then
      SSG( $(p \cup \psi^-(c)) \setminus (\mathcal{R} \cup m \cup \{x\}), m \cup \{x\}, \delta[m] \cup \{(x, c)\}$ );
end

```

2.31. ábra. SSG algoritmus

Az algoritmus egy inicializáló résszel kezdődik, amely beállítja a rekurzív függvény kezdő paramétereit úgy, hogy a függvény hívásakor p tartalmazni fogja az összes terméket ($\mathcal{P}$ elemeit), m halmaz üres lesz (még nem volt döntés egyik anyagról sem) és $\delta[m]$ is üres lesz (mivel nem történt semmilyen döntés). Ezzel egy időben meghívásra kerül az SSG eljárás.

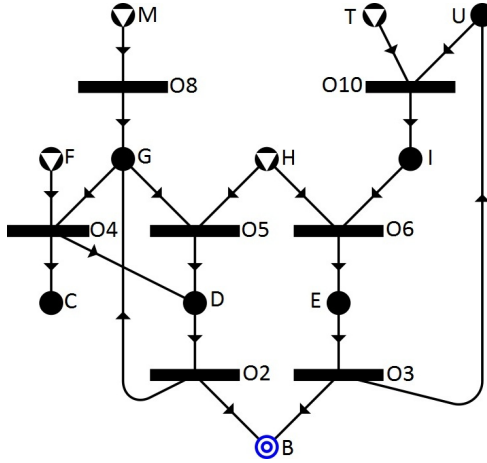
Az SSG eljárás először megvizsgálja, hogy van-e még olyan anyag, amelyről dönteni kell, és ha nincs, akkor kiírja az eredményt. Ha van olyan anyag, amiről még dönteni kell, akkor kiválaszt egyet (x anyag) és összegyűjti, hogy ezt az anyagot hányféle módon lehet előállítani ($\wp(\Delta(x)) \setminus \{\emptyset\}$). Egy anyagot az őt előállítani képes összes műveleti egységek bármely kombinációjával elő lehet állítani, a műveleti egységek halmazának a hatványhalmaza tartalmazza az összes lehetséges kombinációt az üres halmazt is beleértve, de ez utóbbi nem állítja elő a terméket.

Az eljárás minden lehetséges döntésre (c) megvizsgálja, hogy az konzisztens-e vagy sem. A konzisztenciavizsgálat során minden y anyagra, amelyre már korábban döntöttünk, megvizsgálja, hogy az aktuális döntés nem vesz-e be olyan műveleti egységet, amit már kizártunk ($c \cap \bar{\delta}(y) = \emptyset$), valamint nem zár-e ki olyant, amelyet már bevettünk ($(\Delta(x) \setminus c) \cap \delta(y) = \emptyset$). Ha a döntés konzisztens, akkor rekurzívan meghívja önmagát a három paraméterrel.

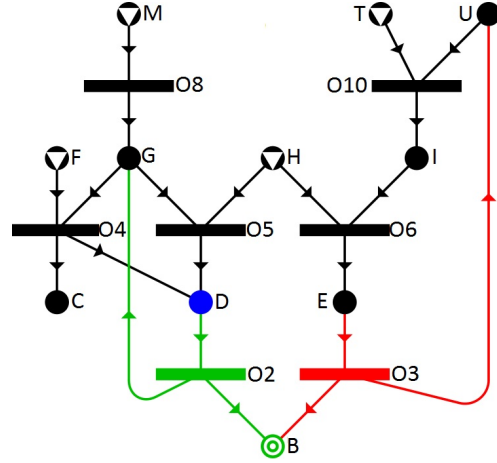
- $(p \cup \psi^-(c)) \setminus (\mathcal{R} \cup m \cup \{x\})$ – Az előállítandó anyagok halmazába belekerülnek azok az anyagok, amelyek a most bevett műveleti egységek bemenetei ($\psi^-(c)$), kivéve ha azok nyersanyagok ($\mathcal{R}$), vagy olyan anyagok, amelyekről korábban már döntöttünk (m), vagy ha éppen most döntöttünk róla (x).
- $m \cup \{x\}$ – Az x anyag belekerül azon anyagok halmazába, amelyről már döntöttünk.
- $\delta[m] \cup \{(x, c)\}$ – A döntési leképezés döntései közé bekerül az x anyaggal kapcsolatos c döntés.

Az algoritmus működése nagyfokú hasonlóságot mutat az MSG algoritmus kompozíciós részéhez, az alapvető különbség az, hogy egy anyagnál nem kizárólag az az egyetlen döntésünk lehet, hogy az összes őt előállító műveleti egységet bevesszük a struktúrába. Az algoritmus működésének bemutatásához tekintsük az MSG algoritmus segítségével az előző fejezetben előállított maximális struktúrát kiindulásnak. Az illusztratív ábrákon a következő színezést használjuk.

- Anyagok
 - Kék szín: azok az anyagok, amelyek előállításáról még dönteni kell (a p halmaz elemei).
 - Zöld szín: azok az anyagok, amelyek előállításáról már döntöttünk (az m halmaz elemei).
 - Fekete szín: a többi anyag, amelyeket még nem értünk el vagy nem kell dönteni az előállításukról (az $\mathcal{M} \setminus (m \cup p)$ halmaz elemei, ahol $\mathcal{M}$ a maximális struktúra anyaghalmaza).



2.32. ábra. SSG példa: inicializálás (X)



2.33. ábra. SSG példa: B anyagot kizárólag $O2$ állítja elő (X1)

• Műveleti egységek

- Zöld szín: a struktúrába bevett műveleti egységek (az $\bigcup_{x \in m} \delta(x)$ halmaz elemei).
- Piros szín: a struktúrából kizárt műveleti egységek (az $\bigcup_{x \in m} \bar{\delta}(x)$ halmaz elemei).
- Fekete szín: a többi műveleti egység, amelyekről még nem született döntés (az $\mathcal{O} \setminus \bigcup_{x \in m} \Delta(x)$ halmaz elemei, ahol $\mathcal{O}$ a maximális struktúra műveleti egységeinek halmaza).

A könnyebb követhetőség kedvéért a döntéseket jelölő gráfokat egy azonosítóval látjuk el, amely azonosítók azt is meghatározzák, hogy hol tartunk az iterációban. Így a rekurzióból való visszatérés során ezekre az azonosítókra hivatkozhatunk anélkül, hogy definiálni kellene a korábbi döntéseinket. Az első gráfhoz az X azonosító tartozik, a többi azonosítót pedig úgy generáljuk, hogy az őt „meghívó” gráf azonosítójához hozzáfűzünk egy számot, ami azt jelöli, hogy hányadik függvényhívásnál keletkezett. Így könnyen követhető, hogy hány függvényhívás történt és melyik előző döntés leszár-mazottja az aktuális gráf (függvényhívás). A továbbiakban a részeseteket a meghívás sorrendjében fogjuk megvizsgálni.

Az algoritmus inicializálása során a folyamatszintézis feladat termékei lesznek azok az anyagok, amelyek előállításáról dönteni kell. Ez jelen példánkban egyetlen anyagot, a B anyagot jelenti. Az eljárás hívását reprezentáló gráf a 2.32-es ábrán látható, amelyre a továbbiakban X-szel hivatkozunk. Az SSG eljárás első futása során a $p = \{B\}$ halmazból kiválasztott x anyag csak B lehet. Ezt a B anyagot két műveleti egység, az $O2$ és az $O3$ képes előállítani. Ez alapján három lehetséges döntésünk van:

- Kizárólag az $O2$ műveleti egység állítja elő B -t.
Ebben az esetben a döntés $c = \{O2\}$, ami konzisztens az előző döntésekkel, mivel nem voltak előző döntések. Így az eljárás a rekurzív hívás során rendre a következő paramétereket adja át:
 - $\{D\}$, mivel ez az $O2$ műveleti egység egyetlen bemeneti anyaga, és erről kell dönteni a következő híváskor,
 - $\{B\}$, mivel B anyagról döntöttünk, és
 - $\{(B, \{O2\})\}$, mivel a B anyagnál $O2$ bevétele volt a döntés.

A gráf a 2.33-as ábrán látható és X1 azonosító tartozik hozzá.

- Kizárólag $O3$ műveleti egység állítja elő B -t.
Ebben az esetben a döntés $c = \{O3\}$, ami szintén konzisztens döntés. A rekurzív hívás során a paraméterek rendre
 - $\{E\}$, mivel ez az $O3$ műveleti egység egyetlen bementi anyaga és erről kell dönteni a következő híváskor,

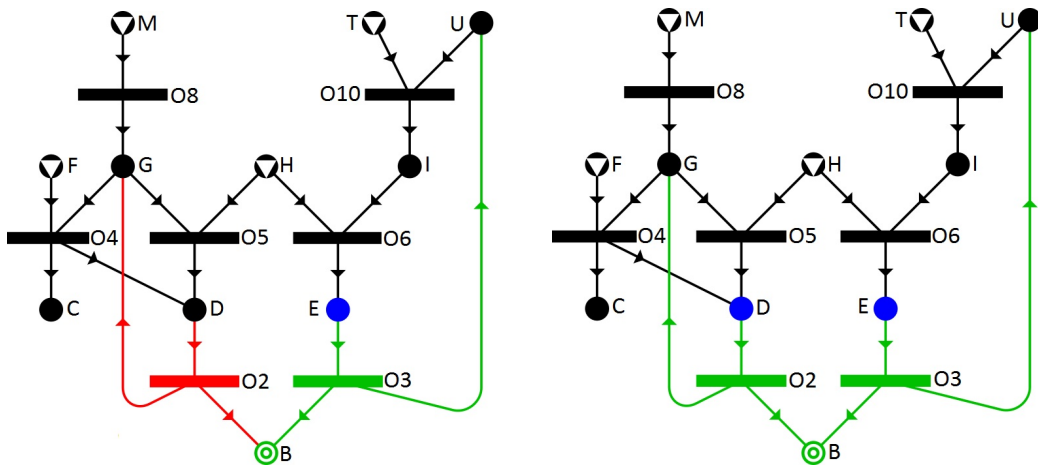
- $\{B\}$, mivel a B anyagról döntöttünk, és
- $\{(B, \{O3\})\}$, mivel a B anyagnál $O3$ bevétele a döntés.

A gráf a 2.34-es ábrán látható és az azonosítója X2.

- Mind a két műveleti egység részt vesz B előállításában.
Ebben az esetben a döntés $c = \{O2, O3\}$, ami konzisztens. A rekurzív hívás során az átadott paraméterek rendre

- $\{D, E\}$, mivel ezek az $O2$ és $O3$ műveleti egységek bementi anyagai és ezekről kell dönteni a következő híváskor,
- $\{B\}$, mivel B anyagról döntöttünk, és
- $\{(B, \{O2, O3\})\}$, mivel a B anyagnál $O2$ és $O3$ együttes bevétele a döntés.

A híváshoz tartozó gráf a 2.35-ös ábrán látható és az azonosítója X3.



2.34. ábra. SSG példa: B anyagot kizárólag $O3$ állítja elő (X2) 2.35. ábra. SSG példa: B anyagot $O2$ és $O3$ állítja elő (X3)

Mind a három részt külön-külön meg kell vizsgálni, mivel mindegyik esetben egy-egy függvényhívás következett. Elsőként vizsgáljuk az X1 esetet, amikor csak $O2$ műveleti egységet választottuk be a struktúrába. Ebben az esetben a hívási paraméterek $p = \{D\}$, $m = \{B\}$ és $\delta[m] = \{(B, \{O2\})\}$. A p halmazban most is csak egyetlen anyag szerepel, ezért $x = D$. Ezt az anyagot szintén két műveleti egységgel lehet előállítani, mégpedig az $O4$ és $O5$ műveleti egységekkel. Mivel az előző esethez hasonlóan itt is két műveleti egység áll rendelkezésre D előállításához, ezért szintén három döntésünk van.

- Kizárólag az $O4$ műveleti egység állítja elő D -t.
Ebben az esetben a döntés $c = \{O4\}$, ami konzisztens az előző döntésekkel, mivel $O4$ -ről még nem volt döntés. A rekurzív hívás során az átadott paraméterek rendre
- $\{G\}$, mivel ez az $O4$ műveleti egységnek kettő bemeneti anyaga van, de F nyersanyag, ezért annak az előállítását nem kell vizsgálni,
- $\{B, D\}$, mivel B és D anyagról döntöttünk eddig és
- $\{(B, \{O2\}), (D, \{O4\})\}$, mivel a B anyagnál $O2$, D anyagnál pedig $O4$ bevétele volt a döntés.

A gráf a 2.36-os ábrán látható és az X11 azonosító tartozik hozzá.

- Kizárólag az $O5$ műveleti egység állítja elő D -t.
Ebben az esetben a döntés $c = \{O5\}$, ami szintén konzisztens döntés. A rekurzív hívás során a paraméterek rendre
- $\{E\}$, az $O5$ műveleti egységnek kettő bemeneti anyaga van, de mivel H nyersanyag, ezért ennek előállításáról nem kell dönteni,

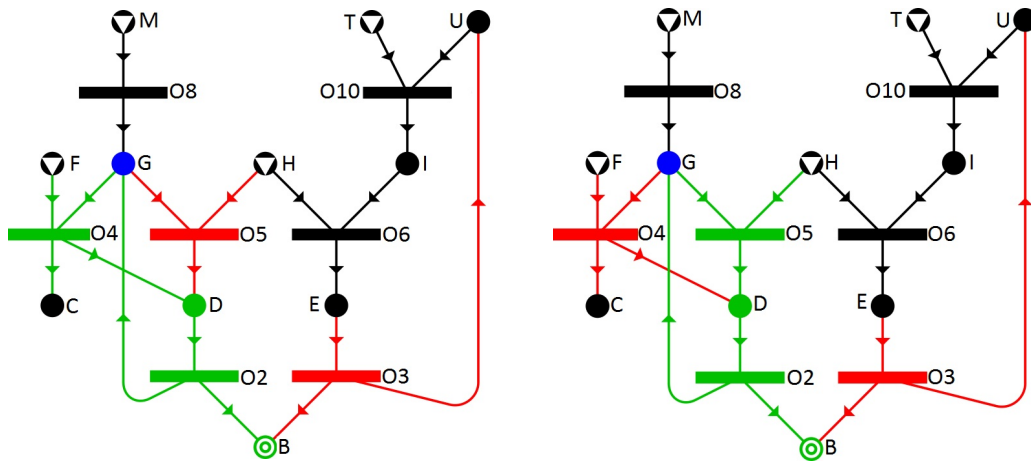
- $\{B, D\}$ mivel a B és D anyagról született eddig döntés és
- $\{(B, \{O2\}), (D, \{O5\})\}$ mivel B anyagnál $O2$, D anyagnál pedig $O5$ bevétele a döntés.

A híváshoz tartozó gráf a 2.37-es ábrán látható és az azonosítója X12.

- $O4$ és $O5$ műveleti egység is részt vesz D előállításában.
Ebben az esetben a döntés $c = \{O4, O5\}$, ami szintén konzisztens a korábbi döntésekkel. Az eljárás a rekurzív hívás során a paraméterek rendre

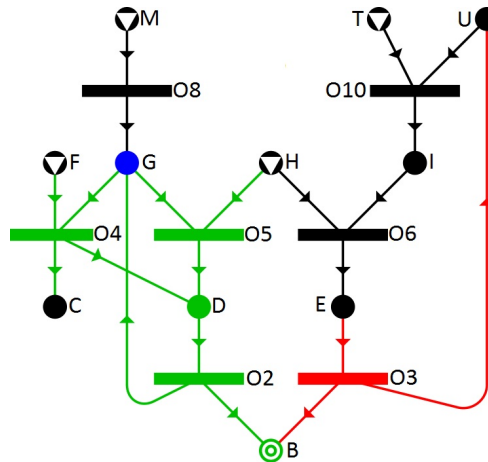
- $\{G\}$, mivel az $O4$ és $O5$ műveleti egységeknek ez az egyetlen olyan bemeneti anyaga, ami nem nyersanyag,
- $\{B, D\}$,
- valamint $\{(B, \{O2\}), (D, \{O4, O5\})\}$.

A gráf a 2.38-as ábrán látható és az azonosítója X13.



2.36. ábra. SSG példa: D anyagot kizárólag $O4$ állítja elő (X11)

2.37. ábra. SSG példa: D anyagot kizárólag $O5$ állítja elő (X12)



2.38. ábra. SSG példa: D anyagot $O4$ és $O5$ is előállítja (X13)

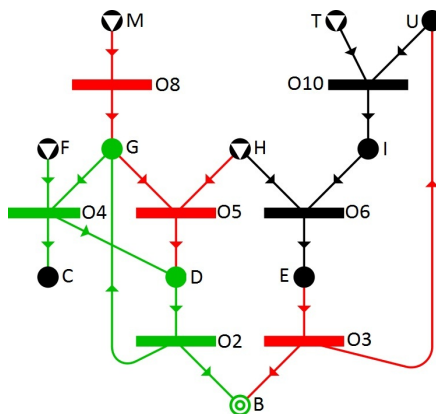
Következőben vizsgáljuk meg, hogy az X11 függvényhívásnál mi történik. Ekkor a hívási paraméterek $p = \{G\}$, $m = \{B, D\}$ és $\delta[m] = \{(B, \{O2\}), (D, \{O4\})\}$. A p halmazban most is csak egyetlen anyag szerepel, ezért $x = G$. A G anyagot a korábbiakhoz hasonlóan két műveleti egységgel lehet előállítani, mégpedig az $O2$ és $O8$ műveleti egységekkel. Mivel két műveleti egység áll rendelkezésre G előállításához, ezért szintén három döntésünk lehet.

- Kizárólag az $O2$ műveleti egység állítja elő G -t.
Ebben az esetben a döntés $c = \{O2\}$, ami konzisztens az előző döntésekkel, mivel $O2$ -t már korábban is bevettük a struktúrába. A rekurzív hívás során a paraméterek rendre
 - $\emptyset$, mivel ez az $O2$ műveleti egységnek a D anyag a bemenete, de arról már döntöttünk,
 - $\{B, D, G\}$, mivel B, D és G anyagról döntöttünk eddig és
 - $\{(B, \{O2\}), (D, \{O4\}), (G, \{O2\})\}$.

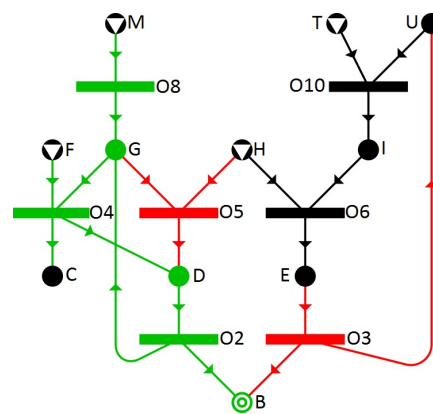
A gráf a 2.39-es ábrán látható és az X111 azonosító tartozik hozzá.

- Kizárólag az $O8$ műveleti egység állítja elő G -t.
Ebben az esetben a döntés $c = \{O8\}$, ami nem konzisztens döntés, mivel ez a döntés az $O2$ műveleti egységet kizárná a struktúrából, miközben B anyagnál már úgy döntöttünk, hogy bevesszük. Így nem kerül sor rekurzív függvényhívásra.
- Mind a két műveleti egység részt vesz G előállításában.
Ebben az esetben a döntés $c = \{O2, O8\}$, ami konzisztens az összes korábbi döntéssel. Az eljárás a rekurzív hívása során a paraméterek rendre
 - $\emptyset$, mivel az $O2$ műveleti egység bementi anyagáról már döntöttünk, $O8$ bementi anyaga pedig nyersanyag,
 - $\{B, D, G\}$ és
 - $\{(B, \{O2\}), (D, \{O4\}), (G, \{O2, O8\})\}$.

A gráf a 2.40-es ábrán látható és az azonosítója X112.



2.39. ábra. SSG példa: G anyagot az $O2$ állítja elő (X111)



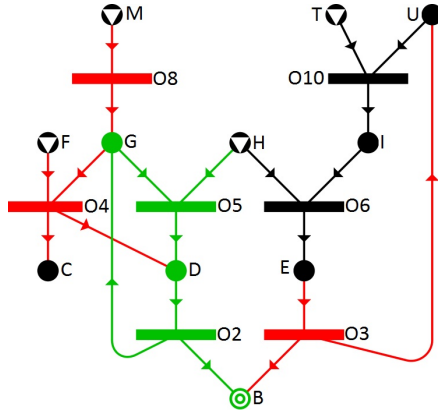
2.40. ábra. SSG példa: G anyagot $O2$ és $O8$ is előállítja (X112)

Az X111-hez tartozó függvényhíváskor a p paraméter egy üres halmaz, ezért az eljárás kiírja a leképezési döntést, mint megoldást. Ebben az esetben a megoldásstruktúra az $O2$ és $O4$ műveleti egységeket (és a hozzájuk tartozó anyagokat, azaz a 2.39-es ábra zölddel jelölt részeihez képest még C és F anyagot is) tartalmazza. X112 esetben szintén üres a megvizsgálandó anyagok halmaza, itt a megoldásstruktúra az $O2$, $O4$ és $O8$ műveleti egységeket tartalmazza.

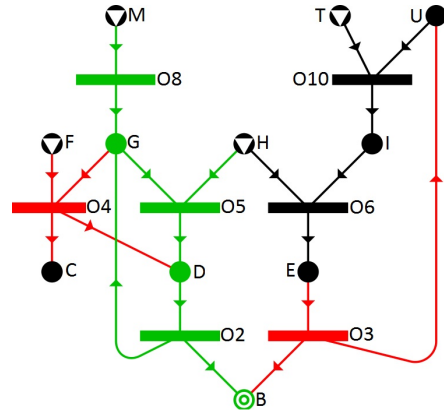
A rekurzióban egy szintet visszalépve az X12 esetet kell tovább vizsgálni. Ez az eset szinte egy az egyben megegyezik az X11 esettel, ezért külön nem is írjuk fel, csak a keletkező két gráfot ábrázoljuk a 2.41-es (X121) és a 2.42-es (X122) ábrákon.

Mind az X121, mind az X122 gráf esetén megoldásstruktúrához értünk el, mivel egyik esetben sincs olyan anyag, amelynek előállításáról még dönteni kell. A keletkezett két megoldásstruktúra az $O2$, $O5$ valamint az $O2$, $O5$, $O8$ műveleti egységeket tartalmazza.

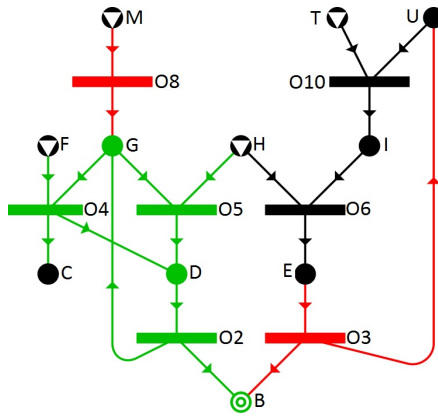
A rekurzióban ismét visszalépve egy szintet az X13-at vizsgáljuk. Ebben az esetben is a G anyagról kell dönteni, és ugyanazok a döntések lesznek konzisztensek, mint az előző kettőnél, ezért nem részletezzük, csak a keletkező két gráfot mutatjuk be a 2.43-as (X131) és a 2.44-es (X132) ábrákon.



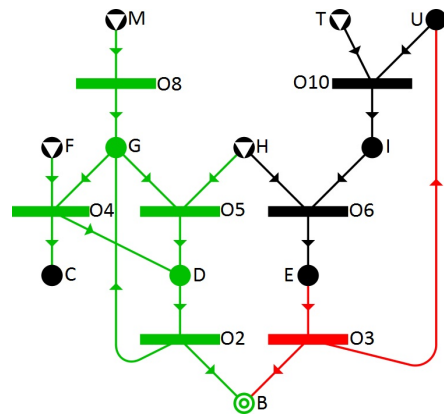
2.41. ábra. SSG példa: G anyagot $O2$ állítja elő (X121)



2.42. ábra. SSG példa: G anyagot $O2$ és $O8$ is előállítja (X122)



2.43. ábra. SSG példa: G anyagot $O2$ állítja elő (X131)



2.44. ábra. SSG példa: G anyagot $O2$ és $O8$ is előállítja (X132)

Mind X131, mind X132 esetén megoldásstruktúrához értünk el. A keletkezett két megoldásstruktúra az $O2$, $O4$, $O5$ valamint az $O2$, $O4$, $O5$, $O8$ műveleti egységeket tartalmazza.

A rekurzióban az X1-ből származó összes függvényhívást megnéztük, most az X2 eset következik. X2-nél a hívás esetén a paraméterek rendre a következők:

- $p = \{E\}$,
- $m = \{B\}$,
- $\delta[m] = \{(B, \{O3\})\}$.

Mivel a p halmaznak egyetlen eleme van, ezért az $x = E$. Az E anyag előállításához egyetlen műveleti egység áll rendelkezésre, ezért csak egyetlen döntésünk lehet: az, hogy ezzel előállítjuk, azaz $c = \{O6\}$. Ez a döntés konzisztens, mivel az $O6$ műveleti egységről korábban nem döntöttünk. Az eljárás a rekurzív hívás során a paraméterek rendre

- $\{I\}$ mivel az $O6$ műveleti egységnek az I anyag az egyetlen olyan bemeneti anyaga, amely nem nyersanyag,
- $\{B, E\}$ és $\{(B, \{O3\}), (E, \{O6\})\}$.

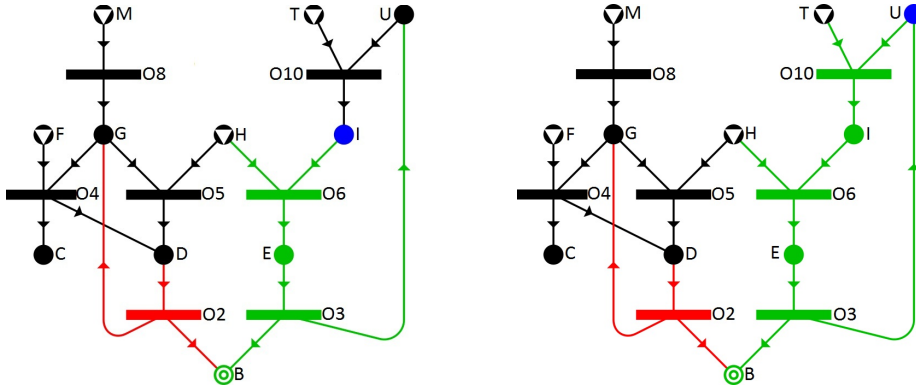
A gráf a 2.45-ös ábrán látható és az azonosítója X21.

Az X21 esetben az I anyagról kell dönteni. Az előzőhöz hasonlóan itt is csak egyetlen döntés lehetséges, amikor $c = \{O10\}$. Ez a döntés konzisztens, mivel az $O10$ műveleti egységről korábban nem döntöttünk. Az eljárás a rekurzív hívása során a paraméterek rendre

- $\{U\}$,

- $\{B, E, I\}$ és
- $\{(B, \{O3\}), (E, \{O6\}), (I, \{O10\})\}$.

A híváshoz tartozó gráf a 2.46-os ábrán látható és az azonosítója X211.

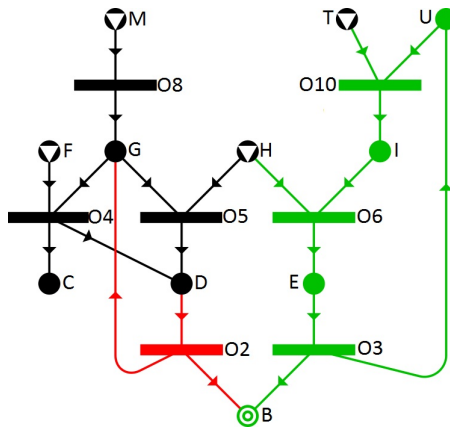


2.45. ábra. SSG példa: *E* anyagot *O6* állítja elő (X21)
2.46. ábra. SSG példa: *I* anyagot *O10* állítja elő (X21)

A rekurzióban tovább menve az X211 esetre az előző két esethez hasonlóan egyetlen anyag előállításáról kell dönteni (*U*), amelyhez egyetlen műveleti egység áll rendelkezésre (*O3*). A rekurzív hívás paraméterei rendre

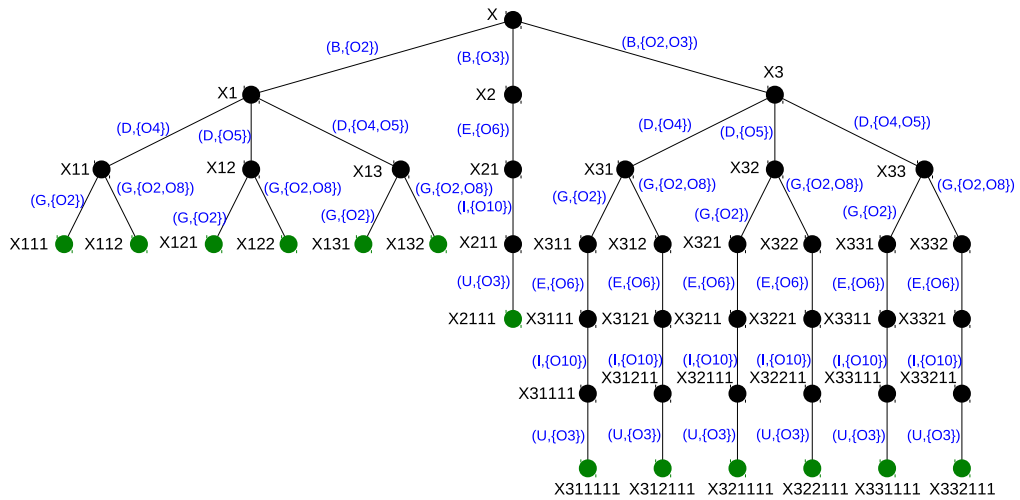
- $\emptyset$,
- $\{B, E, I, U\}$ és
- $\{(B, \{O3\}), (E, \{O6\}), (I, \{O10\}), (U, \{O3\})\}$

A híváshoz tartozó gráf a 2.47-es ábrán látható és az azonosítója X2111. Az X2111 esetben a *p* halmaz üres, azaz elértünk egy megoldásstruktúrához, amely az *O3*, *O6* és *O10* műveleti egységeket tartalmazza.

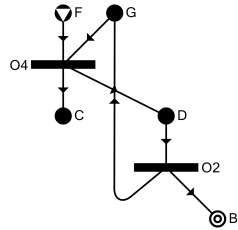


2.47. ábra. SSG példa: *U* anyagot *O3* állítja elő (X2111)

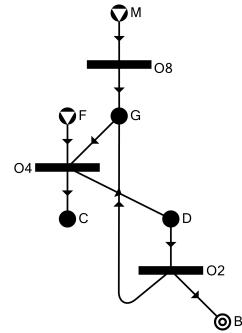
Egyedül az X3 esetet nem vizsgáltuk meg, amely gyakorlatilag az X1 és X2 esetek kombinációja, azaz a két esetből keletkezett megoldásstruktúrák unióját kapjuk meg a további hívások során. X3-hoz a továbbiakban már nem részletezzük a rekurzív függvényhívásokat, de a futás során keletkezett gráfok szülő-gyerekek kapcsolatát a 2.48-as ábrán bemutatjuk. A keresőfa felépítése során feltételeztük, hogy az X3 eset kibontásakor az anyagokról való döntés sorrendje *D, G, E, I, U*. Az ábrán nem csak a szülő-gyerekek kapcsolatok vannak jelölve, hanem kék színnel az aktuális döntések is szerepelnek.



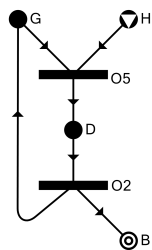
2.48. ábra. SSG példa: Rekurzív függvényhívások



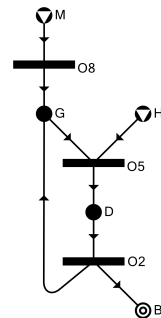
Sorszám: 1
Azonosító: X111
Műveleti egységek: {O2, O4}



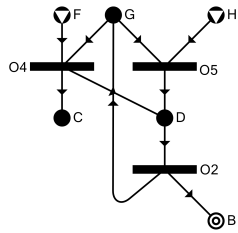
Sorszám: 2
Azonosító: X112
Műveleti egységek: {O2, O4, O8}



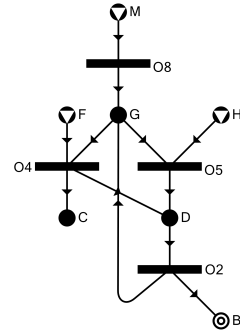
Sorszám: 3
Azonosító: X121
Műveleti egységek: {O2, O5}



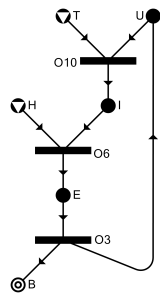
Sorszám: 4
Azonosító: X122
Műveleti egységek: {O2, O5, O8}



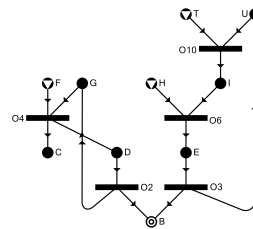
Sorszám: 5
Azonosító: X131
Műveleti egységek: $\{O2, O4, O5\}$



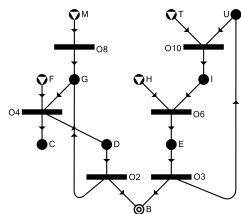
Sorszám: 6
Azonosító: X132
Műveleti egységek: $\{O2, O4, O5, O8\}$



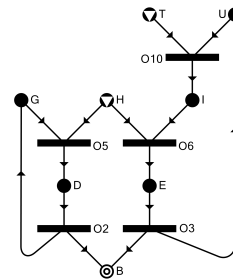
Sorszám: 7
Azonosító: X2111
Műveleti egységek: $\{O3, O6, O10\}$



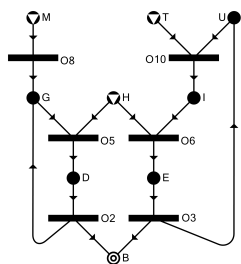
Sorszám: 8
Azonosító: X311111
Műveleti egységek: $\{O2, O3, O4, O6, O10\}$



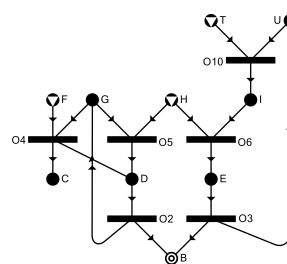
Sorszám: 9
Azonosító: X31211
Műveleti egységek: $\{O2, O3, O4, O6, O8, O10\}$



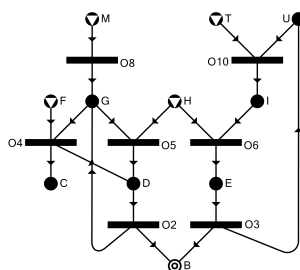
Sorszám: 10
Azonosító: X321111
Műveleti egységek: $\{O2, O3, O5, O6, O10\}$



Sorszám: 11
Azonosító: X322111
Műveleti egységek: $\{O2, O3, O5, O6, O8, O10\}$



Sorszám: 12
Azonosító: X331111
Műveleti egységek: $\{O2, O3, O4, O5, O6, O10\}$



Sorszám: 13

Azonosító: X332111

Műveleti egységek: $\{O2, O3, O4, O5, O6, O8, O10\}$

2.49. ábra. SSG példa: Kombinatorikusan lehetséges megoldásstruktúrák

A keresés során tizenhárom darab kombinatorikusan lehetséges megoldásstruktúrát generálunk az algoritmus segítségével. A 2.49-es táblázatban foglaljuk össze ezeket, valamint a 2.48-as ábrán zöld színnel vannak jelölve azok az esetek, amelyek megoldásstruktúrát eredményeznek.

A fentiekben bemutatott SSG algoritmus a megadott szintézisfeladat összes kombinatorikusan lehetséges megoldásstruktúráját pontosan egyszer generálja, közte a maximális struktúrát, amely maga is megoldásstruktúra. A példánkban az utolsónak generált, X332111 azonosítójú P-gráf a maximális struktúra.

Irodalomjegyzék

- [1] F. Friedler, K. Tarjan, YW Huang, and LT Fan. Combinatorial algorithms for process synthesis. *Computers & chemical engineering*, 16:S313–S320, 1992.
- [2] F. Friedler, K. Tarjan, YW Huang, and LT Fan. Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation. *Computers & chemical engineering*, 17(9):929–942, 1993.
- [3] F. Friedler, JB Varga, Feher E., and LT Fan. Combinatorially accelerated branch-and-bound method for solving the mip model of process network synthesis. In Floudas CA and Pardalos PM, editors, *State of the Art in Global Optimization*, pages 609–626. Kluwer Academic Publishers, 1996.
- [4] F. Friedler, JB Varga, and LT Fan. Decision-mapping: a tool for consistent and complete decisions in process synthesis. *Chemical Engineering Science*, 50(11):1755–1768, 1995.
- [5] N. Nishida, G. Stephanopoulos, and A.W. Westerberg. A review of process synthesis. *AIChE Journal*, 27(3):321–351, 1981.
- [6] J.J. Siirola. Industrial applications of chemical process synthesis. *Advances in Chemical Engineering*, 23:1–62, 1996.
- [7] Hertwig, T.A., A. Xu, A.B. Nagy, R.W. Pike, J.R. Hopper, and C.L. Yaws. Optimal Configuration of Chemical Complexes Based on Economic, Environmental, and Sustainable Costs. *Proceedings of PRES'01, 4-th Conference on Process Integration, Modeling, and Optimisation for Energy Saving and Pollution Reduction*, Florence, May 20-23, 2001.

3. FEJEZET

A strukturális modell kiterjesztései értékelő paraméterekkel: optimalizálás az ABB algoritmussal

Az eddigiekben kizárólag a PNS feladat strukturális kérdéseivel foglalkoztunk. Ahhoz, hogy a feladat optimális megoldását meg tudjuk határozni, definiálni kell, hogy mi a cél, ami alapján optimalizálni szeretnénk, és mik azok a feltételek, amelyeknek meg kell, hogy feleljen egy megoldás. A feladatot természetesen most is matematikai eszközökkel írjuk le, konkrétan egy MILP modellt írunk fel, amihez feltételezzük, hogy a szintézisfeladat a $\mathcal{P}$, $\mathcal{R}$, $\mathcal{O}$ és $\mathcal{M}$ halmazokkal adott.

3.1. Általános matematikai modell

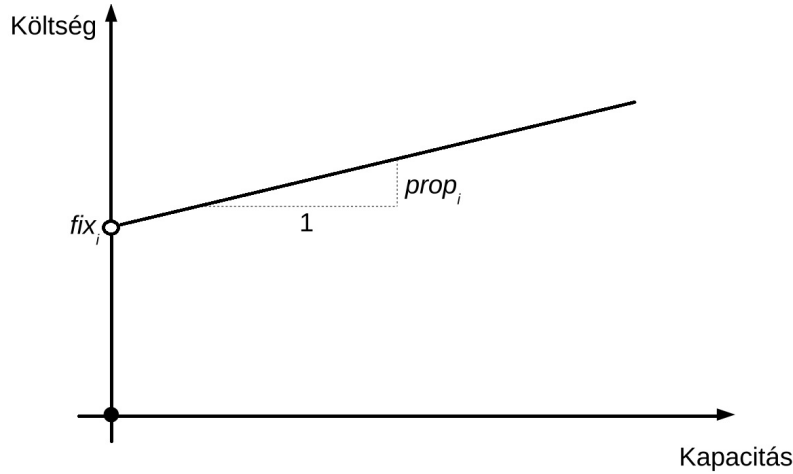
A modellben a célunk a termékek előállítására lesz a szükséges mennyiségben minimális befektetési, működtetési és nyersanyag költséggel. Első lépésben vizsgáljuk meg, hogy milyen döntéseink vannak, mert ezek alapján tudjuk meghatározni a döntési változóinkat. A legegyszerűbb az lenne, ha a kombinatorikusan lehetséges megoldásstrukturák közül kiválasztanánk az optimálisat, amely módszer teljes leszámplálásnak nevezzük. Mi viszont nem akarjuk előre generálni az összes megoldásstrukturát, hanem a feladatnak egy olyan modelljét szeretnénk felírni, amely közvetlenül kiválasztja, hogy mely műveleti egységeket használjuk a termékek előállításához.

Ezek alapján látható, hogy minden $o_i \in \mathcal{O}$ műveleti egységhez tartozik egy y_i bináris változó, melynek értéke 1, ha műveleti egységet bevesszük a megoldásba és 0, ha nem. Ezen kívül a műveleti egységhez tartozik egy x_i pozitív folytonos változó, amely azt adja meg, hogy mekkora műveleti egységet veszünk be a struktúrába, és használunk.

A döntési változók után a célfüggvényt kell meghatároznunk. A célfüggvény értéke függ a műveleti egységek beszerzési árától, a működési költségétől és a felhasznált nyersanyagok mennyiségétől.

- A modellünkben a műveleti egységek beszerzési és működési költsége is ugyanolyan módon függ a műveleti egység méretétől, ezért nem kezeljük őket külön. Ez a gyakorlatban egy elfogadott módszer olyan módon, hogy a beszerzési költséget leosztják az elvárt megtérülési idővel: ha például öt év a megtérülési idő, akkor öttel. Egy műveleti egység költsége nulla, ha nem kerül be a struktúrába, azaz nem vesszük meg, és nem használjuk. Egyébként a költségnek van egy fix része, és ezen felül lineárisan függ a kapacitástól (3.1-es ábra).

Jelölje az $o_i \in \mathcal{O}$ műveleti egység fix költségét fix_i , az arányos (vagy más néven proporcionális) költségét pedig $prop_i$. Ekkor az o_i műveleti egység költségét $y_i(fix_i + prop_i x_i)$ formában írhatjuk fel. Mivel ez a függvény nem lineáris, ezért linearizálni kell. A linearizálás után a függvény $fix_i y_i + prop_i x_i$ formát vesz fel, és a modell kiegészül az $x_i \leq cap_i y_i$ feltétellel, ahol cap_i a műveleti egység kapacitásának felső korlátja. Így, ha $y_i = 0$, akkor $x_i \leq 0$, azaz $x_i = 0$ és a műveleti egység költségének értéke is 0, ha pedig $y_i = 1$, akkor $x_i \leq cap_i$ és a költség értéke $fix_i + x_i prop_i$.



3.1. ábra. Műveleti egység költsége

- A nyersanyagok költségeinek a meghatározásához ki kell számolni, hogy melyik nyersanyagból mennyit használunk fel az adott struktúrában. Ez az anyagmennyiség a nyersanyagot felhasználó műveleti egységek méretétől függ.

Jelölje az egységnyi méretű $o_i \in \mathcal{O}$ műveleti egység által az $m_j \in \mathcal{M}$ anyagból felhasznált anyag mennyiségét ir_{ij} . Ekkor az m_j anyagból felhasznált anyag mennyisége a

$$\sum_{o_i \in \varphi^+(m_j)} x_i ir_{ij}$$

formulával számolható.

Jelölje egy $r_j \in \mathcal{R}$ nyersanyag egységnyi költségét $price_j$. Ekkor az r_j nyersanyag költsége $price_j \sum_{o_i \in \varphi^+(r_j)} x_i ir_{ij}$ módon határozható meg.

Ezek alapján a célfüggvény értéke a műveleti egységek és a nyersanyagok költségeinek az összege.

$$\min \sum_{o_i \in \mathcal{O}} (fix_i y_i + prop_i x_i) + \sum_{r_j \in \mathcal{R}} \left(price_j \sum_{o_i \in \varphi^+(r_j)} x_i ir_{ij} \right) \quad (3.1)$$

A célfüggvény meghatározása után most lássuk a feltételeket.

- A célfüggvény linearizálása során már bevezettünk egy feltételt minden műveleti egységhez.

$$x_i \leq cap_i y_i \quad o_i \in \mathcal{O} \quad (3.2)$$

- Minden $r_j \in \mathcal{R}$ nyersanyagra adott, hogy mennyi érhető el belőle (max_j), azaz ennél többet nem fogyaszthatnak belőle a műveleti egységek.

$$\sum_{o_i \in \varphi^+(r_j)} x_i ir_{ij} \leq max_j \quad r_j \in \mathcal{R}$$

- Minden $p_j \in \mathcal{P}$ termékre adott, hogy mennyit kell legalább előállítani belőle (min_j). Mivel egy nyersanyagot a struktúrában fel is használhatunk, ezért nem az előállított és a felhasznált anyagmennyiség különbsége kell, hogy elérje a kívánt mennyiséget. Egy tetszőleges anyaghoz az előállított anyagmennyiséget a felhasznált anyagmennyiséghez hasonlóan a műveleti egységek mérete alapján lehet meghatározni. Ebben az esetben az ir_{ij} paraméterhez hasonló or_{ij} paraméterre van szükség, amely azt adja meg, hogy az egységnyi méretű $o_i \in \mathcal{O}$

műveleti egység a $m_j \in \mathcal{M}$ anyagból mennyit gyárt.

$$\sum_{o_i \in \varphi^-(p_j)} x_i or_{ij} - \sum_{o_i \in \varphi^+(p_j)} x_i ir_{ij} \geq \min_j \quad p_j \in \mathcal{P}$$

- A nyersanyagokon kívül minden egyes $m_j \in \mathcal{M} \setminus (\mathcal{R} \cup \mathcal{P})$ anyagra igaz az anyagegyensúly feltétel, azaz legfeljebb akkora mennyiséget lehet felhasználni, amennyit előállítottunk. A termékekre adott előző feltételünk ennél szigorúbb, ezért rájuk ezt a feltételt nem kell felírni.

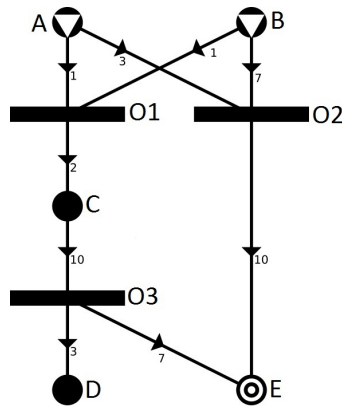
$$\sum_{o_i \in \varphi^-(m_j)} x_i or_{ij} \geq \sum_{o_i \in \varphi^+(m_j)} x_i ir_{ij} \quad m_j \in \mathcal{M} \setminus (\mathcal{R} \cup \mathcal{P})$$

Az anyagokra vonatkozó feltételeket össze lehet vonni egy feltétellé az egyszerűség kedvéért ((3.3)). Ehhez mindössze a bemeneti paramétereket kell megfelelően megadni az alapján, hogy a feltétel középső része a felhasznált és az előállított anyagmennyiség különbsége.

- Nyersanyagoknál az előállított anyagmennyiség nulla, és a felhasználható mennyiség a $\max_j$ értékkel adott. Ez alapján $l_j = -\max_j$ és $u_j = 0$.
- Termékeknél $l_j = \min_j$ egyértelműen adódik. Az u_j paraméter értékét eddig nem definiáltuk, de megadható minden termékből egy maximálisan előállítandó mennyiség is. Ha nem akarunk ilyen értéket megadni, akkor $u_j = \infty$.
- Köztes termékeknél az anyagmegmaradás feltétele $l_j = 0$ paraméter megadásával biztosítható. A felső korlát definiálása az adott anyagból megmaradó anyagmennyiségre jelent feltételt, ami fontos lehet például szennyező anyag esetén.

$$l_j \leq \sum_{o_i \in \varphi^-(m_j)} x_i or_{ij} - \sum_{o_i \in \varphi^+(m_j)} x_i ir_{ij} \leq u_j \quad m_j \in \mathcal{M} \quad (3.3)$$

A modell felírásának bemutatáshoz a 3.2-es ábrán látható PNS feladatot vesszük alapul. Az MSG és SSG algoritmusoktól eltérően itt már fontosak az éleken látható súlyok, amelyek azt jelzik, hogy a kapcsolódó műveleti egység mennyi anyagot fogyaszt vagy állít elő, ha egységnyi méretű. Azaz ezek az értékek pontosan az ir_{ij} és or_{ij} értékeket adják meg.



3.2. ábra. Példa MILP felírásához

A feladathoz tartozó paramétereket táblázatokban foglaltuk össze. A három műveleti egységhez tartozó kapacitáskorlátok, fix és proporcionális költségek a 3.1-es táblázatban találhatóak. A nyersanyagokhoz és termékekhez tartozó paraméterek a 3.2-es táblázatban találhatóak, ahol a termékeknel az előállítandó anyagmennyiség ($\min$), a nyersanyagoknál pedig a rendelkezésre álló mennyiség ($\max$) és az egységnyi nyersanyag ára ($price$) látható. A termékekhez nem definiálunk maximális mennyiséget, köztes anyagokra pedig nem definiáljuk a maximálisan megmaradó anyagmennyiségeket. A 3.3-as táblázatban szerepelnek az egységnyi méretű műveleti egységekhez tartozó bemeneti anyagmennyiségek, a 3.4-es táblázatban pedig a kimeneti anyagmennyiségek. Mivel az (S2)

axióma szerint nyersanyagokat nem állíthatunk elő, ezért a kimeneti anyagáram ezekre nem értelmezhető, így a 3.4-es táblázatban nem szerepelnek.

Műveleti egység	<i>cap</i>	<i>fix</i>	<i>prop</i>
<i>O1</i>	10	5	5
<i>O2</i>	10	10	10
<i>O3</i>	10	5	5

3.1. táblázat. ABB példa: Műveleti egységek paraméterei

Anyag	<i>min</i>	<i>max</i>	<i>price</i>
<i>A</i>		10	2
<i>B</i>		20	1,5
<i>E</i>	5		

3.2. táblázat. ABB példa: Anyagok paraméterei

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>O1</i>	1	1	0	0	0
<i>O2</i>	3	7	0	0	0
<i>O3</i>	0	0	10	0	0

3.3. táblázat. ABB példa: Bemeneti anyagarányok (ir_{ij})

	<i>C</i>	<i>D</i>	<i>E</i>
<i>O1</i>	2	0	0
<i>O2</i>	0	0	10
<i>O3</i>	0	3	7

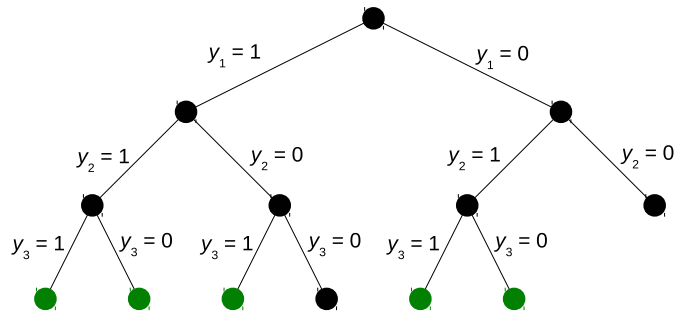
3.4. táblázat. ABB példa: Kimeneti anyagarányok (or_{ij})

A paraméterek segítségével fel tudjuk írni a PNS feladat MILP modelljét. A (3.1) egyenlet szerint a célfüggvény a műveleti egységek és a nyersanyagok költségeinek az összege. Az o_i műveleti egység költsége $fix_i y_i + prop_i x_i$, azaz az összes műveleti egység költsége $fix_1 y_1 + prop_1 x_1 + fix_2 y_2 + prop_2 x_2 + fix_3 y_3 + prop_3 x_3 = 5y_1 + 5x_1 + 10y_2 + 10x_2 + 5y_3 + 5x_3$. Az r_j nyersanyag költsége a nyersanyag ára szorozva a felhasznált anyagmennyiséggel. Az *A* anyagot az *O1* és az *O2* műveleti egység használhatja fel, ahol a bemeneti anyagarányok rendre 1 és 3. Így az *A* nyersanyagból felhasznált anyagmennyiség $x_1 + 3x_2$. Hasonlóan a *B* anyagból $x_1 + 7x_2$ anyagmennyiséget használunk fel. Ezeket az értékeket a költségekkel megszorozva kapjuk, hogy a teljes nyersanyagköltség $2(x_1 + 3x_2) + 1,5(x_1 + 7x_2) = 3,5x_1 + 16,5x_2$. A modellünk teljes célfüggvényét az alábbi egyenletben foglaltuk össze:

$$\min 5y_1 + 8,5x_1 + 10y_2 + 26,5x_2 + 5y_3 + 5x_3$$

A célfüggvényhez hasonlóan az összes feltételt is generálni tudjuk, amelyet most nem írunk fel részletesen, csak felsoroljuk őket. Gyakorlatilag minden műveleti egységhez tartozik egy linearizálási feltétel, és minden anyaghoz egy feltétel. A feltételek előtt szerepel a műveleti egység vagy az anyag neve, amelyhez a feltétel tartozik.

$$\begin{aligned} O1 : 0 &\leq x_1 \leq 10y_1 \\ O2 : 0 &\leq x_2 \leq 10y_2 \\ O3 : 0 &\leq x_3 \leq 10y_3 \\ A : -10 &\leq -x_1 - 3x_2 \leq \infty \\ B : -20 &\leq -x_1 - 7x_2 \leq \infty \\ C : 0 &\leq 2x_1 - 10x_3 \leq \infty \\ D : 0 &\leq 3x_3 \leq \infty \\ E : 5 &\leq 10x_2 + 7x_3 \leq \infty \end{aligned}$$



3.3. ábra. Példa: szétválasztás és korlátozás algoritmus keresési fája

3.2. Gyorsított szétválasztás és korlátozás algoritmus

Az eddigiekben leírt modellt a maximális struktúra alapján lehet felírni, mivel a felírás közben feltételeztük az (S2) axióma teljesülését. A modellt több módszerrel is meg lehet oldani.

- Egy általános MILP megoldó használatával.

Ebben az esetben a bináris változók száma megegyezik a műveleti egységek számával.

- Az SSG algoritmust használva.

Az SSG segítségével generáljuk az összes kombinatorikusan lehetséges megoldásstruktúrát, ahol a modell bináris változói már rögzítettek aszerint, hogy a műveleti egység benne van a struktúrában vagy nincs. Így annyi LP modellt kell megoldani, ahány megoldásstruktúra tartozik a PNS feladathoz.

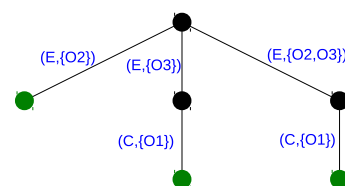
- Szétválasztás és korlátozás algoritmus segítségével.

A szétválasztás és korlátozás (branch and bound B&B) típusú algoritmusok a feladatot iteratíván egyre kisebb részfeladatokra bontják, amíg könnyen megoldható feladatot nem kapunk, és a kisebb részfeladatok megoldásából következtetnek az eredeti feladat megoldására. A részfeladatok generálása a bináris változók különböző értékekre való rögzítésével valósul meg. Az algoritmus legrosszabb esetben rosszabb futási idővel rendelkezik, mint a teljes leszámolás, de a nagyfokú rugalmassága miatt gyorsítható. Az általános MILP megoldók ezen algoritmus alapján működnek.

- ABB algoritmussal.

Az általános szétválasztás és korlátozás algoritmus nem használja ki a feladat strukturális tulajdonságait, ezért érdemes egy olyan algoritmust alkalmazni, amely az axiómák által lecsökkentett keresési téren dolgozik. Ha az előző, három műveleti egységes feladatot szeretnénk megoldani egy általános szétválasztás és korlátozás algoritmus segítségével, akkor az algoritmus keresési fája (legrosszabb esetben) a 3.3-as ábrán látható módon nézne ki. A keresési fa gyökere reprezentálja az eredeti feladatot, amelyből két kisebb részfeladatot generálunk arról döntve, hogy az $O1$ műveleti egység szerepel a struktúrában ($y_1 = 1$), vagy sem ($y_1 = 0$). A két újabb részfeladatban az $O2$ műveleti egységről döntünk, az így keletkezettekben pedig az $O3$ műveleti egységről. A fa leveleiben zölddel jelölve találhatóak azok a részfeladatok, ahol már minden műveleti egységről döntés született, és megvalósíthatóak. Itt már nem MILP, hanem csak LP modelleket kell megoldani, amelyek sokkal könnyebben megoldhatóak.

Az algoritmus futása után nem kell minden részfeladatot megvizsgálni. A nem megvalósíthatóakat, és azokat, amelyekből nem érhető el jobb megoldás, mint a már megtalált, nem kell tovább bontani. Ha például úgy döntünk, hogy sem $O1$, sem $O2$ műveleti egységet nem választjuk be a struktúrába, úgy már nem lehet a terméket előállítani. De így is látható, hogy 5 lehetséges megoldás van, és 13 részfeladatunk van a keresés



3.4. ábra. Példa: SSG keresési fája

alapján, ezzel szemben összesen 3 megoldás-struktúra tartozik a feladathoz. Ebből is látszik, hogy a keresési teret nagy mértékben tudjuk csökkenteni a fejezet elején bemutatott axiómák használatával.

Az alábbiakban bemutatjuk a gyorsított szétválasztás és korlátozás (accelerated branch and bound, ABB) algoritmust, amely az SSG algoritmusnál bemutatott döntési leképezésen alapul. A különbség annyi az SSG-hez képest, hogy a rekurzív hívás előtt egy korlátszámolási eljárást hajtunk végre, amely a részfeladathoz egy alsó korlátot rendel. Az alsó korlát alapján törölhetünk részfeladatokat, így az ABB algoritmus az SSG-hez tartozó fa egy részét generálja ki, legrosszabb esetben a teljes SSG fát (3.4-es ábra). A két keresési fát összehasonlítva láthatjuk, hogy az ABB az axiómák segítségével jelentős mértékben lecsökkenti a keresési teret.

Felmerülhet a kérdés, hogy hogyan lehet az egyik esetben 5, a másik esetben pedig 3 megoldás. Ez úgy fordulhat elő, hogy az általános B&B eljárás redundáns megoldásokat is generál. Például az $O2$ műveleti egység egymagában elegendő a termék előállításához, de a B&B algoritmus előállít olyan megoldást is, amikor az $O2$ és $O1$ valamint az $O2$ és $O3$ műveleti egység is szerepel a struktúrában. Ekkor az $O1$ illetve az $O3$ műveleti egységek nem vesznek részt a tényleges gyártásban, a kapacitásuk nulla lesz, de a fix költségük növeli a teljes költséget.

Nem beszéltünk még a korlátozásról, ahol egy részproblémához próbáljuk megbecsülni a hozzá tartozó optimum értéket. A részproblémához tartozó korlátot a keresési fa más részein használhatjuk fel. Az ABB algoritmusban használt korlát számolási eljárás egy LP modell megoldását jelenti, ahol a még le nem rögzített bináris változókat relaxáljuk, azaz nem csak 0 és 1 értékeket vehetnek fel, hanem a $[0, 1]$ intervallumon bármilyen értéket. A relaxált érték természetesen nem lehet jobb, mivel tartalmazza a teljes szülő feladatot.

A 3.6 ábrán látható eljárás az SSG algoritmusban használt változókon kívül használ még három új változót:

- U – az eddig megtalált legjobb megoldás értéke.
- *currentbest* – az eddig megtalált legjobb megoldás.
- *bound* – az aktuális részfeladat korlátja.

Mint látható, és ahogy korábban említettük is, az ABB algoritmus nagyfokú hasonlóságot mutat az SSG algoritmussal, ezért részletesen nem ismertetjük, csak a két algoritmus közötti különbségekre térünk ki.

- Mivel induláskor még nem találtunk megoldást, ezért az U értéke nem lehet kisebb, mint bármi, amit a keresés során találhatunk, azaz végtelen kell, hogy legyen.
- A *currentbest* értékét nem kell inicializálni, mivel ezt csak a kiírásnál használjuk fel, ami viszont csak akkor történik meg, ha már megtaláltuk az optimális megoldást.
- A korlátszámolás a BOUND eljárás meghívásával történik meg. A döntési leképezés tartalmazza az összes eddigi döntést, azaz ez alapján le lehet rögzíteni a műveleti egységek bináris változóinak egy részét.

Az ABB algoritmushoz több gyorsítás is készült, amelyek közül mi a legelső és legáltalánosabban használt, az úgynevezett neutrális kiterjesztést mutatjuk be. A neutrális kiterjesztés az egymással összefüggő döntések kezelését segíti elő. Ha a példánkban az $O3$ műveleti egységet beválasztjuk a struktúrába, akkor az $O1$ műveleti egységet is be kell választani, valamint ha az $O3$ -at kizárjuk, akkor $O1$ -et sem vehetjük be, mivel ez a két műveleti egység egymás nélkül nem működhet. Az $O1$ állítja elő az $O3$ működéséhez szükséges anyagokat, és ezeket az anyagokat más egység nem fogyasztja.

A formális definícióhoz tegyük fel, hogy egy részfeladatnál az m_i anyaghalmazra történt eddig döntés. Ahhoz hogy ennél a részfeladatnál kombinatorikusan lehetséges megoldásstruktúrát kapjunk, további anyagokról még dönteni kell, konkrétan a $(\psi^-(\varphi(\delta[m_i])) \cup \mathcal{P}) \setminus (m_i \cup \mathcal{R})$ halmazban lévő anyagokról mindenképpen. Ezek azok az anyagok, amelyek valamely már beválasztott műveleti egység bemeneti anyagai $(\psi^-(\varphi(\delta[m_i]))$, vagy termékek, de nem nyersanyagok, és még nem döntöttünk róluk. A következőkben vezessük be a direkt neutrális kiterjesztést, a neutrális kiterjesztést és a maximális neutrális kiterjesztést.

- Direkt neutrális kiterjesztés

A $\delta_x[m \cup \{x\}] = \delta[m] \cup \{(x, d)\}$ a $\delta[m]$ konzisztens döntési leképezés direkt neutrális kiterjesztése, ha $x \in (\psi^-(\varphi(\delta[m])) \cup \mathcal{P}) \setminus (m \cup \mathcal{R})$ és $d \in \Delta[m]$, valamint $\forall c \in \varphi(\delta(x)) \setminus \{\emptyset, d\}$ esetén c nem konzisztens döntést jelent. Azaz az x anyag esetén d az egyetlen konzisztens döntés.

- Neutrális kiterjesztés

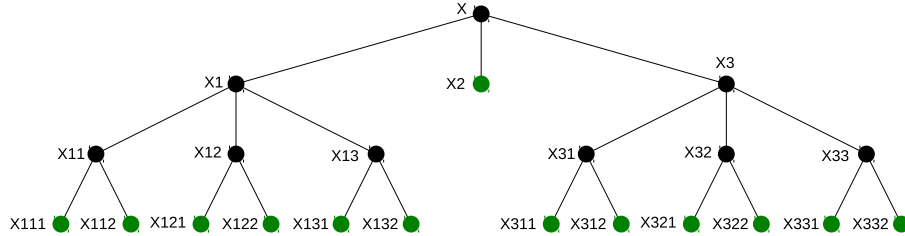
A $\delta_n[m_n]$ ($n = 0, 1, \dots$) döntési leképezés a $\delta_0[m_0]$ konzisztens döntési leképezés neutrális kiterjesztése, ha létezik konzisztens döntési leképezések olyan $\delta_0[m_0], \delta_1[m_1], \dots, \delta_n[m_n]$ sorozata, amelyben $\delta_i[m_i]$ direkt neutrális kiterjesztése a $\delta_{i-1}[m_{i-1}]$ döntési leképezésnek minden $i = (1, 2, \dots, n)$ esetén.

- Maximális neutrális kiterjesztés

A $\hat{\delta}[\hat{m}]$ konzisztens döntési leképezés a $\delta[m]$ konzisztens döntési leképezés maximális neutrális kiterjesztése, ha annak neutrális kiterjesztése, és nincsenek további direkt neutrális kiterjesztései.

A maximális neutrális kiterjesztéssel kiegészített ABB algoritmus pseudokódja a 3.7 ábrán látható. Ez az algoritmus abban különbözik az előzőtől, hogy minden döntés után (és az algoritmus kezdeténél is) előállítjuk az aktuális részfeladat maximális neutrális kiterjesztését ($\hat{\delta}[\hat{m}]$), és azzal dolgozunk tovább. A neutrális kiterjesztés során természetesen megváltozik azon anyagok halmaza, amelyekről már döntöttünk (m), és amelyekről még dönteni kell (p), ezeket a halmazokat ($\hat{m}'$, p') is elő kell állítani.

A gyorsítás mértékének szemléltetéséhez tekintsük az SSG algoritmusnál használt példát. Az alap ABB algoritmus ugyanazt a fát generálja legrosszabb esetben, mint az SSG algoritmus (2.48-as ábra). A maximális neutrális kiterjesztés gyakorlatilag azokat a lépéseket vonja össze, amelyek egymásból következnek, azaz ahol a fában egy szülőnek csak egy gyereke van. A maximális neutrális kiterjesztést használó ABB algoritmus legrosszabb esetének keresőfája a 3.5-ös ábrán látható. A két ábrát összehasonlítva láthatjuk, hogy a fa leveleinek száma nem változik, mivel mind a két algoritmus ugyanazokat a megoldásstruktúrákat vizsgálja, de a csúcsok száma 43-ról 22-re, azaz majdnem a felére csökkent. Ez azt jelenti, hogy fele annyi LP modellt kell megoldani, amely az algoritmus futásának döntő részét teszi ki.



3.5. ábra. Példa: ABB keresési fa maximális neutrális kiterjesztéssel

```

bemenet:  $\mathcal{M}, \mathcal{P}, \mathcal{R}, \Delta[\mathcal{M}]$ ;
kimenet: A  $(\mathcal{P}, \mathcal{R}, \mathcal{O})$  feladat optimális megoldása;
globális változók:  $U, currentbest$ ;

begin
     $U = \infty$ ;
     $ABB(\mathcal{P}, \emptyset, \emptyset)$ ;
    if  $U < \infty$  then
        print  $currentbest$ ;
    else
        print "There is no solution";
end

procedure  $ABB(p, m, \delta[m])$ 
begin
    legyen  $x \in p$ ;
     $C = \wp(\Delta(x)) \setminus \{\emptyset\}$ ;
    for all  $c \in C$  do
        if  $\forall y \in m, c \cap \bar{\delta}(y) = \emptyset$  and  $(\Delta(x) \setminus c) \cap \delta(y) = \emptyset$  then
             $p' = (p \cup \psi^-(c)) \setminus (\mathcal{R} \cup m \cup \{x\})$ ;
             $m' = m \cup \{x\}$ ;
             $\delta[m'] = \delta[m] \cup \{(x, c)\}$ ;
             $bound = BOUND(\delta[m'])$ ;
            if  $U \geq bound$  then
                if  $p' = \emptyset$  then
                     $U = bound$ ;
                     $currentbest = \delta[m']$ ;
                else
                     $ABB(p', m', \delta[m'])$ ;
        end
    end

```

3.6. ábra. ABB algoritmus


```

bemenet:  $\mathcal{M}, \mathcal{P}, \mathcal{R}, \Delta[\mathcal{M}]$ ;
kimenet: A  $(\mathcal{P}, \mathcal{R}, \mathcal{O})$  feladat optimális megoldása;
globális változók:  $U, currentbest$ ;

begin
     $U = \infty$ ;
    Legyen  $\hat{\delta}[\hat{m}]$  a  $\delta[\emptyset]$  maximális neutrális kiterjesztése;
     $p = (\psi^-(\varphi(\hat{\delta}[\hat{m}])) \cup \mathcal{P}) \setminus (\hat{m} \cup \mathcal{R})$ ;
    if  $p = \emptyset$  then
         $U = \text{BOUND}(\hat{\delta}[\hat{m}])$ ;
        update  $currentbest$ ;
    else
         $\text{ABB}(p, \hat{m}, \hat{\delta}[\hat{m}])$ ;
    if  $U < \infty$  then
        print  $currentbest$ ;
    else
        print "There is no solution";
end

procedure  $\text{ABB}(p, m, \delta[m])$ 
begin
    legyen  $x \in p$ ;
     $C = \varphi(\Delta(x)) \setminus \{\emptyset\}$ ;
    for all  $c \in C$  do
        if  $\forall y \in m, c \cap \bar{\delta}(y) = \emptyset, (\Delta(x) \setminus c) \cap \delta(y) = \emptyset$  then
             $p' = (\psi^-(\varphi(\hat{\delta}[\hat{m}']))) \cup \mathcal{P} \setminus (\hat{m}' \cup \mathcal{R})$ ;
             $m' = m \cup \{x\}$ ;
             $\delta[m'] = \delta[m] \cup \{(x, c)\}$ ;
            Legyen  $\hat{\delta}[\hat{m}']$  a  $\delta[m']$  maximális neutrális kiterjesztése;
             $bound = \text{BOUND}(\hat{\delta}[\hat{m}'])$ ;
            if  $U \geq bound$  then
                if  $p' = \emptyset$  then
                     $U = bound$ ;
                     $currentbest = \hat{\delta}[\hat{m}']$ ;
                else
                     $\text{ABB}(p', \hat{m}', \hat{\delta}[\hat{m}'])$ ;
        end
    end

```

3.7. ábra. ABB algoritmus maximális neutrális kiterjesztéssel

3.3. Időkorlátos PNS

Folyamatok tervezésekor gyakran merülnek fel időbeli korlátok, legyen a folyamat akár üzleti, ipari vagy logisztikai. Ezen korlátok jelenthetnek például határidőt, szinkronizálási igényt különböző tevékenységek között. A különböző korlátokat folyamathálózat-szintézisben különböző módokon lehet figyelembe venni. A következő fejezetben tipikus korlát típusok kezelését tárgyaljuk.

3.3.1. Üzleti folyamatok szóhasználata

Mivel időkorlátok leggyakrabban üzleti folyamatokban merülnek fel, így ebben a fejezetben az üzleti folyamatok szóhasználatával élünk: az aktivitásokat nem műveleti egységeknek, hanem tevékenységeknek, az entitásokat pedig nem be- és kimenő anyagoknak, hanem előfeltételeknek és következményeknek nevezzük. Továbbá egy üzleti folyamathoz nem nyersanyagok és termékek, hanem erőforrások és céljai vannak. Az analógiát a 3.5-ös táblázatban foglaljuk össze.

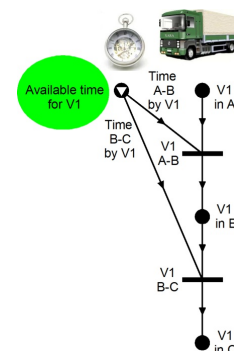
Ipari folyamatok	Üzleti folyamatok
nyersanyag	erőforrás
végtermék	végcél
műveleti egység	tevékenység
bemenő anyag	előfeltétel
kimenő anyag	következmény
műveleti egység terhelése	tevékenység volumene

3.5. táblázat. Ipari és üzleti folyamatok szóhasználata

3.3.2. Működési idő mint erőforrás

Talán a legegyszerűbb időkorlátos feladat, amikor a tevékenységek fixen vagy terhelésüktől függően időt vesznek igénybe, és tevékenységek egy halmazára adott, hogy összesen mennyi a felhasználható idő. Az összes felhasznált idő akkor jelent határidőt is egyben, ha a tevékenységek egymás után következnek be és indulásuk nem függenek más eseményektől, csak az előző tevékenység befejezésétől. Ilyen például egy autó körjárata, ahol az egyes utazások és rakodások követik egymást, és a körjárat teljes menetideje az egyes utazások és pakolások összes ideje.

Ebben az esetben a tevékenységek együttes határideje felírható egy közös erőforrásként, melyet mindegyik tevékenység felhasznál és van a felhasználásnak felső korlátja.



3.8. ábra. Időkorlát megadása erőforrásként

A 3.8-as ábrán a $V1$ gépkocsi teljes futási idejére adunk korlátot. Egyik tevékenységként A -ból B -be utazik, közben $Time_A - B$ mennyiséget használ el a rendelkezésre álló időből. Másik tevékenysége hogy B -ből C -be utazik, és $Time_B - C$ időt használ fel erre.

Az időkorlátot a idő, mint erőforrás felső korlátja adja meg. Ha a felhasznált idő függ a tevékenység terhelésétől, akkor a kettő arányát az adja meg, hogy egységnyi méretben a tevékenység az időből mennyit használ el. Ha egy tevékenységnek terheléstől független fix ideje van, akkor azt olyan tevékenységnek kell elhasználnia, amely méretének van alsó korlátja. Ekkor a fix időt a művelet alsó korlátjának és az általa egységnyi volumenben felhasznált időnek a szorzata adja meg.

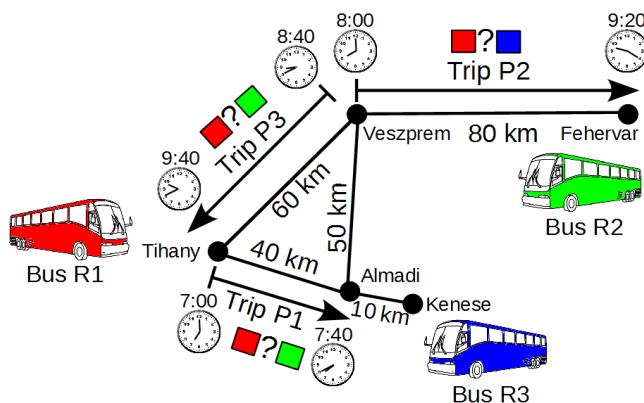
3.3.3. Ismert időzítésű tevékenységek

Az időkorlátos feladatok második osztályát jelentik az olyan folyamatok tervezése, ahol bizonyos tevékenységek ütemezése előre ismert, így kiszámítható, hogy mely tevékenységek követhetik adott sorrendben egymást és melyek nem. Tipikus példái az előre ütemezett feladatoknak a menetrendek, órarendek, munkarendek.

Általában adott az erőforrások rendelkezésre állásának időszaka és a folyamatban tevékenységként jelenik meg minden egyes erőforrás időzített feladathoz rendelése. A feladatok elvégzését pedig a folyamat céljaként határozzuk meg. Egy tevékenység elvégzésével egy erőforrás képessé válhat egy későbbi időpontban kezdődő tevékenység végrehajtására is, ekkor újabb potenciális tevékenység lehet az egyik feladatról egy másikra.

Ilyen feladat például a jármű hozzárendelés, ahol adott a menetrend időzített utazásokkal, és meg kell tervezni, hogy melyik jármű melyik utazásokat milyen sorrendben fogja végrehajtani. Ekkor nem csak azt kell figyelembe venni, hogy egy tevékenység előbb befejeződjön, mint a rákövetkező elkezdődik, hanem azt is, hogy ha egy tevékenység az *A* helyen fejeződik be, és a rákövetkező a *B* helyen kezdődik, akkor a két tevékenység közti időben még *A*-ból *B*-be is el lehessen jutni. Hasonló feltétel vizsgálata minden olyan munkafolyamatnál szükséges, ahol a feladatok végrehajtásának helyszíne távol eshet egymástól, így az idő jelentős részét a feladatok végrehajtása mellett a helyszínek közti utazás teszi ki.

a 3.9-es ábra egy jármű hozzárendelési feladatot szemléltet, ahol a *P1*, *P2* és *P3* túrákhoz kell egy-egy járművet rendelni az *R1*, *R2* és *R3* buszok közül. Az ábráról leolvasható a túrák kezdetének és végének a helye és ideje, hogy melyik busz tudja végrehajtani, és hogy melyik busz honnan indul. Ebben a feladatban a járműveknek nem kell visszatérnie az indulási helyükre. Ilyen feltételek mellett, a feladat folyamatszintézis feladatként átirásának eredményét a 3.10-es ábra mutatja.



3.9. ábra. Jármű hozzárendelési feladat grafikus leírása

3.3.4. Időben egymásra épülő tevékenységek

A legáltalánosabb, de egyben legnehezebben kezelhető időkorlát, amikor egyes tevékenységek időpontja változó, egynél több másik tevékenységtől is függhet, és a teljes folyamat végcéljaira fogalmazunk meg határidőket. A tevékenységek ideje lehet fix és a tevékenység volumenétől függő.

A feladat megadásához a folyamatszintézis két kibővítésére van szükség. Egyrészt bővíteni kell a paraméteres folyamatszintézis feladat [1] paramétereinek a körét, másrészt bővíteni kell a maximális struktúrát, hogy konkurens aktivitások szinkronizációja is kifejezhető legyen strukturális döntésekkel.

Tekintsük a folyamatszintézis 3.1-3.2-es fejezetben megismert paraméteres modelljét. A következő fejezetben ennek időkorlátokkal kibővített változatát mutatjuk be.

3.3.4.1. Időparaméterekkel bővített szintézis feladat

Az időkorláttal kibővített PNS (Time Constrained PNS, TCPNS) feladatban [2][3] az összes végcélú időben el kell érni, miközben az erőforrások időbeli rendelkezésre állása korlátozott, illetve az összes tevékenység időtartama a méretének fix részt tartalmazó lineáris függvénye. A fix részt tartalmazó lineáris költségfüggvényű paraméteres PNS feladatot négy paraméterrel egészítették ki az időkorlátok definiálására, amelyek a következők: a tf_i az egész részű, tp_i pedig az arányos részű komponense annak a függvénynek, amelyik becslést ad minden egyes tevékenység időtartamára a terhelésük alapján. Az Ut_j az egyes végcélok határidejét jelöli, amely a következőképpen definiálható:

$$Ut_j = \begin{cases} \geq 0, & \forall m_j \in \mathcal{P} \\ \max_{m_j \notin \mathcal{P}} \{Ut_j\}, & \text{máskülönben} \end{cases} \quad (3.4)$$

Az Lt_j pedig az egyes erőforrások legkorábbi rendelkezésre állását jelöli, amely a következőképpen definiálható:

$$Lt_j = \begin{cases} \geq 0, & \forall m_j \in \mathcal{R} \\ 0, & \text{máskülönben} \end{cases} \quad (3.5)$$

A cél az, hogy meghatározzuk azt az (m^*, o^*, x^*, z^*) hálózatot, ami kielégíti a 3.2 - 3.3-as és a 3.6 - 3.8-as egyenleteket és a z^* minimális, ahogy az a 3.1-es egyenletben látható.

Bármely m_j entitás kizárólag a legkorábbi Lt_j rendelkezésre állása és a Ut_j határideje közötti időszakban használható fel, azaz:

$$\forall m_j \in \mathcal{M} : Lt_j \leq t_{m_j} \leq Ut_j \quad (3.6)$$

Az $o_i \in o$ tevékenység t_{o_i} kezdőidőpontja nem előzheti meg semelyik m_j előfeltételének rendelkezésre állási t_{m_j} idejét, azaz:

$$o_i = (\alpha, \beta) \in o, \forall m_j \in \alpha_i : t_{o_i} \geq t_{m_j} \quad (3.7)$$

Egy o_i tevékenység bármely m_j következményének rendelkezésre állási t_{m_j} ideje nem előzheti meg a tevékenység t_{o_i} kezdő idejétől számított időtartamának $(tf_i + tp_i * x_i)$ befejeztét, azaz:

$$o_i = (\alpha, \beta) \in o, \forall m_j \in \beta_i : t_{m_j} \geq t_{o_i} + tf_i + x_i * tp_i \quad (3.8)$$

A TCPNS relaxált modelljében a 3.8-as egyenletet a 3.9-es és a 3.10-es egyenletek szerint relaxáljuk, ahol tb a leghosszabb határidőt jelöli a végcélokhoz definiált határidők között.

$$tb = \max_{m_j \in \mathcal{P}} \{Ut_j\} \quad (3.9)$$

$$o_i = (\alpha, \beta) \in \mathcal{I}, \forall m_j \in \beta_i : t_{m_j} \geq t_{o_i} + x_i * tp_i + y_i * (tb + tf_i) - tb \quad (3.10)$$

Az y_i bináris változó jelöli az o_i tevékenység struktúrába történő bevonását, ekkor az $y_i = 1$ értéket vesz fel, illetve kizárását, amikor az $y_i = 0$ értékű lesz.

A megoldás folyamatában mielőtt döntöttünk egy tevékenység bevonásáról vagy kizárásáról a folyamatból, mind a tevékenységek költségét, mind azon időigényét csak becsülni tudjuk.

$$y_i = 0 \Rightarrow t_{m_j} \geq t_{o_i} - tb \quad (3.11)$$

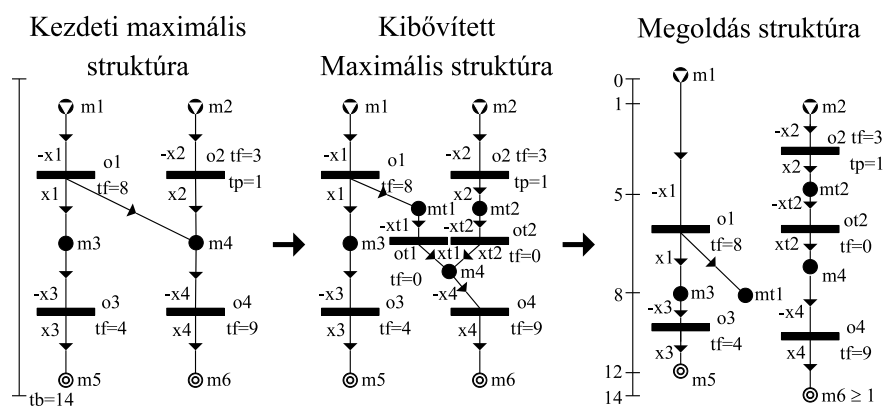
$$y_i = 1 \Rightarrow t_{m_j} \geq t_{o_i} + x_i * tp_i + tf_i \quad (3.12)$$

A becsléshez az y_i döntési változó relaxálható a $[0, 1]$ intervallumon, de ekkor is kapcsolatban van a tevékenység megengedett kapacitásával (ha egy tevékenység nem valósul meg, akkor csak nulla lehet a kapacitása), amely összefüggés a 3.13-as egyenlettel adható meg.

$$x_i \geq u_i * y_i \quad (3.13)$$

A 3.10-es egyenletben található egyenlőtlenség megegyezik a 3.11-es és a 3.12-es egyenletekkel függően attól, hogy az y_i egyenlő nullával vagy egyel. A 3.10-es egyenlet korlátai csak akkor vannak hatással, ha az y_i értéke közel van az egyhez a relaxált modellben.

3.3.4.2. Előidejűség kezelése



3.11. ábra. A szemléltető példa maximális struktúrájának előidejűség szerinti kiterjesztése az időkorlátok kezelésére

A 3.11-es ábrán látható példán fogjuk az újonnan bevezetett P-gráf módszertanbeli előidejűség kezelést szemléltetni. A feladat kombinatorikus komponensei egy hármassal $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ adhatóak meg, ahol

$$\mathcal{R} = \{m1, m2\}$$

$$\mathcal{P} = \{m5, m6\}$$

$$\mathcal{O} = \{o1, o2, o3, o4\} = \left\{ (\{m1\}, \{m3, m4\}), (\{m2\}, \{m4\}), (\{m3\}, \{m5\}), (\{m4\}, \{m6\}) \right\}$$

A feladat entitásainak paramétereit a 3.6-os táblázat, a tevékenységeinek paramétereit pedig a 3.7-es táblázat foglalja magába. Ahogy az a 3.6-os táblázatban látható kettő végcél tartalmaz a feladat, az $m5$ -öt, amit 13 óra alatt és az $m6$ -ot, amelyet 14 óra alatt kell teljesíteni.

Entitások	Típus	Legkorábbi rendelkezésre állás (Lt) [h]	Határidő (tb) [h]
$m1$	erőforrás	0	0
$m2$	erőforrás	1	0
$m5$	végcél	0	13
$m6$	végcél	0	14

3.6. táblázat. Az entitások paramétereit

Tevékenységek	Előfeltételek	Következmények	Fix idő (tf) [h]	Arányos idő (tp) [h]
$o1$	$m1$	$m3, m4$	8	0
$o2$	$m2$	$m4$	3	1
$o3$	$m3$	$m5$	4	0
$o4$	$m4$	$m6$	9	0

3.7. táblázat. A tevékenységek előfeltételei, következményei és idő paramétereit

A feladat bonyolultságát az adja, hogy minden entitás bevárja az összes őt eredményező tevékenység teljesülését. Az $m5$ végcél egyértelműen csak az $o1$ és az $o2$ tevékenységek által állhat elő, míg az $m6$ -hoz az $o4$ közbeiktatásával az $o1$ és az $o2$ tevékenységen keresztül is vezet út. Mivel az $o2$ -es tevékenységnek mindenképpen szerepelnie kell a megoldásban, hogy az $m6$ teljesíthető legyen és az $m4$ mind az $o1$ és mind az $o2$ tevékenységek befejeztét bevárja, mielőtt felhasználható lenne az $o4$ -es tevékenységhez, ezért csak a nyolcadik óra után használható fel az $o4$ -es által. Így viszont a $m6$ -os végcél soha nem állhatna elő a 14 órás határidőn belül, mivel az $o4$ -es tevékenység 9 órán át tart ($8 + 9 > 14$). Viszont, ha csak a $o2$ -es tevékenységet vennénk figyelembe az $m4$ -es entitás elérésénél, akkor teljesíthető lenne az $m6$ -os tevékenység határideje ($1 + (3 + 1) + 9 = 14$). Ezért az $m4$ -es entitás és az őt eredményező $o1$ és $o2$ tevékenységek be kell illeszteni az $mt1$ és az $mt2$ kiegészítő entitásokat és az $ot1$ és az $ot2$ kiegészítő tevékenységeket ahogy a 3.11-es ábrán látható. Így már szét tud válni úgy a hálózat, hogy az $o1$ befejezte előtt aktiválódhasson az $o4$, ahogy a 3.11-es ábra jobb oldalán a „Megoldás struktúra” részben látható. Tehát a fenti kiegészítések segítségével megoldhatóvá vált ez az előidejűség problémáját felvető feladat.

Összefoglalva elmondható, hogy a folyamatszintézis feladat a 3.3.4.1-es fejezetben leírt kiterjesztései csak a paraméteres feladat megoldása során használt LP modell kibővítését igénylik, az algoritmusok strukturális része nem változik. A 3.3.4.2-es fejezetben ismertetett kiegészítése a maximális struktúrának sem más, mint az előidejűség szemléltetése P-gráffal, ami egyben lehetővé is teszi, hogy PNS algoritmusok strukturális döntései az előidejűséget meghatározzák. Tehát ezen kiegészítések után a korábban megismert PNS algoritmusok változatlanul használhatóak.

Irodalomjegyzék

- [1] Bertok, B., M. Barany, and F. Friedler, Generating and Analyzing Mathematical Programming Models of Conceptual Process Design by P-graph Software, *Industrial & Engineering Chemistry Research*, 52(1), 166–171 (2013).
- [2] Kalauz, K., Z. Sule, B. Bertok, F. Friedler, and L. T. Fan, Extending process-network synthesis algorithms with time bounds for supply network design, *Chemical Engineering Transactions*, 29, 259–264 (2012).
- [3] Kalauz, K., PhD dolgozat, Pannon Egyetem (2013).

4. FEJEZET

Fenntarthatósági mértékek

Fenntarthatósági mértékek energiatermelő ellátási láncokban

Ebben a fejezetben a fenntarthatósággal, az energiatermelő ellátási láncokkal, és a kettő kapcsolatával foglalkozunk. Célunk egy módszertan bemutatása fenntartható energiaellátó rendszerek tervezésére. Amennyiben egy térségben több fajta megújuló energiahordozó érhető el, például fa, biogáz, stb., akkor ezek felhasználásának mikéntje komplex kombinatorikus feladat. Ennek a feladatnak a megfogalmazása és a megoldási módszer bemutatása a célunk.

Tegyük fel például, hogy egy burgenlandi régióban kukoricát termesztene, de számos erdőterület is található ott. A kukorica magból bio-etanol állítható elő, de azt takarmányként is fel lehet használni. Ekkor a megmaradt kukoricacsutkákból és szárból biogáz állítható elő erjesztéssel. Az állatok trágyáját szintén fel lehet használni erre a célra. A gázt kétfajta kazánban lehet elégetni, az egyik csak hőt, a másik hőt és elektromosságot is termel. A hőt el lehet adni, de egy részére szükség van a kukoricacsutka és a fa szárításához. A fát közvetlenül el lehet égetni, vagy előállíthatunk belőle pelletet illetve faaprítékot. A közvetlen égetéshez több manuális munka kell az adagolás miatt, a pellet és az apríték adagolása viszont automatikus, de az előállításukhoz áram szükséges. Jelen van még a rendszerben földgáz, amellyel a lakossági és egyéb hőigényeket bármikor el lehet látni, és az elektromos hálózat, amely az áramigényt tudja fedezni. Lehetnek még más, itt le nem írt lehetőségek, és adottak az egyes technológiákhoz kapcsolódó technikai és költségparaméterek. Ezt a feladatot fejben nem lehet optimálisan megoldani, hiszen nagyon sok az egymásra hatás, a feltétel és a lehetséges elrendezés. Szisztematikusan kell meghatározni minden lehetőséget, és azokat adott szempont szerint kell elemezni.

Olyan módszert szeretnénk megvalósítani, amely képes az optimális struktúra mellett számos más alternatíva meghatározására is, így a döntéshozók olyan feltételeket is figyelembe tudnak venni, amelyek az eredeti modellben nem voltak megfogalmazva. A módszer következő újdonsága, hogy az optimalizálás során nem csak a költséget veszi figyelembe, hanem több más fenntarthatósági mértéket is számol modell szinten. Így például az "emergy"-t vagy az ökológia lábnyomot. Ez lehetővé teszi, hogy a megoldásokból meghatározzuk a Pareto frontot, vagy, hogy ezen értékekre fogalmazzunk meg feltételeket.

A fejezet célja új módszertan bemutatása fenntartható energiaellátó rendszerek tervezésére. További cél a helyben illetve a régióban található erőforrások minél hatékonyabb felhasználása. Fenntartható energia ellátó rendszerek esetében a fosszilis energiahordozók használatát a minimálisra kell szorítani, és fokozatosan egyre nagyobb szerepet kell adni a megújuló energiahordozóknak. Ez nemcsak környezetvédelmi kérdés, hanem hosszabb távon fontos gazdasági, politikai kérdés is, hiszen így jelentősen csökken a külföldi energiahordozóktól való függés. A régiókat azért hangsúlyozzuk külön, mert a helyi energiahordozókat lehet leggazdaságosabban felhasználni. Például egy szénátüzelésű erőműnek a begyűjtési távolsága nem lehet túl nagy, mert ellenkező esetben a szállítás, olyan mértékű környezetszennyezéssel jár, amely kioltja a megújuló energiahordozó használatának az előnyét. Célunk annak eldöntése, hogy a számtalan egymással versengő energiahordozó illetve energiaátalakítási technológia közül melyeket érdemes használni.

4.1. A fenntarthatóság fogalma

A fenntarthatóság napjaink egyik kiemelten fontos kutatási területe. Az a felismerés ad neki különös fontosságot, hogy bolygónk erőforrásai végesek, ezért azokat a lehető legnagyobb körültekintéssel kell felhasználni. Egyre több ember fogadja el, hogy környezettudatosság nem csak egy jól csengő divatirányzat, hanem szükségszerűség. Hiába termelünk most hatalmas profitot, ha a gyerekeinknek nemcsak nyersanyagok, de még tiszta víz és levegő sem jut majd. A fenntarthatóság hétköznapi nyelven megfogalmazott definíciója, az Egyesült Nemzetek Szervezetének 1987-es Brundtland jelentésében található:

„Kielégíteni a jelen szükségleteit anélkül, hogy csökkentjük a jövő generációk azon képességét, hogy kielégítsék a saját szükségleteiket.”

A fenntarthatóság és a környezetvédelem egymással összefüggő, egymást kiegészítő fogalmak. A környezetvédelem arra törekszik, hogy a ne károsítsuk az erdőket, a levegőt, a vizet és a termőföldet. Ennek érdekében különféle kibocsátási szabályozásokat vezettek be. Megszabták például, hogy mennyi lehet a benzin kéntartalma, mekkora lehet a kipufogógáz koromtartalma, illetve mekkora lehet egy gyártási tevékenység során képződő szennyvíz szennyezőanyag koncentrációja. A környezetvédelem fogalma jóval korábban keletkezett, mint a fenntarthatóságé. Már az ipari forradalomban, a 19. században, szembe kellett nézni azzal, hogy a nagyobb volumenű termelés, káros hatásokkal, például szmoggal, is jár. Később felismerték, hogy ezek a hatások nem csak a közérzetet, hanem magát a termelést is károsan befolyásolják.

A fenntarthatóság annyival több, mint a környezetvédelem, hogy az előbbi nem engedi meg, hogy a problémák kezelését a bizonytalan jövőre halasszuk. Földünk hatalmas, így számos fenntarthatósági probléma jelenleg nem okoz gondot, mivel hatalmasak a tartalékok. Ha egy erdőt kivágtak, akkor ott van egy másik, ha egy olajkút kimerült, akkor feltárnak egy újat, ha a szemétkukák hely megtelt, akkor betemetik és nyitnak egy újat. A fenntarthatóság azt hangsúlyozza, hogy ne halogassuk a problémák megoldását, hanem kezeljük őket felelősen most a jelenben.

Azzal is tisztában kell lenni, hogy a fejlett világban jelenleg semmi sem száz százalékgig környezetbarát vagy fenntartható. Egy napelem-cella gyártásához például szükség van ritkafémekre, amelyekből korlátozottak a készletek, az újrafeldolgozásuk pedig nem oldható meg költséghatékonyan. Vagyis a napelem, amit fenntarthatónak hívunk, mégsem az teljes mértékben. A célunk tehát a lehető legfenntarthatóbb rendszerek, technológiák létrehozása.

4.2. A fenntarthatóság mérése

Számos nemzetközi kutatócsoport foglalkozik azzal, hogy a fenntarthatóság általános fogalmát, pontosítsa úgy, hogy különböző mértékeket definiálnak, amelyek számszerűen bemutatják, hogy egy termék, vagy szolgáltatás mennyire fenntartható. Ezeknek a mértékeknek jelentős tudományos irodalma van. Egy objektív mérték nélkül nem lehet például összehasonlítani, hogy két ellátási lánc közül melyik a fenntarthatóbb. A probléma az, hogy a téma újszerűsége miatt még nincsenek széleskörűen elfogadott mérőszámok vagy módszertan. Számos mérték ismert ugyan, azonban mind-egyiknek megvan a saját szabályrendszere, mindegyiket folyamatosan fejlesztik, például létrehozhatnak adatbázisokat, de csak az idő fogja eldönteni, hogy melyik lesz közülük általánosan elfogadott. A helyzetet tovább nehezíti az, hogy a különféle mértékek különböző esetekben hatékonyak, így talán nem is létezik egy minden szempontból legjobb mérték. Sokan foglalkoznak azzal, hogy ezeket a mértékeket összegyűjtsék, rendszerezék és valamilyen egyesített mértéket definiáljanak. Ez is egy nagyon nehéz feladat, hiszen valamilyen módon meg kell határozni, hogy az egyesített mérték milyen súlyokkal veszi figyelembe az elemi mértékeket. Ebben a fejezetben a mértékek közül bemutatjuk a Fisher információt, az SPI-t, az ökológiai lábnyomot és az emergiát. Az utóbbi kettőt fogjuk beleépíteni a P gráf módszertanba.

4.3. Fenntarthatóság ellátási láncokban

A fenntarthatóság témakörében az ellátó rendszerek fenntarthatósága egyre nagyobb figyelmet kap. Az ellátási lánc (supply chain) egymással összefüggő folyamatok és erőforrások összessége, amely az alapanyagok beszerzésével kezdődik és a végtermék fogyasztóhoz való leszállításával végződik. Résztét képezik a beszállítók, gyárak, logisztikai szolgáltatók, raktárak, disztribútorok, nagykereskedők és minden egyéb résztvevő, amely hozzájárul a végtermék fogyasztóhoz való eljuttatásához. Szélesebb perspektívában beleértendők a beszállítók beszállítói és a vevők vevői is. Vagyis az ellátási lánc nem csak egy egyszerű termékre, egy darab gyártási folyamatra koncentrál, hanem a termelést annak teljességében vizsgálja.

Az energiatermelő ellátási láncok az ellátási láncok egy speciális területe, amely az energiatermeléssel, a különböző energiatípusok átalakításával foglalkozik. Hagyományosan az ellátási láncok arra a kérdésre keresik a választ, hogy mit honnan kell szállítani, hol kell megvenni, illetve eladni. Lényegében egy nagyméretű optimalizálási feladat megoldása a cél, ahol biztosítani kell az anyagegyensúlyokat, és a lehető leggazdaságosabb megoldást kell azonosítani. A szakirodalom egy kisebb szelete külön foglalkozik a különféle termékek, termékcsoporthoz, iparágak specifikus ellátó rendszereivel. Az energiaellátó és termelő rendszerek a mindennapi életben alapvető fontosságúak. Ezt a területet a fenntarthatóság, mint plusz feltétel megjelenése, rendkívüli módon befolyásolja.

Energiaellátó rendszerek esetén különösen fontos a rendszerszemlélet, a különböző folyamatok együttes vizsgálata, mert csak így lehet hitelesen összehasonlítani két különböző energiatermelő ellátási láncot. Mint minden ellátási láncban, itt is meg kell határozni, hogy hol vannak a rendszer határai. A vizsgált példákban a rendszer bemenetei az elsődleges energiahordozók, mint a földgáz, a fa, a kukorica, vagy az elektromos hálózathoz elérhető áram. A rendszer kimenete pedig adott mennyiségű hő és elektromosság.

A koncepcionális tervezés megvalósításához a P-gráf módszertant fogjuk felhasználni. A módszertant Friedler és munkatársai dolgozták ki folytonos üzemű vegyipari rendszerek szintézisére. Ezt a módszert azóta kiterjesztették számos más tématerületre: például molekulatervezés, szétválasztási hálózat szintézis, hibadetektálás, stb. Ezek az eredmények megmutatják, hogy a módszertan kellően rugalmas. Célunk, hogy egy újabb területre terjesszük ki ezt a módszertant.

Az optimális energiatermelő ellátási láncok meghatározásához fel fogjuk használni a P-gráf módszertanhoz kifejlesztett szoftver eszközöket: PNS Solutions, (Varga et al., 2010), PNS Draw, PNS Studio.

4.4. Szakirodalmi áttekintés

A fenntarthatóság, illetve a fenntarthatóság az energiatermelő ellátási láncokban, a tudományos szakirodalomban is egyre nagyobb figyelmet kap. Hassini et al. (2012), Gupta et al. (2011), és Seuring et al. (2008) részletes áttekintést adtak erről a témakörrel. Chaabane és szerzőtársai (2012) matematikai programozási modell segítségével vizsgálták és optimalizál ellátási láncokat az alumíniumipar területén. A módszer egyensúlyt próbál teremteni a költség, és a szennyezőanyag kibocsátás között. Külön hangsúlyt kap a szennyezés kibocsátási kvótarendszer vizsgálata, és hatásainak az elemzése. Winkler (2011) a teljesen zárt termelés fontosságát hangsúlyozta az úgynevezett "tisztítás a rendszerből való kilépéskor" elv helyett. Azt a következtetést vonja le, hogy egy cég önmagában, még ha jelentősebb is, nem képes a zárt termelést költséghatékonyan megvalósítani, de cégek csoportjára ez már igaz lehet. Büyüközkan és Berkol (2011) egy szakértői rendszert valósítottak meg fenntartható ellátó láncok tervezésére. A rendszer több különböző, a gyakorlatban bizonyos feladatokra hatékony módszert használ fel. Ilyen a minőségfüggvény alkalmazás, az analitikus hálózati feldolgozás és a bináris célprogramozás. Brito és szerzőtársai (2008) a ruha- és textil ipart, azon belül a minőségi termékek ellátó láncát vizsgálta. Az ipar szereplőivel interjúkat készített, és ezek segítségével feltérképezte az adott terület a sajátosságait és kihívásait. Részletes irodalomjegyzéket gyűjtött össze, amely tartalmazza az eddigi eredményeket, valamint megvizsgálta, hogy a fenntarthatóság, hogyan alakítja át az eddigi ellátó láncokat.

Az energiaellátó rendszerek területén a folyamathálózat szintézisnek nagy szerepe van. Friedler és szerzőtársai (1993) bemutatták a folyamathálózat szintézis témakörébe tartozó P-gráf módszertant

alapját. A keretrendszer egy új reprezentálási módszeren, strukturális tulajdonságokat leíró axiómákon és hatékony algoritmusokon alapszik. Friedler és szerzőtársai (1995) bemutatták a P gráfok ábrázolásának egy formális módját. Ez az ábrázolás lehetővé teszi, a P-gráfokat felhasználó algoritmusok hatékony implementálását. Heckl és szerzőtársai (2010) a P-gráf módszertant kiterjesztették a szétválasztási hálózatok szintézisére. Ez a munka világosan mutatja, hogy a P-gráf módszertan egy általános eszköz a folyamathálózat területén, amelyet kellő körültekintéssel, nagyon sokféle esetben lehet alkalmazni. Célunk, hogy a módszertant egy újabb területre, a fenntartható energiaellátó rendszerekre is kiterjesszük.

4.5. A módszer lépéseinek rövid bemutatása

Egy energiatermelő ellátási lánc meghatározása több, egymással összefüggő lépésből áll. Először röviden bemutatjuk ezeket a lépéseket és a lépések kapcsolatait, később részletesen kifejtjük a lépéseket.

A probléma megoldása a következő részfeladatokból áll: Az adatgyűjtés során feltérképezzük a fenntartható energiaellátó rendszerek fő tulajdonságait, információkat gyűjtünk a jellemző energia átalakító technológiákról, és igyekszünk ezeket strukturáltan tárolni. Hasonló fontosságú a fenntarthatóság különböző jellemzőinek a feltérképezése is. Döntést kell hozni arról, hogy a különböző mértékek közül melyek a leghasznosabbak számunkra. A következő részfeladat a strukturális modell megalkotása a P-gráf módszertan segítségével. A modellt fokozatosan fogjuk felépíteni, egyre több opciót hozzáadva. A modell tesztelése során ellenőrizzük, hogy a megoldás megegyezik-e az előzetes feltevésünkkel és szükség esetén javítjuk a hibákat, finomítjuk a modellt és megállapítjuk az alkalmazhatóságának a határait. A konkrét részfeladatok a következők.

4.5.1. Adatgyűjtés a feladathoz kapcsolódóan

A legelső feladat az esettanulmányhoz szükséges paraméterek összegyűjtése. Az iparban jelenleg is használt rendszerek különböző tulajdonságai, például az ár, sokszor nem nyilvános, vagy egyenesen üzleti titok. E nehézségek ellenére különböző nyilvános adatforrásokból össze kell gyűjteni, annyi információt, amely egy közepes méretű gyakorlati példához elégséges. Adatokat kell szerezni a különböző anyagokhoz illetve az átalakítási technológiákhoz, azaz műveleti egységekhez. Figyelembe kell venni, hogy az adatok sokszor nem ugyanahhoz az évhez tartoznak. Az anyagokhoz tartozó információk könnyebben elérhetőek, de a műveleti egységek esetén becslő eljárást kell kidolgozni, a hiányzó értékek meghatározására.

Probléma, hogy a P-gráf módszertan szerint a műveleti egységek mérete tetszőlegesen változtatható, vagyis egy folytonos, pozitív változóval írható le. Ez a gyakorlatban sokszor nem igaz, például egy reaktorokból általában csak diszkrét méretek szerezhetőek be. A probléma kezelhető linearizálással, de ekkor megoldás után egy korrekció eljárás segítségével, módosítani kell a kapott változó értékeket.

4.5.2. A strukturális modell átalakítása: költség nyersanyagként történő reprezentálása

A P-gráf módszertan szerint költség paraméter a nyersanyagokhoz, potenciálisan a köztes anyagokhoz, negatív előjellel a termékekhez és a műveleti egységekhez kapcsolódhat. Célunk, hogy a költség csak egyik jellemzője legyen a javasolt megoldásoknak. Bemutatunk egy módszert, amelynek a segítségével a műveleti egységekre vonatkozó költségek (beruházási költség fix része, beruházási költség arányos része, működési költség fix része, működési költség arányos része) anyagpontként ábrázolható a P-gráf módszertanon belül. Az új formalizmus alkalmas arra, hogy teljes költségre vonatkozó megszorításokat fogalmazzunk meg magában a PNS feladatban.

4.5.3. *P-gráf modell kibővítése a nyersanyagokra vonatkozó ökológiai lábnyommal*

A következő feladat meghatározni, hogy egy adott megengedett megoldáshoz, mekkora ökológiai lábnyom tartozik. Az ökológiai lábnyom egy fenntarthatósági mérték. Bemutatjuk azt az eljárást, amelynek segítségével tetszőleges hálózat ökológiai lábnyoma számolható. A módszer nem lehet utólagos, vagyis az nem kielégítő, ha a megoldás meghatározása után számoljuk ki a kívánt értéket. Cél, hogy már az optimalizálás folyamán elérhető legyen ez az érték, így hatékonyabb legyen a megoldás folyamata, és lehessen korlátozó feltételt megadni az ökológiai lábnyomra vonatkozóan.

4.5.4. *P-gráf modell kibővítése a nyersanyagokra vonatkozó emergy feltételekkel*

Az emergy szintén egy a fenntarthatóságban vizsgált mérőszám, ami az exergy-vel van szoros kapcsolatban. A terület alapú mérőszámok, mint például az előző pontban bemutatott ökológiai lábnyom, nem minden esetben adnak a gyakorlatban is összehasonlítható értékeket, ahogy például földgáz esetében sem. Bemutatjuk, hogy hogyan lehet emergy alapú feltételeket is megfogalmazni az adott modell keretein belül.

4.5.5. *Modellezési eljárás kidolgozása fosszilis és megújuló energiahordozók felhasználásának arányára*

A fenntarthatósághoz kapcsolódó hasznos érték a megújuló és a fosszilis energiahordozók aránya. Ismertetünk egy modellezési eljárást az említett feltétel megvalósítására. Ez nem olyan egyszerű, mint egy általános MINLP (mixed integer nonlinear programming) modell esetén, ahol a meglévő feltételek után csak be kell illeszteni egy újat. A P-gráfon alapuló módszer képes kihasználni a feladat strukturális tulajdonságait, így a futási ideje kisebb, viszont a plusz feltétel megadáshoz egy modellezési trükköt kell bevezetni. Az utóbbi két lépés nem feltétlenül szükséges a módszer végrehajtásához, így azokat nem fogjuk részletezni.

4.6. Sustainable process index

Az SPI (sustainable process index) egy fenntarthatósági mérték, amely azt fejezi ki, hogy egy termék vagy szolgáltatás kvantitatív módon mennyire fenntartható. Az SPI megalkotóinak (Krotscheck, Narodslawsky, 1996) elsőként azt a kérdést kellett megválaszolniuk, hogy milyen mértékegységet használjanak a fenntarthatóság mérésére. A választás a területre esett. Egy fenntartható társadalomban az egyetlen hosszútávon felhasználható energiaforrás a napenergia. Minden más energiaforma például a szél, a hullám, a fa, a biogáz és a fosszilis energiahordozók, a napenergia átalakult formái. Három kivétel van csupán: a nukleáris energia, az árapály energia és a geotermikus energia. A napenergia használata a földfelszínhez kötött, mi több, a terület egy korlátozott erőforrás egy fenntartható társadalomban, hiszen bolygónk nagysága adott.

Bolygónk képes a megújulásra, de ehhez idő és terület kell. Tegyük fel, hogy az idő adott, méghozzá egy év. Ekkor meghatározhatjuk, hogy mekkora terület szükséges ahhoz, hogy feldolgozza azokat a kipufogógázokat, amelyeket például egy autó egy kilométer megtétele közben kibocsát. Egy másik példa növénytermesztés. Tegyük fel, hogy búzát termesztünk, amihez nem csak az aktuális termőterületre van szükség, hanem valamennyivel többre, hiszen egy bizonyos részt parlagon kell hagyni, hogy ne merüljön ki a termőréteg. Hasonló módon terület kell a művelés során beszennyezett levegő és víz regenerálásához is.

A területnek mindkét funkcióját figyelembe vesszük, vagyis azt, hogy fogadja a napenergiát és azt is, hogy a megújulásnak az egyik tényezője. Az SPI képes egy folyamat ökológiai hatásának mérésére és összehasonlítására az alapján, hogy az mekkora energia és anyagáramokat indukál. Azok a folyamatok, amelyek több területet igényelnek ugyanannak a terméknek vagy szolgáltatásnak az előállításához, kevésbé kívánatosak a fenntartható gazdaságban.

4.6.1. Az SPI számolása

Azért, hogy meghatározzuk egy folyamat SPI értékét, meg kell határozni a folyamathoz tartozó anyag és energia áramokat. A termelő eszközöket és az ezeket működtető munkásokat külön-külön vesszük figyelembe. Az SPI számolása az A_{tot} meghatározásával kezdődik, amely egy olyan érték, ami megadja, hogy mekkora terület kell ahhoz, hogy egy folyamat fenntartható módon beépülhessen az ökoszférába:

$$A_{tot} = A_R + A_E + A_I + A_S + A_p + A_D \quad [\text{m}^2] \quad (4.1)$$

Az előbbi képletben A_R jelenti a nyersanyagok területszükségletét, A_E az energiához tartozó terület érték, A_I a berendezések terület értéke, A_S a munkásokhoz tartozó terület érték, A_p a termékekhez illetve melléktermékekhez tartozó érték és A_D a disszipáció értéke. Ezeket az értékeket a referencia periódusra, vagyis egy évre számolják. Ebben az évben S_{tot} számú egységet (terméket vagy szolgáltatást) hoz létre az adott folyamat. a_{tot} jelöli a fajlagos (fenntartható) szolgáltatási területet. Y_{tot} -ot úgy definiáljuk, hogy az a fajlagos szolgáltatási terület reciprokja.

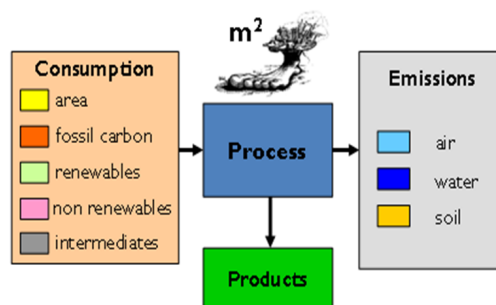
$$a_{tot} = A_{tot} / S_{tot} = 1 / Y_{tot} \quad [\text{m}^2 / \text{év egység}] \quad (4.2)$$

A fajlagos terület önmagában egy lehetséges mértéke a fenntarthatóságnak. Azért, hogy hatásosabb mértéket kapjunk, a fajlagos területet elosztjuk az adott régióra jellemző egy főre jutó területtel (a_{in}). Ez a terület az egy lakosra jutó elméleti átlagérték, amelyet figyelembe lehet venni a termékek és energia előállításánál.

$$SPI = a_{tot} / a_{tin} \quad (4.3)$$

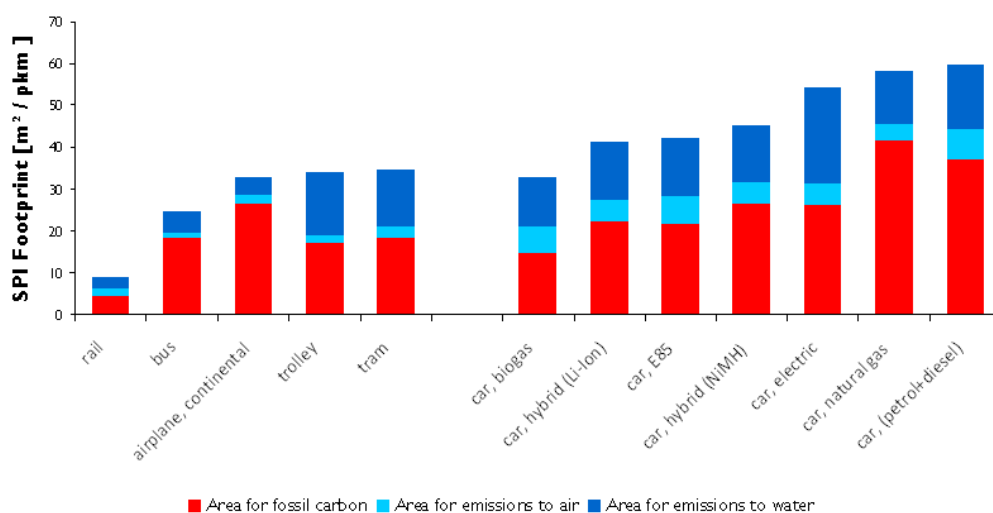
Az SPI egy olyan érték, amely az előbbi két terület arányán alapszik. A cél az, hogy ellássunk egy lakost adott termékkel vagy szolgáltatással. Az első terület azt fejezi ki, hogy mekkora terület szükséges ahhoz, hogy a kérdéses előállítás fenntartható módon illeszkedhessen az ökoszférába. A második terület a statisztikailag egy főre jutó terület. Az SPI tehát az előállításhoz szükséges és a rendelkezésre álló terület hányadosa.

Az SPI meghatározásánál fontos, hogy annak értéke miből adódik. Ezért az SPI értéket gyakran szétbontják hét alkotórészre, amelyek a következők: terület, fosszilis energiahordozók, megújuló anyagok, nem megújuló, levegő-, víz- és talajszennyezés.

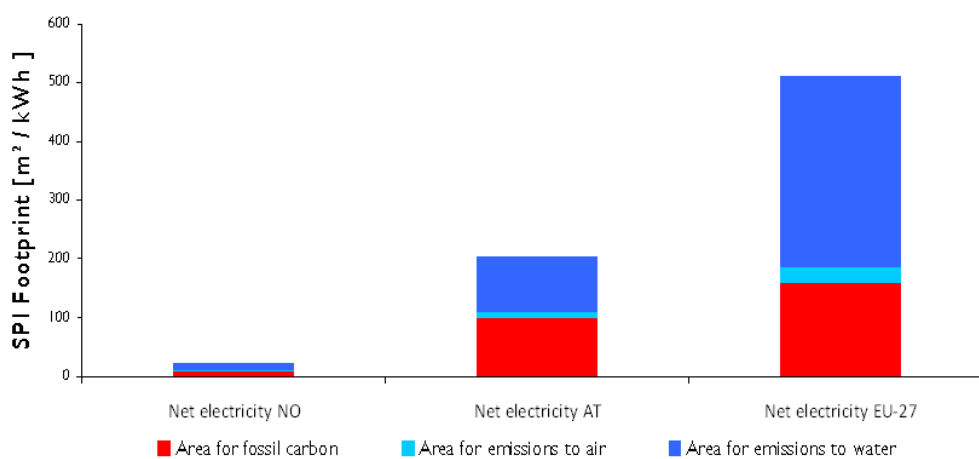


4.1. ábra. Az SPI számolás alkotóelemei

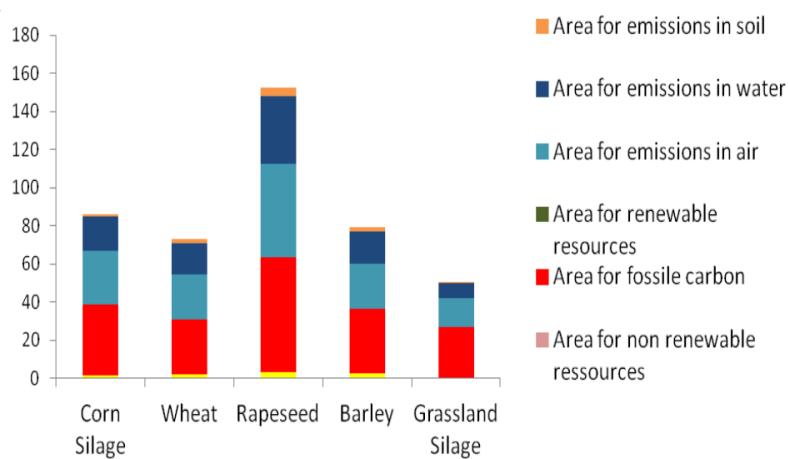
A gyakorlatban kiderült, hogy sokszor célszerűbb az SPI érték helyett az a_{tot} -ot használni. A módszeren és a mérték már SPI-ként ismert mindenhol, ezért sokszor a_{tot} helyett az SPI elnevezést használják. A 4.2-4.5-ös ábrákon néhány szolgáltatás és termék SPI értéke látható. Fontos emlékezni arra, hogy az SPI egy adott területre vonatkozik. Például nem mindegy, hogy egy elektromos meghajtású autóhoz hol termelték az áramot. Minden országban más az energia mix, így a hozzá tartozó SPI is.



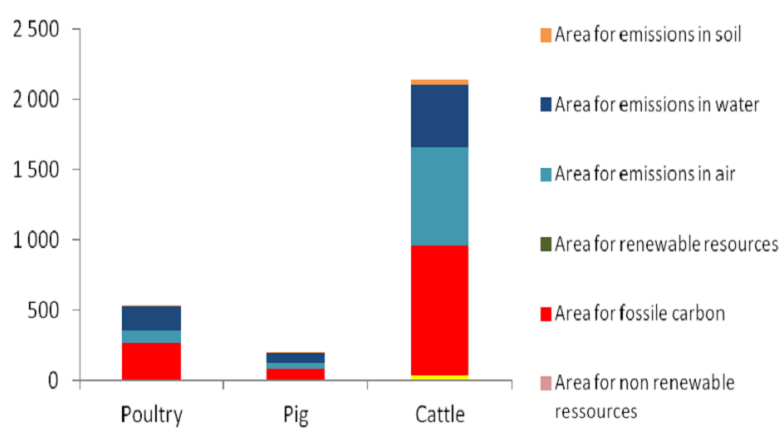
4.2. ábra. Különböző közlekedési módokhoz tartozó SPI érték egy kilométerre vonatkoztatva.



4.3. ábra. Egy kWh előállításához szükséges SPI különböző országokban.



4.4. ábra. Mezőgazdasági termékekhez tartozó SPI értékek.



4.5. ábra. Állattenyésztéshez tartozó SPI értékek (ide beépült a takarmány SPI értéke is; a disznó és a szárnyasok húskihozatala sokkal kedvezőbb, mint a szarvasmarháé).

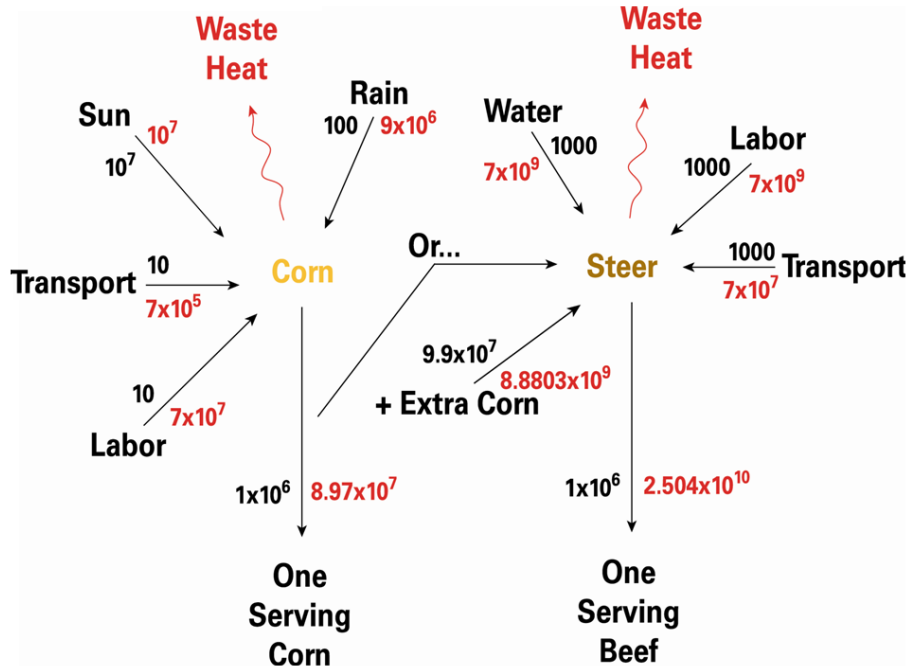
4.7. Emergia

A cím nem elírás, nem az energiáról van szó. Az emerggia analízis egy olyan módszertan, amely egyenként számításba veszi az összes olyan napenergia igényt, amely egy termék vagy szolgáltatás előállításához – közvetve vagy közvetlenül - szükséges. Az emerggia az összes termodinamikai munka, amelyet felhasználtak ahhoz, hogy egy terméket vagy szolgáltatást, esetünkben ellátási láncot létrehozzanak. A be és kimeneti anyagáramokat egy átváltási faktor, az úgynevezett szoláris transzformitás segítségével váltják szoláris energiára (seJ). Az emerggia kifejez egyfajta energiaminőséget is. Például egy joule hőenergia emerggiája kisebb, mint egy joule elektromos energiáé, mert egységnyi elektromos energia előállításához több napenergia szükséges, mint egységnyi hőenergia előállításához. Azt a folyamatot tekintjük fenntarthatóbbnak, amelynek kisebb az energiafogyasztása.

Az emerggia analízist hivatalosan nem tekintik fenntarthatósági mértéknek, a gyakorlatban mégis ilyen szerepet tölt be. Az emerggia analízis során nem csak a természeti, hanem az emberi gazdaság tényezőit is figyelembe kell venni, az emerggia összekapcsolja a gazdasági és az ökológiai rendszereket. Ez a fajta megközelítés nem mentes a nehézségektől, mert amikor egy rendszer számos bemenettel és kimenettel rendelkezik, akkor nehéz meghatározni, hogy a bemeneti emerggiákat, hogyan kell szétosztani a kimenetek között.

Az emerggia analízis elméleti alapja az, hogy a Földön az energia alapvetően a következő forrásokból származhat: napenergia, árapály energia, geotermikus energia, nukleáris energia. Ezen források közül messze a napenergiáé a legnagyobb részesedés. A beérkezett napenergia többszörösen alakul át geológiai és biológiai folyamatokban. Például a növények a fejlődésükhöz napenergiát használnak fel, majd miután elpusztulnak, sok-sok év alatt kőolaj, földgáz vagy szén képződhet belőlük. A napenergia azonban szélenergiává is átalakulhat az általa felmelegített levegő segítségével. Az emerggia számolásánál figyelembe kell venni a helyi adottságokat, így például egységnyi szalma emerggiája nem ugyanannyi Magyarországon, mint az USA-ban.

Az emerggia fogalmát talán az exergiával való összehasonlítás világíthatja meg. Az exergia az termodinamikai munka, amelyet valamilyen közeg végezni tud, miközben egyensúlyba kerül a környezetével. Például egy pohár forró teának nagyobb az exergiája az északi sarkon, mint a sivatagban. Fenntarthatóság szempontjából annál jobb a folyamat, minél kisebb az exergia változás. Ezzel szemben az emerggia az összes munkát tartalmazza, ami egy adott dolog előállításához szükséges. A 4.6 ábrán egy adagnyi kukorica és marhasült exergiája és emerggiáját (pirossal szedve a jobb oldalon) hasonlítjuk össze (Mayer et al., 2004). Mindkét fogással nagyjából hasonló mértékben lehet jól lakni, segítségével a szervezet hasonló mennyiségű fizikai munkát tud végezni, így a két étel exergiája nagyjából megegyezik (10^6). Viszont a marhasült előállításához sokkal több energia kell, például rengeteg takarmányra van szüksége a marhának, így a marhasült emerggia értéke jóval nagyobb lesz.



4.6. ábra. Egy adagnyi kukorica és marhasült exergiája és emergiája.

4.8. Ökológiai lábnyom

Az ökológiai lábnyom egy fenntarthatósági mérték, amely azt fejezi ki, hogy mekkora terület szükséges ahhoz, kielégítse adott termék, szolgáltatás, emberi közösség szükségleteit, oly módon, hogy lehetővé teszi a szükséges termelést és a hulladék környezetbe való visszaforgatását. Ez a mérték lényegében segít megmutatni, hogy egy közösség illetve a gazdaság, addig nyújtózott-e, amíg a takarója ért, annyi terület ténylegesen rendelkezésére áll-e, mint amennyire szüksége van. Ezt a mértéket eredetileg globális szinten használták és megállapították, hogy már 1977-ben a föld népessége, az akkori fogyasztási szinten, több mint egy Földnyi területet igényelne, ha fenntarthatóan akar működni. A későbbiekben a mérték számos módosításon ment keresztül, ami lehetővé tette, hogy kisebb, akár egyedi szinten is alkalmazni lehessen.

Az ökológiai lábnyom egy összesített, terület dimenziójú mérték, de az is hasznos információ, hogy milyen típusú területekből tevődött össze a végső érték. Hét típust különböztetünk meg:

- művelhető terület
- erdő
- legelő
- tenger
- energiatermelésre, és -hasznosításra fordított terület
- beépített terület

Az ökológiai lábnyom egy additív mérték, ami azt jelenti, hogy egy rendszer egészéhez tartozó lábnyomot megkaphatjuk a részeihez tartozó lábnyomok összegeként. Az ökológiai lábnyom – mint mérték - használata egyre inkább elterjed, mert a természeti hatásoknak nem csak egy-egy aspektusát, hanem összességét veszi figyelembe, nem úgy mint a széndioxid lábnyom vagy a víz lábnyom.



4.7. ábra. Ökológiai lábnyom reprezentálása

4.9. Műveleti egységek új reprezentálása

Célunk fenntartható energiatermelő ellátási láncok tervezése. Ehhez szükséges, hogy a P-gráfon alapuló modellezési technikát kiterjesszük úgy, hogy figyelembe tudjunk venni a multi-periódusos műveleti egységeket is. A több-periódusos működés lényege, hogy a peremfeltételek minden periódusban állandóak, de a különböző periódusok eltérhetnek egymástól, például az egyikben több terméket kell előállítani, mint a másikban.

4.9.1. A multi-periódusos működés jellemzése

A vegyiparban a folyamatos rendszerű termelés (a batch alapú termeléssel szemben) igen elterjedt. Ekkor a működési feltételek a gyártás megkezdése után nem változnak, így ezt lehet egy periódusú működésnek tekinteni. A P-gráf módszertan eredeti definíciója szerint egy műveleti egység folyamatosan, teljes (100%) terhelésen működik. Ennek oka, hogy eleve úgy lett tervezve, hogy pontosan teljesíteni tudja az elvárt mennyiségű kimenetet, amelynek nagyságát szintén állandónak tekintjük. Mivel az egység mérete és terhelése sosem vált el egymástól ezért nem is különböztettük meg őket, hanem a kapacitás kifejezést használtuk.

A folyamatos, állandó nagyságú terhelés gyakori a vegyiparban, de más területeken, például mezőgazdaság, élelmiszeripar egyáltalán nem ez a helyzet az évszakok szezonális hatása miatt. Ha elegendő raktárkapacitás áll rendelkezésre, illetve a tárolt anyag nem romlandó, akkor a műveleti egység továbbra is lehet egy periódusú, mert a raktározás el tudja simítani az igényekben jelentkező fluktuációt, így elegendő az évi átlagra tervezni. Ha a raktározás nem oldható meg gazdaságosan, akkor multi-periódusú működésre kell felkészülni. Ekkor a legyártandó termék mennyisége periódusról periódusra változhat. Sőt, esetlegesen a felhasznált bemenetek és a keletkezett kimenetek is változhatnak az egyes periódusok között.

Egy kisméretű szemléltető példát mutatunk be a multi-periódusos működésre három periódussal. A költség adatokat a következő egyenletekben mutatjuk be:

$$cc = 140 + 20m \quad [\text{\$}] \quad (4.4)$$

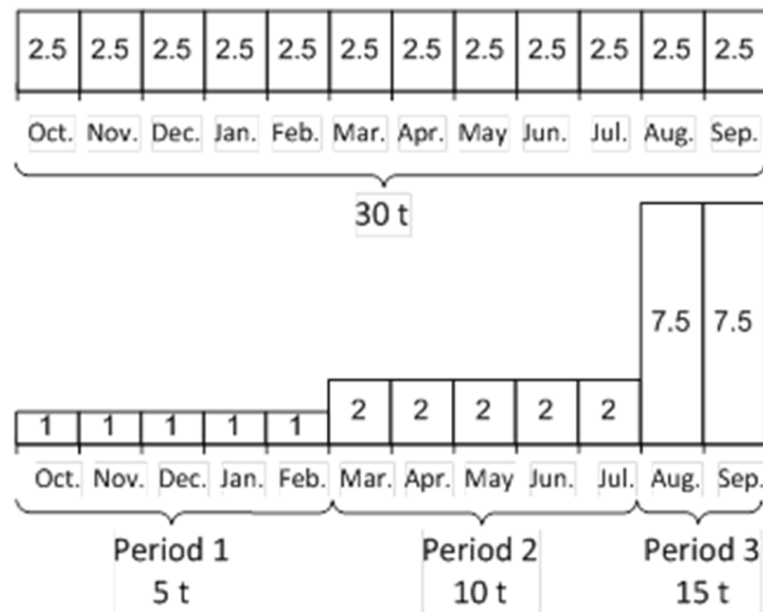
$$acc = 140/10 + (20/10)m \quad [\text{\$/év}] \quad (4.5)$$

$$oc_i = (6 + 3a_i)pl_i \quad [$/\acute{e}v] \quad (4.6)$$

$$a_i \leq m \quad [t/\acute{e}v] \quad (4.7)$$

$$tc = acc + \sum oc_i \quad [$/\acute{e}v] \quad (4.8)$$

A fenti egyenletekben cc , acc , oc_i , pl_i és tc rendre a következő jelentéssel bír: beruházási költség, évesített beruházási költség, az i . periódus működési költsége, az i . periódus hosszának aránya és a teljes költség. Ezen kívül a_i és m jelöli az i . periódus aktuális kapacitását illetve a maximális kapacitást. Az illusztráló példában, ami almahámozásra vonatkozik, i értéke 1, 2 vagy 3 lehet. Az összehasonlítás végett az egy és a több-periódusos működést is bemutatjuk.



4.8. ábra. Havi feldolgozandó mennyiségek egy és több periódusos működésre. A feldolgozás közbeni veszteség elhanyagolható, így a bemenet és kívánt termék mennyisége megegyezik.

A 4.8 ábrán a havi feldolgozandó mennyiségek láthatóak, amelyek megegyeznek a hámozandó alma mennyiségével feltételezve, hogy az almahéj mennyisége elhanyagolható. Mindegyik működési módban 30 tonna almát dolgoznak fel.

Az összehasonlíthatóság kedvéért az egy periódusos működést mutatjuk be először. Itt csak egy periódus van, vagyis i maximális értéke 1, és ebben a periódusban a hámozó berendezés, E , kapacitása, m , folyamatosan teljesen kihasznált. A 4.7-es egyenlet formája ebben az esetben a következő:

$$m = a_1 \quad (4.9)$$

Az előző (4.9) egyenlet azt fejezi ki, hogy az aktuális kapacitás megegyezik a maximális kapacitással, ha egy év alatt mind a 30 tonna almát fel akarjuk dolgozni. Vagyis az egyetlen periódusban, $m = a_1 = 30$ t/év, és, $mf = md = 2.5$ t/hó minden hónapra, ahol mf és md a havi bemenetet és a havi feldolgozandó mennyiséget jelöli. A berendezés hasznos élettartam 10 év. Az évesített befektetési költség, acc , a működési költség, oc , és a teljes költség, tc , a 2., 3. és 5. egyenlet segítségével számolhatóak. Ezek az értékek rendre 74, 96 és 170 \$/év az egy periódusos esetre, ahogy ez a 4.1 táblázatból látszik.

Periódus hossz, pl_i [év]	1
Havi bemenet, mf_j [t/hó]	2.5
Periódus bemenete, pf_i [t/periódus]	30
Aktuális kapacitás, a [t/év]	30
Évesített befektetési költség, acc [\$/év]	74
Működési költség, oc [\$/év]	96
Teljes költség, tc [\$/év]	170

4.1. táblázat. Az egy-periódusos esetre vonatkozó adatok

Az egy-periódusos működéssel szemben, a több-periódusos működés három periódust tartalmaz, amelyek egyenként 5, 5 és 2 hónap hosszúak, ahogy ez a korábbi ábrán is látszik. Az egy-periódusos működéssel szemben, most a műveleti egység, az almahámozó, kapacitása nincs teljes mértékben kihasználva minden periódusban. Az i -dik periódusban a feldolgozó egység csak az aktuális kapacitással dolgozik, a_i , $i = 1, 2$ vagy 3 , és az alábbi egyenletnek teljesülnie kell.

$$a_i \leq m, i = 1, 2 \text{ vagy } 3 \quad [\text{t/év}] \quad (4.10)$$

Multi-periódusos működés során az aktuális kapacitásoknak, a_i , elegendően nagyoknak kell lenniük, hogy fel tudják dolgozni a periódus bemenetet, pf_i [t/periódus], ami megegyezik a legyártandó termékkel, ha a veszteségtől eltekintünk. Más szavakkal, ha a periódus bemenetet normalizáljuk, vagyis elosztjuk a hozzátartozó periódus hosszával, pl_i , akkor megkapjuk az aktuális kapacitás, a_i , értékét.

$$a_i = pf_i / pl_i \quad \text{quad} [\text{t/év}] \quad (4.11)$$

Az egyedi a_i értékek a következők, $i = 1, 2, 3$,

$$a_1 = pf_1 / pl_1 = 5 / (5/12) = 12 \quad [\text{t/év}] \quad (4.12)$$

$$a_2 = pf_2 / pl_2 = 10 / (5/12) = 24 \quad [\text{t/év}] \quad (4.13)$$

$$a_3 = pf_3 / pl_3 = 15 / (2/12) = 90 \quad [\text{t/év}] \quad (4.14)$$

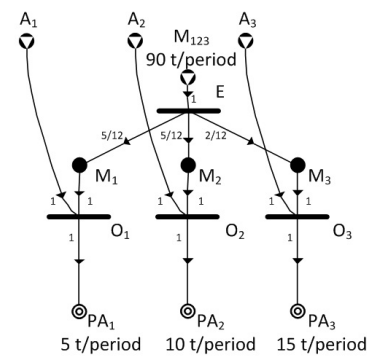
Nyilvánvalóan $m = a_3 = 90$ t/év kell, hogy legyen, hogy a műveleti egység a csúcs periódusban is működni tudjon. Az aktuális kapacitás, a_i , sosem lehet több, mint a maximális kapacitás, m . A beruházási költség (cc) a maximális kapacitás (m) értékéhez, a működési költség (a_i) az aktuális kapacitás értékéhez kapcsolódik. Az évesített befektetési költség (acc) a működési költség (oc) és a teljes költség (tc) a 4.5, a 4.6 és a 4.8 egyenlet segítségével számolhatóak. Az eredmény az egyéb paraméterekkel együtt, a 4.2 táblázatban található a több-periódusos esetre.

	Periódus 1	Periódus 2	Periódus 3
Periódus hossz, pl_i [év]	5/12	5/12	2/12
Havi bemenet, mf_i [t/hó]	1	2	7.5
Periódus bemenete, pf_i [t/periódus]	5	10	15
Aktuális bemenete, a_i [t/év]	12	24	90
Évesített befektetési költség, acc [\$/év]	194	194	194
Működési költség, oc [\$/év]	17.5	32.5	46
Teljes költség, tc [\$/év]	290	290	290

4.2. táblázat. A több periódusos esetre vonatkozó adatok.

Az egyes periódusokra vonatkozó működési költség fix részét, 6, eloszthatjuk egyenletesen, ekkor mindegyik periódusnál 2-t kell figyelembe venni vagy lehetőség van ezen fix rész időarányos elosztására is: 2, 5, 2, 5 és 1. A teljes működési költség mindkét esetben ugyanaz lesz.

A több-periódusos eset azért drágább, mert a beruházási költség magasabb amiatt, hogy a maximális kapacitás 30 helyett, 90. Erre azért van szükség, mert a harmadik periódusban gyorsan és sok anyagot kell feldolgozni. A gyakorlatban azt is figyelembe kell venni, hogy az egy-periódusos esetben a raktározási költségek sokkal jelentősebbek.

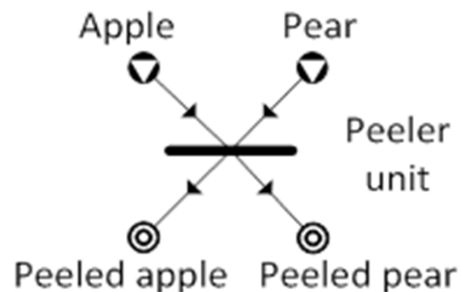


4.9. ábra. A szemléltető példa P-gráf reprezentálása a három-periódusos esetre.

4.9.2. A több-periódusos műveleti egység modellezése

Az előző ábra a szemléltető példában az almahámozó műveleti egységét modellezi három periódussal. Ez a P-gráf megvalósítás egyetlen almahámozó egységet, E , tartalmaz, amely mind a három periódust, vagyis összesen 12 hónapot kiszolgál, valamint E három egyedi megvalósulását az egyes periódusokban, O_i , $i = 1, 2, 3$. E éves feldolgozási kapacitását, M_{123} , ami most egy anyagpont, szét kell osztani, az egyes periódusokra O_i , $i = 1, 2, 3$ számára, hogy azok működni tudjanak. A felosztás alapja a periódus hosszak aránya, pl_i , $5/12$, $5/12$ és $2/12$ év. Így jönnek létre a periódusra vonatkozó kapacitások, M_1 , M_2 és M_3 , amelyek szintén anyagpontok.

M_1 , M_2 és M_3 , bevezetésével azt biztosítjuk, hogy E -t nem használjuk többet, mint az fizikailag lehetséges. Első olvasásra zavaró lehet, hogy több mindent a kapacitás kifejezéssel illetünk. Az elemi műveleti egységeknek van kapacitása, a multi-periódusos műveleti egység-nél aktuális és maximális kapacitásról beszélünk, most pedig anyagpontokat is kapacitásnak neveztünk el. Azért a hasonló elnevezés, mert ezek a fogalmak kapcsolódnak egymáshoz.



4.10. ábra. Az alternatív bemenettel rendelkező műveleti egység helytelen ábrázolása.

A több-periódusos műveleti egység csak akkor működhet az i -edik periódusban, ha nemcsak a hámozatlan alma, A_i , hanem a hámozási kapacitás, M_i , is rendelkezésre áll elegendő mértékben. O_i be és kimeneteihez tartozó arányokból illetve a periódusokra vonatkozó követelményekből kiszámítható, hogy O_1 , O_2 és O_3 működéséhez szükséges $5 M_1$, $10 M_2$ és $15 M_3$. Ezen értékek legyártásához az E műveleti egységnek $90 M_{123}$ -ra van szüksége. A számoláshoz az igények és E éleihez tartozó arányok szükségesek. Érdemes megjegyezni, hogy E előállítja és O_i fogyasztja M_i -t, de ezek a mennyiségek nem feltétlenül egyeznek meg. Példánkban E a 90 t/periódus bemenetből 37.5 t/periódus M_1 -t, 37.5 t/periódus M_2 -t és 15 t/periódus M_3 -t állít elő. Ugyanakkor a fogyasztás csak 5 , 10 , illetve 15 t/periódus.

Amennyiben az igények és a több-periódusú műveleti egység modellje, azaz az E eszköz és annak különböző megjelenései az egyes periódusokban O_1 , O_2 és O_3 adottak, akkor a feladat megoldható az ABB algoritmussal, amely megadja a kapacitásokat és a költségeket. Az eredmény megegyezik a kézi számolás eredményével, ami az előző táblázatban látható. Ez azt mutatja, hogy az alkalmazott modellezési eljárás helyes. Fontos megjegyezni, hogy ugyanazt az ABB algoritmust használjuk mind az egy, mind a több-periódusú esetre. A különbség a modellezésben és nem annak megoldásában van. Ez azt mutatja, hogy a P-gráf módszertan kellően rugalmas ahhoz, hogy egy újabb területre, a több-periódusú működésre is alkalmazzuk.

	Periódus 1	Periódus 2	Periódus 3
Periódus hossz, pl_i [év]	4/12	4/12	4/12
Havi bemenet, mf_i [t/hó]	5	10	15
Periódus bemenete, pf_i [t/periódus]	1.25	2.5	3.75
Aktuális bemenete, a_i [t/év]	15	60	45
Évesített befektetési költség, acc [\$/év]	17	122	47
Működési költség, oc [\$/év]	134	134	134
Teljes költség, tc [\$/év]	320	320	320

4.3. táblázat. A több fajta bemenetet tartalmazó példára vonatkozó adatok.

4.9.3. Műveletek különböző bemenetekkel illetve költségekkel

Egy másik gyakran felmerülő probléma az, amikor egy adott műveleti egység több fajta bemenettel tud dolgozni. Például, egy kazán égethet fa aprítékot, szalmát, pelletet, stb. Ha az előző példánkat tekintjük, akkor pedig mondhatjuk, hogy a hámozó egység almát és körtét is tud hámozni. Az előző (4.10-es) ábra nem modellezi helyesen ezt a fajta működést, mert valójában egy olyan műveleti egységet reprezentáltunk, amely akkor tud működni, ha mind alma, mind körte egyszerre megjelenik a bemenetén a megfelelő arányban.

A megoldás ismét az, hogy különválasztjuk a működési módokat és a fizikai megvalósulást. Minden működési mód, vagy periódus, külön műveleti egységgel reprezentált. Ez lehetővé teszi, hogy minden módozatban más legyen a bemenet illetve a kimenet. Ezen kívül az egyes élekekhez tartozó arányok és a költségek is változhatnak periódusról periódusra.

Tegyük fel, hogy most az periódushosszak megegyeznek. Az igények a következők: 5 t hámozott körte az első periódusban, 10 t hámozott barack a második periódusban és 15 t hámozott alma a harmadik periódusban. A körtehámozás működési költsége kétszerese a barackhámozás költsége duplája az almáénak. Az alma és a körtehámozás költsége megegyezik. Mivel a barack puha és kicsi, ennek hámozása nagyon sok hulladékot eredményez. A bemenet fele veszendőbe megy. Ezekből az következik, hogy a második periódus lesz a csúcs periódus. Habár a harmadik periódusban több terméket állítunk elő, mint a másodikban, de a másodikban a műveleti egység terhelése nagyobb, mert a veszteség miatt több nyersanyagot kell feldolgozni. A feladathoz tartozó adatok az előző táblázatban találhatók.

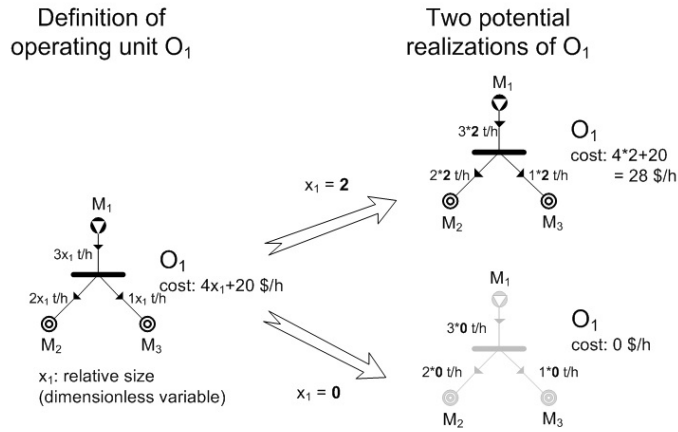
4.9.4. Több-periódusú működés a fenntartható energiatermelő ellátási láncokban

Ebben az alfejezetben bemutattuk, hogy miként lehet a műveleti egységeket modellezni, ha nem folyamatos, hanem több-periódusú a működés. Ez a fenntartható energiatermelő ellátási láncokban azért fontos, mert ott tipikusan előfordulnak olyan megújuló energiahordozók, fa, szalma, stb., amelyeket nem lehet a hagyományos módon modellezni.

4.10. Költség modellezése nyersanyagként

4.10.1. Eddigi módszer

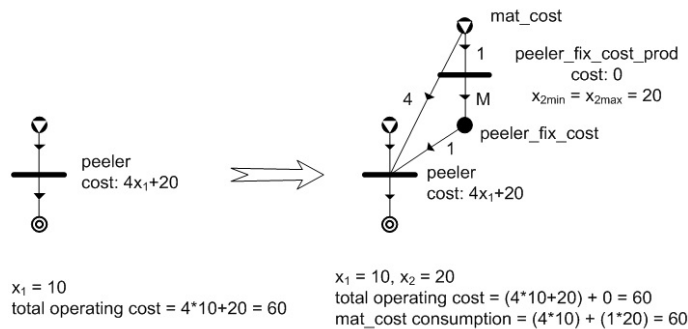
A korábbiakban leírtuk, hogy különféle fenntarthatósági mértékek folyamatosan egyre fontosabbak lesznek. Emellett a költség továbbra is egy meghatározó tényező marad. Először röviden átvesszük, hogy a költséget hogyan kezelik hagyományosan a P-gráf módszertan keretében, ezután egy új modellezési technikát mutatunk be, amelynek segítségével a teljes költségre be tudunk vezetni egy korlátozási feltételt.



4.11. ábra. Egy műveleti egység definíciója és két lehetséges realizálása

Egy hálózat költsége magába foglalja a nyersanyagok és a műveleti egységek költségét, amiből le kell vonni a termékek árát. Ha ez a költség negatív, akkor a hálózat profitot generál. Egy műveleti egység költségének két része van, beruházási költség és működési költség. Mindkét tényező további két részre bontható, fix rész és arányos rész. Általában a beruházási költséget elosztják a megtérülési idővel, így kapják meg az évesített beruházási költséget. Ekkor a két fix és két arányos költséget egybe lehet vonni és ezzel megkapjuk a teljes költség függvényt, amely szintén rendelkezik fix és arányos résszel. A teljes költség függvény és a relatív méret segítségével határozható meg a műveleti egység tényleges költsége. A 4.11-es ábra bal oldalán egy műveleti egység definíciója látható a teljes költség függvénnyel együtt. Jobb oldalt felül a kettő relatív méretű műveleti egység látható a hozzá tartozó áram értékekkel és a költséggel. Ezen értékeket úgy kaptuk, hogy 2-t behelyettesítettük x_1 helyére. Az ábra jobb alsó része azt a speciális esetet mutatja, amikor a relatív méret 0. Ez azt jelenti, hogy a műveleti egység egyáltalán nem szerepel egy adott hálózatban, így a fix részt sem kell kifizetni, vagyis a teljes költség 0.

Ez a fajta költségrepresentálás megfelelő egészen addig, amíg a költség az egyetlen célfüggvény. Ha léteznek más célfüggvények is, például különböző fenntarthatósági mértékek, akkor szükségünk lehet egy olyan eljárásra, amely képes a költséget korlátként kezelni.



4.12. ábra. Műveleti egység átalakítása úgy, hogy a költség új anyagpontként jelenjen meg.

4.10.2. Új eljárás

Egy modellezési technikát mutatunk be, amely képes arra, hogy a költséget egy egyszerű anyagként kezelje és így lehetőség nyílik arra, hogy korlátot vezessünk be rá. Egy új anyagot, *mat_cost*, kell bevezetnünk a hálózat számára, amely a struktúra teljes költségét reprezentálja. A korábbi költség paramétereket továbbra is használjuk, azok segítségével történik az optimalizálás, az új anyag, *mat_cost*, viszont lehetővé teszi, hogy a P-gráf módszertan eszközeit felhasználva korlátozzuk a keresést *mat_cost* alapján. A hálózat költsége és a felhasznált *mat_cost* anyag mennyisége megegyezik, mert ugyanannak a dolognak kétféle reprezentálásáról van szó. Az előző ábra azt is bemutatja,

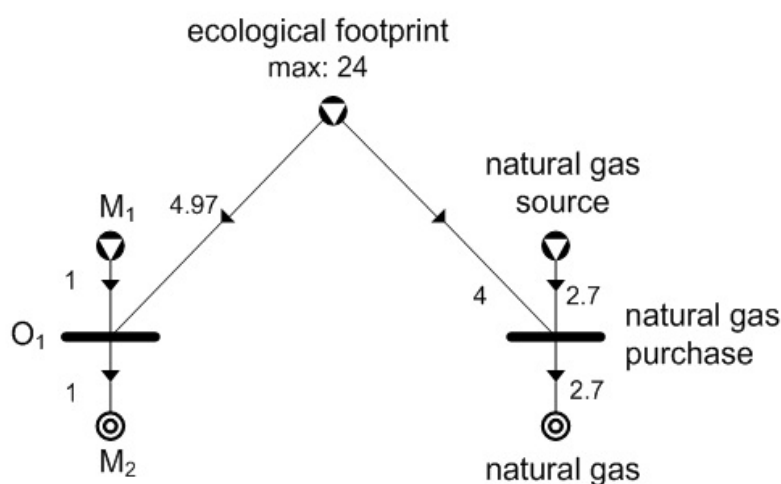
hogy az eredeti hálózat minden műveleti egységéhez be kell vezetni egy új anyagot és egy új műveleti egységet. Az ábra bal oldalán álló hámozó egységet úgy modellezzük, hogy megjelenik két új bemenet, *mat_cost* és *peeler_fix_cost*, 4 és 1 aránnyal. Az első arány értéke azért 4, mert az megegyezik a hámozó egység teljes költségvényében lévő arányossági együtthatóval. A másik új anyagot, *peeler_fix_cost*, csak az új műveleti egység *peeler_fix_cost_prod*, tudja előállítani, vagyis ennek az egységnek benne kell lennie a megoldásban, ha azt akarjuk, hogy a hámozónk működjön. Viszont, ha *peeler_fix_cost_prod* benne van a megoldásban, akkor annak relatív mérete pontosan 20 kell, hogy legyen. Azért 20, mert az eredeti költségfüggvénynek ez a fix része.

Az előző ábrán a hámozó relatív mérete 10. Ezt az értéket behelyesítve a teljes költségfüggvénybe 60-at kapunk. Az átalakított hálózat költsége szintén 60, mert a hámozó költsége nem változott és az új műveleti egységnek, *peeler_fix_cost_prod*, nincs költsége.

Az új elrendezésben 40 egységnyi *mat_cost*-t fogyaszt közvetlenül a hámozó egység és 20-at az új műveleti egység *peeler_fix_cost_prod*. Az új műveleti egység mindig pontosan 20 *mat_cost*-t fogyaszt, ha a hámozó egység aktív, függetlenül annak méretétől. A teljes *mat_cost* fogyasztás tehát 60, ami megegyezik az eredeti hálózat költségével. Ha egy nyersanyagnak van költsége és ezt akarjuk modellezni *mat_cost* segítségével, akkor biztosítani kell, hogy a nyersanyag fogyasztása együtt jár *mat_cost* fogyasztásával. Vezessünk be egy új műveleti egységet, amely az adott nyersanyag vásárlását reprezentálja. Ekkor reprezentálni kell az eredeti nyersanyagot, a vásárlást műveleti egységgel és a már megvásárolt nyersanyagot.

Ekkor az új műveleti egységnek legyen bemenete *mat_cost* is. Például, ha áramot veszünk a hálózathoz 149Euro/MWh-ért, akkor az új műveleti egység és a *mat_cost* között lévő él aránya 149 legyen.

4.11. Az ökológiai lábnyom modellezése



4.13. ábra. Az ökológiai lábnyom kezelése

Az ökológiai lábnyom egy gyakran használt eszköz a fenntarthatósági vizsgálatokban, egy a különböző fajta lábnyomok sorában. Ezek a lábnyom additív abban az értelemben, hogy a hálózat lábnyoma számítható a komponensek lábnyomainak az összegeként. A célunk az, hogy képesek legyünk meghatározni az egyes megoldásokhoz tartozó ökológiai lábnyomokat, és hogy legyen egy olyan módszerünk, amely a P-gráf technika keretein belül képes korlátozni a szóba jöhető lehetőségeket azokra a hálózatokra, amelyeknek az ökológiai lábnyoma nem halad meg egy adott szintet.

Az előbb említett célt úgy érjük el, hogy egy átalakítási eljárást mutatunk be. Az eljárást az előző ábrán szemléltetjük. Egy új anyagpontot vezetünk be, amelyet ökológiai lábnyomnak hívunk, és ami

Név	Költség	Max. [ha/év]
electricity_grid	149 €/MWh	
natural_gas	0.5 €/m ³	
area_corn		300
area_corn_silage		400
area_grass_silage		1200
area_wood		600

4.4. táblázat. Nyersanyagok tulajdonságai

Név	Min. [MWh/év]
hot_utility	5000
electric_utility	2000

4.5. táblázat. Termékek tulajdonságai

bemenete lesz minden olyan műveleti egységnek, amely működésével hozzájárul az ökológiai lábnyom növekedéséhez. Az új anyagpont tehát, nevéhez híven, a ténylegesen felmerülő ökológiai lábnyom nagyságát jellemzi. Például az előző ábrán, ha O_1 működik és relatív mérete 1, akkor az 4,97 hektár ökológiai lábnyomot jelent. Összekötjük O_1 -t az új anyagponttal és ennek az élnek az arányszáma 4,97 lesz, így O_1 működése együtt jár az új anyagpont fogyasztásával a kívánt mértékben. Ha a lábnyom egy nyersanyaghoz tartozik, akkor egy új műveleti egységet kell felvenni, amely ennek a nyersanyagnak a beszerzését jelképezi. Az ökológiai lábnyom anyagpont, ennek az új műveleti egységnek is a bemenete lesz.

Az átalakított hálózat tényleges ökológiai lábnyoma ugyanaz, mint az eredeti esetben és megegyezik az azonos nevű, új, hipotetikus anyagpont fogyasztásával. Ennek az anyagnak beállíthatunk egy felső korlátot, hiszen ezt a P-gráf módszertan lehetővé teszi. Ez biztosítja azt, hogy az optimalizálási eljárás során csak olyan hálózatokat vegyünk számításba, amelyek elérnek egy adott fenntarthatósági szintet.



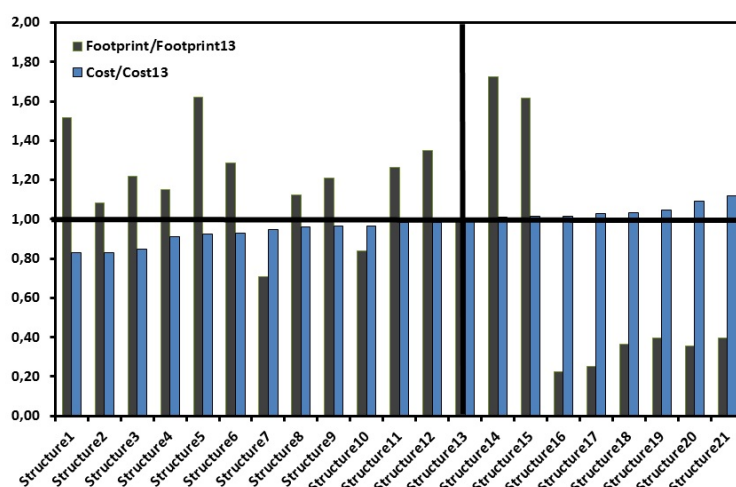
Név	Bemenet	Arány		Kimenet	Arány	
electricity_feeder	electricity_grid	1	MWh	electricity	1	MWh
biogas_plant				biogas_plant_c	8000	h
biogas_prod_corn	biogas_plant_c corn_silage electricity	13 1 0.13	h t MWh	biogas	600	m ³
biogas_prod_grass	biogas_plant_c grass_silage electricity	12 1 0.13	h t MWh	biogas	550	m ³
corn_silage_prod	area_corn_silage	1	ha	corn_silage	12	t
grass_silage_prod	area_grass_silage	1	ha	grass_silage	12	t
biogas_CHP_plant				biogas_CHP_plant_c	8000	h
biogas_CHP_corn	biogas_plant_c corn_silage	4 0.694	h t	electricity heat	1 0.65	MWh
biogas_CHP_grass	biogas_plant_c grass_silage	4 0.758	h t	electricity heat	1 0.65	MWh
electricity_utility_prod	electricity	1	ha	electricity_utility	1	MWh
gas_burner				gas_burner_c	8000	h
biogas_burning	gas_burner_c biogas	1 153	h m ³	heat	0.85	MWh
natural_gas_burning	gas_burner_c natural_gas	1 91.9	h m ³	heat	0.85	MWh
corn_prod	area_corn	1	ha	corn corn_straw	9 14	t
pelletizer				pelletizer_c	8000	h
corn_straw_pellet_prod	electricity heat pelletizer_c corn_straw	0.15 0.5 0.5 1	MWh MWh h t	corn_straw_pellet	1	t
wood_pellet_prod	electricity heat pelletizer_c wood	0.1 0.85 0.5 1	MWh MWh h t	wood_pellet	1	t
wood_chips_prod	electricity heat wood	0.03 0.48 1	MWh MWh t	wood_chips	1	t
wood_prod	area_wood	1	ha	wood	3	t
feeder				feeder_c	8000	h
burner				burner_c	8000	h
corn_straw_pellet_burning	corn_straw_pellet feeder_c burner_c	0.25 4 4	t h h	heat	1	MWh
wood_pellet_burning	wood_pellet feeder_c	0.25 4	t h	heat	1	MWh

4.6. táblázat. A műveleti egységek be és kimenetei

Név	Fix rész [€/év]	Arányos rész [€/év]
electr_feeder	0	0
corn_silage_prod	0	960
grass_silage_prod	0	960
corn_prod	0	960
wood_prod	0	180
biogas_plant	35000	49286
biogas_CHP_plant	131237	81298
biogas_prod_corn	3680	10
biogas_prod_grass	3680	10
biogas_CHP_corn	9822	4
biogas_CHP_grass	9822	4
gas_burner	1000	2000
biogas_burning	0	0
natural_gas_burning	0	0
burner	15578	15692
wood_burning	7347	4
wood_chips_burning	7347	3
wood_pellet_burning	7347	3
corn_straw_pellet_burning	7347	3
hot_utility_prod	0	0
wood_chips_prod	30820	3
wood_pellet_prod	10400	2
corn_straw_pellet_prod	10400	2
electric_utility_prod	0	0
pelletizer	30000	185000
feeder	100	0

4.7. táblázat. A műveleti egységekhez tartozó költségegyütthatók

4.12. Szemléltető példa



4.15. ábra. 21 a feltételeknek megfelelő ellátási lánc összehasonlítása a 13. megoldás (nem megújuló energiahordozókat használ) alapján.

Egy olyan esettanulmányt választottunk a korábban bemutatott módszer szemléltetésére, amelyben a feladat, hogy egyszerre gyártsunk hőt és elektromosságot egy adott régióban. Az ellátási lánc maximális hálózata, amely tartalmazza az összes figyelembe vehető lehetőséget a 4.14-es ábrán látható, alatta, a hozzá tartozó adatok táblázatos formában adottak (4.4-4.7 táblázat).

A cél, hogy teljesítsük az elektromosság és hő termelési követelményeket, 2000 és 5000 MWh/év. Természetesen rendelkezésre állnak nem megújuló technológiák, például elektromosság vásárlása külső forrásból, illetve hő termelés földgáz segítségével. A vizsgált körzet rendelkezik megújuló forrásokkal, mint szilázs kukorica (összegyűjtött és erjesztett), szilázs fű, kukoricacsutka (a szárat és a levelet is beleértve) és fa. Emellett több energia konverziós technológia áll rendelkezésre, például biogáz erőmű, CHP biogáz erőmű, gázkazán, pellet gép, hagyományos kazán. Érdemes megjegyezni, hogy néhány műveleti egység, például a pellet gép, csak adott méretekben kapható. Ezekhez a

méretekhez tartozó paraméterekből állítottuk elő a költségfüggvényt. Az eredeti adatok (Luttenberger et al.)-ban találhatóak.

Az előző ábrán néhány műveleti egység nem egy tényleges berendezést, hanem egy tevékenységet jelent. Például az *electricity_feeder* az elektromos áram vásárlását jelenti külső forrásból. Ez a műveleti egység nem rendelkezik költséggel, a bemeneti és a kimeneti arányszáma egyaránt 1. A *wood_production* nevű műveleti egység a fakitermelést jelképezi. A bemenete az erre szánt terület nagysága. A szemléltető példában egy konkrét korlát adott az előbbi területre, a kukoricatermelésre két külön terület is adott. Az egyik a szilázs kukorica, a másik a kukorica pellet előállításához tartozik.

Ha egy műveleti egységnek két működési módozata van, például a gázkazán használható földgáz és biogáz égetésére is, akkor egy műveleti egység reprezentálja az eszközt (*gas_burner*), ami előállítja a hipotetikus anyagot (*gas_burner_capacity*), egy műveleti egység reprezentálja a biogáz égetést és egy másik a földgáz égetését.

Az előző ábra az eredeti modellt szemlélteti, mielőtt még végrehajtottuk volna a korábban ismertetett költség és lábnyom modellezési eljárást. A biogáz hagyományos és CHP típusú erőműben is előállítható. Mindkét erőmű bemenete lehet kukorica és fű szilázs is. Pellet előállítható kukoricaszárból és fából. A fából fa apríték is készíthető, de a fa közvetlenül is elégethető. A pellet és apríték égetése egy adagoló egységet kíván meg, ami plusz befektetés, de a fa közvetlen égetése nagyobb élő munkát igényel, ami pedig arányos költségként jelentkezik.

4.8. táblázat. Nyersanyagok, költség, ökológiai lábnyom a 21 struktúrához

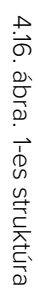
	Electricity_grid	Natural_gas	area_corn	area_corn_silage	area_grass_silage	area_wood	cost	Cost Divided By cost(13)	Cost Change%	Footprint	Footprint Divided by footprint(13)	Footprint Change%
	[TJ/év]	[m ³ /év]	[ha/év]	[ha/év]	[ha/év]	[ha/év]	[€ /év]			[global ha]		
Struktúra 1	7,37						500,00	0,83	-16,86	1046,81	1,52	51,57
Struktúra 2				117,69			476433	0,83	-16,85	749,28	1,08	8,50
Struktúra 3							367,73	0,85	-15,03	840,65	1,22	21,73
Struktúra 4	7,57			120,03	128,54		521283	0,91	-9,02	796,21	1,15	15,29
Struktúra 5							539,13	0,93	-7,46	1119,21	1,62	62,06
Struktúra 6		399272				393,66	531909	0,93	-71,6	889,40	1,29	28,78
Struktúra 7				116,30			542711	0,95	-5,10	489,71	0,71	-29,09
Struktúra 8			72,96	126,77			551507	0,96	-3,74	774,86	1,12	12,20
Struktúra 9		399272		124,78		380,69	553388	0,97	-3,42	835,02	1,21	20,91
Struktúra 10					127,02		554007	0,97	-3,31	580,00	0,84	-16,02
Struktúra 11			72,96		138,46		562730	0,97	-3,31	580,00	0,84	-16,02
Struktúra 12					136,29	380,69	564435	0,99	-1,49	931,90	1,35	34,94
Struktúra 13	7,25	540588					572956	1,00	0,00	690,61	1,00	0,00
Struktúra 14	7,99						579343	1,01	1,11	1190,91	1,72	72,44
Struktúra 15	8,17		102,04				580439	1,01	1,31	1114,72	1,61	61,41
Struktúra 16				214,45			581310	1,01	1,46	154,24	0,22	-77,67
Struktúra 17				125,12	98,01		589975	1,03	2,97	175,19	0,25	-74,63
Struktúra 18				90,05	135,88		592323	1,03	3,38	250,82	0,36	-63,38
Struktúra 19	7,95				234,67		601052	1,05	4,90	272,33	0,39	-60,57
Struktúra 20				125,00			626618	1,09	9,37	244,35	0,35	-64,62
Struktúra 21	8,02				136,36		640159	1,12	11,73	274,22	0,40	60,29

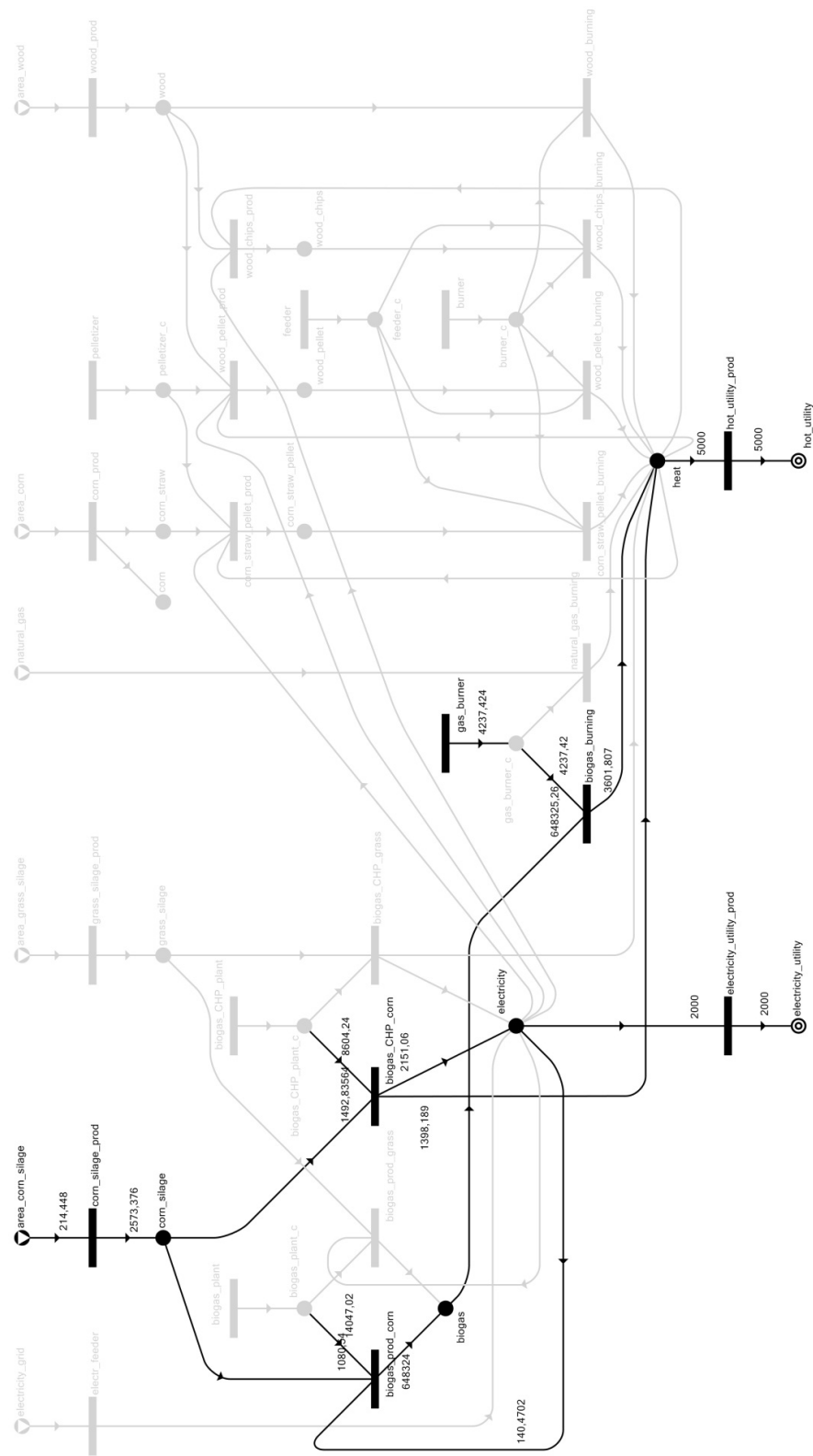
4.13. A szemléltető példa eredményei

A korábban ismertetett esettanulmány fő eredménye a 21 különböző ellátási lánc, amelyeket az előző ábra hasonlít össze és további részleteket tartalmaz a 4.8-as táblázat. A szemléltető példa relatív kicsi, így összesen csak 21 lehetséges megoldás létezik. A 13-as struktúra jelenti azt az esetet, amikor hagyományos, azaz nem megújuló, energiahordozókat használunk. Ekkor az áramot (7,2 TJ/év) a hálózathoz vesszük, a hőt (18 TJ/év) pedig földgáz segítségével állítjuk elő. A 13-as hálózatot tekintjük referencia pontnak, ezen hálózat költségéhez és ökológiai lábnyomához hasonlítjuk a többi hálózat hasonló tulajdonságait. Az előző táblázatban a struktúrákat költség alapján rendeztük. Az előző ábrán a vastag függőleges vonal jelöli a 13-as struktúrát, a vízszintes vonal pedig a hozzá tartozó relatív költséget és lábnyomot.

Azok a struktúrák, amelyek kevésbé költségesek illetve környezetbarátabbak, mint a 13-as struktúra a vastag vízszintes vonal alatt találhatók. Az ábrát megvizsgálva két struktúrára keltheti fel a figyelmünket, a 7-es és a 10-es. A 7-es struktúra 5%-al olcsóbb, mint az alapeset, ugyanakkor az ökológiai lábnyoma 29%-al kedvezőbb. A 10-es struktúra esetében a költségmegtakarítás 3% és a lábnyom 16%-al kedvezőbb, mint az alapeset. Mindkét struktúra földgázt használ alapvetően, de a 7-es megoldás kukorica szilázst, a 10-es fű szilázst is hasznosít.

Ha hajlandóak vagyunk mérsékelt költségnövekedést tolerálni az alacsonyabb ökológiai lábnyomért, akkor a 16, 17, 18 és 19-es struktúra is fókuszba kerülhet. A 16-os struktúra költsége körülbelül 1, 5%-al nagyobb, mint az alapeset, de az ökológiai lábnyom 78%-al kisebb. Itt kukorica szilázs az egyedül felhasznált nyersanyag. A 17-es struktúra 3%-os költségemelkedést, de 75%-os ökológiai lábnyomcsökkenést valósít meg. Ekkor kukorica és fű szilázst is felhasználunk. A 18-as struktúrában szintén 3%-al nagyobb a költség, amíg a lábnyom 64%-al kisebb. Végül a 19-es struktúrában 5%-al nőtt a költség és 60%-al csökkent az ökológiai lábnyom. Itt csak fű szilázst használtunk fel. A feladatnak nincs abszolút optimuma, mert költség szerint az 1-es, lábnyom szerint viszont a 16-os struktúra a legjobb (lásd a kapcsolódó ábrákat). A felsorolt lehetőségek közül, a döntéshozóknak kell választani annak függvényében, hogy mekkora súlyt tulajdonítanak a költségnek és a lábnyomnak. Azért is fontos, hogy több alternatíva legyen, mert így olyan tényezőket is figyelembe lehet venni, amelyeket nem tudtunk modellezni.





4.17. ábra. 16-os struktúra

Irodalomjegyzék

- [1] Brito, M. P., Valentina Carbone, Corinne Meunier Blanquart, Towards a sustainable fashion retail supply chain in Europe: Organisation and performance, *International Journal of Production Economics*, 114, 534-553, 2008.
- [2] Büyüközkan, G., Çiğdem Berkol, Designing a sustainable supply chain using an integrated analytic network process and goal programming approach in quality function deployment, *Expert Systems with Applications*, 38, 13731-13748, 2011.
- [3] Chaabane, A., A. Ramudhin, M. Paquet, Design of sustainable supply chains under the emission trading scheme, *International Journal of Production Economics*, 135, 37-49, 2012.
- [4] Friedler, F., J. B. Varga, and L. T. Fan, Decision-Mapping: A Tool for Consistent and Complete Decisions in Process Synthesis, *Chem. Engng Sci.*, 50, 1755-1768, 1995.
- [5] Friedler, F., K. Tarjan, Y. W. Huang, and L. T. Fan, Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation, *Computers Chem. Engng*, 17, 929-942, 1993.
- [6] Gupta, S., Omkar D. Palsule-Desai, Sustainable supply chain management: Review and research opportunities Original Research Article *IIMB Management Review*, 23, 234-245, 2011.
- [7] Hassini, E., Chirag Surti, Cory Searcy: A Literature Review and a Case Study of Sustainable Supply Chains with a Focus on Metrics, *International Journal of Production Economics*, In Press, 2012.
- [8] Heckl, I., F. Friedler, and L. T. Fan, Solution of separation network synthesis problems by the P-graph methodology, *Computers et. Chemical Engineering*, 34(5), 700-706, 2010.
- [9] Krotscheck, C., M. Narodoslawsky, The Sustainable Process Index - A new Dimension in Ecological Evaluation, *Ecological Engineering*, 6/4, 241-258, 1996.
- [10] Luttenberger, C.; Birnstingl-Gottinger, B.; Puchas, K.; Riebenbauer, L.; Krotscheck, C.; Stubenschrott, Jauschnegg, E.; Mandl, M.; Lauer, M.; Narodoslawsky, M.; Gwehenberger, G. KOMEOS – Konzeption von Modellen multifunktionaler Energiezentren für die Ost-Steiermark, project report, 2008.
- [11] Mayer, A.L., Thurston, H.W., Pawlowski, C.W., The Multidisciplinary Influence of Common Sustainability Indexes, *Front. Ecol. Environ*, 2(8), 419-426, 2004.
- [12] Seuring, S., Martin Müller, From a literature review to a conceptual framework for sustainable supply chain management, *Journal of Cleaner Production*, 16, 1699-1710, 2008.
- [13] Varga, V., I. Heckl, F. Friedler, and L. T. Fan, PNS Solutions: A P-graph based programming framework for process-network synthesis problems, *Chemical Engineering Transactions*, 21, 1387-1392, 2010.
- [14] Winkler, H., Closed-loop production systems — A sustainable supply chain approach, *CIRP Journal of Manufacturing Science and Technology*, 4, 243-246, 2011.

5. FEJEZET

Több célfüggvény kezelése

Többcélú optimalizálásról beszélünk, amikor egy feladat megoldása során több, különböző szempontot is figyelembe kell venni, amelyek mindegyikét egy-egy a feladat lehetséges megoldásain értelmezett célfüggvény írja le. Bizonyos esetekben ilyenkor van egy kiemelt célfüggvény, amelynek a legnagyobb a jelentősége és a többi célfüggvény csak azon megoldások rangsorolására szolgál, amelyek a kiemelt célfüggvény szerint optimálisak. De az esetek többségében nincs ilyen rangsorolás a célfüggvények között, hanem mindegyik célfüggvényt egyformán, vagy hasonló mértékben figyelembe kell vennünk. A probléma az, hogy ezekben az esetekben gyakran előfordul, hogy bizonyos megoldások, amelyek nagyon jó eredményt adnak az egyik célfüggvény szempontjából rosszul teljesítenek a másik célfüggvény alapján.

Ebben a fejezetben a hálózati folyamatok szintézise során felmerülő többcélú problémákkal foglalkozunk. Elsőként bemutatjuk hol fordulnak elő és miként definiálhatóak a többcélú problémák a PNS probléma esetén. Ezt követően röviden ismertetjük általában milyen megközelítéseket használnak többcélú problémák vizsgálatára és áttekintjük ezek egymással való kapcsolatát. Majd bemutatjuk miként használhatóak ezek az általános módszerek a PNS probléma esetén. A módszerek közül elsőként a Pareto optimális megoldások generálásának problémáját, aztán az aggregált célfüggvények kezelését, majd a korlátozott optimalizálási modellek vizsgálatát mutatjuk be. Végül a fejezet második részében a többcélú optimalizáláshoz kapcsolódó robusztus optimalizálást tárgyaljuk.

5.1. Többcélú problémák a hálózati folyamatok szintézisében

A gyártási folyamatok szintézisében és gyakran az egyéb optimalizálási problémákban - amelyeket a P-gráf módszertan segítségével is sikerül megoldani - az alapköltségek mellett egyéb célpontok is felmerülnek. A legalapvetőbb célpont a gyártás költségeinek minimalizálása mellett a gyártási folyamatnak a környezetre vonatkozó esetleges káros hatásainak minimalizálása. Ilyen modelleket vizsgálnak a [2] és [12] tanulmányokban. Gyakran törvények és egyéb szabályozások (például Európai Unió pályázatok elnyerése során tett vállalások) kötelezik a vállalkozásokat környezetvédelmi szempontok figyelembevételére is. Sok esetben a tisztább technológiák drágábbak, így a többcélú optimalizálásnál azzal az általában felmerülő problémával kerülünk szembe, mely szerint a különböző szempontok különböző típusú megoldásokat helyeznek előnybe. A környezeti ártalmak minimalizálása mellett más szempontok is felmerülnek, például a gyártás folyamatának gyorsasága. Az időkorlátokkal felírt PNS modellt ezen tankönyv egy korábbi fejezetében tárgyaltuk részletesebben, de a gyártási folyamatra fordított időt többcélú optimalizálási folyamatként is figyelembe lehet venni. További szempont lehet a gyártás stabilitása, melyben a gyártási folyamatban részt vevő eszközöknek nem csak a költségeit, de a megbízhatóságát is figyelembe vesszük.

A továbbiakban a módszerek áttekintésénél tetszőleges számú (k darab) célfüggvény kezelését ismertetjük, de a speciális, PNS problémára vonatkozó részekben már csak az egyszerűbb két célfüggvényes esetet tárgyaljuk. Ennek célja a jobb áttekinthetőség. A megoldások - esetleg némi technikai nehézséget követően - kiterjeszthetők több célfüggvény esetére is. Szintén a könnyebb érthetőség érdekében a megoldásokat elsőként az egyszerűbb strukturális modellre (ahol nincsenek

az anyagok mennyiségétől függő költségek) mutatjuk be. Ez az egyszerűbb eset már elegendő az alapvető ötletek megértéséhez, és csak ennek bemutatása után tárgyaljuk miként terjeszthetők ki az algoritmusok az általánosabb fix részt tartalmazó lineáris költségfüggvényű PNS feladat változatra.

A strukturális modell vagy más néven fix költséges modell esetén minden o_i műveleti egységnek van egy cf_i költsége, és a cél annak a lehetséges megoldás struktúrájának a megtalálása, amelyre a benne szereplő műveleti egységek költségeinek összege minimális. Egy S lehetséges struktúra esetén a benne levő műveleti egységek költségeinek összegét z_1 -el jelöljük. Az általunk részletebben vizsgált kétcélú PNS feladat esetén két fix költséget rendelünk egy o_i műveleti egységhez cf_i -t és df_i -t mindkét szempont szerint egyet-egyet. Ennek megfelelően a lehetséges megoldások terén is két célfüggvényt definiálunk: a z_1 és z_2 függvényeket. Az általánosabb modellben a műveleti egységeknek további költségei is vannak, amelyek függnek a műveleti egység által használt x_i anyagmennyiségtől, ezt a költséget a cp_i arányossági konstansokkal írhatjuk le. A kétcélú fix részt tartalmazó lineáris költségfüggvényű modellben ezekből az arányossági konstansokból is kettő van cp_i és dp_i , és egy struktúrával és anyagáramokkal megadott lehetséges megoldás esetén két költséget számolunk, az egyiket z_1 -et a kiválasztott műveleti egységek cf_i , cp_i értékei alapján, a másikat z_2 -t pedig a kiválasztott műveleti egységek df_i és dp_i értékei alapján. Fontosnak tartjuk megjegyezni, hogy hibrid modelleket is definiálhatunk. Előfordulhatnak olyan esetek, amikor az egyik szempont szerint fix költségfüggvényekkel adható meg a probléma, míg a másik szempont szerint a bonyolultabb fix részt tartalmazó lineáris költségfüggvényű modellt kell használnunk.

5.2. Többcélú optimalizálásban használt modellek

Elsőként tömören összefoglaljuk milyen megközelítéseket szokás használni a több célfüggvényt is figyelembe vevő modellekben, majd részletesebben bemutatjuk miként használhatóak ezek a hálózati folyamatok szintézisének esetén. Mivel a hálózati folyamatok témakörében többnyire minimalizálási feladatokat kell megoldanunk, ezért a továbbiakban feltételezzük, hogy minden szempont egy minimalizálandó célfüggvény által van megadva. Ennek megfelelően minimum feladatokra adjuk meg az összes definíciót. Mindezt az általánosság megszorítása nélkül feltehetjük, hiszen egy maximum feladat áttéríthető egy minimum feladatra ha a célfüggvényt megszorozzuk -1 -el. De megjegyezzük, hogy a definíciók maguk is minden nehézség nélkül kiterjeszthetők maximalizálási feladatokra is.

Az alapvető megközelítés, hogy olyan megoldásokat keresünk, amelyek nem javíthatóak egyik célfüggvény szerint sem anélkül, hogy más célfüggvényekben rontanánk a megoldás hatékonyságán. A matematikailag precíz definíció érdekében tegyük fel, hogy k darab minimalizálandó célfüggvényünk van, amiket $f_1, \dots, f_k$ jelöl, továbbá a probléma lehetséges megoldásainak a halmazát jelölje S .

Két megoldást összehasonlítva egy nyilvánvaló gondolat azt mondanunk, hogy egy $x \in S$ megoldás akkor jobb, mint egy $y \in S$, ha minden szempont szerint jobb, azaz minden $i = 1, \dots, k$ értékre teljesül, hogy $f_i(x) < f_i(y)$. Ezen fogalom alapján definiálhatjuk a gyenge efficiens megoldásokat. Egy megoldást *gyenge efficiensnek* nevezünk, ha nincsen nála jobb megoldás. Tehát egy $x^* \in S$ megoldást akkor nevezünk gyenge efficiens megoldásnak, ha nincs olyan $x \in S$ megoldás, amelyre teljesülne $f_i(x) < f_i(x^*)$ minden $i = 1, \dots, k$ esetén.

Egy másik, az előzőnél megengedőbb, és szélesebb körben használt fogalom két megoldás összehasonlítására a következő. Azt is mondhatjuk, hogy egy $x \in S$ megoldás akkor jobb, mint egy $y \in S$, ha egyik szempont szerint sem rosszabb és legalább az egyik szempont szerint jobb. Tehát minden $i = 1, \dots, k$ értékre teljesül, hogy $f_i(x) \leq f_i(y)$ és van olyan $j \in \{1, \dots, k\}$ index, amelyre $f_j(x) < f_j(y)$. Az ebben az értelemben vett legjobb megoldásokat szokás erősen efficiens vagy *Pareto optimális* megoldásoknak nevezni. Tehát egy $x^* \in S$ megoldást akkor nevezünk Pareto optimális megoldásnak, ha nincs olyan $x \in S$ megoldás, amelyre teljesülne $f_i(x) \leq f_i(x^*)$ minden $i = 1, \dots, k$ esetén és $f_j(x) < f_j(x^*)$ valamely $j \in \{1, \dots, k\}$ indexre. Mindkét esetben szokás használni a dominancia kifejezést. Amennyiben egy megoldás jobb egy másiknál, akkor azt mondjuk dominálja azt.

Többcélú optimalizálási feladatok esetén az egyik megközelítés az, hogy meghatározzuk a Pareto optimális megoldásokat. Másrészt az összes Pareto optimális megoldás generálásán kívül más

módszereket is szokás használni többcélú optimalizálási feladatok megoldására. Az egyik lehetőség, hogy a különböző célfüggvényekből egy aggregált célfüggvényt hozunk létre. Ennek a legegyszerűbb és legelterjedtebb módja az, hogy vesszük a célfüggvények egy pozitív súlyokkal képzett súlyozott összegét, és erre az aggregált függvényre keressük a minimális megoldást. Másrészt indokolt esetben más, minden változóban szigorúan monoton függvény is használható aggregált célfüggvényként. Így a problémát visszavezetjük egy egyetlen célfüggvényes optimalizálási feladatra. A két megközelítés közötti kapcsolatot adja meg az alábbi állítás.

5.1. Tétel. *Minden optimális megoldás amit egy, az eredeti célfüggvények pozitív súlyokkal vett lineáris kombinációjával képzett aggregált költségfüggvény alapján kaptunk Pareto optimális megoldása a többcélú optimalizálási feladatnak.*

Bizonyítás: Legyen $g(x) = \sum_{i=1}^k c_i f_i(x)$ az aggregált célfüggvény, ahol a c_i értékek mindegyike egy pozitív konstans. Továbbá legyen $x^* \in S$ az optimális megoldása a feladatnak erre az aggregált célfüggvényre nézve. Igazoljuk, hogy x^* Pareto optimális megoldása a többcélú optimalizálási feladatnak. Ezt indirekt bizonyítjuk. Tegyük fel, hogy x^* nem Pareto optimális, azaz van olyan $x \in S$ megoldás, amelyre teljesül $f_i(x) \leq f_i(x^*)$ minden $i = 1, \dots, k$ esetén és $f_j(x) < f_j(x^*)$ valamely $j \in \{1, \dots, k\}$ indexre. Másrészt ekkor

$$g(x) = \sum_{i=1}^k c_i f_i(x) < \sum_{i=1}^k c_i f_i(x^*) = g(x^*).$$

Tehát azt kaptuk, hogy x szigorúan jobb megoldás, mint x^* a g függvényre nézve, ami ellentmond a kiindulási feltételünknek miszerint x^* optimális. Így ellentmondáshoz jutottunk, amivel az tétel állítását igazoltuk.

Fontosnak tartjuk megjegyezni, hogy a fenti állítás teljesen hasonlóan igazolható minden olyan aggregált célfüggvényre, ami szigorúan monoton növekvő az összes változóban. Bizonyos alkalmazások esetén szoktak a lineáris kombinációnál bonyolultabb aggregált függvényeket is vizsgálni.

Egy további megközelítése a többcélú optimalizálási problémáknak, amelyben kiemelünk egy célt és ezen cél szerint keressük az optimális megoldást, miközben a többi szempontot korlátozási feltételként írjuk elő. Az egyszerűbb leírás érdekében tegyük fel, hogy az f_1 függvényt emeltük ki, amely szerint optimalizálni szeretnénk, ez a célfüggvények átjelölésével biztosítható. Tehát egy korlátozásos egycélú redukciónál adottak $C_2, \dots, C_k$ konstansok és az optimalizálási feladatunk a következő:

$$\begin{aligned} \min f_1(x) \\ f_i(x) &\leq C_i, \quad i = 2, \dots, k \\ x &\in S \end{aligned}$$

Egy ilyen korlátozásos egycélú redukciónál nem feltétlenül teljesül, hogy az optimális megoldás az eredeti többcélú feladatnak Pareto optimális megoldása lesz, hiszen előfordulhatnak olyan megoldásai is a feladatnak, amelyek f_1 szerint ugyanazt az értéket veszik fel, mint a kiválasztott optimális megoldás, de a többi célfüggvény szerint jobbak. Másrészt egy kicsit gyengébb állítást kimondhatunk.

5.2. Tétel. *Ha egy korlátozásos egycélú redukált feladatnak van optimális megoldása, akkor az az eredeti feladatnak gyenge efficiens megoldása lesz.*

Bizonyítás: Legyen $x^* \in S$ egy optimális megoldása a korlátozásos egycélú redukált feladatnak. Igazoljuk, hogy x^* gyenge efficiens megoldása a többcélú optimalizálási feladatnak. Ezt indirekt bizonyítjuk. Tegyük fel, hogy x^* nem gyenge efficiens, azaz van olyan $x \in S$ megoldás, amelyre teljesül $f_i(x) < f_i(x^*)$ minden $i = 1, \dots, k$ esetén. Ekkor $f_i(x) < f_i(x^*) \leq C_i$ ($i = 2, \dots, k$) alapján kapjuk, hogy x lehetséges megoldása lesz a korlátozásos feladatnak is. Továbbá $f_1(x) < f_1(x^*)$, így azt kaptuk, hogy x szigorúan jobb megoldása a korlátozásos feladatnak, mint x^* , ami ellentmond a kiindulási feltételünknek miszerint x^* optimális. Így ellentmondáshoz jutottunk, amivel a tétel állítását igazoltuk.

A továbbiakban csak ezekkel a megközelítésekkel fogunk foglalkozni, de fontosnak tartjuk megemlíteni, hogy vannak más többcélú modellek is.

Ezek közül talán a legfontosabbak a kompromisszum programozási modellek, ahol minden szem-

pont szerint meg kell adnunk egy célértéket, ami azt mutatja milyen értékű (vagy legfeljebb milyen értékű) megoldást szeretnénk kapni. Ezt követően egy megoldás kiértékelésekor meg kell néznünk, hogy az egyes szempontokban mennyire térünk el (mennyivel léptük túl) a célértéket és ezen különbségeknek egy függvénye fogja az adott megoldást minősíteni.

A különbségeknek szokás venni a maximumát, az abszolút értékek súlyozott összegét vagy a négyzetösszegét. A modellek többségére itt is teljesül, hogy a kompromisszum programozással kapott megoldások Pareto optimálisak vagy gyenge efficiensek. Az érdeklődő olvasó további részleteket találhat a többcélú optimalizálásról a [8, 9] könyvekben.

5.3. Pareto optimális megoldások generálása PNS modellek esetén

Pareto optimális megoldások generálására a legegyszerűbb módszer az, hogy generáljuk az összes lehetséges megoldást, és minden lépésben a kapott megoldást megvizsgáljuk, hogy az eddigi megoldásokhoz képest Pareto optimális-e. Ha igen, akkor bővítjük az új megoldással a Pareto optimálisnak jelölt megoldások halmazát. Másrészt egy új megoldás esetén azt is meg kell néznünk, hogy az eddig Pareto optimálisnak jelölt megoldások közül dominálja-e valamely megoldásokat. Ha igen, akkor törölnünk kell ezeket a dominált megoldásokat a jelöltek listájából. Természetesen ez a módszer csak akkor működik, amikor véges sok lehetséges megoldásunk van, és azokat valamely algoritmus-sal fel tudjuk sorolni.

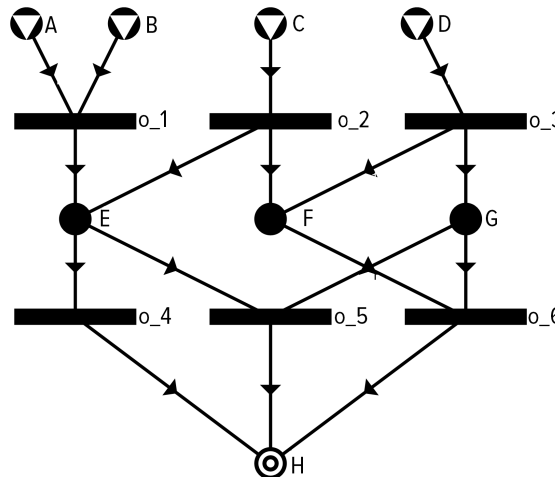
Mindezek alapján azt kaptuk, hogy az egyszerűbb fix költséges modellben könnyen kiterjeszthető az előző fejezetekben bemutatott SSG algoritmus a Pareto optimális megoldások megtalálására. Annyit kell tennünk, hogy az algoritmus mellé konstruálunk egy J jelölthalmazt, és minden esetben, amikor találtunk egy lehetséges megoldást az SSG algoritmussal, akkor frissítjük a fentiek alapján ezt a J jelölthalmazt. Az algoritmus működését az alábbi példán mutatjuk be:

Példa Az 5.1-es ábrán látható feladatban hat műveleti egység van az $o_1, \dots, o_6$ egységek. Az o_1 egység az A, B anyagokból gyártja az E anyagot, az o_2 egység a C anyagból gyártja az E és F anyagokat, az o_3 egység az D anyagból gyártja az F és G anyagokat, az o_4 egység az E anyagból gyártja a H anyagot, az o_5 egység az E, G anyagokból gyártja a H anyagot végül az o_6 egység az F, G anyagokból gyártja a H anyagot. Az A, B, C, D anyagok a nyersanyagok a H a végtermék. A műveleti egységek költségeit a $cf_1 = 3, df_1 = 2, cf_2 = 1, df_2 = 3, cf_3 = 2, df_3 = 2, cf_4 = 9, df_4 = 8, cf_5 = 6, df_5 = 2, cf_6 = 3, df_6 = 6$ értékek adják meg. Végrehajtva az SSG algoritmust megkaphatjuk az összes lehetséges megoldást. Ezek a következők $x_1 = \{o_1, o_4\}$, $x_2 = \{o_2, o_4\}$, $x_3 = \{o_1, o_2, o_4\}$, $x_4 = \{o_1, o_3, o_5\}$, $x_5 = \{o_2, o_3, o_5\}$, $x_6 = \{o_1, o_2, o_3, o_5\}$, $x_7 = \{o_3, o_6\}$, $x_8 = \{o_2, o_3, o_6\}$, $x_9 = \{o_1, o_3, o_4, o_5\}$, $x_{10} = \{o_2, o_3, o_4, o_5\}$, $x_{11} = \{o_1, o_2, o_3, o_4, o_5\}$, $x_{12} = \{o_1, o_3, o_4, o_6\}$, $x_{13} = \{o_2, o_3, o_4, o_6\}$, $x_{14} = \{o_1, o_2, o_3, o_4, o_6\}$, $x_{15} = \{o_1, o_3, o_5, o_6\}$, $x_{16} = \{o_2, o_3, o_5, o_6\}$, $x_{17} = \{o_1, o_2, o_3, o_5, o_6\}$, $x_{18} = \{o_1, o_3, o_4, o_5, o_6\}$, $x_{19} = \{o_2, o_3, o_4, o_5, o_6\}$, $x_{20} = \{o_1, o_2, o_3, o_4, o_5, o_6\}$. Az egyes megoldásokra számolt költségfüggvényeket az 5.1-es táblázat tartalmazza.

Tehát a megoldások generálása során a Pareto optimális megoldások tárolására használt jelöltilista a következőképpen változik. Az első megoldás x_1 bekerül a listába. Ezt követően az x_2 megoldást x_1 nem dominálja, így az bekerül e jelöltilistába. Majd x_3 -at dominálja x_1 , így az nem kerül be. Ezt követően x_4 dominálja x_1 -et de x_2 -t nem, így x_1 -et töröljük a jelöltek listájából és a helyére x_4 kerül. Az x_5 megoldást nem dominálja sem x_2 sem x_4 , így az is bekerül a jelöltek listájába. Másrészt x_5 do-

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
z_1	12	10	13	11	9	12	5	6	20	18
z_2	10	11	13	6	7	9	8	11	14	15
	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
z_1	21	17	15	18	14	12	15	23	21	24
z_2	17	18	19	21	12	13	15	20	21	23

5.1. táblázat. A megoldások költségei



5.1. ábra. A Pareto optimális megoldásokhoz tartozó példa

minálja x_2 -t, így azt törölnünk kell. Az x_6 megoldást dominálja x_4 , így az nem Pareto optimális. Az x_7 megoldást nem dominálja egyik jelölt sem, így az is bekerül a jelölthalmazba, és x_7 sem dominálja az ottani megoldásokat, így senkit sem kell törölni. Tehát a Pareto optimális jelöltek halmaza $\{x_4, x_5, x_7\}$ lesz. A többi megoldás mindegyikét dominálja x_7 , továbbá egyikük sem dominálja x_4 -et vagy x_5 -öt, így a jelöltek halmaza nem változik tovább. Következésképpen az x_4, x_5 és x_7 megoldások lesznek a Pareto optimálisak.

A fenti példa alapján látszik, hogy sok olyan lehetséges megoldás lehet, amelyeket a meglevő Pareto optimálisnak jelölt megoldások valamelyike dominál. Következésképpen az SSG algoritmus sok esetben olyan megoldásokat vizsgál, amelyek nem lesznek Pareto optimálisak. Az algoritmust gyorsíthatja, ha nem kell legenerálnunk az összes ilyen megoldást, hanem egy korlátozás és szétválasztáson alapuló algoritmusnak (mint az előző fejezetekben ismertetett ABB) egy kiterjesztésével levágjuk a megoldásokat tartalmazó fa néhány olyan ágát, amelyekben csak nem Pareto optimális megoldások vannak.

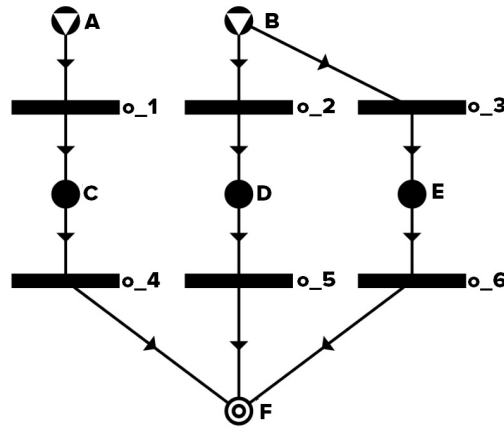
A korlátozás és szétválasztás algoritmusánál a lehetséges megoldások egy részhalmazára kiszámoljuk egy alsó korlátját a célfüggvénynek, és ha ez az alsó korlát nagyobb, mint egy már ismert megoldás költsége, akkor a részhalmazban levő megoldásokat nem kell tovább vizsgálnunk. Pareto optimális megoldások keresése során is használhatjuk ezt a megközelítést. Mindkét célfüggvény szerint ki kell számolnunk az alsó korlátot és amennyiben ezeknek a korlátoknak egyike sem kisebb, mint valamely már meglevő Pareto optimálisnak jelölt megoldás esetén számolt megfelelő értékek, akkor nem kell tovább vizsgálnunk a részhalmazba eső megoldásokat. Ezzel a kiterjesztéssel nagymértékben javíthatunk az előzőekben bemutatott egyszerű algoritmus hatékonyságán.

A fenti SSG-n alapuló megközelítés csak véges sok lehetséges megoldás esetén működik. Amennyiben a bonyolultabb fix részt tartalmazó lineáris költségfüggvényű modellt tekintjük, akkor az anyagáramok figyelembe vétele miatt végtelen sok lehetséges megoldásunk van. Másrészt ekkor is csak véges sok, a gyártást leíró lehetséges struktúra létezik és az SSG algoritmussal ezeket felsorolhatjuk. Így a feladat tulajdonképpen redukálódik arra, hogy egy rögzített struktúra mellett melyek lesznek a Pareto optimális anyagáramok, hiszen ezeket minden struktúrára meghatározva használhatjuk a fentiekben bemutatott algoritmust. Amennyiben egy hibrid modellt vizsgálunk, azaz az első szempont szerint fix gépköltségek vannak, a második szempont szerint pedig a fix költségek mellett az anyagáramoktól lineárisan függő költségek is, akkor az első cél szerinti költség adott és a második költség fix része is adott, így a feladatunk csak annyi, hogy meghatározzuk a rögzített struktúrára a második cél szerint optimális anyagáramot, ami pedig egy lineáris programozási feladat megoldása. A feladat jóval érdekesebb, ha mindkét szempont fix részt tartalmazó lineáris költségfüggvény. Ekkor a rögzített struktúra mellett egy kétcélú lineáris programozási feladat Pareto optimális megoldásait kell meghatároznunk. Kétcélú lineáris programozási feladatok kezelésére több módszer ismert, de azoknak a tárgyalása túlmutat jelen tankönyv keretein, az érdeklődő olvasó többek között a [10, 13] tanulmányokban találhat a témával kapcsolatos részleteket.

Miként a fix költséges modellben itt is van lehetőségünk az alapalgorithmus gyorsítására. Ugyanazt az alapötletet használhatjuk az algoritmus kiterjesztésére. Ebben az esetben is kiszámíthatjuk minden lépésben a korlátokat ezúttal a bonyolultabb költségfüggvényre vonatkozó eljárások alapján és lezárhatjuk azokat a részhalmazokat, amelyekre a korlátok alapján tudjuk, hogy minden megoldást dominál valamely már ismert Pareto optimális megoldás.

5.4. Aggregált célfüggvények

Elsőként azt a gyakran használt megoldást tárgyaljuk, ahol az aggregált célfüggvény az eredeti célfüggvények egy pozitív súlyokkal számolt lineáris kombinációja. Ez a módszer nagyon könnyen kezelhető mind a fix költséges mind pedig a lineáris költséges esetben. A teendőnk annyi, hogy ha a $z(X) = r_1 \cdot z_1(x) + r_2 \cdot z_2(x)$ célfüggvény szerinti optimális megoldást keressük, akkor definiálunk egy új feladatot a műveleti egységekhez az $ef_i = r_1 cf_i + r_2 df_i$ illetve $ep_i = r_1 cf_i + r_2 df_i$ költségeket rendelve. Ennek az egycélú feladatnak az optimális megoldása fogja adni az aggregált célfüggvényre nézve az optimális megoldást, hiszen az új súlyokkal definiált probléma célfüggvénye pontosan az aggregált feladat célfüggvénye lesz.



5.2. ábra. Az aggregált célfüggvényhez tartozó példa

Példa

Az 5.2-es ábrán látható példában hat műveleti egység van $o_1, \dots, o_6$. Az o_1 egység az A anyagból gyártja az C anyagot, az o_2 egység a B anyagból gyártja az D anyagot, az o_3 egység a B anyagból gyártja az E anyagot, az o_4 egység a C anyagból gyártja az F anyagot, az o_5 egység a D anyagból gyártja az F anyagot végül az o_6 egység az E anyagból gyártja az F anyagot. Az A, B anyagok a nyersanyagok az F a végtermék. A műveleti egységek költségeit a $cf_1 = 6, df_1 = 2, cf_2 = 3, df_2 = 4, cf_3 = 1, df_3 = 7, cf_4 = 9, df_4 = 4, cf_5 = 5, df_5 = 5, cf_6 = 3, df_6 = 7$ értékek adják meg. A lehetséges megoldások közül három olyan van, amelyek mindegyike része valamely megoldásnak, így csak ezek jöhetnek szóba optimális megoldásként. Ezek az $x_1 = \{o_1, o_4\}, x_2 = \{o_2, o_5\}, x_3 = \{o_3, o_6\}$ megoldások. Az egyes megoldásokra kiszámítva a célfüggvényeket azt kapjuk, hogy $z_1(x_1) = 15, z_2(x_1) = 6, z_1(x_2) = 8, z_2(x_2) = 9, z_1(x_3) = 4, z_2(x_3) = 14$. Tehát a z_1 célfüggvény szerint az x_3 megoldás az optimális, a z_2 célfüggvény szerint az x_1 az optimális megoldás. Másrészt ha az aggregált $z(X) = z_1(x) + z_2(x)$ célfüggvényt tekintjük, akkor $z(x_1) = 21, z(x_2) = 17$ és $z(x_3) = 18$, így erre a célfüggvényre az optimális megoldás már az x_2 lesz. Érdekes megjegyezni, hogy ebből következik, hogy az x_1, x_2, x_3 megoldások mindegyike Pareto optimális, de ez látszik a Pareto optimális megoldások definíciója alapján is. Az aggregált költségfüggvényre itt az optimális megoldást úgy kaptuk, hogy minden esélyes megoldásra kiszámoltuk a költségeket. Mint a fentiekben írtuk bonyolultabb feladatok esetén van hatékonyabb módszer, mint az összes lehetséges megoldásra a célfüggvények kiszámítása és ebből az aggregált érték képzése. Ebben a példában a z aggregált célfüggvény esetén az új $ef_1 = 8, ef_2 = 7, ef_3 = 8, ef_4 = 13, ef_5 = 10, ef_6 = 1$ értékeket kell a műveleti egységekhez rendelni, és ezen költségekkel megoldani a feladatot. Ezekkel a költségekkel is azt kapjuk, hogy az optimális megoldás x_2 és a költség 17.

Sokkal bonyolultabb a helyzet, ha az aggregált függvény egy nemlineáris g monoton függvénye az eredeti célfüggvényeknek. Amennyiben megvannak a lehetséges megoldásaink, akkor egy ilyen x megoldásra kiszámolhatjuk a célfüggvény értékét mindkét szempont szerint, majd vehetjük ezen $f_1(x), f_2(x)$ értékekre a $g(f_1(X), f_2(X))$ értéket. De ebben az általános esetben már nem tudjuk új

súlyok megadásával visszavezetni az aggregált költségfüggvényű feladat megoldását egy klasszikus feladat megoldására. Ekkor a korlátozás és szétválasztás elvén alapuló eljárásokat kell kiterjesztenünk az aggregált feladat megoldására. A lehetséges megoldások ugyanazok, mint az eredeti feladatnál, így a korlátozás és szétválasztás alapján működő ABB algoritmusban csak a korlátozó függvényt és a lehetséges megoldások értékét meghatározó függvényt kell újra definiálnunk.

Mint fent már említettük a lehetséges megoldások esetén a célfüggvényértékek megadása egyszerű, mindkét célfüggvény kiszámítható és ezen célfüggvényértékeket behelyettesítve a g aggregációs függvénybe megkapjuk a megoldásokat. A korlátozó függvényénél is van egy egyszerű megoldás. Megtehetjük, hogy mindkét célfüggvény szerint kiszámítjuk az optimális értékre vonatkozó L_1 és L_2 alsó korlátokat, majd ezekből képezzük a $g(L_1, L_2)$ alsó korlátot. Ez a megközelítés az alsó korlátnál egymástól függetlenül kezeli az egyes szempontokat.

A következő példa jól mutatja ennek a megközelítésnek a hátrányát. Tegyük fel, hogy van egy megoldáshalmazunk, ami 3 megoldást tartalmaz, legyenek ezek x_1, x_2 , és x_3 , a $z_1(x_1) = 12, z_2(x_1) = 40, z_1(x_2) = 25, z_2(x_2) = 25$ és $z_1(x_3) = 35, z_2(x_3) = 15$ célfüggvényértékekkel. Továbbá tegyük fel, hogy az aggregációs függvény $g(x, y) = x^2 + y^2$. Ekkor a z_1 szerinti alsó korlát legfeljebb 12 lehet, a z_2 szerinti alsó korlát legfeljebb 15, így a fenti módszerrel nem kaphatunk jobb korlátot, mint $12^2 + 15^2 = 369$. Másrészt az aggregált célfüggvényre a költségek rendre 1744, 1250, 1450, így lényegesen jobb korlát is elképzelhető, mint amit az egyes szempontok egymástól függetlenül való kezelésével kapunk.

Másrészt itt is, mint általában a korlátozás és szétválasztás elvén alapuló algoritmusok esetében ügyelni kell a korlát élessége és a korlát kiszámításához szükséges idő egyensúlyára. A klasszikus, egy célú PNS modellek esetén a korlátszámítás többnyire egy egyszerű és gyors algoritmussal megoldható. A fix gépköltséges modellben az egyik legelterjedtebb korlátot úgy kapjuk, hogy a már kiválasztott gépek költségeihez hozzáadjuk a maximális struktúrában a kiválasztott gépektől nyersanyagig vezető legrövidebb út költségét. Ennek a módszernek a kiterjesztése nem könnyű, hisz egy nemlineáris aggregált függvény esetén az aggregált költségre vonatkozó alsó korlát nem kapható meg egy legrövidebb utat kereső gráfalgoritmussal. Amennyiben a fix részt tartalmazó lineáris költségfüggvényű modellt vizsgáljuk ott is nehézségekbe ütközünk egy nemlineáris aggregált függvény közvetlen kezelésével, hiszen nem elegendő egy lineáris programozási feladatot megoldanunk, hanem nemlineáris feladattal kerülünk szembe.

5.5. Korlátozott feladatok

Ebben a modellben a lehetséges megoldások halmaza is függ a korlátoktól, amelyeket az egyes célfüggvényekre teszünk, így a feladat megoldására vonatkozó eljárásokat több szempontból is ki kell terjesztenünk. Az alábbiakban feltesszük, hogy mint az eddigiekben két célfüggvényünk van z_1 és z_2 . A z_2 függvényre adott egy C_2 felső korlát, amely korlátnál nagyobb költséggel rendelkező megoldásokat nem tekintünk lehetségesnek.

Tehát az eredeti feladat megoldásai közül néhány megoldást ki kell zárunk. Amennyiben az összes megoldást szeretnénk legenerálni, akkor azt megtehetjük az SSG algoritmus egyszerű módosításával, amely minden generált megoldás esetén ellenőrzi, hogy a $z_2(x) \leq C_2$ feltétel teljesül-e, és csak a feltételt kielégítő megoldásokat veszi be a lehetséges megoldások halmazába.

Példa

Tekintsük ismét az aggregált célfüggvényeknél bemutatott, az 5.2. ábrán látható példát. Ha a $C_2 = 14$, akkor az x_1, x_2, x_3 megoldások mindegyike lehetséges megoldása a korlátozott feladatnak, így az optimális megoldás x_3 lesz, és az optimális z_1 érték 4. Amennyiben a $C_2 = 9$ szigorúbb feltétel mellett keressük a feladat optimális megoldását, akkor csak az x_1 és x_2 megoldások elégítik ki a feltételt, így a korlátozott feladat optimális megoldása x_2 lesz, az optimális z_1 érték pedig 8-ra növekedik. Végül, ha a $C_2 = 6$ korlátot alkalmazzuk, akkor kizárjuk az x_2 megoldást is, így az optimális megoldás x_1 lesz és az optimális z_1 érték 15.

A lehetséges megoldások halmazának felhasználásával megkaphatjuk azt az elemet, amelyre a halmazban a legkisebb a z_1 célfüggvény értéke és ezzel megtalálhatjuk a korlátozott feladat optimális megoldását. Másrészt - miként a klasszikus feladatban is - egy hatékonyabb megoldó algoritmust keresünk. Ehhez ki kell terjesztenünk a korlátozás és szétválasztás elvén működő eljárásokat (mint pl. az előző fejezetekben tárgyalt ABB) az általánosabb feladat megoldására. Az aggregált célfüggvényekre felépített modellel ellentétben, itt nem változnak a lehetséges megoldások célfüggvény értékei, de változik a megengedett lehetséges megoldások halmaza. Tehát itt nem a korlátozó függvény változásait kell megvizsgálnunk, hanem a szétválasztó függvényt is.

A szétválasztó függvény esetén egy lépésben megvizsgáljuk, hogy egy adott anyagot mely műveleti egységek gyárthatnak, és a lehetséges döntések szerint bontjuk tovább a lehetséges megoldások halmazát. A célunk az, hogy azon megoldáshalmazokat, amelyekben nem teljesülhet a $z_2(X) \leq C_2$ feltétel ne vizsgáljuk tovább. Vegyük észre, hogy ilyen halmazok kizárására használhatjuk a z_2 cél szerinti korlátozó függvényt is, hiszen azon megoldáshalmazokat, amelyekre ez a függvény biztosítja, hogy nem adnak jobb megoldást C_2 -nél kizárhatjuk. Tehát az algoritmus minden új részhalmaz generálása során kiszámolja a z_1 és z_2 szerinti korlátozó függvényeket, és azon megoldáshalmazokat, amelyek minden elemére kiderül, hogy $z_1(x)$ rosszabb egy már ismert megoldásnál, vagy $z_2(x) > C_2$ kizárjuk, azaz az ilyen kezdeti megoldások további kiterjesztéseit nem vizsgáljuk.

A korlátozás és szétválasztás algoritmusának ezen kiterjesztésének az az előnye, hogy közvetlenül használhatjuk az egyes célfüggvények szerinti korlátozó függvényeket. Másrészt ennek a megközelítésnek az a hátránya, hogy a két célpont szerint egymástól függetlenül vesszük a korlátokat. Amennyiben mindkét szempontot egyszerre figyelembe vesszük a korlátok számításánál, akkor sok esetben hatékonyabban zárhatunk ki megoldáshalmazokat. Ennek illusztrálására vegyük a következő példát. Tegyük fel, hogy van egy megoldáshalmazunk, ami 3 megoldást tartalmaz, legyenek ezek x_1, x_2 , és x_3 , a $z_1(x_1) = 15$, $z_2(x_1) = 40$, $z_1(x_2) = 21$, $z_2(x_2) = 26$ és $z_1(x_3) = 30$, $z_2(x_3) = 24$ célfüggvényértékekkel. Továbbá tegyük fel, hogy a legjobb ismert megoldásunk költsége 20 és legyen $C_2 = 25$. Ekkor a z_1 szerinti alsó korlát legfeljebb 15 lehet, a z_2 szerinti alsó korlát legfeljebb 24, így a korlátozás és szétválasztás algoritmusának fentiekben leírt kiterjesztése esetén ezt a megoldáshalmazt tovább kell vizsgálnunk, hisz egyik szempont szerint sem zárhatjuk ki ezt a halmazt.

Itt is ugyanazzal a problémával kerülünk szembe, mint az aggregált célfüggvények esetén. Különböleg vizsgáljuk a célfüggvényekre vonatkozó korlátokat, nem pedig együttesen és ezáltal csökken a korlát hatékonysága. Ilyen esetekben segíthet, ha a z_1 -re és z_2 -re vonatkozó korlátok mellett felhasználunk még egy aggregált korlátot is, ami a két függvény valamely lineáris kombinációja. Mint az aggregált célfüggvények esetén láttuk lineáris kombinációk esetén nem okoz extra nehézséget a korlát számítása. Amennyiben a fenti példában használjuk a $z = z_1 + z_2$ függvényt is, akkor arra kaphatunk egy 47-es alsó korlátot. Másrészt csak olyan megoldások érdekelnek minket, amelyekre $z_1(x) < 20$ és $z_2(x) \leq 25$, így ezen aggregált z -re vonatkozó korlátozással már kizárhatjuk ezt a halmazát a megoldásoknak.

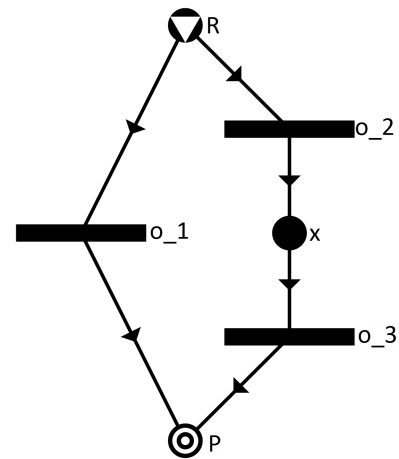
5.6. Robusztus optimalizálás

5.6.1. A robusztus modell

Ebben a részben a PNS probléma robusztus változatát tekintjük át. Mint a fejezet elején említettük ebben a változatban nem a klasszikus értelemben van több különböző célfüggvény, amiket egyszerre kell figyelembe vennünk, hanem tulajdonképpen egy célfüggvényünk van, de nem tudjuk pontosan a benne szereplő paramétereket. Ez a bizonytalanság kezelhető fuzzy módszerekkel, ahogy a P-gráfok workflow problémákra való alkalmazása során tették a [15] cikkben, vagy a P-gráfot elérhetőségi valószínűségekkal kiterjesztve, ahogy a [14] cikkben tették, ellátási láncok modellezése esetén. Ezekben az esetekben kell valami előzetes információ az egyes valószínűségekről. Ezekkel a sztochasztikus megközelítésekkel szemben a robusztus optimalizálás teljesen determinisztikus, itt nem eloszlásokkal dolgozunk csak egy intervallumra van szükségünk a paraméterek lehetséges értékeiről. Több robusztus megközelítést kidolgoztak kombinatorikus feladatokra, részletes áttekintés található a [3, 5] tanulmányokban.

Itt azt a modellt mutatjuk be, amelyben van egy előzetes információnk a paraméterek azon értékeire, amelyet felvehetnek, ilyen modelleket vizsgálnak többek között a [6] munkában. Ez a kombinatorikus PNS probléma esetén azt jelenti, hogy minden műveleti egységnek két különböző költsége van egy nominális és egy kiterjesztett költsége, továbbá ismert még egy b korlát is, amely azt adja meg, hogy legfeljebb b műveleti egység vehet fel kiterjesztett költséget, a többieknek marad a nominális költsége. Ezen feltételek mellett keressük az optimális megoldást. Ezt a modellt vizsgálják a [1, 16] tanulmányok.

Az o_i műveleti egységnek a nominális költségét $c(o_i)$ -vel jelöljük, a kiterjesztett költség pedig $c(o_i) + e(o_i)$, ahol $e(o_i)$ -t extra költségnek hívjuk. Ezt a jelölést kiterjesztjük műveleti egységek halmazaira is, egy Q halmazra $c(Q)$ jelöli a Q -ban levő műveleti egységek nominális költségeinek az összegét. Mivel a legrosszabb esetre vonatkozó célfüggvényt vizsgálunk, ezért egy lehetséges megoldás esetén a megoldás költségét úgy kapjuk meg, hogy vesszük a megoldásban szereplő műveleti egységek nominális költségeit és azt megnöveljük a megoldásban szereplő b darab legnagyobb extra költséggel. A robusztus modellben a b értékétől függően egészen különböző optimális megoldásokat kaphatunk, mint azt az alábbi példa mutatja.



5.3. ábra. A robusztus optimalizáláshoz tartozó példa

Példa: Tekintsük azt az 5.3-as ábrán ábrázolt példát, amiben R a nyersanyag, P a végtermék és egy további anyagunk van, amit X jelöl. Három műveleti egység van: o_1 közvetlenül gyártja P -t R -ből és $c(o_1) = 5$, $e(o_1) = 2$, o_2 a közbenső X anyagot gyártja R -ből és $c(o_2) = 2$, $e(o_2) = 2$, végül o_3 pedig P -t gyártja X -ből és $c(o_3) = 2$, $e(o_3) = 2$. Ha a feladatot a nominális költségekkel tekintjük, akkor az optimális megoldás o_2 -t és o_3 -at tartalmazza és az optimális költség 4. Ha a robusztus modellt tekintjük a $b = 1$ korláttal, akkor még mindig o_2 -t és o_3 adja az optimális megoldást, de ekkor a költség már 6. Végül ha a robusztus modellt tekintjük a $b = 2$ korláttal, akkor az optimális megoldás o_1 lesz és az optimális költség 7.

5.6.2. A fix költséges modell megoldása korlátozás és szétválasztással

Ahogy a fentiekben tárgyalt modellekben itt is ugyanaz a lehetséges megoldások halmaza, mint az eredeti feladat esetén és csak a lehetséges megoldások esetén számolt költség különbözik az eredetitől. Tehát a feladat megoldható az SSG algoritmus végrehajtásával, az összes lehetséges megoldás legenerálásával és azokból a legkisebb költségű kiválasztásával. Itt is egy hatékonyabb algoritmust szeretnénk, így a korlátozás és szétválasztás módszerén alapuló eljárások lehetséges kiterjesztését kell megvizsgálnunk. Mivel a lehetséges megoldások halmaza ugyanaz, ezért itt csak a korlátozó függvény változik, azt kell külön kezelnünk. Tehát egy olyan függvényt kell definiálnunk, amely minden részmegoldás esetén egy alsó korlátot ad azokra a lehetséges megoldásokra, amelyek ezen részmegoldás kiterjesztésével kaphatóak.

A legegyszerűbb függvényt akkor kapjuk, ha vesszük azokat a műveleti egységeket, amelyekre már eldöntöttük, hogy szerepelnek a megoldásban és ehhez hozzáadjuk az ezen műveleti egységek között található legnagyobb b extra költséget. Másrészt bonyolultabb költségfüggvényeket is definiálhatunk. Egy további gyakran használt függvény az, amelyben a már kiválasztott műveleti egységek költségeinek összegéhez hozzáadjuk a maximális struktúrában keresett legrövidebb út hosszát (az út hossza a benne szereplő műveleti egységek költségeinek összeg), amely a kiválasztott műveleti egységekből nyersanyaghoz vezet. Azért a nyersanyagokhoz vezető utakat vizsgáljuk, mert a megoldás építését a végtermékeket gyártó egységekkel kezdjük. Ez az alsó korlát is kiterjeszthető a robusztus modellre. Számíthatjuk azt a korlátot, amelyet úgy kapunk, hogy ehhez a korláthoz hozzáadjuk a kiválasztott műveleti egységek között található legnagyobb b extra költséget.

Végül fontosnak tarjuk megjegyezni, hogy a strukturális alapon végrehajtott gyorsítások, mint az előző fejezetekben ismertetett NX és RSG részalgoritmusok a robusztus modellre kifejlesztett korlátozás

és szétválasztás eljárásban is ugyanabban a formában használhatóak, hiszen azok a lehetséges megoldások halmazán alapulnak, amely ebben a feladatban is ugyanaz.

5.6.3. Heurisztikus algoritmusok a fix költséges modell megoldására

A korlátozás és szétválasztás algoritmusai legrosszabb esetben exponenciális futási idejű, így nagyon nagyméretű feladatok esetén szükséges lehet heurisztikus algoritmusok használata, amelyek nem feltétlenül találják meg az optimális megoldást, de gyorsan adnak egy remélhetőleg optimálishoz közeli célfüggvényértékkel rendelkezőt. Továbbá az ilyen algoritmusok felhasználhatóak a korlátozás és szétválasztáson alapuló egzakt megoldó algoritmusok gyorsítására is, hiszen egy jó kezdeti megoldás lehetővé teszi, hogy kisebb értékkel kelljen a számított korlátokat összehasonlítani és ezáltal hamarabb abbahagyhatjuk az egyes optimális megoldást nem tartalmazó részhalmazok átvizsgálását.

Az alábbiakban bemutatunk két heurisztikus algoritmosztályt, amelyek képesek gyorsan egy megoldást adni a feladatnak. Az algoritmusok a [7] cikkben a klasszikus fix költséges problémára bemutatott algoritmusok kiterjesztései, amelyeket a [1] cikkben publikáltak. Mindkét algoritmosztály hasonló ötleten alapul. Az algoritmusok egy *an* költségfüggvényt használnak, amely minden anyaghoz egy nemnegatív költséget rendel hozzá, ami arról tartalmaz valamilyen információt, hogy mekkora az illető anyag gyártásának költsége. Az algoritmusok elve az, hogy mindig azt a gépet választjuk, amelyre a gép becsült *be* költségének és a gép input anyagai gyártásához szükséges költségnek az összegét, elosztva a gyártandó anyagok közül a gép által gyártott anyagok számával, a kapott hányados minimális. Kezdetben a gyártandó anyagok halmaza a P halmaz. Az eljárás során, amennyiben egy gépet felveszünk a kiválasztott gépek közé, akkor a gép azon input anyagai, amelyeket még nem gyártunk és amelyek nem nyersanyagok bekerülnek a gyártandó anyagok közé. Az algoritmus akkor ér véget, ha a gyártandó anyagok halmaza üressé válik. A két algoritmus között a különbség annak becslésében van, hogy mennyi egy gép input anyagai gyártásának költsége. Az első esetben, ezt a költséget az anyagok *an* által felvett értékeinek összegével, a második esetben az anyagok ezen értékeinek maximumával becsüljük.

A fentiekben vázolt algoritmusok az alábbi formában írhatók le.

Asum_c algoritmus

- *Előkészítő rész.* Legyen $N_0 = P$, $O_0 = \emptyset$, $K_0 = R$, és $i = 0$.
- *Iterációs rész* (i -edik iteráció).
 - 1. lépés. Amennyiben $N_i = \emptyset$, akkor lépünk a 3. lépésre. Egyébként számoljuk ki minden $u \notin O_i$ gépre, amely N_i -beli anyagot gyárt a

$$\frac{be(u) + \sum_{X \in \psi^-(u)} an(X)}{u(N_i)}$$

hányados, ahol $u(N_i)$ azon N_i beli anyagok száma, amelyek u output anyagai. Válasszunk egy olyan gépet, amelyre ez a hányados minimális, legyen ez a gép v .

- 2. lépés. Legyen $O_{i+1} = O_i \cup \{v\}$, és $K_{i+1} = K_i \cup \psi^+(v)$ $N_{i+1} = N_i \cup \psi^-(v) \setminus K_{i+1}$. Növeljük i értékét eggyel és térjünk rá a következő iterációra.
- 3. lépés. Legyen $o = O_i$, és $m = \psi(o)$.

Amax_c algoritmus

- *Előkészítő rész.* Legyen $N_0 = P$, $O_0 = \emptyset$, $K_0 = R$, és $i = 0$.
- *Iterációs rész* (r -edik iteráció).

- 1. lépés. Ha $N_i = \emptyset$, akkor lépünk a 3. lépésre. Egyébként számoljuk ki minden $u \notin O_i$ gépre, amely N_i -beli anyagot gyárt a

$$\frac{be(u) + \max\{an(X) : X \in \psi^-(u)\}}{u(N_i)}$$

hányadost, ahol $u(N_i)$ azon N_i belüli anyagok száma, amelyek u output anyagai. Válasszunk egy olyan gépet, amelyre ez a hányados minimális, legyen ez a gép v .

- 2. lépés. Legyen $O_{i+1} = O_i \cup \{v\}$, és $K_{i+1} = K_i \cup \psi^+(v)$ $N_{i+1} = N_i \cup \psi^-(v) \setminus K_{i+1}$. Növeljük i értékét eggyel és térjünk rá a következő iterációra.
- 3. lépés. Legyen $o = O_i$, és $m = \psi(o)$.

Az algoritmusok helyességéhez elsőként igazoljuk, hogy azok a c függvénytől függetlenül mindig lehetséges megoldást eredményeznek. Ezt követően azt tárgyaljuk, hogy milyen an és be becsléseket használhatunk. Az $m = \psi(o)$ definíció alapján egyből adódik, hogy az o és m halmazok egy olyan hálózati gráfot adnak meg, amelyre az (S5) tulajdonság teljesül. Mivel csak olyan elemeket választunk ki a megoldásban, amelyek a szintézis feladatban definiáltak, ezért nyilvánvalóan teljesül az (S3) tulajdonság is.

Az (S1) tulajdonság igazolásához elsőként vegyük észre, hogy K_{i+1} az R -beli anyagokat, és azon anyagokat tartalmazza, amelyekre van O_{i+1} -ben olyan gép, amelynek az illető anyag output anyaga. Ez az észrevétel i szerinti indukcióval K_{i+1} és O_{i+1} konstrukciója alapján egyből adódik. Viszont ez $N_{i+1} = N_i \cup \psi^-(v) \setminus K_{i+1}$ alapján azt jelenti, hogy az N_i halmazból csak olyan elemek kerülnek ki, amelyekhez van O_{i+1} -ben olyan gép, amelynek az illető anyag output anyaga. Másrészt $N_0 = P$, így ha valamely i -re teljesül $N_i = \emptyset$, akkor minden $X \in P$ -re van olyan gép O_i -ben, amelynek X output anyaga. Következésképp, $P \subseteq m$ valóban teljesül.

Az (S2) tulajdonság igazolásához vegyük észre, hogy mivel $PG(M, O)$ maximális struktúra, ezért teljesül rá (S2). Így O -ban nincs olyan gép, amely nyersanyagot gyárt. Viszont $o \subseteq O$, így o -ban sincs nyersanyagot gyártó gép. Másrészt m konstrukciója alapján, minden $X \in m$ anyagra $X \in \psi^-(u)$ vagy $X \in \psi^+(u)$ teljesül valamely $u \in o$ gépre. Amennyiben $X \in \psi^+(u)$, akkor nyilvánvalóan vezet X -be él az u gépből a $PG(m, o)$ gráfban. Tegyük fel, hogy $X \notin R$ és $X \in \psi^-(u)$ egy olyan u gépre, amely az i -edik iteráció során került bele az O_i halmazba. Ekkor, amennyiben $X \in K_i$, akkor van olyan gép O_i -ben, amely X -et gyártja, így ismét vezet él X -be a $PG(m, o)$ gráfban. Végül, ha $X \notin K_i$, akkor $X \in N_i$ teljesül az N_i halmaz konstrukciója alapján. Ebben az esetben (S1) igazolásához teljesen hasonlóan adódik, hogy $N_r = \emptyset$ esetén lesz O_r -ben olyan gép, amely X -et gyártja. Tehát megmutattuk, hogy $X \notin R$ esetén vezet él X -be a $PG(m, o)$ gráfban, amivel az (S2) tulajdonságot is igazoltuk.

Az (S4) tulajdonság igazolásához az O_i -beli gépekre megmutatjuk i szerinti indukcióval, hogy O_i minden gépéből vezet út valamely végtermékhez. Mivel O_1 egyetlen gépet tartalmaz, egy olyan gépet, amelynek van végtermék output anyaga, ezért az állítás $i = 1$ esetén teljesül. Most legyen $i \geq 1$ és tegyük fel, hogy az állítás teljesül O_i -re. Igazoljuk, hogy amennyiben nem ér véget az i -edik iterációs részben az eljárás, akkor az állítás igaz O_{i+1} -re is. Mivel $O_{i+1} = O_i \cup \{v\}$, ezért az indukciós feltevés alapján elegendő az algoritmus $i + 1$ -edik iterációs részében választott v gépre igazolni, hogy v -ből vezet út valamely végtermékbe. Ezen v gépnek van N_i -beli X output anyaga. Ez az output anyag vagy végtermék, vagy input anyaga valamely $u \in O_i$ gépnek. Az első esetben v -ből él vezet egy végtermékbe, így az állítás teljesül. A második esetben az indukciós feltevés alapján u -ból vezet út egy végtermékbe, és egy ilyen út elejére felvéve a v gépet és az X anyagot, egy megfelelő utat kapunk.

Igazoltuk, hogy az eljárás végén kapott m, o halmazokra $PG(m, o)$ kielégíti az (S1), ..., (S5) axiómák mindegyikét, tehát a bemutatott algoritmusok valóban lehetséges megoldásokat szolgáltatnak.

Most tekintsük át milyen közelítéseket használhatunk az an és be függvények esetén. A gépek költségének becslésénél csak a robusztusság problémájával kerülünk szembe, az adott gép kiválasztásakor még nem tudhatjuk, hogy a megoldásban a nominális vagy a kiterjesztett költség lesz majd figyelembe véve. Ezért valamilyen módon mindkét költséget figyelembe kell vennünk. Az alábbi függvényeket dolgozták ki a probléma kezelésére:

- *Átlagolt költség*: Ebben az esetben a nominális és a kiterjesztett költség egy súlyozott átlaga

lesz a becslés a gép költségére. Azaz $be_A(o) = \lambda c(o) + (1 - \lambda)(c(o) + e(o))$ valamely $0 < \lambda < 1$ értékre. A választott λ érték függhet a feladat b paraméterétől. Minél több esetben kell a kiterjesztett költséggel számolnunk, annál kisebb λ értéket érdemes használni.

- *Legrosszabb költség:* Ebben az esetben minden esetben a kiterjesztett költséggel számolunk, amikor nem tudjuk biztosan, hogy az adott gép esetén a nominális költség kerül beszámításra. Azaz $be_{LR}(o) = c(o) + e(o)$, amennyiben nincs b darab már kiválasztott műveleti egység, akiknek az extra költsége legalább $e(o)$ és $be_{LR}(o) = c(o)$ különben.
- *Hibrid költség:* Ebben az esetben minden esetben az átlagolt költséggel számolunk, amikor nem tudjuk biztosan, hogy az adott gép esetén a nominális költség kerül beszámításra. Azaz $be_H(o) = \lambda c(o) + (1 - \lambda)(c(o) + e(o))$ valamely $0 < \lambda < 1$ értékre, amennyiben nincs b darab már kiválasztott műveleti egység, akiknek az extra költsége legalább $e(o)$ és $be_H(o) = c(o)$ különben.

Most tekintsük át miként becsülhetjük meg az anyagok gyártásának költségét. Ez a költség függ a gépek költségétől de mivel a robusztus költséget vesszük figyelembe, így a költség kiszámításánál itt a $be_A(o)$ értékeket használjuk a c értékek helyett. Becslésnek a minimális előállítási költségre vonatkozó korlátot használhatjuk, amelyet először a [11] dolgozatban vezették be, ahol ezt a függvényt a PNS probléma megoldására szolgáló korlátozási és szétválasztási eljárásban használták. A függvényt kiszámító eljárás igen bonyolult, ezért itt csak az egyszerűsített változatát ismertetjük, amely körmentes gráfok esetén működik.

Mielőtt formálisan definiálnánk az algoritmust ismertetjük az alapötleteket.

Minden iterációs lépésben két halmazunk lesz, az $I^{(r)}$ és $J^{(r)}$ halmazok. $I^{(r)}$ lesz azon anyagok halmaza, amelyekre már kiszámítottuk az *an* függvényértékeket, $J^{(r)}$ pedig ennek a halmaznak a komplementere lesz, amelyre még ki kell számolnunk ezeket az értékeket. Ennek megfelelően $I^{(0)}$ kezdetben csak a nyersanyagokat fogja tartalmazni 0 költséggel. Az algoritmus r -dik lépésében választunk egy olyan X anyagot $J^{(r)}$ -ből, amelyet csak olyan műveleti egységek gyártanak, amelyeknek az összes bemeneti anyaga az $I^{(r)}$ halmazban van. Ezt követően határozzuk meg az X anyagra a költségfüggvény értékét, majd átrakjuk X -et a $J^{(r)}$ halmazból az $I^{(r)}$ halmazba. Most megadjuk a pontos algoritmust, ami kiszámolja az *an* függvényt.

Az *an* függvényt kiszámító algoritmus

- *Előkészítés*

Legyen $I^{(0)} = R$, $J^{(0)} = M \setminus R$, $r = 0$, és $an(X) = 0$, minden $X \in M$.

- *Iterációs rész (r -dik iteráció)*

Az eljárás véget ért, ha $J^{(r)} = \emptyset$. Egyébként válasszunk egy olyan $X \in J^{(r)}$ anyagot, amelyre az anyagot gyártó $u_1, \dots, u_j$ műveleti egységek mindegyike olyan, hogy a bemeneti anyagai az $I^{(r-1)}$ halmazba esnek. Minden ilyen u_t gépre számítsuk ki a

$$c_t = be_A(u_t) + \max\{an(V) : V \in \psi^-(u_t)\},$$

értéket és legyen $an(X)$ ezen értékek minimuma. Továbbá legyen $I^{(r+1)} = I^{(r)} \cup \{X\}$ és $J^{(r+1)} = J^{(r)} \setminus \{X\}$. Növeljük r értékét 1-el és térjünk át a következő iterációra.

Fontos megjegyezni, hogy az algoritmus működéséhez szükséges, hogy minden lépésben, amikor $J^{(r)} \neq \emptyset$ teljesül legyen olyan anyag, amelyre az anyagot gyártó $u_1, \dots, u_j$ műveleti egységek mindegyike olyan, hogy a bemeneti anyagai az $I^{(r-1)}$ halmazba esnek. A körmentesség miatt ez könnyen igazolható egy indirekt bizonyítással. Ehhez tegyük fel, hogy az állítás nem igaz, azaz minden $X \in J^{(r)}$ esetén van olyan műveleti egység, aminek van bemeneti anyaga a $J^{(r)}$ halmazban. Válasszunk egy A_0 anyagot a $J^{(r)}$ halmazból. Az indirekt feltevés alapján van olyan o_1 műveleti egység, amely gyártja A_0 -t és van egy A_1 bemeneti anyaga a $J^{(r)}$ halmazban. Ugyanez az állítás igaz A_1 -re, így lesz egy o_2 műveleti egység, ami gyártja A_1 -et és van egy $A_2 \in J^{(r)}$ bemeneti anyaga. Ha folytatjuk ezt a konstrukciót, akkor az anyagoknak és műveleti egységeknek egy olyan $A_0, A_1, \dots$ és $o_1, o_2, \dots$ sorozatát kapjuk, ahol A_{i-1} és A_i rendre bemeneti és kimeneti anyagai az o_i műveleti

egységnek. Másrészt az anyagok halmaza véges, így teljesülni fog $A_\ell = A_k$ valamely $k > \ell$ párosra. Ekkor $A_\ell, o_{\ell+1}, A_{\ell+1}, \dots, o_k, A_k$ egy kört alkot feladat P-gráfjában, amivel ellentmondáshoz jutunk.

5.6.4. A lineáris költségfüggvény robusztus modelljei

Az általánosabb, fix részt tartalmazó lineáris költségfüggvényű modell esetén két lehetséges robusztus kiterjesztés is használható. Az egyszerűbb modellben csak a fix költségrészben használjuk a robusztus kiterjesztést, azaz ebben az esetben az arányossági konstansok rögzített értékűek. Ekkor ki tudjuk terjeszteni a tankönyv előző részeiben bemutatott ABB algoritmust, csak annyit kell hozzátennünk, hogy a korlátozó függvényhez hozzáadjuk kiválasztott műveleti egységek között található legnagyobb b extra költséget. Amennyiben az arányossági konstansokat is robusztus modellben keressük, akkor a korlátozó függvények számítása során egy robusztus lineáris programozási feladatot kell megoldanunk. Ennek megoldása túlmutat jelen tankönyv keretein, az érdeklődő olvasó a [3, 4] tanulmányokban megtalálhatja a téma tárgyalását.

Irodalomjegyzék

- [1] D. Almási, Cs. Imreh, T. Kovács, J. Tick, Heuristic algorithms for the robust PNS problem, *Acta Polytechnica Hungarica*, **11(4)**, 169–181, 2014.
- [2] M. Bárány, B. Bertók, Z. Kovács, F. Friedler, L. T. Fan, Solving vehicle assignment problems by process-network synthesis to minimize cost and environmental impact of transportation, *Clean Technologies and Environmental Policy*, **13(4)**, 637–642, 2011.
- [3] A. Ben-Tal, L. El Ghaoui, A. Nemirovski, *Robust Optimization*, Princeton University Press, 2009.
- [4] A. Ben-Tal, A. Nemirovski, Robust Solutions of Uncertain Linear Programs, *Operations Research Letters*, **25(1)**, 1–13, 1999.
- [5] D. Bertsimas, D. Brown, C. Caramanis, Theory and applications of robust optimization, *SIAM Review*, **53**, 464–501, 2011.
- [6] D. Bertsimas, M. Sim, Robust Discrete optimization and Network Flows, *Mathematical Programming Series B*, **98**, 49–71, 2003.
- [7] Z. Blázsik, Cs. Holló, Cs. Imreh, Z. Kovács, Heuristics for the PNS problem, *Applied Optimization Vol 59, Optimization Theory, Mátrahaza 1999* eds. F. Gianessi, P. Pardalos, T. Rapcsák, Kluwer Academic Publishers, Dordrecht, Boston, London, 1–18, 2001.
- [8] Y. Collett, P. Siarry, *Multiobjective Optimization: Principles and case studies*, Springer, 2003.
- [9] M. Ergotth, *Multicriteria Optimization*, Springer 2005.
- [10] J.P. Evans, R.P. Steuer, A revised simplex method for multiple objective programs, *Mathematical Programming*, **5**, 54–72, 1973.
- [11] B. Imreh, 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*, editors: I. M. Bonze, T. Csendes, R. Horst, P. M. Pardalos, Kluwer Academic Publisher, Dordrecht, Boston, London, 301–348, 1996.
- [12] Cs. Imreh, Z. Kovács, On pollution minimization in the optimization models of process network synthesis, *Chemical Engineering Transactions*, **7(2)**, 565–570, 2005.
- [13] H. Masri, S. Krichen, A. Guitouni, Generating efficient faces for multiobjective linear programming problems *International Journal of Operational Research*, **15(1)**, 1–15. 2012.
- [14] Z. Süle, B. Bertók, F. Friedler, L.T. Fan, Optimal Design of Supply Chains by P-graph Framework Under Uncertainties, *Chemical Engineering Transactions*, **25**, 453–458, 2011.
- [15] Tick, J.: Fuzzy Extension to P-Graph Based Workflow Models , *Proceedings of the 7th IEEE International Conference on Computational Cybernetics, ICC3 2009*, 109–112, 2009.
- [16] J. Tick, Z. Kovács, Cs. Imreh, Business Process Modeling and the robust PNS problem, *Acta Polytechnica Hungarica*, **10(6)**, 193–204 2013.

6. FEJEZET

Nevezetes feladatok megoldása PNS-el

A folyamatszintézis eszközeivel megoldhatóak azok a feladatok is, melyek elemi gráf algoritmusokkal. A gráf leképezhető folyamatgráfként, a keresés célja pedig optimalizálási célként.

A gráf minden esetben hasonló módon írható át a folyamatszintézis nyelvére. A csúcsokhoz anyagokat, az irányított élekhez pedig műveleti egységeket rendelünk. A csúcsok egy $\{1, 2, 3, 4, 5, 6\}$ halmazát például az $\mathcal{M} = \{m_1, m_2, m_3, m_4, m_5, m_6\}$ anyaghalmaz jelöli, míg az irányított élek egy $\{(1, 2), (1, 3)(2, 3)\}$ halmazát az

$$\mathcal{O} = \{O_{12} = (\{m_1\}, \{m_2\}), O_{13} = (\{m_1\}, \{m_3\}), O_{23} = (\{m_2\}, \{m_3\})\}$$

műveleti egység halmaz. Irányítatlan gráf esetén az élek mindkét lehetséges irányát külön-külön rögzíteni kell. Tehát irányítatlan élek egy $\{\{1, 2\}, \{1, 3\}\}$ halmazát az

$$\mathcal{O} = \{(\{m_1\}, \{m_2\}), (\{m_2\}, \{m_1\}), (\{m_1\}, \{m_3\}), (\{m_3\}, \{m_1\})\}$$

műveleti egység halmaz adja meg. Az algoritmusok célját az anyagokra és műveleti egységekre vonatkozó korlátokkal és költség paraméterekkel tudjuk kijelölni.

6.1. Minimális feszítőfa szintézise

Minimális feszítőfa egy olyan összefüggő gráf, mely minden csúcsot tartalmaz, és az élek súlyának összege minimális. A gráf összefüggőségét biztosítja, ha olyan folyamat szintetizálunk, melyben egyik csúcsot forrásként azaz nyersanyagként, minden más csúcsot pedig nyelőként azaz kötelező termékként definiálunk. A termékekre szükséges pozitív mennyiségi alsó korlát beállítása, hogy a hozzá vezető műveletek legalább egyikének mérete ne lehessen nulla. Ezután az élek súlyát, mint a leíró műveleti egységek fix költségét adjuk meg. Kruskal algoritmus bemutatásánál szereplő feladat folyamatszintézis átírása a 6.1-es és a 6.2-es táblázatokban látható.

A feszítőfát azok az élek adják, melyekhez tartozó műveleti egységek szerepelnek a szintézis feladat optimális megoldásában. Ilyen megoldást ad például az ABB algoritmus szoftver megvalósítása.

6.2. Legrövidebb út szintézise

A legrövidebb út hasonlóan határozható meg mint a feszítőfa, azzal a különbséggel, hogy a legrövidebb úthoz csak a kiindulási és a cél csúcsot összekötő gráfot kell meghatározni, melyben az élek súlyának összege minimális. Ennek megfelelően, a folyamatszintézis feladatban a kiindulási csúcsot adjuk meg nyersanyagként, a cél csúcsot pedig kötelező termékként valamely pozitív előállítandó mennyiséggel. Az élek súlyát ismét a műveleti egységek fix költségként definiáljuk. A Dijkstra algoritmusnál bemutatott példára a folyamatszintézis feladat a 6.3-as és a 6.4-es táblázatokban látható.

Név	Típus	Alsó korlát	Felső korlát
<i>m1</i>	nyersanyag		5
<i>m2</i>	termék	1	
<i>m3</i>	termék	1	
<i>m4</i>	termék	1	
<i>m5</i>	termék	1	
<i>m6</i>	termék	1	

6.1. táblázat. Anyagok

Név	Bemenő anyag	Kimenő anyag	Fix költség
<i>O12</i>	<i>m1</i>	<i>m2</i>	2
<i>O21</i>	<i>m2</i>	<i>m1</i>	2
<i>O13</i>	<i>m1</i>	<i>m3</i>	3
<i>O31</i>	<i>m3</i>	<i>m1</i>	3
<i>O23</i>	<i>m2</i>	<i>m3</i>	1
<i>O32</i>	<i>m3</i>	<i>m2</i>	1
<i>O24</i>	<i>m2</i>	<i>m4</i>	4
<i>O42</i>	<i>m4</i>	<i>m2</i>	4
<i>O25</i>	<i>m2</i>	<i>m5</i>	3
<i>O52</i>	<i>m5</i>	<i>m2</i>	3
<i>O35</i>	<i>m3</i>	<i>m5</i>	2
<i>O53</i>	<i>m5</i>	<i>m3</i>	2
<i>O45</i>	<i>m4</i>	<i>m5</i>	3
<i>O54</i>	<i>m5</i>	<i>m4</i>	3
<i>O46</i>	<i>m4</i>	<i>m6</i>	6
<i>O64</i>	<i>m6</i>	<i>m4</i>	6
<i>O56</i>	<i>m5</i>	<i>m5</i>	1
<i>O65</i>	<i>m6</i>	<i>m5</i>	1

6.2. táblázat. Műveleti egységek

Név	Típus	Alsó korlát	Felső korlát
<i>m1</i>	nyersanyag		5
<i>m2</i>	köztes anyag		
<i>m3</i>	köztes anyag		
<i>m4</i>	köztes anyag		
<i>m5</i>	köztes anyag		
<i>m6</i>	termék	1	

6.3. táblázat. Anyagok

Név	Bemenő anyag	Kimenő anyag	Fix költség
<i>O12</i>	<i>m1</i>	<i>m2</i>	1
<i>O21</i>	<i>m2</i>	<i>m1</i>	1
<i>O13</i>	<i>m1</i>	<i>m3</i>	3
<i>O31</i>	<i>m3</i>	<i>m1</i>	3
<i>O23</i>	<i>m2</i>	<i>m3</i>	1
<i>O32</i>	<i>m3</i>	<i>m2</i>	1
<i>O24</i>	<i>m2</i>	<i>m4</i>	4
<i>O42</i>	<i>m4</i>	<i>m2</i>	4
<i>O25</i>	<i>m2</i>	<i>m5</i>	3
<i>O52</i>	<i>m5</i>	<i>m2</i>	3
<i>O35</i>	<i>m3</i>	<i>m5</i>	2
<i>O53</i>	<i>m5</i>	<i>m3</i>	2
<i>O45</i>	<i>m4</i>	<i>m5</i>	3
<i>O54</i>	<i>m5</i>	<i>m4</i>	3
<i>O46</i>	<i>m4</i>	<i>m6</i>	6
<i>O64</i>	<i>m6</i>	<i>m4</i>	6
<i>O56</i>	<i>m5</i>	<i>m5</i>	1
<i>O65</i>	<i>m6</i>	<i>m5</i>	1

6.4. táblázat. Műveleti egységek

A legrövidebb utat azok az élek adják, melyekhez tartozó műveleti egységek szerepelnek a szintézis feladat optimális megoldásában. Ilyen megoldást ad például az ABB algoritmus szoftver megvalósítása.

6.3. Maximális folyam szintézise

A maximális folyamoknál a gráf éleihez rendelt tulajdonság nem a költség, hanem a kapacitás. Ezen kapacitás mellett kell a lehető legnagyobb mennyiséget egy forrásból a nyelőbe juttatni. A gráf leírása hasonló az előző két alfejezetben tárgyaltakhoz, de az éleket reprezentáló műveleti egységeknek itt nem költsége lesz, hanem maximális kapacitása, mely megegyezik az megfelelő él kapacitásával. A maximális folyamot az motiválja, hogy a nyelőn megjelenő anyagmennyiséget egy olyan termékkel írjuk le, aminek van egy pozitív ára, tehát a minél nagyobb termelése, egyre nagyobb profitot hoz. Annak érdekében, hogy az éleken ne folyjon olyan anyagmennyiség, melynek nincs szerepe a maximális folyamban, mert csak közbülső csúcshoz vezet, korlátozzuk nullára a közbülső anyagpontokon megmaradó anyagmennyiséget az anyag felső korlátjával. A Ford-Fulkerson algoritmusnál megismert példát megadó szintézis feladat leírása a 6.5-ös és a 6.6-os táblázatokban látható.

Név	Típus	Alsó korlát	Felső korlát	Ár
<i>m1</i>	nyersanyag		∞	0
<i>m2</i>	köztes anyag		0	
<i>m3</i>	köztes anyag		0	
<i>m4</i>	köztes anyag		0	
<i>m5</i>	köztes anyag		0	
<i>m6</i>	termék		∞	1

6.5. táblázat. Anyagok

A maximális folyamot a szintézis feladat optimális megoldásában szereplő műveleti egységek által reprezentált élek adják, felhasznált kapacitásuk pedig megegyezik a megfelelő műveleti egységek optimális kapacitásával. Ilyen megoldást ad például az ABB algoritmus szoftver megvalósítása.

Név	Bemenő anyag	Kimenő anyag	Kapacitás felső korlát
<i>O12</i>	<i>m1</i>	<i>m2</i>	1
<i>O13</i>	<i>m1</i>	<i>m3</i>	3
<i>O32</i>	<i>m3</i>	<i>m2</i>	1
<i>O24</i>	<i>m2</i>	<i>m4</i>	4
<i>O25</i>	<i>m2</i>	<i>m5</i>	3
<i>O35</i>	<i>m3</i>	<i>m5</i>	2
<i>O45</i>	<i>m4</i>	<i>m5</i>	3
<i>O46</i>	<i>m4</i>	<i>m6</i>	6
<i>O56</i>	<i>m5</i>	<i>m5</i>	1

6.6. táblázat. Műveleti egységek

7. FEJEZET

Integrált szintézis feladatok

A fejezetben a folyamatszintézis feladatot nem önmagában akarjuk megoldani, hanem más önmagában is nehéz feladatokkal kombinálva. Két integrációt mutatunk be, a hőintegrációt és az ütemezést.

Hőintegráció esetében a megtervezett hálózat költségei nem kizárólag a megvásárolt műveleti egységek méretétől és a felhasznált nyersanyagok mennyiségétől függenek. A hálózatban vannak olyan anyagáramok, amelyeket hűteni vagy fűteni kell, amelynek befektetési és működtetési költsége is van, így az egész rendszer költségére hatással van.

A P-gráfban eddig folytonos műveleti egységekkel foglalkoztunk, ahol az anyag folyamatosan áramlott a műveleti egységek között. A gyakorlatban viszont általánosan előfordulnak szakaszos működésű műveleti egységek is, amelyek működésének elején felhasználja az összes bemeneti anyagát és a működésének végén keletkeznek a kimeneti anyagok. Ilyen feladatoknál egy műveleti egységet a hálózat több pontján is fel lehet használni időben természetesen eltolva. A műveleti egységek feladatokhoz való rendelését és időzítését nevezzük ütemezésnek.

7.1. Folyamat- és hőcserélő hálózat szintézise

A gyártási folyamatok általában tartalmazznak olyan anyagáramokat, melyek hőmérsékletét változtatni kell. Az anyagáramok hőmérsékletének változtatása energia befektetéssel jár. Gazdasági és környezetvédelmi szempontból fontos, hogy olyan folyamatokat szintetizáljunk, amelyekben az eddig használt befektetési költség mellett az energia fogyasztást (és az ahhoz kapcsolódó befektetési költségeket) is figyelembe vesszük. Hőcserélő hálózatokban lehetőség van a rendszer hőmérséklet változtatást igénylő anyagáramának hőcserélő berendezésben való összekapcsolásával a termelés költségeinek csökkentésére.

A PNS és a hőintegrációs feladat megoldható egymás után. Így például először a PNS feladat optimális megoldását keressük meg, majd az így kapott megoldásra leszűkítve keressük a hőintegrációs feladat optimális megoldását, vagy fordított sorrendben. Mivel az egyik feladat megoldása befolyásolja a másik feladat megoldását, ezért a szekvenciális megközelítés nem vezethet mindkét feladat szempontjából optimális megoldáshoz. Például egy optimális PNS megoldás esetén könnyen előfordulhat, hogy az nagyon kevés lehetőséget biztosít a hőintegráció számára, így magas lesz a gyártás energiaköltsége. Ugyanakkor ha az optimumhoz képest kicsivel rosszabb befektetési költségeket is megengedünk, akkor ezeknél a megoldásoknál megnövekedhet a hőintegráció lehetősége és így csökken a szükséges energia mennyisége.

A hőintegráció az iparban hőcserélő berendezések segítségével valósítható meg. A hőcserélő berendezések a gyakorlatban igen drágák, a tisztításuk nehéz és költséges, esetleg lehetetlen.

7.1.1. Hőcserélő hálózat szintézise

A hőcserélő hálózat szintézisének feladatáról részletesen már írtunk az 1.4-es fejezetben. Egy általános HENS problémát a bemenetek segítségével definiálhatunk, ahol a bemenetek a következők:

- A melegáramok halmaza (F^H);
- A hidegáramok halmaza (F^C);
- A anyagáramok nagysága és hőkapacitása;
- A felhasználható hőforrások (U^H) és hőnyelők (U^C) és ezek hőmérséklete és költsége;
- A hőcserélők költsége.

7.1.2. Integrált folyamat- és hőcserélő hálózat szintézise

Az integrált szintézis során a teljes folyamathálózat és hőcserélő hálózat meghatározása azonos időben történik. Szekvenciális módszerek esetében először meghatározzák magát az optimális folyamathálózatot és utána határozzák meg az optimális hőcserélő hálózatot. Könnyen belátható, hogy ez utóbbi nem vezet optimális megoldáshoz, hiszen az optimális folyamathálózat meghatározásakor figyelmen kívül hagyjuk a hőcserével kapcsolatos információkat, költségeket.

A PNS feladatot vesszük alapul a feladat részletes ismertetéséhez. A PNS feladatot kibővítjük új paraméterekkel, amelyek szükségesek a HENS feladathoz.

- Anyagáramhoz kapcsolódó hőáram

A kiterjesztett feladatban az anyagáramokhoz tartozhat hőmérséklet is. Ha egy műveleti egység kimenő anyagárama közvetlenül kapcsolódik egy másik műveleti egység bemenő anyagáramára és a definiált hőmérsékletek nem egyeznek meg, akkor hőbevitel válik szükségessé. Az itt fellépő hő mennyisége az anyagárammal arányos.

- Rejtett hő

Rejtett hőnek nevezzük azokat a hőigényeket, amikor műveleti egységgel úgy valósítunk meg hőforgalmat, hogy a műveleti egység hőmérséklete közben állandónak kell maradnia. Azaz a műveleti egységből egy adott mennyiségű hőt kell elvonni vagy adott mennyiségű hőt kell vele közölni, de közben a műveleti egység hőmérséklete nem változik. Ilyen lehet például a halmazállapot változását vagy a kémiai átalakulást kísérő entalpiaváltozás.

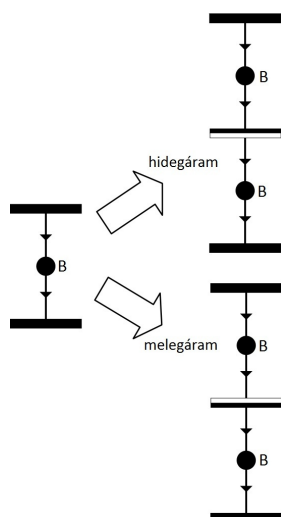
Egyes műveleti egység típusoknál a művelet során betáplált vagy elvont rejtett hő mennyisége többszöröse lehet annak a hőforgalomnak, amit a műveleti egységbe belépő és kilépő anyagáramok képviselnek. Erre jó példa egy erősen exoterm vagy endoterm folyamatot végrehajtó berendezés.

Mindezek miatt a folyamatszintézis hőintegrációval való kiterjesztésekor a rejtett hőt figyelembe kell venni és a matematikai modellbe be kell építése. A rejtett hőt (típusától függetlenül) úgy definiáljuk a modellben, hogy minden egyes esetben megadjuk a hőmérsékletét és azt a hőmennyiséget is, ami a rejtett hőforrásához tartozik a műveleti egység egységnyi kapacitása mellett. Egy műveleti egységhez több rejtett hő is tartozhat. Ilyenkor egy nagyobb méretű műveleti egység egyes pontjait fűteni, másokat pedig hűteni kell.

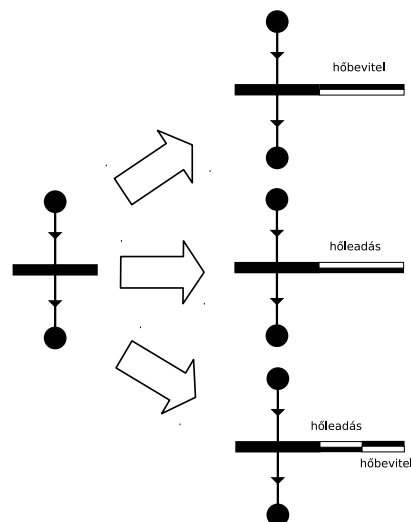
7.1.3. hP-gráf

A korábban bevezetett P-gráf reprezentációt ki kell bővíteni ahhoz, hogy reprezentálni tudja a lehetséges hőbevitelket és hőelvonásokat. Ahogy az előző fejezetben leírtuk az anyagáramokhoz tartozó hőáramot valamint a műveleti egységekhez tartozó rejtett hőt is reprezentálni kell.

Kezdjük a rejtett hő reprezentálásával. Egy hP-gráfban, ha egy műveleti egységhez rejtett hő tartozik, akkor az adott műveleti egységet jelölő vízszintes téglalapot meghosszabbítjuk. A meghosszabbított rész nem egyszínű fekete lesz, hanem az alsó vagy a felső fele fehér, attól függően, hogy hőbevitel vagy hőelvonás van. A 7.1-es ábrán a műveleti egység három különböző módon lett kiterjesztve.



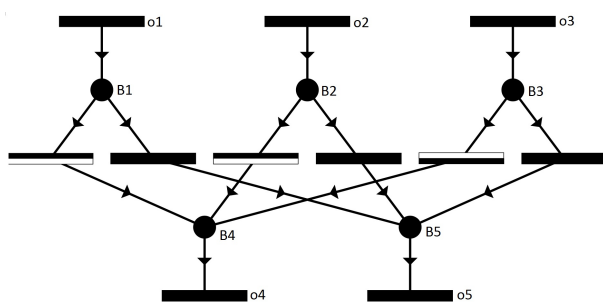
7.1. ábra. Műveleti egység rejtett hőjének ábrázolása



7.2. ábra. Hideg és meleg anyagáram ábrázolása

Az anyagáramok hőáramának ábrázolását hP-gráfban úgy oldottuk meg, hogy a különböző hőmérsékletű anyagokat különböző anyagcsomópontoknak tekintjük. Az azonos, de különböző hőmérséklettel rendelkező anyagcsomópontok közé pedig egy hőmérséklet változást (vagy anyagáramot) reprezentáló műveleti egységet helyezünk el. Attól függően, hogy hideg- vagy melegáramról van szó (fűteni vagy hűteni kell) a műveleti egységet reprezentáló téglalap alsó vagy felső fele fehér (7.2-es ábra).

Gyakran előfordul, hogy egy anyagot nem csak egy műveleti egység gyárt, illetve nem csak egy műveleti egység használ fel. Az anyag hőmérséklete minden műveleti egységnél különböző lehet. Ha az eredeti P-gráf leírást használnánk, akkor az azt jelentené, hogy a műveleti egységek kimenetén lévő anyagok összekeverésre kerülnének és ezek után a hőmérsékleti paraméterek torzulnának. Ennek elkerülésére a hP-gráfban minden műveleti egységhez egy külön anyagcsomópont tartozik. Az anyagáramokat ezek után műveleti egységekkel reprezentálni tudjuk. Az anyagcsomópontokat vagy hőmérséklet változást reprezentáló műveleti egység vagy fiktív műveleti egység kapcsolja össze, attól függően, hogy az anyagáramnál szüksége van-e hőcserére.



7.3. ábra. B anyagcsomópont kiterjesztése

A 7.3 ábrán látható példa esetében a B anyagot három műveleti egység ($o1$, $o2$ és $o3$) képes előállítani és két műveleti egység ($o4$ és $o5$) fogyaszthatja. A kiterjesztés után mindegyik műveleti egységhez tartozik egy B anyag, amelyek nevei alapján rendre $B1$, $B2$, ..., $B5$. Az $o1$ és az $o2$ műveleti egység által előállított B anyag hidegebb, mint amit az $o4$ műveleti egység bemenetén elvárunk, ezért $B1$ - $B4$ és $B2$ - $B4$ hidegáramnak minősül. Az $o3$ műve-

leti egység melegebb anyagot gyárt, mint amire $o4$ -nek szüksége van, ezért $B3$ - $B4$ melegáram. Az $o5$ műveleti egységnél nem specifikált a bemeneti anyag hőmérséklete, ezért a $B5$ anyag hőmérséklete tetszőleges lehet, így nincs szükség hőcserére. Ettől függetlenül a könnyebb kezelhetőség érdekében ezekre az anyagáramokra is felvesszünk fiktív műveleti egységeket.

7.1.4. Hőmérsékleti adatok, halmazok

Első lépésben vezessünk be jelöléseket a hőmérsékletekkel kapcsolatos adatok kezeléséhez. Hőmérséklet tartozhat az anyagokhoz és a műveleti egységekhez is.

- Anyagok hőmérséklete

Jelölje az o_j műveleti egység m_i bemeneti anyagához tartozó hőmérsékletet t_{ij}^{in} , a kimenetihez tartozót pedig t_{ij}^{out} . Azért kell a ki- és a bemeneti anyaghoz külön hőmérsékletet megadni, mert egy műveleti egységnek egy anyag lehet egyszerre bemenete és kimenete is.

- Rejtett hő

A rejtett hő megadása bonyolultabb, mivel egy műveleti egységhez több rejtett hő is tartozhat. Jelölje az o_j műveleti egységhez tartozó rejtett hőforrások számát l_j . Jelölje az egységnyi méretű o_j műveleti egység i , a rejtett hőforráshoz időegység alatt betáplálendő vagy elvonandó hő mennyiségét h_{ji} , ahol h_{ji} pozitív, ha hőt kell elvonni és negatív, ha hőt kell közölni. Továbbá jelölje TM_{ji} a rejtett hő hőmérsékletét.

Ezen paraméterek alapján már meg lehet határozni, a hideg- és melegáramok valamint a hideg és meleg hőforrások halmazát.

- Hőáramok

Egy anyagáram megadásához három paraméter szükséges, melyik anyag melyik műveleti egységből melyik műveleti egységbe kerül át. A hőáramok pedig azok az anyagáramok, amelyeknél mind a két műveleti egységhez adottak a hőmérséklet paraméterek. Ezen belül pedig az a hidegáram, amelyet fűteni kell és az a meleg, amelyet hűteni kell. Ezek után a hideg- és melegáramok halmazát definiálni lehet.

$$\begin{aligned} F^H &= \{(i, j, k) : t_{ij}^{out} > t_{ik}^{in}, m_i \in \mathcal{M}, o_j \in \varphi^-(m_i), o_k \in \varphi^+(m_i)\} = \\ &= \{FH_1, FH_2, \dots, FH_{n_{FH}}\} \\ F^C &= \{(i, j, k) : t_{ij}^{out} < t_{ik}^{in}, m_i \in \mathcal{M}, o_j \in \varphi^-(i), o_k \in \varphi^+(m_i)\} = \\ &= \{FC_1, FC_2, \dots, FC_{n_{FC}}\} \end{aligned}$$

Az egyszerűbb felírás kedvéért a t_{ij}^{out} és a t_{ik}^{in} hőmérsékleteket az (i, j, k) hármas segítségével is lehet paraméterezni. Azaz a továbbiakban jelölje $t_0^{(i,j,k)}$ és $t_1^{(i,j,k)}$ a két hőmérsékletet úgy, hogy mindig $t_0^{(i,j,k)}$ jelentse a kisebb hőmérsékletet.

- Rejtett hőforrások

Egy rejtett hőforrás megadásához kettő paraméter elég, melyik műveleti egység hányadik hőforrásáról van szó. Meleg hőforrásnak azt tekintjük, ahol hőt kell elvonni, hidegnek pedig azt, amikor hőt kell közölni a műveleti egységgel. Ezek alapján a hideg és meleg hőforrások halmazai már definiálhatók.

$$\begin{aligned} L^H &= \{(j, i) : h_{ji} > 0, o_j \in \mathcal{O}, i \in \{1, \dots, l_j\}\} = \{LH_1, LH_2, \dots, LH_{n_{LH}}\} \\ L^C &= \{(j, i) : h_{ji} < 0, o_j \in \mathcal{O}, i \in \{1, \dots, l_j\}\} = \{LC_1, LC_2, \dots, LC_{n_{LC}}\} \end{aligned}$$

Itt is definiáljuk az (j, i) párral indexelhető hőmérsékletet, a rejtett hőforrás hőmérsékletét, ami legyen $t^{(j,i)}$.

Mint már korábban említettük nem lehet akármilyen hőmérsékletű hideg és meleg hőáramot vagy hőforrást párosítani. Csak akkor lehet közöttük hőcsere végrehajtani közöttük, ha meg van a minimális MT hőmérséklet különbség. Azért, hogy ezzel a modell felírását ne bonyolítsuk feltesszük, hogy az összes hidegáram és hideg hőforrás értéke ezzel a MT értékkel meg van növelve. Ezek után mondhatjuk, hogy két hőáram illetve hőforrás akkor cserélhet hőt, ha van közös tartománya a hőintervallumuknak. Nem kell, hogy a teljes intervallum egyforma legyen, mivel egy hőáram illetve hőforrás többször is cserélhet hőt, több hőcsereelő berendezésen is keresztül mehet. Ahhoz, hogy az ilyen jellegű hőcsereket modellezni tudjuk le kell tudnunk írni, hogy melyik hőcsere melyik hőmérséklet intervallumban megy végbe.

Első lépésben meg lehet határozni az úgynevezett elemi hőmérsékleti intervallumokat, amelyek a legszűkebb hőmérsékleti intervallumok, amelyeken hőcsere létrejöhet. Ezt úgy tehetjük meg, hogy az összes hőáramhoz és rejtett hőforráshoz tartozó hőmérsékleteket sorba rendezzük úgy, hogy egy érték sem ismétlődhet benne. Így kapunk egy $t_1, t_2, \dots, t_{n_e+1}$ sorozatot, ahol teljesül az, hogy ha $i < j$, akkor $t_i < t_j$. Ezek alapján definiálni tudjuk az elemi hőmérsékleti intervallumokat, ahol $E_p = [t_p, t_{p+1}]$, $p \in \{1, \dots, n_e\}$ egy elemi intervallum.

Az elemi hőmérsékleti intervallumok után fel tudjuk bontani a hőáramokat elemi hőáramokká, amelyek már közvetlenül tudnak egymással hőt cserélni a teljes intervallumon. A hideg és a meleg elemi hőáramok halmazát is definiálhatjuk.

$$\begin{aligned} E^H &= \left\{ (i, j, k, p) : (i, j, k) \in F^H, t_p \geq t_0^{(i,j,k)}, t_{p+1} \leq t_1^{(i,j,k)}, p \in \{1, \dots, n_e\} \right\} = \\ &= \{FSH_1, FSH_2, \dots, FSH_{n_{FSH}}\} \\ E^C &= \left\{ (i, j, k, p) : (i, j, k) \in F^C, t_p \geq t_0^{(i,j,k)}, t_{p+1} \leq t_1^{(i,j,k)}, p \in \{1, \dots, n_e\} \right\} = \\ &= \{FSC_1, FSC_2, \dots, FSC_{n_{FSC}}\} \end{aligned}$$

A hőcserék viszont a gyakorlatban nem feltétlenül elemi hőáramok között jönnek létre. Ha például két hőáram intervalluma teljesen lefedi egymást, akkor ezzel a módszerrel ezt az intervallumot szétbontottuk több részre és egy hőcsere helyett több keletkezett. Mivel a hőcsere költségek nem lineárisak, ezért ez a megoldás nem elfogadható, így bevezetjük a részhőáramok fogalmát, amely részhőáramok szomszédos elemi hőáramok összevonásából keletkeznek.

Tegyük fel, hogy az (i, j, k) hőáram hőmérséklet intervallumát $d - 1$ elemi hőmérsékleti intervallumra lehet felbontani, azaz $[t_0^{(i,j,k)}, t_1^{(i,j,k)}] = [t_p, t_{p+d}]$. Ebben az esetben $\binom{d+1}{2}$ részintervallumot lehet generálni. Bármely intervallum $I_{qs} = [t_q, t_{s+1}]$ formában felírható, ahol $p \leq q \leq s \leq p + d - 1$. Ez alapján fel lehet írni a hideg és meleg részhőáramokat.

$$\begin{aligned} I^H &= \left\{ (i, j, k, q, s) : (i, j, k) \in F^H, I_{qs} \subseteq [t_0^{(i,j,k)}, t_1^{(i,j,k)}] \right\} = \{SSH_1, SSH_2, \dots, SSH_{n_{SSH}}\} \\ I^C &= \left\{ (i, j, k, q, s) : (i, j, k) \in F^C, I_{qs} \subseteq [t_0^{(i,j,k)}, t_1^{(i,j,k)}] \right\} = \{SSC_1, SSC_2, \dots, SSC_{n_{SSC}}\} \end{aligned}$$

A hideg és meleg részhőáramok lehetséges összekapcsolásait is halmazokkal definiáljuk. A megadásnál elegendő csak az egyik irányba definiálni a lehetséges kapcsolatokat, mivel az szimmetrikus. Mi a meleg részhőáramokhoz és hőforrásokhoz párosítható hideg részhőáramokat és hőforrásokat adjuk meg.

$$\begin{aligned} JFF(SSH_l) &= \{SSC_{l'} = (i', j', k', q', s') \in I^C : q \leq q', s \leq s'\} \\ SSH_l &= (i, j, k, q, s) \in I^H \\ JFL(SSH_l) &= \{LC_{l'} = (j', i') \in L^C : t_q \leq T_{i'j'}\} \\ SSH_l &= (i, j, k, q, s) \in I^H \\ JLF(LH_l) &= \{SSC_{l'} = (i', j', k', q', s') \in I^C : T_{ji} \leq t_{s'+1}\} \\ LH_l &= (j, i) \in L^H \\ JLL(LH_l) &= \{LC_{l'} = (j', i') \in L^C : T_{ij} \leq T_{i'j'}\} \\ LH_l &= (j, i) \in L^H \end{aligned}$$

A hőáramok nem csak másik hőárammal és rejtett hőforrással cserélhetnek hőt, hanem külső hőforrással is. A következőkben minden elemi hőáramhoz és az összes rejtett hőforráshoz definiáljuk, hogy mely külső hőforrással párosíthatóak.

$$\begin{aligned} JFU(FSC_l) &= \{u \in U^H : UT_u \geq t_{p+1}\} & FSC_l &= (i, j, k, p) \in E^C \\ JFU(FSH_l) &= \{u \in U^C : UT_u \leq t_p\} & FSC_l &= (i, j, k, p) \in E^H \\ JLU(LC_l) &= \{u \in U^H : UT_u \leq T_{ji}\} & LC_l &= (j, i) \in L^C \end{aligned}$$

$$\text{JLU}(LH_l) = \{u \in \mathcal{U}^C : UT_u \geq T_{ji}\}$$

$$LH_l = (j, i) \in L^H$$

Lehetnek a rendszerben olyan hőáramok, amelyek technológiai okok miatt nem párosíthatók egymással. Ezeket kivesszük a megfelelő halmazokból.

7.1.5. Matematikai modell

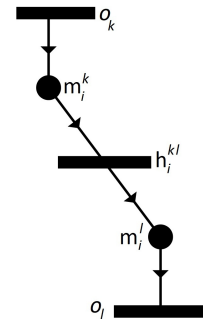
A PNS lineáris modelljét már korábban ismertettük a gyorsított szétválasztás és korlátozás algoritmus leírásánál (3.1)-(3.3). Az ott ismertett MILP modell feltételei továbbra is érvényesek, hiszen a nyersanyagkorlátok, az előállítandó termékmennyiségek nem változnak és az anyagmegmaradás elve továbbra is érvényben marad. A jelen fejezetben a HENS által létrejövő új feltételeket és a célfüggvényre gyakorolt hatását mutatjuk be.

Az eredeti P-gráfot kiterjesztettük hP-gráffá, ami azt jelenti, hogy az eredetihez képest több anyagcsomópontunk és műveleti egységünk van. Jelölje $T (\subseteq \mathcal{M})$ azon anyagokat, amelyekhez adottak hőmérsékletek. Legyen $m_i \in T$ és legyen $o_k \in \varphi^-(m_i)$, valamint $o_l \in \varphi^+(m_i)$, ekkor vezessük be a következő jelöléseket, amelyeket a 7.4-es ábrán összefoglaltunk.

- Jelölje m_i^k és m_i^l az o_k és az o_l műveleti egységhez tartozó új „mesterséges anyagpontokat”.
- Jelölje t_{ik}^{out} az m_i^k és t_{il}^{in} az m_i^l anyag hőmérsékletét.
- Jelölje h_i^{kl} az o_k műveleti egységből o_l műveleti egységbe áramló m_i anyaghoz tartozó mesterséges műveleti egységet.
- Jelölje a h_i^{kl} műveleti egységen átáramló anyag mennyiségét w_i^{kl} .

A modell nem változik azoknál az anyagcsomópontoknál, amelyekhez nem tartozik hőmérséklet, azaz itt ugyanazok maradnak a feltételek. A T halmazban lévő anyagcsomópontokhoz tartozó feltételeket ki kell cserélni. Jelölje az m_i anyaghoz újonnan bevezetett anyagok halmazát M_i , és a műveleti egységek halmazát pedig K_i . A bevezetett műveleti egységek egy bemenettel és egy kimenettel rendelkeznek és annyi anyagot bocsátanak ki, amennyi a kimenetükön megjelenik, azaz minden $o_k \in \bigcup_{i \in T} K_i$ és $j \in \psi(o_k)$ esetén $ir_{kj} = 1$ és $or_{kj} = 1$.

Ezek után mondhatjuk, hogy a hőmérsékleti adatokkal nem rendelkező anyagokhoz tartozó feltételek nem változnak, a hőmérsékleti adatokkal rendelkező eredeti anyagokhoz nem tartozik feltétel, és az új anyagcsomópontokhoz szintén tartoznak az eredetihez hasonló feltételek, ahol φ'^- operátor a hP-gráf egy anyagcsomópontjához megadja az őt előállító műveleti egységek halmazát, a φ'^+ operátor pedig az őt felhasználó műveleti egységek halmazát. Azaz értelmezve van az új anyagcsomópontokra és a visszaadott halmazban mesterséges műveleti egységek is lehetnek.



7.4. ábra. Mesterséges műveleti egység

$$l_i \leq \sum_{o_k \in \varphi^-(m_i)} x_k or_{ki} - \sum_{o_k \in \varphi^+(m_i)} x_k ir_{ki} \leq u_i$$

$$m_i \in \mathcal{M} \setminus T$$

$$l_j \leq \sum_{o_k \in \varphi'^-(m_j)} x_k or_{kj} - \sum_{o_k \in \varphi'^+(m_j)} x_k ir_{kj} \leq u_j$$

$$m_j \in M_i, m_i \in T$$

A hőegyensúlyi feltételek definiálásához meg kell határozni az elvonandó valamint a betáplálendő hő mennyiségét. Meleg áramoknál és hőforrásoknál ez az érték pozitív, hideg áramoknál és hőforrásoknál pedig negatív.

- Egy hőáramnál ez a hőmennyiség az áramló anyag mennyiségének és a hőátadási tényezőnek a szorzata. (Az anyagmennyiséget a kapcsolódó műveleti egységek kapacitásából és a

műveleti egységekhez tartozó ki- és bemeneti anyagarányokból lehet meghatározni.)

$$\begin{aligned} QFH_l &= c_i w_i^{jk} (t_{p+1} - t_p) & FSH_l &= (i, j, k, p) \in E^H \\ QFC_l &= c_i w_i^{jk} (t_{p+1} - t_p) & FSC_l &= (i, j, k, p) \in E^C \end{aligned}$$

- Rejtett hőforrásnál ez a műveleti egység méretének és az időegység alatt betáplálendő vagy elvonandó hő mennyiségének a szorzata.

$$\begin{aligned} QLH_l &= h_{ji} x_j & LH_l &= (j, i) \in L^H \\ QLC_l &= h_{ji} x_j & LC_l &= (j, i) \in L^C \end{aligned}$$

Vezessünk be változókat a hőcserélésekhez. A változók mindig az átvitt hőmennyiséget jelölik. Mivel ugyanannyi hőmennyiség megy át a melegből a hidegbe, mint a hidegből a melegbe, ezért elegendő csak az egyik irányba definiálni a változókat. A változók első indexe mindig a meleg hőáramot vagy hőforrást, a második indexe pedig a hideg hőáramot vagy hőforrást jelöli, így a változók értéke nem lehet negatív.

$$\begin{aligned} QFF_{ij} : & \quad SSH_i \in I^H, SSC_j \in JFF(SSH_i) \\ QFL_{ij} : & \quad SSH_i \in I^H, LC_j \in JFL(SSH_i) \\ QLF_{ij} : & \quad LH_i \in L^H, SSC_j \in JLF(LH_i) \\ QLL_{ij} : & \quad LH_i \in L^H, LC_j \in JLL(LH_i) \end{aligned}$$

A külső hőforrásokat az elemi hőáramokhoz és a rejtett hőforrásokhoz rendeljük. Ebben az esetben a hideg és meleg hőáramokhoz valamint a hideg és meleg rejtett hőforrásokhoz is veszünk fel külön változókat. Itt az indexek sorrendje a hőátadás irányát jelzi, azaz mindig a meleg van elől, és a változók ebben az esetben is nem-negatívak.

$$\begin{aligned} QFU_{iu} : & \quad FSH_i \in E^H, u \in JFU(FSH_i) \\ QUF_{ui} : & \quad FSC_i \in E^C, u \in JFU(FSC_i) \\ QLU_{iu} : & \quad LH_i \in L^H, u \in JLU(LH_i) \\ QUL_{ui} : & \quad LC_i \in L^C, u \in JLU(LC_i) \end{aligned}$$

A hőegyensúlyi feltételek adják meg, hogy minden hőigényt pontosan ki kell elégíteni.

- Minden (LH_l) meleg rejtett hő esetén az elvonandó hőmennyiség meg kell, hogy egyezzen a hideg hőáramok, a hideg rejtett hőforrások és a külső hőforrások felé közölt hőmennyiségek összegével.

$$QLH_l = \sum_{SSC_{l'} \in JLF(LH_l)} QFL_{l'l} + \sum_{LC_{l'} \in JLL(LH_l)} QLL_{l'l} + \sum_{u \in JLU(LH_l)} QLU_{lu} LH_l \in L^H$$

- Minden (LC_l) hideg rejtett hő esetén a felvett hőmennyiség meg kell, hogy egyezzen a hideg hőáramoktól, a hideg rejtett hőforrásoktól és a külső hőforrásoktól felvett hőmennyiségek összegével. Mivel a változók a hőátadást jelölik és mindig pozitívak, és a hőfelvételt jelölő QLC_l érték negatív, ezért egy „-” előjelre van szükség.

$$-QLC_l = \sum_{LC_{l'} \in JFL(SSH_{l'})} QFL_{l'l} + \sum_{LC_{l'} \in JLL(LH_{l'})} QLL_{l'l} + \sum_{u \in JLU(LC_l)} QUL_{ul} LC_l \in L^C$$

- Minden elemi $(FSH_l = (i, j, k, p))$ meleg hőáram esetén figyelembe kell venni az összes olyan meleg rész hőáramot $(SSH_j = (i, j, k, q, s) \in I^H)$, amely tartalmazza őt, és az ezen rész hőáramok által a hideg rész hőáramok és hideg rejtett hőforrások felé közölt hőmennyiséget kell összeadni. (A képletben szereplő $\frac{t_{p+1}-t_p}{t_{q+1}-t_s}$ szorzótényező azt határozza meg, hogy az elemi

hőáram intervallumának szélessége hányad része az aktuális rész hőáram intervallumának szélességével.) Valamint ehhez hozzá kell adni az elemi meleg hőáram által a külső hőforrások felé közölt hőmennyiséget is.

$$QFH_l = \sum_{SSH_n=(i,j,k,q,s) \in I^H, s \leq p \leq q} \frac{t_{p+1} - t_p}{t_{q+1} - t_s} \left[\sum_{SSC_{n'} \in JFF(SSH_n)} QFF_{nn'} + \sum_{LC_{n'} \in JFL(SSH_n)} QFL_{nn'} \right] + \sum_{u \in JFU(FSH_l)} QFU_{lu},$$

$$FSH_l = (i, j, k, p) \in E^H$$

- Minden elemi ($FSC_l = (i, j, k, p)$) hideg hőáramra a képlet az elemi meleg hőáramokhoz hasonlóan keletkezik, a különbség a QFC_l előjelében van.

$$-QFC_l = \sum_{SSC_n=(i,j,k,q,s) \in I^C, s \leq p \leq q} \frac{t_{p+1} - t_p}{t_{q+1} - t_s} \left[\sum_{SSC_n \in JFF(SSH_{n'})} QFF_{n'n} + \sum_{SSC_n \in JLF(LH_{n'})} QLF_{n'n} \right] + \sum_{u \in JFU(FSC_l)} QUF_{ul}$$

$$FSC_l = (i, j, k, p) \in E^C$$

A hőcserékhez költségek is tartoznak, amely különbözik hőáramok és rejtett hő esetén. A modellben minden hőcsere költsége a hőcserélő felületével arányos.

- Hőáramok hőcseréjének költséggyűthetősége
Tegyük fel, hogy m_i és $m_{i'}$ anyagok között valósul meg hőcsere. Jelölje $A_{ii'}$ az anyagok közötti egységnyi felületű hőcserélő költségét, $U_{ii'}$ pedig az anyagok közötti fajlagos hőátadási tényezőt. Ez után az SSH_l és $SSC_{l'}$ rész hőáram közötti hőcsere költsége az alábbi módon számolható.

$$CFF_{ll'} = A_{ii'} \frac{1}{U_{ii'} LMTD(t_s, t_{q+1}, t_{s'}, t_{q'+1})}$$

$$SSH_l = (i, j, k, q, s) \in I^H, SSC_{l'} = (i', j', k', q', s') \in JFF(SSH_l)$$

Ahol $LMTD(t_s, t_{q+1}, t_{s'}, t_{q'+1})$ jelöli a rész áramok közötti közötti logaritmus középhőmérsékletet. Az érték kiszámolásának általános formája a következő.

$$LMTD(x_1, x_2, y_1, y_2) = \frac{(x_1 - y_1) - (x_2 - y_2)}{\ln \frac{x_1 - y_1}{x_2 - y_2}}$$

- Rejtett hőforrások hőcseréjének költséggyűthetősége
A rejtett hőforrás műveleti egységhez és nem anyaghoz tartozik, de egy hőcserélő berendezésben két anyag cserél hőt. Azaz itt szükség van egy köztes anyagra, amelyen keresztül a hőátvitel megtörténik. Az előzőekben megismert anyag függő paraméterek a köztes anyagra vonatkoznak, amelyet jelöljünk m -mel.

$$CFL_{ll'} = A_{im} \frac{1}{U_{im} LMTD(t_s, t_{q+1}, T_{j'i'}, T_{j'i'})},$$

$$SSH_l = (i, j, k, q, s) \in I^H, LC_{l'} = (j', i')$$

$$CLF_{ll'} = A_{mi'} \frac{1}{U_{mi'} LMTD(T_{ji}, T_{ji}, t_{s'}, t_{q'+1})}$$

$$LH_l = (j, i) \in L^H, SSC_{l'} = (i', j', k', q', s') \in JLF(LH_l),$$

$$CLL_{ll'} = A_{mm} \frac{1}{U_{mm} (T_{ji} + T_{j'i'})} \frac{2}{(T_{ji} + T_{j'i'})}$$

$$LH_l = (j, i) \in L^H, LC_{l'} = (j', i') \in JLL(LH_l)$$

- Külső hőforrások költségegyütthatója

A modellben legyen ez a költség lineáris függvénye az elvitt hőnek. Jelölje az u külső energiaforráshoz tartozó lineáris költségegyütthatót UC_u .

A hőcsere együtthatók alapján definiálni lehet a modell célfüggvényét, ami a PNS feladat és a HENS feladat együttes célfüggvénye.

$$\begin{aligned} \min \sum_{o_i \in \mathcal{O}} (fix_i y_i + prop_i x_i) + \sum_{r_j \in \mathcal{R}} \left(price_j \sum_{o_i \in \varphi^+(j)} x_i ir_{ij} \right) \\ + \sum_{QFF_{jj'}} CFF_{jj'} QFF_{jj'} + \sum_{QFL_{jj'}} CFL_{jj'} QFL_{jj'} + \sum_{QLF_{jj'}} CLF_{jj'} QLF_{jj'} + \\ + \sum_{QFU_{iu}} UC_u QFU_{iu} + \sum_{QLU_{iu}} UC_u QLU_{iu} \end{aligned}$$

7.1.6. Megoldási módszer

A feladat az ABB algoritmus segítségével megoldható. Az algoritmus kombinatorikus része (szétválasztás, kiterjesztések) változatlan. A korlátszámítási lépés változik, a matematikai modell bővül ki a hőcseréhez tartozó változókkal, egyenletekkel. A feltételrendszereket a már leírtak szerint kell generálni.

Az algoritmus az iterációk során dönt a műveleti egységek beviteléről és kizárásáról. A modell felírásakor a fixen kizárt műveleti egységek jelentenek változásokat, az ahhoz kapcsolódó hőáramok, rejtett meleg és hideg források nem kerülnek bele a modellbe.

7.2. PNS és ütemezés integrációja

A termelő rendszerek általában szakaszos és folytonos működésű berendezéseket is alkalmaznak. A szakaszos működésű eszközöket egymással konkuráló folyamatok használják, melyek ütemezése nagyban befolyásolja a folyamatok és a teljes rendszer hatékonyságát, ugyanis nem mindegy, hogy egy termékcsoporthoz legyártásakor az eszközök mennyire kihasználtnak, illetve, hogy mekkora a gyártó üzem termelékenysége. Ezért az ütemezési feladatok optimális megoldása napjaink fontos feladatává vált.

Az ütemezési feladat megoldásakor a berendezések optimális működési sorrendjének a meghatározása a cél. Az optimális ütemezést általában egy vegyes egész lineáris programozási feladat (MILP) megoldásaként kaphatjuk meg. Ipari méretű ütemezési problémák kiszámítására az általános megoldó szoftverek általában alkalmatlanok, ezért szükséges hatékony megoldó módszerek kidolgozása ütemezési feladatok kezelésére.

7.2.1. Ütemezési feladat

Az ütemezési feladatot szakaszos üzemű termelő rendszerek esetén általában a gyártási folyamat struktúrájával (az úgynevezett recepttel), és a felhasználható gépek (berendezések) halmazának megadásával definiálják. A receptben szereplő munkafázisok (művelet, taszk) elvégzésére több különböző berendezés is használható. Természetesen a használt berendezéstől függ a működési ideje és a működési költsége az adott munkafázisnak.

Az ütemezési feladatban mindegyik művelethez hozzá kell rendelni egy berendezést, amely majd ezt a műveletet végre fogja hajtani. A berendezések több műveletet is végrehajthatnak, azaz áthelyezhetőek, de természetesen egy időben csak egy műveletet végezhetnek. Egy berendezés áthelyezésekor berendezéstől és műveletektől függően egy váltási időt figyelembe kell venni. A váltási idő az az időtartam, amennyinek legalább el kell telnie a berendezéshez rendelt két egymás utáni művelet között. Általában a váltási idő a berendezések konfigurálására, tisztítására, vagy esetleg szállítására kell. Az összes berendezésnek meg kell határozni az optimális műveleti sorrendjét,

azaz ha egy berendezés végrehajtott egy műveletet akkor ezután melyik az a következő művelet amit az adott berendezésnek el kell végeznie.

Szakaszos rendszerekben a termelés **batch**ekben (adagokban) történik. Azaz ha az előállítandó termék mennyisége több, mint amennyi a recept egyszeri végrehajtása során keletkezik, akkor a recept többszöri megismétlésével állítják elő a kívánt termékmenyiséget. A recept egyszeri végrehajtását, ütemezését jelenti egy batch-nyi termék előállítása.

Az ütemezési feladatok megoldása során valamely szempont szerinti optimális ütemezést keresünk. Példák ütemezési célokra:

- Adott mennyiségű termék legyártása minél rövidebb idő alatt.
- Adott idő alatt minél több termék legyártása.
- Termékek legyártása határidőre.
- Működési költségek minimalizálása.

7.2.2. S-gráf keretrendszer

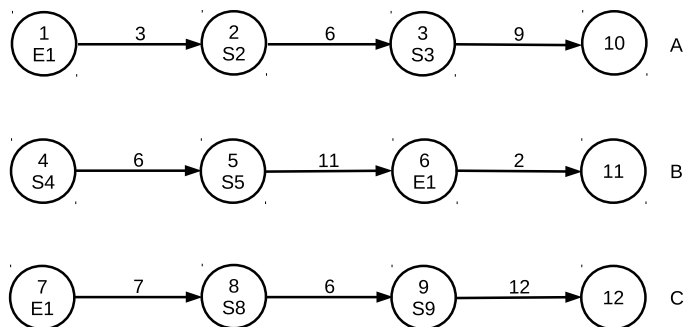
A továbbiakban mi az S-gráf leírást használjuk a feladat leírására és polinomiális idejű gráf-algoritmusokat használunk a szétválasztás és korlátozás módszer keretein belül. A feladatot leíró receptet egy irányított gráffal lehet ábrázolni, amelyet recept-gráfnak nevezünk.

Egy termék előállításának receptjét egy irányított gráffal lehet leírni, ahol csúcsok jelölik a műveleteket és a termékeket, az élek pedig megmutatják a műveletek sorrendjét. Az élekhez súlyokat is rendelhetünk, ahol az él súlya mutatja meg a két kapcsolódó művelet kezdési ideje közötti minimális különbséget. Azaz a recept végrehajtási idejét jelölő élék súlya megegyezik az él kiinduló csomópontjához tartozó művelet végrehajtásához szükséges idővel. Abban az esetben, ha egy művelethez több berendezés is rendelkezésre áll, akkor az él súlya megegyezik a leggyorsabb berendezés működési idejével. Az algoritmus futása közben, amikor egy lassabb berendezést hozzárendelünk egy művelethez, akkor a kapcsolódó élék súlya megváltozik.

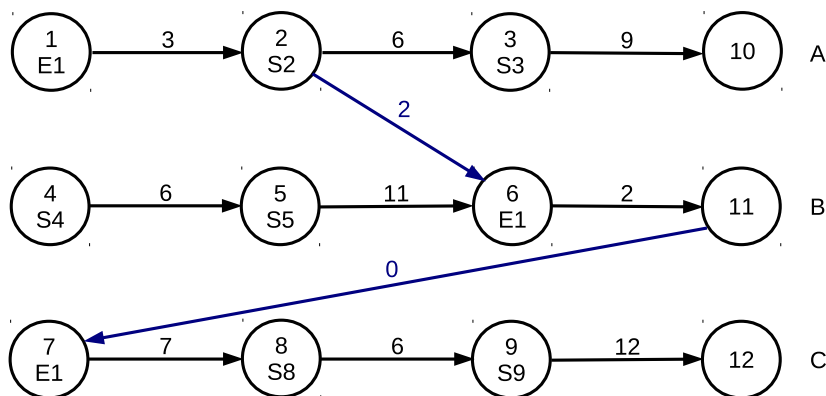
A receptet leíró gráfot **recept-gráfnak** nevezzük. Egy ütemezési feladat receptje megadja minden termékhez a hozzá tartozó műveletek sorrendjét, a köztük lévő anyagáramokat, valamint az egyes műveletekhez felhasználható berendezések halmazát a hozzájuk tartozó működési idővel. Mindezeknek az információknak a recept-gráfban szerepelniük kell ahhoz, hogy a receptet megfelelően le lehessen vele írni.

- Minden művelethez tartozik egy csomópontot (taszk-csomópont) és minden termékhez is egy (termék-csomópont).
- Minden taszk-csomópont-hoz tartozik egy halmaz, amely azokat a berendezéseket tartalmazza, amelyekkel a műveletet végrehajtani lehet hajtani.
- A receptben egymást követő műveletek egy úgynevezett recept-éllel vannak összekötve, ahol az él irányítása megegyezik a működési sorrenddel.
- Minden terméket előállító műveletből egy recept-él mutat a megfelelő termék-csomópontba.
- Egy recept-él súlya az él kiindulási taszk-csomópontjához tartozó berendezések működési idői közül a legkisebbel egyenlő.
- Ha egy termékből nagyobb mennyiségre van szükség, mint amennyit a recept alapján egyszerre elő lehet állítani (több batch-re van szükség), akkor a termék előállításában résztvevő műveletek taszk-csomópontjai, a termék-csomópont, valamint a köztük lévő élek annyiszor kerülnek be a gráfba, ahány adag már minimálisan elegendő a szükséges anyagmennyiséghez.

A 7.5-ös ábrán látható recept-gráfon három termék (*A*, *B*, *C* termék) előállítása látható. Mind a három termék előállítása három egymást követő lépésben történik meg. A műveletek végrehajtási ideje a taszk-csomópont utáni éleken olvasható és a műveletek végrehajtásához felhasználható berendezéseket a taszk-csomópontokban feltüntetett $S_1, S_2, \dots, S_9$ halmazok tartalmazzák. Nem definiáljuk,



7.5. ábra. Recept-gráf példa



7.6. ábra. Ütemezési-él példa

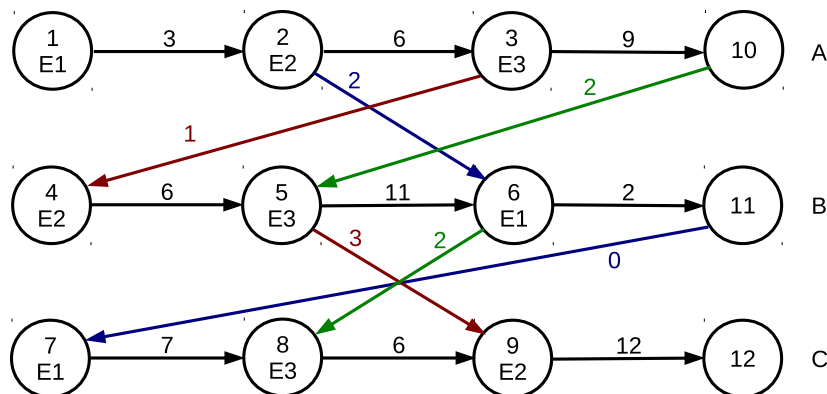
hogy melyik művelethez melyik berendezést lehet felhasználni, mert a példa megértéséhez nincs rá szükség.

Az ütemezés során a berendezéseket és műveleteket egymáshoz kell rendelni, valamint a közös berendezéshez tartozó műveleteken definiálni kell a végrehajtási sorrendet. Az egy berendezéshez tartozó műveletek végrehajtási sorrendjét az egy termékhez tartozó műveletek sorrendjéhez hasonlóan élek jelölik. Egy úgynevezett ütemezési-él kiinduló csomópontja az első művelethez tartozó taszk-csomópontot a recept-gráfban követő csomópont, a végpontja pedig a következő művelethez tartozó taszk-csomópont. Az így felvett, egy berendezés működési sorrendjét jelölő éleket ütemezési-éleknek nevezzük. Ha egy taszk-csomópont után több csomópont is van a recept-gráfban, akkor minden követő csomópontból kell ütemezési-élet húzni. Az ütemezési-él súlya megegyezik a berendezési váltási idejével, azaz azzal az idővel, ami szükséges ahhoz, hogy az egyik műveletről a másikra átváltson (például tisztítási idő).

Az ütemezési-élek ilyen fajta behúzása azt jelenti, hogy az adott műveletet végrehajtó berendezés a művelet befejezése után nem áll rendelkezésre egészen addig, amíg át nem tölti a keletkezett anyagot a következő csomópontban szereplő berendezésbe. Ha így húzzuk be az ütemezési-éleket, akkor a keletkezett gráfot S-gráfnak nevezzük.

A 7.6-os ábrán az *E1* berendezés ütemezése látható. Az *E1* berendezés először az 1-es műveleten dolgozik, az elkészült anyagot áttölti a 2-es műveletet végző berendezésbe, majd 2 időegységnyi átállási idő után kezdheti el a 6-os műveletet. A 6-os művelet elvégzése után elkészül a B termék, amely után a berendezés átállási idő nélkül kezdheti el a munkáját a 7-es műveleten. Az *E1* berendezés működési sorrendjét leíró ütemezési-élek az ábrán kékkel vannak jelölve. Azoknál a taszk-csomópontoknál, ahol megtörtént a berendezés hozzárendelése, a gráfon a berendezések halmazát kicseréltük a hozzárendelt berendezésre.

Az **ütemezési-gráf** olyan speciális körmentes S-gráf, amely egy megoldást reprezentál. Az ütemezési feladat minden megoldásához létezik egy ütemezési-gráf és ez a gráf minden megoldáshoz különböző. Egy ütemezési-gráfban a feladathoz tartozó recept-gráf minden művelete (taszk-csomópontja) ütemezve van, figyelembe véve a berendezés-művelet hozzárendelést.



7.7. ábra. Ütemezési-gráf példa

A 7.7-es ábrán egy (nem feltétlenül optimális) ütemezési-gráf látható, ahol az $E1$ berendezéshez tartozó ütemezési-élek kék, az $E2$ -höz tartozóak piros és az $E3$ -hoz tartozóak zöld színnel vannak jelölve.

A legáltalánosabb cél ütemezési feladatoknál a rendszer teljes működési idejének minimalizálása a rendelkezésre álló erőforrások segítségével. Ebben az esetben adottak a legyártandó termékek mennyiségei, melyek előállítását a recept írja le. Az optimális ütemezés meghatározására szétválasztás és korlátozás típusú algoritmust használunk, amely rész ütemezéseken keresztül generálja az optimális ütemezési-gráfot. Az algoritmus minden szétválasztási lépésben hozzárendel egy berendezést egy művelethez, majd eldönti, hogy a berendezés az adott műveletet melyik művelet után hajtja végre. Az összes csúcs és berendezés beütemezése után levél szinten vannak a lehetséges ütemezésekhez tartozó ütemezési-gráfok.

Az optimális ütemezést a legrövidebb értékű leghosszabb utat tartalmazó ütemezési-gráf adja. A nem megvalósítható részütemezések kiszűrésére az ütemező algoritmus körkereső függvényeket használ.

7.2.3. Integrált feladat

Folyamatszintézis feladatoknál a cél egy gyártási folyamat megtervezése, ütemezési feladatoknál viszont egy meglévő gyártási folyamat ütemezése a cél. Az integrált feladatban a cél egy termelési struktúra és a struktúrához tartozó ütemezés meghatározása, amely adott időtartam alatt előállítja a kívánt termékeket az adott mennyiségben minimális költség mellett.

Hálózatszintézis feladatok esetén a műveleti egységek a gyártási folyamat során elvégzett anyagokon végrehajtott elemi transzformációkat jelölik. Ütemezési feladatoknál a recept írja le a transzformációk egymáshoz való viszonyát, a receptben szereplő transzformációk a műveletek, ahol egy műveletet egy berendezéssel lehet végrehajtani.

Az integrált feladatban jelölik a műveletek azokat a kémiai, biológiai, vagy fizikai átalakításokat, amelyeket PNS feladatoknál a műveleti egységek reprezentáltak. A műveleteket itt is berendezésekkel lehet elvégezni, a berendezés típusától függően különböző hatékonysággal és költséggel. A műveletek a termelési struktúrában szereplő egységek, amelyeknek egy optimális megoldásstruktúráját és az optimális megoldásstruktúrához tartozó berendezések egy lehetséges ütemezését kell meghatározni.

A hálózatszintézis feladathoz tartozó megoldásstruktúrát reprezentáló P-gráf egy receptnek felel meg az ütemezés szempontjából, így ezen egy olyan ütemezést meghatározni, amellyel az időkorláton belül végrehajtható a termelési folyamat. Ebből következik, hogy a maximális struktúrához tartozó P-gráf nem tartalmazhat kört, ugyanis a P-gráfból származtatott S-gráfnak körmentesnek kell lennie. Azok a P-gráfok, amelyek kört tartalmaznak batch-ek bevezetésével és a kör felbontásával visszavezethetők a körmentes P-gráfok osztályához.

Ütemezés során minden műveleti egységhez hozzá kell rendelni egy megfelelő berendezést, ami az adott műveletet fogja elvégezni. A hozzárendelésektől függően egy adott műveletet eltérő költ-

séggel, végrehajtási idővel és kapacitással lehet teljesíteni. A műveletek költségei a korábbiakhoz hasonlóan két részből tevődnek össze: a fix méretű beruházási költségből, és a működtetéssel egyenes arányban növekvő lineáris költségből. A hálózat teljes költsége a benne szereplő műveleti egységek költségeinek az összege, a nyersanyagköltségektől eltekintünk. Egy berendezés különböző műveletek elvégzésére is alkalmas lehet, tehát a berendezések áthelyezhetők. Az áthelyezhetőség miatt lehet egy berendezést több művelet elvégzésére is használni, a berendezéseket ütemezni kell. Vannak kezdetben rendelkezésre álló berendezéseink, melyekhez nem tartozik beruházási költség, de ha célszerű, új eszközök beszerzésére is lehetőség van. A kezdetben rendelkezésre álló berendezések miatt van a feladatnak retrofit jellege, az áthelyezhetőség miatt pedig ütemezési részfeladatokat kell megoldani.

7.2.4. Paraméterek

A hálózatszintézis és az ütemezés két önmagában is bonyolult feladat, amelyek együttes megoldására megfelelő matematikai eszköztár szükséges. Felhasználjuk a hálózatszintézis feladatoknál bemutatott P-gráf reprezentációt, az ütemezési feladatokat leíró S-gráf reprezentációt, valamint az ezekhez kapcsolódó hatékony kombinatorikus algoritmusokat, megoldási módszereket.

Legyen E halmaz a berendezéstípusok halmaza, amelyeket a gyártási folyamat során alkalmazni lehet. Az E halmaz tartalmazza az összes berendezéstípust, ami vagy rendelkezésünkre áll vagy megvásárolhatunk. Két berendezés ugyanolyan típusú, ha a termelés szempontjából fontos paramétereik megegyeznek. A továbbiakban jelölje k_j az e_j típusú berendezésből a rendelkezésre álló mennyiséget.

Az OE függvény írja le, hogy melyik művelet milyen berendezéstípusokkal végezhető el, azaz $OE : \mathcal{O} \rightarrow \wp(E)$ típusú. Ha $o_i \in \mathcal{O}$ akkor $OE(o_i)$ halmaz azokat a berendezéstípusokat tartalmazza, amelyekkel a o_i művelet végrehajtható.

Egy művelet végrehajtása berendezéstől függően más költséggel, kapacitással és sebességgel történhet. Tegyük fel, hogy az o_i műveletet az e_j berendezéssel el lehet végezni. Ha az o_i műveletet az e_j berendezés végzi a működési költség legyen $a_{ij}x_i$, nagyságú, ahol az a_{ij} a művelettől és a berendezéstípustól függő paraméter, az x_i pedig a PNS modellben megismert, a művelet méretét leíró változó. Az x_i értéke explicit nem ismert, a korlátozás lépésben felírt modell megoldásával kap konkrét értéket.

A berendezéseken átfolyó anyag mennyiségére korlátozások vannak. Az anyagmennyiség nem haladhatja meg a berendezés fizikai méretét viszont egy nagy méretű berendezésben nem lehet kevés anyagot sem feldolgozni. Ez a korlátozás függ a berendezés típusától és a művelettől is. Az i műveletet méretének az e_j berendezéstípust használva legalább low_{ij} -nek és legfeljebb upp_{ij} nagyságúnak kell lennie, azaz $low_{ij} \leq x_i \leq upp_{ij}$.

A feladat retrofit jellegét a kezdetben meglevő berendezések lehetősége adja, ahol k_j jelöli az e_j berendezéstípusból rendelkezésre álló mennyiséget. A feladatban feltételezzük, hogy bármelyik berendezéstípusból bármennyit be lehet szerezni, azaz berendezés vásárlásra nincsen elvi korlát. Továbbá jelölje $cost_j$ egy darab e_j típusú berendezés beszerzési árát.

Egy művelet végrehajtási ideje függ a felhasznált berendezéstől, de mivel a berendezések szakaszos működésűek, ezért a működési idő nem függ a berendezésben feldolgozott anyag mennyiségétől. Jelölje $time_{ij}$ azt az időt, amely alatt az e_j berendezés végrehajtja az o_i műveletet. Ütemezés szempontjából a működési idő mellett a váltási idő a másik fontos adat, amire szükség van. Ha $o_i, o_{i'} \in \mathcal{O}$ és $e_j \in OE(o_i) \cap OE(o_{i'})$, azaz az e_j berendezéstípus alkalmas az o_i és az $o_{i'}$ műveletek elvégzésére is, akkor $ctime_{ii'j}$ legyen az az idő, amennyit egy e_j típusú berendezésnek várakoznia kell az o_i és az $o_{i'}$ műveletek végrehajtása között. Továbbá jelölje $TIME$ a gyártási folyamat befejezésére kapott határidőt.

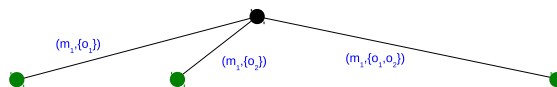
7.2.5. Megoldási módszer

A feladat megoldásához egy szétválasztás és korlátozás algoritmust használunk. Az algoritmus a PNS feladatok megoldásánál megismert ABB algoritmuson alapszik. Az algoritmust természetesen

ki kell terjeszteni, hiszen több döntésünk is lehetséges. Az eredeti ABB algoritmusban csak arról kell dönteni, hogy egy műveleti egység benne legyen-e a megoldásstruktúrában. Az integrált feladatban szükség van még döntésre arról is, hogy egy művelethez milyen típusú berendezést használjunk és hogy ehhez már meglévő berendezést használjunk vagy vegyünk egy újat.

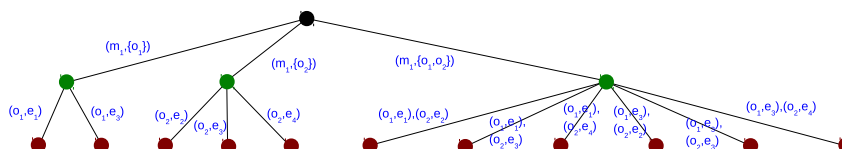
A különböző döntések különböző szinteket jelentenek a keresési fában. Az első szinten mindig az ABB anyagok műveletekkel való előállításának döntése, a második szinten berendezés típus választás és a harmadik szinten a berendezések vásárlására vonatkozó döntés történik. Az érthetőség kedvéért vizsgáljuk meg egy ilyen egymásra épülést egy példán keresztül. Legyen egy m_1 anyag, amely előállításáról az ABB algoritmus a következő lépésben dönteni fog, és amit o_1 és o_2 művelet segítségével lehet végrehajtani. Ezek után tegyük fel, hogy $OE(o_1) = \{e_1, e_3\}$ és $OE(o_2) = \{e_2, e_3, e_4\}$.

Az m_1 anyagot három módon állíthatjuk elő az o_1 és az o_2 műveleti egységek (műveletek) segítségével. Döntési leképezés segítségével ezeket a döntéseket $(m_1, \{o_1\})$, az $(m_1, \{o_2\})$ vagy az $(m_1, \{o_1, o_2\})$ módon írhatjuk le. Ez a keresőfában egy hármas elágazást jelent, amint a 7.8-as ábrán látható.



7.8. ábra. Döntések m_1 anyag előállításáról

A döntések következő szintje a használt berendezések típusát határozza meg. Amikor egy műveletről kell dönteni, hogy milyen berendezés típussal valósítjuk meg, akkor annyi döntésünk lehet, ahány különböző berendezéstípust fel lehet hozzá használni. Ilyen a 7.8-as ábrán látható keresőfa két bal oldali ága, amelyhez 2 és 3 döntés tartozik, mint a 7.9-es ábrán látható. A jobb oldali ág esetén az o_1 és az o_2 művelet esetén is dönteni kell a berendezéstípusról és ez $|OE(o_1)| \times |OE(o_2)| = 2 \times 3 = 6$ számú lehetséges döntést jelent.



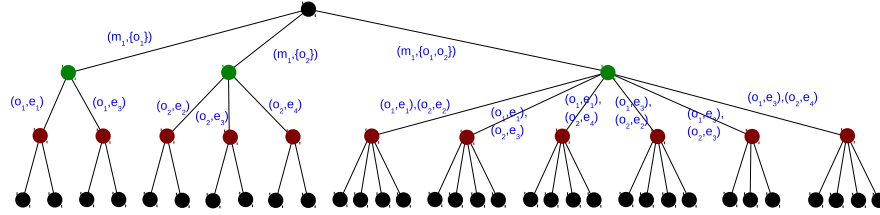
7.9. ábra. Döntések berendezéstípusokról

Berendezések vásárlásánál is végig kell nézni minden lehetőséget. Legegyszerűbb vásárlási lehetőség, ha nem veszünk új berendezést, és meglévő berendezésekkel végezzük el a gyártási folyamatot. A redundancia elkerülése miatt minden vásárlási lehetőségnek pontosan egyszer szabad csak szerepelnie a részdöntések között. Egyszerűen meggondolható hogy, ha az $e_i \in E$ berendezéstípus kettő művelethez van hozzárendelve egy döntésben, akkor ez vásárlás szempontjából háromfajta vásárlási lehetőséget jelent: az e_i berendezéstípusból vagy egyet sem, vagy egyet, vagy két darabot vásárolhatunk (ütemezés szempontjából nem lényeges, hogy melyik berendezéstípushoz vesszük meg a berendezést).

A példánkban három fajta döntést mutathatunk be.

- Egy művelethez egyetlen berendezés típus tartozik.
Ezek a 7.9-es ábrán a bal oldali 5 levél. Ebben az esetben két fajta döntés lehet, vagy veszünk új berendezést vagy nem.
- Két művelethez két különböző berendezéstípus tartozik.
Ezek a 7.9-es ábrán a jobb oldali 6 levél kivéve az utolsó előtti. Itt négy különböző döntés lehet, nem veszünk berendezést, o_1 művelethez veszünk berendezést, o_2 művelethez veszünk berendezést, mind a két művelethez veszünk új berendezést.
- Két művelethez ugyanolyan berendezéstípus tartozik.
Ez a 7.9-es ábrán jobb oldalról az utolsó előtti levél, amikor az o_1 és az o_2 művelethez is e_3 típusú berendezést választunk. Ekkor három döntés lehet, nem veszünk berendezést, egy berendezést veszünk, kettő berendezést veszünk.

A teljes keresőfa a 7.9-es ábrán látható. A harmadik szintű döntéseket az ábrán nem jelöltük, csak a döntések által generált részfeladatokat jelöltük a keresőfában.



7.10. ábra. Döntések berendezések vásárlásáról

Az integrált feladat megoldására kidolgozott branch-and-bound keretalgoritmus korlátozás lépése két részből épül fel: alsó korlát számítás a részfeladat költségére, majd a részfeladat ütemezhetőségi vizsgálata.

Első lépésben az aktuális részfeladathoz tartozó költségre egy alsó korlátot határozunk meg. Ennek a korlátnak minél élesebbnek kell lennie, hogy csökkenteni lehessen a keresési teret a szétválasztás és korlátozás jellegű algoritmusok filozófiájának megfelelően. Azonban ügyelni kell arra, hogy hogyan származtatjuk az korlátot, ugyanis a keresési fa minden csomópontjában meg kell határozni, így meglehetősen sokszor kell végrehajtani az alsó korlát számítást. A PNS feladatokhoz kidolgozott MILP feladat megoldásaként kapott eredményt használva megfelelő alsó korlátot kaphatunk az LP-ben relaxált részfeladat költségére.

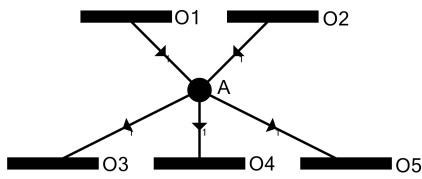
Az integrált feladat modellje abban különbözik az eredeti PNS modelltől, hogy alsó korlát is lehet a műveletekre.

- Az o_i művelet a döntés szerint kikerült a struktúrából. Ekkor a művelet méretének alsó és felső korlátja 0, azaz $0 \leq x_i \leq 0$
- Az o_i művelet a döntés szerint bekerült a struktúrába, és a döntés szerint az e_i típusú berendezés lesz hozzá használva. Ekkor a művelet méretének alsó és felső korlátja megegyezik a korábbiakban definiált alsó és felső korláttal, azaz $low_{ij} \leq x_i \leq upp_{ij}$.
- Az o_i művelet a döntés szerint bekerült a struktúrába, de a használt berendezés típusáról még nem történt döntés. Ebben az esetben művelet méretének alsó korlátja a felhasználható berendezéstípusok közül a legkisebb alsó korlátja, a művelet felső korlátja pedig a legnagyobb felső korlátja lesz, azaz $\min_{e_j \in OE(o_i)} low_{ij} \leq x_i \leq \max_{e_j \in OE(o_i)} upp_{ij}$.
- A műveletről még nem történt döntés. Ekkor az alsó korlát nulla, a felső korlát pedig a felhasználható berendezéstípusok közül a legnagyobb felső korláttal lesz egyenlő, azaz

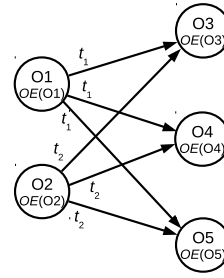
$$0 \leq x_i \leq \max_{e_j \in OE(o_i)} upp_{ij}$$

A részfeladat ütemezhetőségi vizsgálata során meg kell határozni, hogy a struktúrához tartozó gyártási folyamatot végre lehet-e hajtani az előzetesen megadott időkorlát alatt. A gyártási időt az integrált feladathoz tartozó megoldásstruktúrából lehet számolni, ezért először az aktuális részfeladatból egy megfelelő megoldásstruktúrát kell származtatni, majd ebből a recept-gráfot. Ha egy részfeladatból nem származtatható olyan ütemezés, amelyhez tartozó gyártási idő a határidőn belüli, akkor a részfeladat eldobható, nem tartalmazza az integrált feladat megoldását. Ugyanakkor ha van legalább egy határidőn belüli ütemezés (aminek nem kell optimálisnak lennie), akkor a részfeladatot tovább kell vizsgálni.

A korlátozás lépésben a megoldásstruktúra származtatása az SSG algoritmus segítségével végezhető el. Az integrált feladat eddigi strukturális döntéseiből kiindulva egy megoldásstruktúrát kell meghatározni. Előfordulhat, hogy az eddigi döntések nem vezetnek megoldásstruktúrához, ekkor a részfeladatot nem kell tovább vizsgálni, nem lehet megoldása az integrált feladatnak. Azokhoz a műveletekhez, amelyeket az SSG algoritmus rakott be a megoldásstruktúrába, fiktív, egyedi berendezéseket rendel az algoritmus. Ezeknek a fiktív berendezéseknek nincsen típusa, nem kell ütemezni őket. A fiktív berendezéshez tartozó műveleti idő egyenlő annak a berendezéstípusnak a műveleti



7.11. ábra. Példa: P-gráf átalakítása



7.12. ábra. Példa: átalakított S-gráf

idejével, amely a lehető legrövidebb idő alatt tudja végrehajtani az aktuális műveletet. Az SSG algoritmust abban az esetben lehet befejezni, ha találtunk egy olyan megoldásstruktúrát, amelyhez van időkorláton belüli ütemezés, így könnyen előfordulhat, hogy nem kell az összes megoldásstruktúrát meghatározni.

Az SSG algoritmus segítségével kapott megoldásstruktúrából recept-gráfot azért kell készíteni, hogy az ütemezhetőségi vizsgálatot el lehessen végezni. A recept gráf származtatása egyszerűen elvégezhető, a műveletek közti időbeli függőségeket vizsgálva.

A megoldásstruktúra P-grájában szereplő mindegyik művelethez rendeljünk egy csúcsot az S-gráfban, továbbá mindegyik $\mathcal{P}$ halmazbeli termékhez rendeljünk egy S-gráfbeli termék-csomópontot. Az S-gráf recept-éleit a P-gráf éleinek megfelelően húzzuk be. Ha egy anyagot például az o_i művelet termeli, az o_j művelet pedig fogyasztja, ez azt jelenti, hogy az o_j művelet csak akkor kezdődhet el, ha az o_i művelet befejeződött. Ez a függőség az S-gráfban az o_i és az o_j művelethez tartozó S-gráf csúcsok közé behúzott megfelelő irányítású recept-él reprezentálható. Az él súlya a művelethez rendelt berendezés működési idejével egyezik meg. A termékcsúcsok és a terméket gyártó műveletek közti élek hasonlóan vannak behúzva az S-gráfban. Egy S-gráfbeli műveletet a művelethez rendelt berendezéstípusból rendelkezésre álló berendezések valamelyikével lehet elvégezni.

Tegyük fel, hogy egy megoldásstruktúrában egy anyagot két művelet gyárt és három művelet fogyaszt. Ez a struktúra látható a 7.11-es ábrán. A 7.11-es ábrán lévő P-gráfból a fent leírt módon transzformált S-gráf a 7.12-es ábrán található, ahol t_1 az $O1$ művelet végrehajtásához szükséges legrövidebb idő az $OE(O1)$ halmazban lévő berendezések valamelyikével, t_2 pedig az $O2$ művelet végrehajtásához szükséges legrövidebb idő az $OE(O2)$ halmazban lévő berendezések valamelyikével.

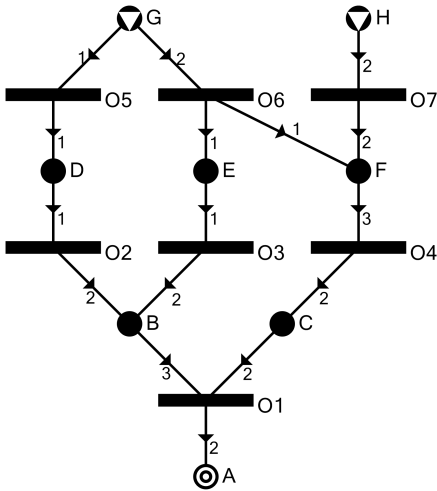
7.2.6. Példa

Az előző fejezetben bemutatott algoritmus működését egy feladaton keresztül szemléltetem. Defináljuk a $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ szintézisfeladatot a feladathoz tartozó P-gráffal, ahol legyen $\mathcal{R} = \{G, H\}$, $\mathcal{P} = \{A\}$, az $\mathcal{O}$ halmaz műveletei, továbbá a ki- és bemeneti anyagarányok a 7.13-as ábra P-gráfból, mint a feladat maximális struktúrájából leolvashatóak. Például az $O3$ művelet 1 egységnyi E anyagot 2 egységnyi B anyaggá transzformál.

A feladat 12 egységnyi A termék legyártása az $e_1, e_2, e_3, \dots, e_4$ típusú berendezések használatával 80 időegységen belül. Azt, hogy egy művelet milyen típusú berendezéssel végezhető el, a 7.1-es táblázat definiálja. A táblázatból kiolvasható, hogy például az $O3$ művelet e_1 vagy e_3 típusú berendezéssel végezhető el.

A kezdetben rendelkezésre álló berendezések mennyiségét valamint egy berendezés megvásárlásának a költségét a 7.2-es táblázat tartalmazza. Például az e_1 berendezésből már 1 darab rendelkezésre áll és egy új beszerzése 130 egységbe kerül.

A berendezéstípusok és a műveletek kapcsolatát a 7.3, 7.4 és a 7.5-ös táblázatok mutatják meg. Például a 7.3-as táblázat tartalmazza a kapacitáskorlátokat, azaz a különböző típusú berendezések méretének alsó és felső korlátját adott műveleten való felhasználás esetén. Ha például az $O3$ műveletet e_3 típusú berendezés végzi, akkor a berendezés legalább 2 egységnyi és legfeljebb 10 egységnyi méretű műveletnek felel meg.



7.13. ábra. Példa: P-gráf maximális struktúra

Művelet	Berendezések
$O1$	e_1, e_2
$O2$	e_2
$O3$	e_1, e_3
$O4$	e_1, e_4
$O5$	e_4
$O6$	e_3, e_4
$O7$	e_1

7.1. táblázat. Példa: műveletek paramétere

Berendezés	k_j	$cost_j$
e_1	1	130
e_2	0	160
e_3	0	160
e_4	1	100

7.2. táblázat. Példa: berendezések paramétere

Berendezés	Művelet						
	$O1$	$O2$	$O3$	$O4$	$O5$	$O6$	$O7$
e_1	[0,4]		[2,6]	[2,5]			[3,9]
e_2	[2,10]	[0,16]					
e_3			[2,10]			[2,10]	
e_4				[0,10]	[2,10]	[1,5]	

7.3. táblázat. Példa: kapacitáskorlátok

Berendezés	Művelet						
	$O1$	$O2$	$O3$	$O4$	$O5$	$O6$	$O7$
e_1	21		23	27			30
e_2	24	20					
e_3			23			20	
e_4				20	15	23	

7.4. táblázat. Példa: működési idők

Berendezés	Művelet						
	O1	O2	O3	O4	O5	O6	O7
e_1	80		170	50			90
e_2	60	43					
e_3			150			80	
e_4				100	80	70	

7.5. táblázat. Példa: lineáris költségparaméterek

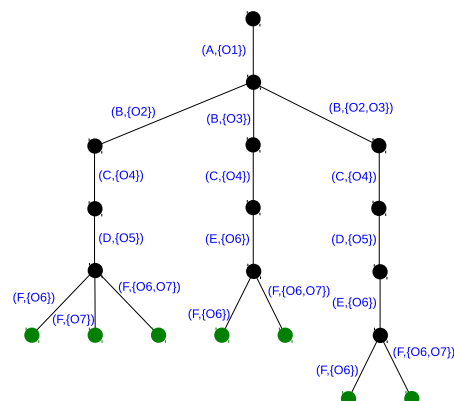
A 7.4-es táblázat tartalmazza a műveletekhez tartozó működési időket. A működési idő függ a berendezés típusától. Például az $O3$ művelet e_3 berendezéssel hajtjuk végre, akkor az 23 időegységig tart. Továbbá tegyük fel, hogy a berendezésekhez tartozó váltási idő nulla.

A költségek meghatározásához meg kell adni a műveletek és a nyersanyagok költségparamétereit. A 7.5-ös táblázat tartalmazza a műveletekhez tartozó működési költségek meghatározásához szükséges lineáris paramétereket. Ha például az O3 műveletet e_3 típusú berendezés végzi, és a berendezés 5 egységnyi mérettel működik, akkor a működési költség $5 \cdot 150 = 750$. Egységnyi tömegű G illetve H nyersanyag költsége legyen 40 illetve 65 egység.

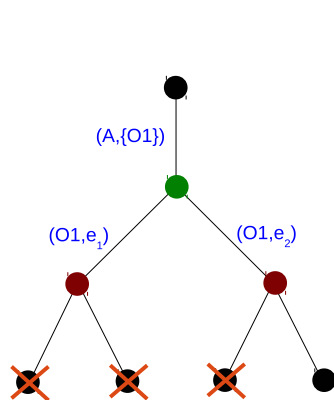
A PNS feladathoz az ABB algoritmus alapján a 7.14-es ábrán látható keresőfát kapjuk, ha nem használjuk a maximális neutrális kiterjesztést. A keresőfa élei mellett a döntési leképezés segítségével vannak megadva a döntések. A hálózatszintézis feladathoz hét megoldásstruktúra tartozik, amelyeket a fa levelei jelölnek. A levelekből a gyökérbe vezető úton haladva és az élek mellett feltüntetett műveletek azonosítót leolvasva megkapjuk a levélhez tartozó megoldásstruktúrában szereplő műveleteket.

Az integrált feladat megoldásakor három különböző részdöntés tartozik egy döntéshez: a struktúrára vonatkozó, a berendezéstípusra vonatkozó és a berendezés vásárlására vonatkozó döntés. A 7.14-es ábra csak a strukturális döntések kapcsolatát szemlélteti.

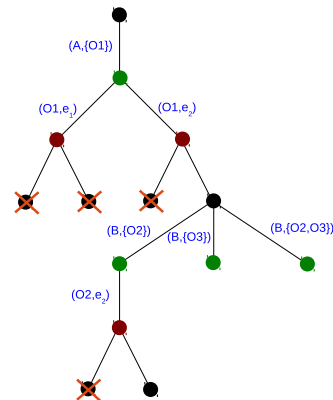
Az algoritmus működésének bemutatásához (ugyanúgy, ahogy az ABB algoritmusnál) legyen p halmaz azoknak az anyagoknak a halmaza, amelyeket gyártani kell, hogy megoldásstruktúrát kapjunk. Azaz kezdetben $p = \{A\}$.



7.14. ábra. Példa: ABB keresőfa



7.15. ábra. Példa: integrált algoritmus döntési fája az első lépés után



7.16. ábra. Példa: integrált algoritmus döntési fája a második lépés után

Első lépésben először az A anyag termeléséről kell döntést hozni. Mivel kizárólag az $O1$ művelettel lehet A anyagot előállítani, ezért az $O1$ műveletet bele kell venni a megoldásstruktúrába. Ez egy új részfeladatot jelent, amit a 7.15-ös ábrán zöld színnel jelöltünk. Ha az $O1$ művelet benne van a megoldásstruktúrában, akkor biztosítani kell azoknak az anyagoknak a gyártását, amelyeket az $O1$ művelet használ. Azaz a későbbiekben dönteni kell ezek gyártásáról, ezért a következő lépésben $p = \{B, C\}$ halmazzal kell majd dolgozni.

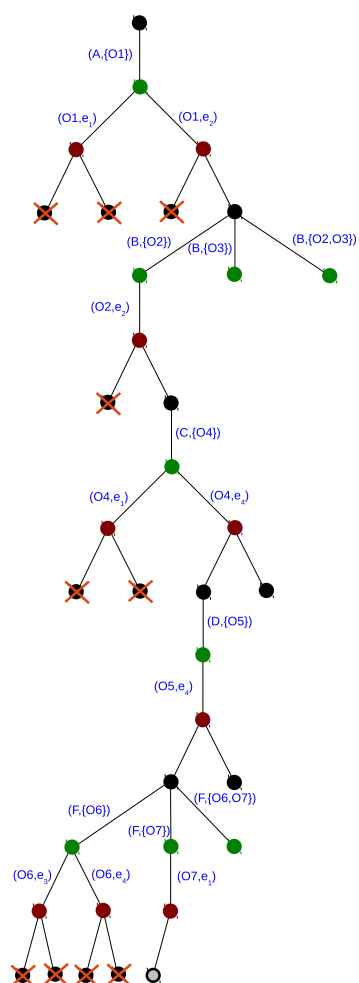
Az $O1$ műveletet vagy e_1 vagy e_2 típusú berendezéssel lehet elvégezni a 7.1-es táblázat alapján. Ez két lehetséges döntést jelent, azaz a fában egy kettes elágazás van, ahogy ez a 7.15-ös ábrán látható. A korábbiakhoz hasonlóan az így keletkezett részfeladatokat piros színnel jelöltük. Abban az esetben, ha az $O1$ műveletet a e_1 típusú berendezéssel hajtjuk végre, akkor ezt vagy a meglévő e_1 típusú berendezéseinkkel (a 7.2-es táblázat szerint egy darab van belőle), vagy egy új e_1 típusú berendezéssel végezhetjük el. Erre vonatkozik a következő részdöntés, amely két új részfeladatot jelent. Ha az $O1$ művelethez az e_2 típusú berendezést választjuk, akkor is ugyanez a két lehetséges döntésünk van, amely szintén két új részfeladatot jelent (fekete csomópontok a 7.15-ös ábrán).

Az LP modell megoldása és a beütemezhetőségre vonatkozó vizsgálatok után a négy részfeladat közül háromra kiderült, hogy nem tartalmazhat optimális megoldást, amelyeket az ábrán áthúzással jelöltünk. Ezek alapján mondhatjuk, hogy ha a feladatnak létezik optimális megoldása, akkor az abban a részfeladatban van, amelyben az A anyagot az $O1$ művelettel és e_2 berendezéstípussal állítjuk elő, és vásárolunk egy darab e_2 típusú berendezést a gyártási folyamat elvégzésére.

A második lépésben az előző lépéshez hasonlóan itt is először a strukturális döntéseket kell meghozni a p halmaz egy elemére. Válasszuk a B anyagot a halmazból. Mivel a B anyagot két művelettel lehet előállítani, ezért ez háromfajta strukturális döntési lehetőséget jelent. A 7.16-os ábrán látható, hogy az $O2$ művelet kiválasztása után az e_3 típusú berendezést lehet csak használni és itt is két részdöntés van a meglévő vagy új berendezés használatára. A többi művelet végrehajtását jelenleg nem vizsgáljuk, mivel az egyik keletkezett részfeladat megvalósíthatónak bizonyult, így annak a vizsgálatával folytathatjuk az algoritmust.

A további lépéseket nem elemezzük ki egyenként, hanem a 7.17-es ábrán lévő döntési fán megmutatjuk az optimális megoldáshoz vezető döntéssorozatot. Az ábrán az optimális megoldást a szürke csomópont jelöli. Az optimális megoldáshoz a fa inorder (bal-szülő-jobb) bejárásával jutottunk. Az algoritmus nem ér véget az optimális megoldás megtalálásánál, hiszen amíg a keresőfa összes nyitott részfeladatáról nem bizonyítjuk, hogy azok nem tartalmazhatnak jobb megoldást, addig azokat tovább kell bontani. Viszont a jobb áttekinthetőség érdekében nincsenek az ábrán feltüntetve az optimális megoldás után generált részfeladatok.

Az optimális megoldás az $O1$, $O2$, $O4$, $O5$ és $O7$ műveleteket tartalmazza, melyeket rendre e_2 , e_2 , e_4 , e_4 és e_1 típusú berendezésekkel hajtjuk végre. A meglévő e_1 és e_4 berendezések mellé két darab e_2 típusú berendezést kell vásárolni, hogy a termelési folyamat a határidőn belül befejeződjön. Az optimális megoldásban a rendszer költsége 4727 egység, a rendszerhez pedig tartozik egy olyan ütemezés, amelynek ideje 74 időegység, amely határidőn belüli.



7.17. ábra. Példa: integrált algoritmus döntési fája az optimum megtalálása után

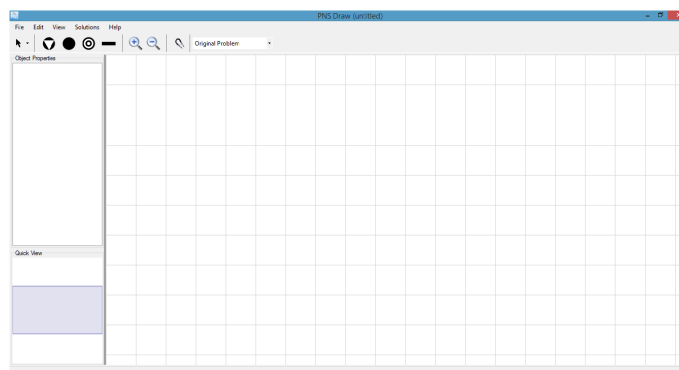
8. FEJEZET

PNS szoftverek

A korábbi fejezetekben bevezetett és megismert folyamathálózat-szintézis feladatok modellezésére, grafikus reprezentálására és az optimális megoldás meghatározására saját készítésű szoftverek kerültek kidolgozásra. A feladatok modellezésében és szemléletes vizualizációjában a PNS Draw rajzolóprogram, míg az optimális megoldás keresésében a PNS Studio keretrendszer lehet segítségünkre. Mindkét szoftver ingyenesen elérhető a www.p-graph.com weboldalról, valamint jelen jegyzet mellékleteként is letölthetőek ezen alkalmazások olyan videókkal együtt, amelyek segítenek a következő ismertető mellett is alkalmazni a szoftvereket. Elsőként a „PNS-graph drawer” (röviden: PNS Draw) kerül bemutatásra.

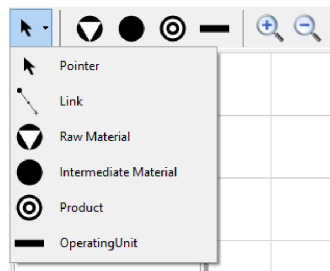
8.1. PNS Draw

A PNS Draw szoftver elsődleges célja folyamathálózat-szintézis feladatok modellezése és grafikus reprezentálása. Olyan eszköztárat biztosít a felhasználók számára, amellyel a PNS feladatok könnyedén leírhatók és ábrázolhatók. Természetesen P-gráfok rajzolására a legtöbb általános rajzolóprogram is alkalmas lehet, azonban igencsak időigényes lehet egy ilyen páros gráf megrajzolása, ha a szabványos jelölésekkel akarjuk leírni a feladatot. Mindazonáltal azt se hagyjuk figyelmen kívül, hogy az egyes P-gráfokkal leírt feladatok egyúttal optimalizálási feladatként is tekinthetők, ahol mind az anyag, mind a műveleti egység típusú csomópontokhoz paramétereket rendelhetünk. Ezen paraméterek nem csak vizuálisan, hanem logikailag is az adott P-gráf elemhez kell, hogy tartozzon. A program futtatásának minimális feltételei a következők: legalább 800 MHz-es Pentium III-as (vagy azal egyenértékű) processzor és 256 MB RAM. A program Windows környezetben futtatható Windows 2000 SP3, Windows XP SP2, Windows Vista, Windows 7 vagy Windows 8 operációs rendszerek megléte esetén. Az első két operációs rendszer esetén .NET 2.0 keretrendszer megléte is szükséges a program futtatásához. A szoftver nyitóképernyője a 8.1-es ábrán látható.



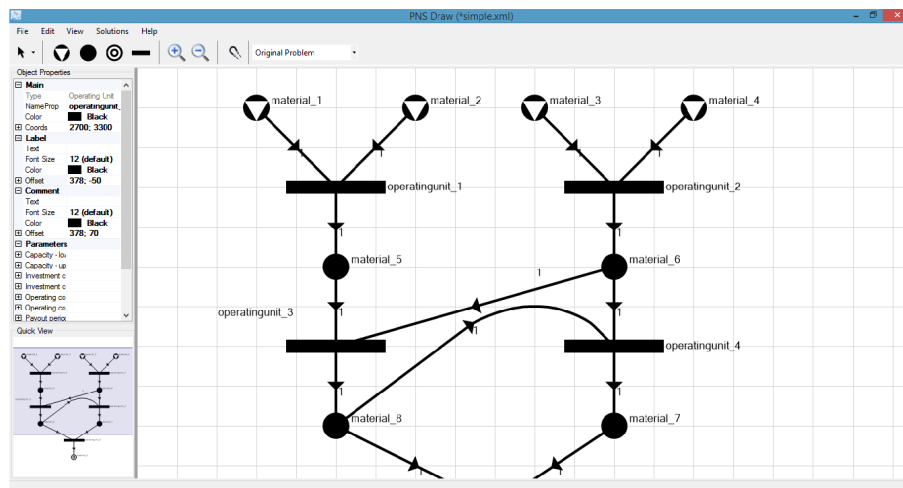
8.1. ábra. PNS Draw nyitóképernyője

Mint az látható, a menüsor és az eszköztár mellett három részre osztható a program nyitóképernyője. A jobb oldali — legnagyobb részt elfoglaló — ablak maga a rajzterület, ahol a modellezendő P-gráf kerül megrajzolásra. A bal oldali ablaksáv függőlegesen két részre osztható: az „Object Properties” rész tartalmazza a rajzterületen kijelölt komponens tulajdonságait: ezek mutathatják egy anyag jellegű csomópont jellemzőit, de akár egy műveleti egység, vagy az ezeket összekötő élek sajátosságai is ezen ablakban fognak megjelenni. Az alatta elhelyezkedő „Quick View” ablak a megrajzolt gráf egy gyorsnézetét mutatja majd, amely lekicsinyítve mutatja az elkészült struktúrát. Az eszköztár legfontosabb elemeit az alábbi ábra mutatja. Mint az sejtethető, a rajzoló program drag&drop módon működik, azaz kiválasztjuk az eszköztárról a megfelelő P-gráf elemet és azt az egér bal gombjának nyomva tartása mellett a rajzterületre húzhatjuk. Ilyen elem lehet a háromféle anyag típusú csomópont, vagy a műveleti egység. A nagyítóba írt plusz-, és mínuszjelek a különféle alkalmazói programoknál már megszokott nagyítást és kicsinyítést jelzik. Elemet kijelölni a „Pointer” mutató eszközzel tudunk, míg a komponensek közötti kapcsolatot a „Link” rajzoló mód kiválasztásával tudjuk megadni. Elsőként a kiindulási komponensre, majd a célkomponensre kell kattintanunk. Ekkor egy olyan élet adtunk hozzá a gráfhoz, amely a kiindulási komponensből a célkomponensre mutat.



8.2. ábra. A PNS Draw eszköztár elemei

Egy PNS Draw-val rajzolt P-gráf a 8.3-as ábrán látható. A csomópontoknak a szoftver automatikusan egyedi nevet ad, és mint az a 8-as anyag csomópont és a 4-es műveleti egység között is látszik, az élek görbülete változtatható, nem feltétlenül szakasz köt össze két összeillő csúcst.



8.3. ábra. Egy a PNS Drawban megrajzolt P-gráf

Kijelölve egy komponenst megadhatók annak tulajdonságai. Anyag jellegű csomópontoknál ezek például a kinézetet leíró paraméterek, mint például a szín, az anyag megnevezése, annak színe, betűmérete, stb., valamint rögzíthetők az optimalizálás szempontjából fontos értékek is: így például egy anyag ára, az abból rendelkezésre álló, vagy a legyártandó mennyiség. Műveleti egységet reprezentáló csomópontnál hasonlóan az előbb felsoroláshoz szintén rögzíthetők a kinézetet leíró jellemzők, azonban itt is szerepeltethetők egyéb optimalizáláshoz szükséges tulajdonságok is: megadhatók például a műveleti egységek kapacitáskorlátai alsó- és felsőkorlátok, a fix és arányos költségek, amelyek a működtetés során szintén fontosak, de akár a működési időre is szabhatunk

megkötéseket.

Ezen adatokat a PNS Draw egy XML formátumú fájlba menti, amely rögzíti mindazon adatokat, amelyek a rajzolás során megadtunk. Nem kitöltött adat esetén az ahhoz tartozó alapértelmezett érték kerül eltárolásra. Egy ilyen minta XML fájlt mutat a következő pár sor. Elsőként tekintsük egy anyaghoz elmentett információkat:

```
<Material ID="1" Name="material_1" Title="" Type="0">
  <ParameterList>
    <Parameter Name="price" Prefix="Price: "
      Value="-1" MU="" Visible="false" />
    <Parameter Name="reqflow" Prefix="Required flow: "
      Value="-1" MU="" Visible="false" />
    <Parameter Name="maxflow" Prefix="Maximum flow: "
      Value="-1" MU="" Visible="false" />
  </ParameterList>
  <Coords>
    <X>900</X>
    <Y>300</Y>
  </Coords>
  <Label Text="">
    <Offset>
      <X>103</X>
      <Y>-100</Y>
    </Offset>
    <FontSize>-1</FontSize>
    <Color>-16777216</Color>
  </Label>
  <Comment Text="">
    <Offset>
      <X>103</X>
      <Y>20</Y>
    </Offset>
    <FontSize>-1</FontSize>
    <Color>-16777216</Color>
  </Comment>
  <Parameters Text="">
    <Offset>
      <X>103</X>
      <Y>140</Y>
    </Offset>
    <FontSize>-1</FontSize>
    <Color>-16777216</Color>
  </Parameters>
  <Color>-16777216</Color>
</Material>
```

Látható, hogy ezen XML fájlrészletben minden anyaghoz elmentésre kerülnek a paraméterek és azok értékei: ilyenek tehát a tekintett anyag ára, az abból előállítandó anyagmennyiség, valamint az abból rendelkezésre álló maximális mennyiség. Az ezt követő kódsorok a megjelenésre vonatkozó információkat tárolják, amelyek részletes ismertetésétől most eltekintünk, azok a megadott kódrészletből már könnyen kikövetkeztethetők.

Egy műveleti egység jellegű csomóponthoz az alábbi XML leírás tartozik:

```
<OperatingUnit ID="3" Name="operatingunit_1" Title="">
  <ParameterList>
    <Parameter Name="caplower" Prefix="Capacity, lower bound: "
Value="-1" MU="" Visible="false" />
    <Parameter Name="capupper" Prefix="Capacity, upper bound: "
Value="-1" MU="" Visible="false" />
    <Parameter Name="investcostfix" Prefix="Investment cost, fix: "
Value="0" MU="" Visible="false" />
    <Parameter Name="investcostprop" Prefix="Investment cost, proportional: "
Value="0" MU="" Visible="false" />
  </ParameterList>
</OperatingUnit>
```

```
<Parameter Name="opercostfix" Prefix="Operating cost, fix: "
Value="0" MU="" Visible="false" />
<Parameter Name="opercostprop" Prefix="Operating cost, proportional: "
Value="0" MU="" Visible="false" />
<Parameter Name="payoutperiod" Prefix="Payout period: "
Value="-1" MU="" Visible="false" />
<Parameter Name="workinghour" Prefix="Working hours per year: "
Value="-1" MU="" Visible="false" />
</ParameterList>
<Coords>
<X>900</X>
<Y>900</Y>
</Coords>
<Label Text="">
<Offset>
<X>378</X>
<Y>-50</Y>
</Offset>
<FontSize>-1</FontSize>
<Color>-16777216</Color>
</Label>
<Comment Text="">
<Offset>
<X>378</X>
<Y>70</Y>
</Offset>
<FontSize>-1</FontSize>
<Color>-16777216</Color>
</Comment>
<Parameters Text="">
<Offset>
<X>378</X>
<Y>190</Y>
</Offset>
<FontSize>-1</FontSize>
<Color>-16777216</Color>
</Parameters>
<Color>-16777216</Color>
</OperatingUnit>
```

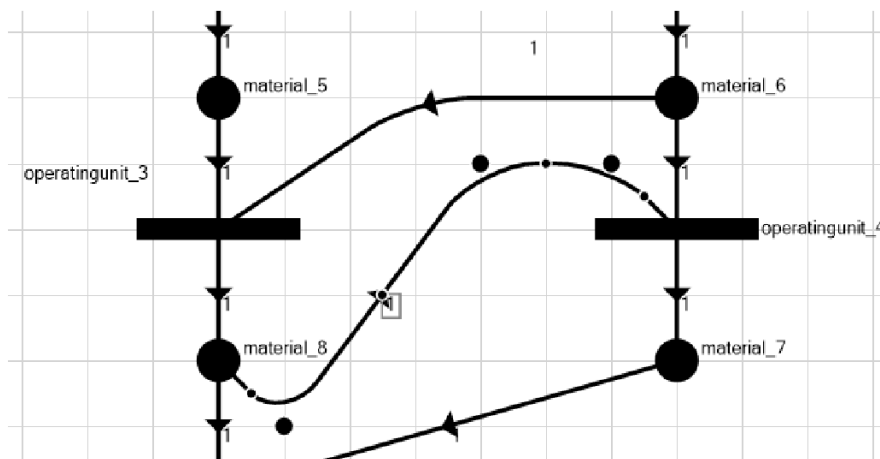
Műveleti egységet reprezentáló csomópontnál 8 paraméterérték kerül rögzítésre: a műveleti egység kapacitásának alsó- és felső korlátja, a befektetési és a működési költségek fix és arányos részei, a kifizetési időszak, valamint a működési idő értéke. Minden paraméterértéknél természetesen megadható annak mértékegysége is. A paraméterlistát követően – hasonlóan az anyag jellegű csomópontokhoz – a kinézetet leíró kód részletet találjuk meg.

Az anyag jellegű csomópontokat és a műveleti egységeket élekkel kötjük össze. Minden él esetén tudnunk kell annak kezdőpontját, azt a csomópontot, amely annak kiindulópontja, valamint tudnunk kell a végpontot meghatározó csomópontot is. E két jellemzőt a BeginID és az EndID attribútumok tárolják. Fontos jellemző még egy él esetén annak súlya, amely egy P-gráf esetén egy arányszám (flowérték). Ezt az értéket a Rate attribútum adja meg. Alapértelmezésben ez az érték 1-gyel egyenlő.

```
<Edge ID="13" BeginID="1" EndID="3" Rate="1"
Title="" ArrowOnCenter="true" ArrowPosition="50">
<Nodes />
<Label Text="">
<Offset>
<X>5</X>
<Y>0</Y>
</Offset>
<FontSize>-1</FontSize>
<Color>-16777216</Color>
</Label>
<Color>-16777216</Color>
```

</Edge>

A PNS Draw kapcsán egy fontos funkcióról beszélnünk kell még. Ez az élek rajzolását és görbületét meghatározó jellemző. Jogos igényként merül fel, hogy a megrajzol P-gráf a lehető legkevesebb él-keresztvezést tartalmazza és ne csupán szakaszokkal legyen összeköthető a gráf két megfelelő komponense. Ezt elősegítendő szerkeszthetők az élek tulajdonságai is: így a rajtuk át haladó flow értéke pozicionálható, de akár színekkel is kiemelhetünk egy élet a többi közül. Ezen túlmenően töréspontokat is definiálhatunk az éleken, amelyek kétféleképpen lehetnek: létrehozhatunk ideiglenes és állandó töréspontokat. Az előbbieket apró körök, míg az utóbbiakat nagyobb körök jelzik. A töréspontoknak is megadhatjuk néhány tulajdonságát, amelyek alapvetően annak simaságára utalnak. A töréspontok jelölésére a 8.4-es ábra mutat példát.

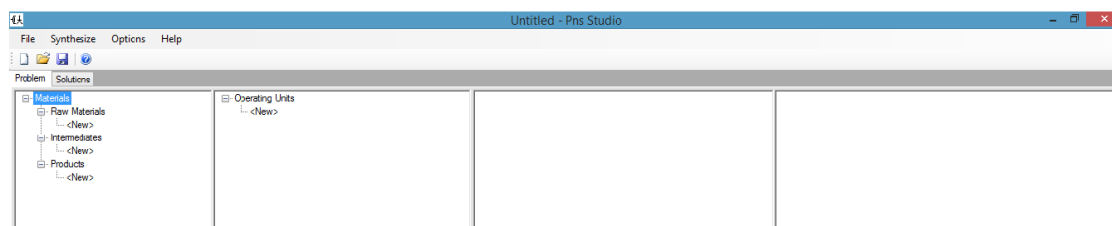


8.4. ábra. Töréspontok jelölése a szoftverben

Végül szólnunk kell a megrajzott gráf mentési és exportálási lehetőségeiről is. Ahogy azt az előbbiekben részleteztük a program XML formátumban menti el a megrajzott P-gráfot. Van azonban arra is lehetőség, hogy az eredményként előállt gráfot exportáljuk PNG, SVG vagy PNS formátumba. Az előbbi két formátum képként menti el a rajzterület elemeit, míg az utóbbi PNS formátum módot arra, hogy a következő alfejezetben bemutatásra kerülő PNS Studio azt beolvassa és elvégezhető legyenek különféle vizsgálatok: így futtathatók legyen többek közt pl. az MSG, SSG, vagy ABB algoritmusok.

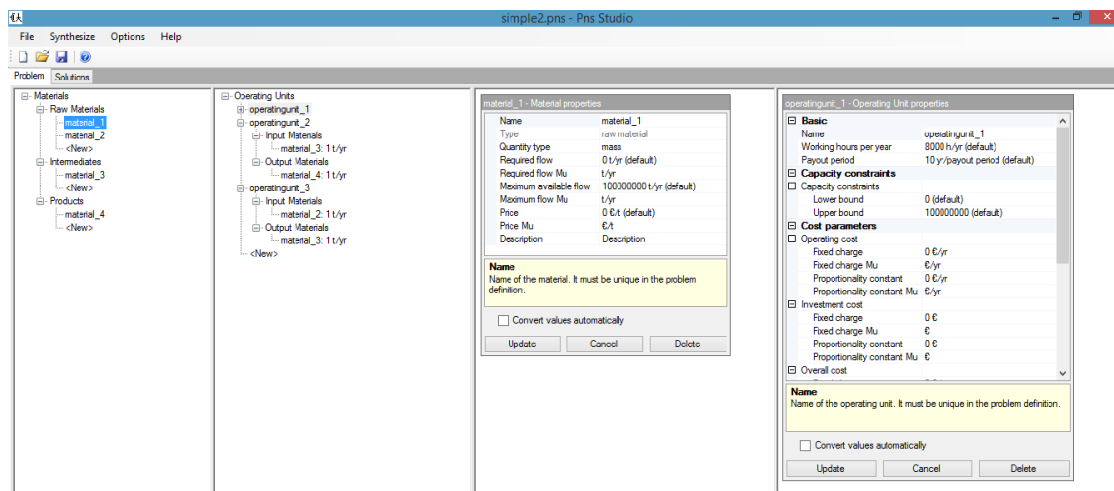
8.2. PNS Studio

Folyamathálózat-szintézis feladatok vizsgálatára és az optimális megoldás meghatározására a PNS Studio nyújt lehetőséget. Rendszerkövetelményei megegyeznek a PNS Draw szoftver követelményeivel, azaz legalább 800 MHz-es Pentium III-as (vagy azzal egyenértékű) processzor és 256 MB RAM szükséges a Windows környezet mellett ahhoz, hogy a szoftver futtatható legyen. A programot elindítva a következő nyitóképernyő fogad bennünket.



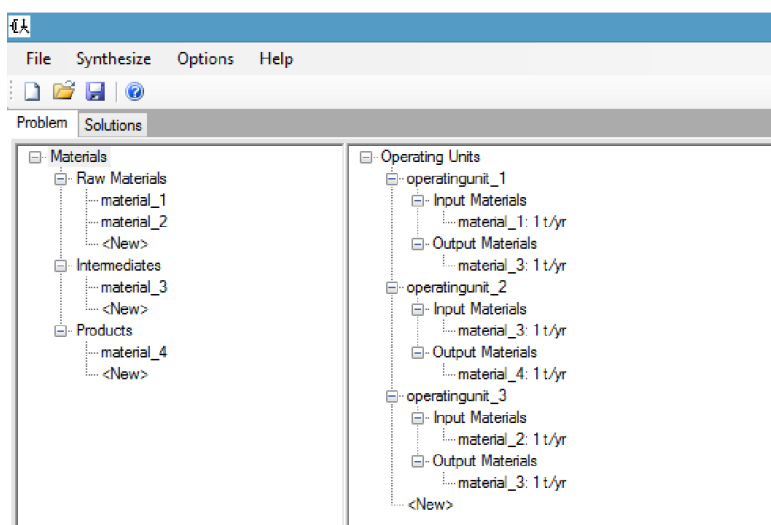
8.5. ábra. A PNS Studio nyitóképernyője

Az ablak négy nagyobb részre osztható: a bal oldali függőleges sáv az anyagok felvitelét biztosítja, ezt követi jobbra haladva a műveleti egységek rögzítését biztosító kisebb ablak, majd további két függőleges sáv zárja a program felületét: az előbbi a kiválasztott anyagok paramétereit mutatja, míg az utóbbi a műveleti egységek jellemzőit ismerteti. Betöltve egy mintafájl a következő séma szerint jelennek meg a P-gráf adatai:



8.6. ábra. Egy anyag és műveleti egység típusú csomópont adatait mutató ablak

Új anyag és műveleti egység jellegű csomópontot az első két ablakban vihetünk fel a '<New>' szövegre kattintva. A program ekkor automatikusan új, egyedi névvel látja el a létrehozott anyagot és műveleti egységet. Az ábráról látható (8.7), hogy a nyersanyagokat, köztes anyagokat és a termékeket külön helyen tudjuk felvinni, biztosítva így rögtön ezen anyag jellegű csomópontok fajta szerinti besorolását. A műveleti egységek megadása további műveletet is megkövetel. Minden ilyen csomópont a gráfban rendelkezik bemenetekkel és kimenetekkel, azaz definiálnunk kell, hogy mely anyagokból, mely más anyagokat állít elő a tekintett komponens. Ezt a legegyszerűbb módon drag&drop technikával valósították meg a fejlesztők, azaz az anyagokat tartalmazó Materials listából az egér nyomógombjának nyomvatartásával a megfelelő műveleti egység kimenetére vagy bemenetére húzható a kiválasztott anyag. A következő ábrán látható, hogy egy három műveleti egységet tartalmazó P-gráfban 4 anyag jellegű csomópont található: két nyersanyag (material_1 és 2), egy köztes anyag (material_3) és egy végtermék (material_4). A három műveleti egység közül az elsőnek (operatingunit_1) a bemenetén a material1, kimenetén a material_3 nevű anyag jelenik meg. Hasonlóan a többi műveleti egység is könnyen értelmezhető.



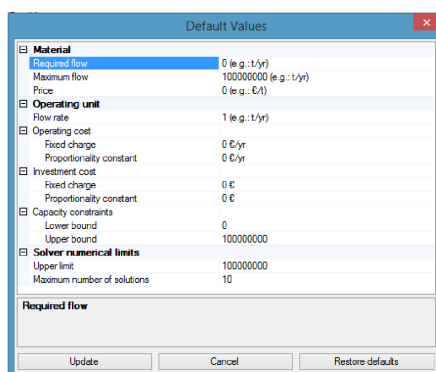
8.7. ábra. Műveleti egységek bemenetei és kimenetei a szoftverben

Természetesen módon merül fel az igény, hogy a PNS Draw-val megrajzolt P-gráfot a PNS Studio bemenetén fogadni tudja. Mint azt korábban láttuk egy megrajzolt gráf PNS formátumban is menthető, amely egyúttal a PNS Studio által elvárt input formátumnak felel meg. Így minden ott megrajzolt folyamat-gráf bemenete lehet a PNS Studio szoftvernek. A betöltött fájlok természetesen tetszőlegesen módosíthatóak: így új csomópontokat adhatunk a feladathoz, vagy akár törölhetünk is belőle.

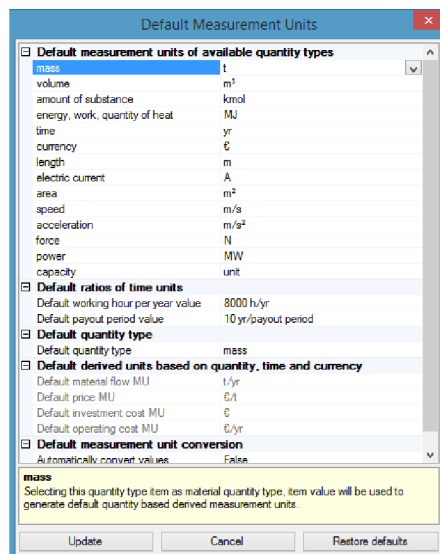
Egy gráf beolvasását vagy elemeinek megadását követően különféle PNS algoritmusokkal (MSG, SSG, SSG+LP, ABB) vizsgálható a feladat. Az MSG és SSG algoritmus nem követel meg semmilyen paraméterértéket, a gráf strukturális felépítése alapján végez műveleteket. Előbbi a maximális struktúrát, utóbbi az összes lehetséges megoldásstruktúrát állítja elő. Ezzel szemben az SSG+LP és ABB algoritmusok a költség szerinti optimális struktúra megkeresését végzik el. Míg az utóbbi algoritmus irányított kereséssel dolgozik, addig az előbbi minden lehetséges megoldásstruktúrára elvégzi a szükséges számításokat. E két algoritmus jellegénél fogva néhány paraméterérték (pl. kapacitáskorlátok, elvárt termékmennyiség, nyersanyagkorlátok) meglétét várja, így végzi el az optimalizálást.

Néhány előre beállított értéket a szoftver már tartalmaz, ezeket természetesen szabadon módosíthatjuk. Mindezt az „Options” menüpont „Default Values ...” menüpontban tehetjük meg, ahogy azt az ábra is mutatja. Külön felhívjuk a figyelmet a „Maximum number of solutions” beállítási lehetőségre, ahol - mint annak neve is mutatja – megadhatjuk, hogy az első hány darab legjobb megoldást szeretnénk a szoftverrel kiszámoltatni. Így tehát arra is van lehetőség, hogy nem csak az optimális megoldást jelenítsük meg, hanem megvizsgáljuk a 2., 3., vagy általában az N . legjobb megoldás részleteit is.

Ugyancsak az „Options” menüpontban találunk arra módot, hogy az alapértelmezett mértékegységeket változtassuk. Ezt mutatja a 8.9-es ábra.

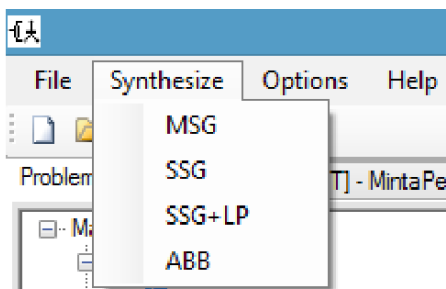


8.8. ábra. Alapértelmezett értékek beállítása a szoftverben



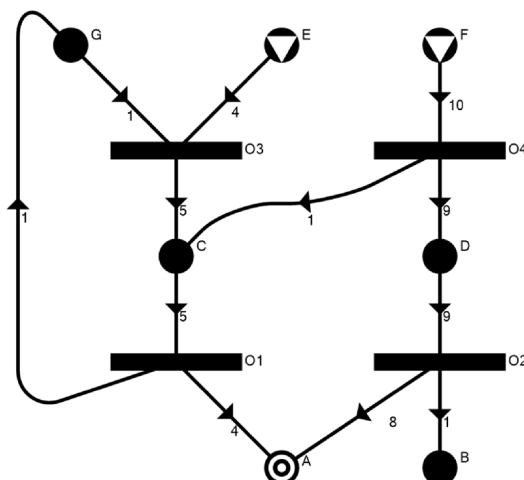
8.9. ábra. Mértékegységek rögzítése a szoftverben

A PNS algoritmusok futtatására a „Synthesize” menüpont ad lehetőséget. Mint az látható a 8.10-es ábrán a maximális struktúra generálására itt található az MSG algoritmus, leszámíthatók a lehetséges megoldásstruktúrák az SSG algoritmus segítségével, és optimalizálásra is mód nyílik az SSG+LP illetve az ABB algoritmusok segítségével.



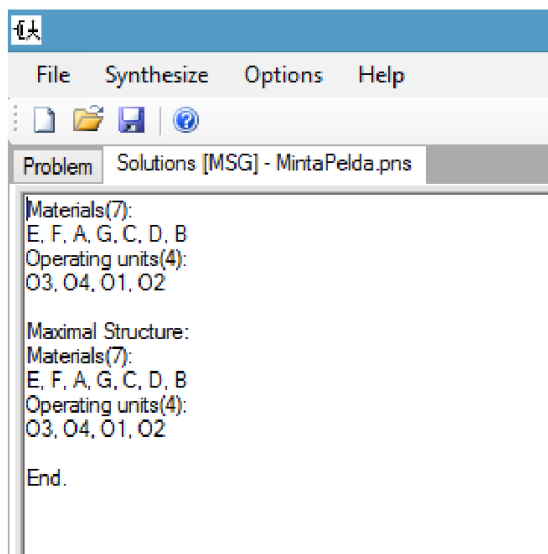
8.10. ábra. A Synthesize menü

Mintaként tekintünk a 8.11-es ábrán látható P-gráfot.



8.11. ábra. Egy mintapélda gráfja

Az MSG algoritmust futtatva a gráfon a megoldó szoftver a 8.12-es ábrán látható kimenetet adja.



8.12. ábra. Az MSG algoritmus kimenete a mintapéldában megadott gráfra

Látható, hogy a bemenetként megadott P-gráf egyúttal maximális struktúrája is a feladatnak, hiszen abban megtalálható az összes anyag és műveleti egység.

A kombinatorikusan lehetséges struktúrák száma 6, amelyet az SSG algoritmus futási eredményéből olvashatunk ki (8.13-as ábra).

Szövegesen ez az alábbi:

Materials(7):
E, F, A, G, C, D, B
Operating units(4):
03, 04, 01, 02

Maximal Structure:
Materials(7):
E, F, A, G, C, D, B
Operating units(4):
03, 04, 01, 02

Solution structure #1:
Materials(4):
E, A, G, C
Operating units(2):
03, 01

Solution structure #2:
Materials(5):
F, A, G, C, D
Operating units(2):
04, 01

Solution structure #3:
Materials(6):
E, F, A, G, C, D
Operating units(3):
03, 04, 01

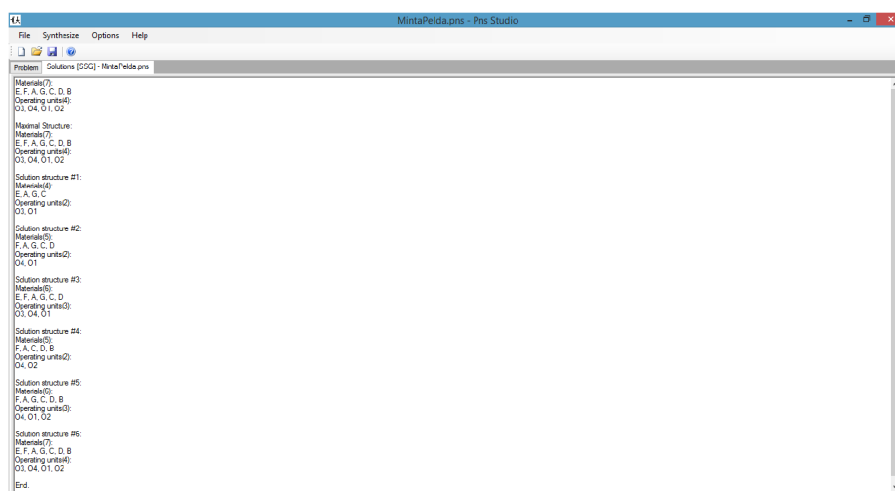
Solution structure #4:
Materials(5):
F, A, C, D, B

Operating units(2):
04, 02

Solution structure #5:
Materials(6):
F, A, G, C, D, B
Operating units(3):
04, 01, 02

Solution structure #6:
Materials(7):
E, F, A, G, C, D, B
Operating units(4):
03, 04, 01, 02

End.



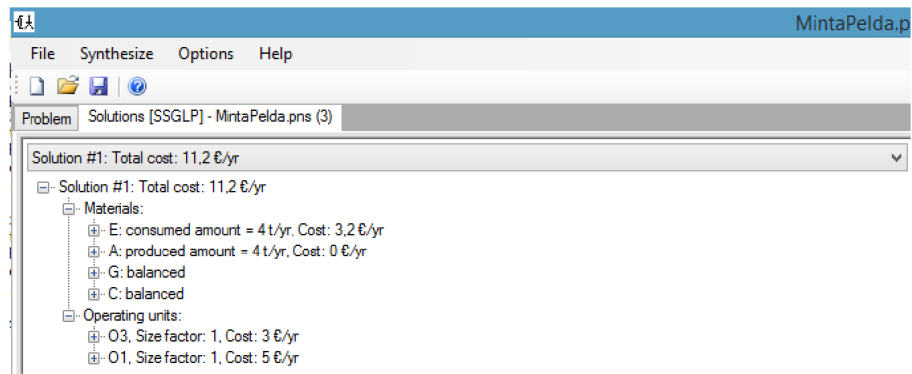
8.13. ábra. Az SSG algoritmus kimenete a mintapéldában megadott gráfra

Az SSG+LP és ABB algoritmusok futtatásához néhány paraméterérték bevezetése szükséges. A műveleti egység költségparamétereit a 8.1-es táblázat értékei mutatják.

Műveleti egység	Fixköltség	Arányos költség
O1	4	1
O2	3	1
O3	2	1
O4	2	0.5

8.1. táblázat. Műveleti egységes költségparamétereit

Az „E” nyersanyag ára legyen 0.8 EUR, míg az „F” 1.6 EUR-ba kerül. Az előbbi nyersanyagból korlátos készlettel rendelkezünk, csupán 10 egységnyi áll rendelkezésre. Ugyanakkor a termékből legalább 4 egységnyiit elő kell állítanunk. Vizsgáljuk meg, hogy ilyen feltételek és a P-gráfon feltüntetett flow értékek mellett miként teljesíthető minimális költséggel a termelési feladat. Megoldva a feladatot a két megoldó valamelyikével megkapjuk a feladat optimális megoldását (8.14-es ábra), és láthatók az abban szereplő műveleti egységek és költségértékek is. Mint az az ábráról leolvasható ott az optimális megoldást láthatjuk, de a legördülő listából kiválaszthatjuk akár a 2. vagy 3. legjobb megoldást is.



8.14. ábra. Az optimális megoldás részletei

Mint azt a jelen fejezetben bemutattuk a PNS Draw és PNS Studio programcsomagok segítségével könnyedén modellezhetők, reprezentálhatók és vizsgálhatók P-gráf alapokon megadott folyamathálózat-szintézis feladatok, megismerve így azok strukturális és egyéb jellemzőit.

9. FEJEZET

PNS algoritmusok implementálása

A PNS algoritmusoknak alapvetően két fontos komponense van. Egyik a folyamatok struktúrájának elemzése egy P-gráfon, másik a paraméteres modell feltételeinek teljesülés vizsgálata. A strukturális elemzéseket tipikusan halmazműveletekkel fogalmazzuk meg, ahogy a szintézis feladat strukturális része és egy P-gráf szerkezete is halmazokkal adott. A paraméteres modell megoldhatóság vizsgálatára pedig egy lineáris programozási (LP) modellt használunk.

9.1. Halmazok implementációja

A strukturális elemzések alapját halmazműveletek adják, ezért ezek végrehajtásának hatékonysága kulcs kérdés egy PNS algoritmus implementációjában. A legtöbb magas szintű programozási nyelv biztosít olyan adatszerkezeteket, melyek halmazként használhatóak, például C++-ban az STL set osztály. Ezek hatékonyságának korlátja egyrészt, hogy maga az adatszerkezet tetszőleges objektumok tárolására készült oly módon, hogy azok valamely rendezés szerint (< operátor alapján) kikereshetőek legyen. Így a megvalósításuk gyakorlatban valamilyen keresőfát jelent. A keresőfák egy-egy elem beillesztését és kivételét hatékonyan támogatják, ám egy teljes fa, vagy azzal modellezett halmaz másolása elemenként, az elemek közti kapcsolatok újraépítésével tud csak megvalósulni. A feladat még komplexebb, ha közben egy másik fa elemeit is figyelembe kell venni, például két halmaz metszetének kialakításához.

9.1.1. Objektumok sorszámozása

A PNS megoldó algoritmusok implementálása során arra legjobb gyakorlatra jutottunk, hogy a halmazműveleteket úgy célszerű kialakítani, hogy a halmazban szereplő objektumokat (például anyagok, műveleti egységek) nem maga a halmaz tárolja, hanem egy indexelhető tároló (például C++-ban STL vector), a halmaz pedig csak a benne levő elemek indexét. A halmazok efféle leírásának egyértelműségéhez a teljes programban biztosítani kell, hogy egyazon objektum mindig ugyanazt az azonosítót kapja. Ennek két kulcsa van. Egyik, hogy a sorszám hozzárendelést egy globális (statikus) függvény végezze az objektumok valamilyen egyértelmű rendezése alapján. Mivel különböző osztályba tartozó objektumok nehezen hasonlíthatók egymáshoz, szükség lehet arra, hogy a rendező függvény egy minta (template) szerint bármilyen osztályba tartozó objektumokat tudjon sorszámozni, akár osztályonként más-más módon. A 9.1-es ábra egy sorszámozó osztály deklarációját mutatja C++-ban.

A sorszámozó egyrészt sorszám szerint azonnal visszaad egy objektumot egy indexelhető tárolóból (a példában vector) és tetszőleges objektumban ki tudja keresni a sorszámát rendezés szerint egy asszociatív tárolóból (a példában map). Fontos felismerni, hogy egy objektumnak már létezik azonosítója, nehogy többet is generáljunk neki. Ezért a sorszámok kiosztásához használt rendezésben olyan attribútumait érdemes figyelembe venni az objektumnak, melyek egyrészt nem változnak

```
template <class elem>
class indexer{
public:
    typedef map<elem,unsigned int> mymap;
    typedef typename mymap::iterator mymapiterator;
    typedef vector<mymapiterator> myvector;
private:
    static myvector v;
    static mymap m;
public:
    const elem& operator[](const unsigned int i) const;
    unsigned int Find(const elem& e) const;
    unsigned int FindOrAdd(const elem& e);
    unsigned int size() const {return v.size();};
    unsigned int end() const {return size();};
};
```

9.1. ábra. Objektum sorszámozó osztály deklarációja C++-ban halmazműveletekhez

(legalábbis egy megoldó aktuális futása során), másrészt egyértelműek. Erre a célra például az objektumok azonosítójának (névének) lexikografikus rendezése alkalmas.

9.1.2. Egy elem – egy bit

Miután sikerült objektumokhoz sorszámot rendelni, akkor sorszámmal azonosított elemek halmazainak leírása már nagyon hatékony lehet. Ennek példája az úgynevezett bitvektoros megvalósítás. A bitvektor gyakorlatban nem más, mint egy előjel nélküli egész szám (C++-ban például unsigned int), melynek minden bitje külön állítható. Halmazok leírásában egy-egy bit azt jelenti, hogy az adott sorszámu elem szerepel-e a halmazban. Ha például egy 1. és 4. elem szerepel a halmazban akkor az egész szám értéke: binárisan 10010 decimálisan $2^1 + 2^4 = 2 + 16 = 18$. Ha 1-es helyi értéket nem akarjuk elveszíteni, akkor érdemes a sorszámozónak a 0 sorszámot is kiosztania, ami egybevág például azzal, hogy C++-ban egy indexelhető tároló első eleme a 0 indexű.

Ha egy objektumot beteszünk egy halmazba, akkor a sorszámanak megfelelő bit 1-re változik. Például ha a fenti 1-es és 4-es sorszámu elem mellé bekerül a 0 sorszámu, akkor a szám már binárisan $10010 + 1 = 10011$ lesz. Ha kiveszünk egy elemet, akkor pedig a sorszámanak megfelelő helyi érték nullává válik. Például a fenti 0-ás, 1-es, 4-es sorszámu elemet tartalmazó halmazt reprezentáló 10011 vektorból 10001 lesz a 1-es sorszámu elem kivétele után.

Egyetlen egész szám így tehát leírhat egy olyan elemszámú halmaz bármely részhalmazát, melynek legfeljebb annyi eleme van, amint szám bithossza. Egy szám bithossza megegyezik a futtató környezet (CPU, operációs rendszer) bithosszával. Manapság a személyi számítógépek processzorának bithossza általában 64 bit, az operációs rendszereké pedig 32 vagy 64 bit. Így, ha nem kívánunk egy halmazban 32 vagy 64 elemnél többet megkülönböztetni, akkor az elemek bármely halmaza leírható egyetlen egész számmal. Ha ennél több elemre van szükségünk, akkor több egészet kell egymás után fűznünk. Ezután minden művelet két lépésre bomlik: a megfelelő egész kikeresése, majd annak módosítása.

A 9.2-es ábrán látható eljárás először ellenőrzi, hogy az új elem beszúrásához kell-e az egészek tárolóját nagyobbra méretezni. Ha igen, akkor megteszi. Ezután kikeresi a megfelelő sorszámu egészet. Mivel ez egy 32 bites megvalósítás, így minden egész 32 elemet tárol, tehát a szükséges egész sorszáma, a kívánt elem sorszáma osztva 32-vel. Kettes számrendszerben úgy osztunk 32-vel, hogy a bináris számot eltoljuk 5 helyi értékkel jobbra, tehát levágjuk a 32-nél kisebb (1-es, 2-es, 4-es, 8-as, és 16-os) helyi értékeket. A megfelelő egész kiválasztása után biztosítanunk kell, hogy a betendő elem sorszámat jelentő helyi értéken biztosan 1 van. A módosítandó bit helye az egészen belül a az objektum sorszámanak 32-es maradéka, amit úgy kapunk hogy a számnak kivágjuk az utolsó 5 helyi értékét. A kivágáshoz a bitenkénti ÉS műveletet használjuk bináris 11111-el vagyis decimális 31-el. Ekkor megkaptuk a számon belül a módosítandó bit sorszámat. Erre pozícióra

```

smallset smallset::operator +=(const unsigned int element)
{
    if (element>maxelem)
        Resize(element+1);
    bitvector[element>>5]=bitvector[element>>5]|( 1<<(element&31) );
    // (element mod 32)-th bit of bitvector[element div 32]
    // number of used bytes = maxelem div 8 +1
    return *this;
}

```

9.2. ábra. Egyetlen elem hozzáadása bitvektorhoz C++-ban

```

smallset smallset::operator |(const smallset& set2) const{
    if (maxelem>=set2.maxelem){
        smallset value(*this);
        for(unsigned int i=0;i<=set2.lastword;i++){
            value.bitvector[i]=set2.bitvector[i];
        }
        return value;
    }
    else{
        smallset value(set2);
        for(unsigned int i=0;i<=lastword;i++){
            value.bitvector[i]=bitvector[i];
        }
        return value;
    }
}

```

9.3. ábra. Két halmaz unióját számomló bitműveletek C++-ban

egy 1-est eltolunk balra, végül bitenkénti VAGY művelettel hozzátesszük a bitvektorhoz. Korábbi példánkhoz visszatérve, ha a 10001 bitvektorhoz hozzá kívánjuk tenni a 3. bitet, akkor eltolunk egy 1-es 3-al, eredményül 1000-t kapva. Ezután a 10001 bitenként VAGY 1000 = 11001 biztosan olyan szám, melynek 3. bitje 1-es.

9.1.3. Halmazműveletek implementációja

Az elemek beszúrása és kivétele mellett alapvető halmazműveletek az unió és a metszet. A bitvektoros reprezentációban ezen műveleteknek a bitenkénti VAGY és ÉS műveletek felelnek meg. Tehát a 4-es, 1-es elemet reprezentáló 10010 és a 4-es, 2-es elemet reprezentáló 101000 számok bitenkénti VAGY műveletének eredménye 10110, ami pontosan a fenti két halmaz uniójának megfelelő 4-es, 2-es, 1-es elemeket azonosítja. Hasonlóan a 10010 és 10100 számok bitenkénti ÉS műveletnek eredménye 10000, vagyis pontosan a két halmaz metszetét jelentő egyetlen 4-es sorszámú elemet adja. A bitenkénti ÉS azt jelenti, hogy az eredmény számban pontosan az a bit lesz 1-es amely mindkét argumentumban szereplő számban is 1-es volt. A bitenkénti VAGY eredménye pedig egy olyan szám, ahol pontosan azok a bitek 1-esek, melyek legalább az egyik argumentumban szereplő számban is azok voltak.

A 9.3-as ábra két halmaz uniójának számítását reprezentáló bitműveleteket mutat. Az elágazás oka, hogy figyelembe kell venni két különböző méretű halmaz közül, hogy melyikben van a legnagyobb sorszámú elem, és az határozza meg a tárolási méretet.

A bitműveletek előnye, hogy egyetlen órajelciklus alatt végrehajthatóak, tehát, egy gigahertzes processzor egymilliárd ilyen műveletet tud végrehajtani minden másodpercben. Ezen bitvektoros implementációjú halmazok uniójának és metszetének számítása átlagosan körülbelül 3-4 nagyságrenddel (1000-10000-szer) gyorsabb, mint a keresőfában tárolt halmazoké.

9.2. LP megoldó könyvtárak

A paraméteres folyamatszintézis feladatban adott korlátozási feltételek kielégíthetőségének vizsgálatára és a folytonos paraméterek legjobb értékének megadására valamilyen folytonos optimalizáló könyvtárra van szükség. Mivel a jegyzetben bemutatott korlátozási feltételek és célfüggvények lineárisak, így egy Lineáris Programozási (LP) könyvtárt kell használnunk.

Számos ilyen könyvtár létezik különböző programozási nyelvekhez. Mivel egy PNS feladat megoldásához, a legjobb struktúra vagy struktúrák megtalálásához, néha több ezer vagy millió kisebb LP feladat megoldása szükséges, így olyat kell választanunk mely a strukturális keresést végrehajtó függvényekhez közvetlenül hozzákapcsolható.

9.2.1. Nyílt és zárt forráskódú megoldók

Ipari méretű feladatok esetén a gyakorlatban gyors futási idő hatékony implementációt igényel, általában valamilyen gép közeli nyelven (C/C++). C++-ban hozzáférhető LP könyvtárak közül nyílt forrásban hozzáférhető, Linux rendszerekben pedig az operációs rendszer részeként opcionálisan telepíthető. Ilyenek például a GLPK, az LP-Solve, és COIN-OR CLP. Utóbbi óriási fejlődésen ment át az elmúlt években, korszerű algoritmusokat tartalmaz, és főállású fejlesztő csapat gondozza, nagy multinacionális vállalatok támogatásának köszönhetően, mint például az IBM (<http://www.coin-or.org/members.html>). A nyílt forrásnak köszönhetően ezek a könyvtárak lefordíthatók bármely környezetben és könnyen illeszthetők szoftvereinkhez.

A piaci zárt forrású megoldók előnye, hogy jelentős áruknak köszönhetően gyakran nagyságrenddel több emberév van a fejlesztésükben, mint a nyílt forrásúakéban és ez a sebességükön is megmutatik. Nagy feladatok esetén például egy IBM-ILOG CPLEX megoldó nem ritkán 3-10-szer volt gyorsabb, mint a COIN-OR CLP ugyanarra a feladatra. Az IBM-ILOG CPLEX az egyik legjobb megoldó a világon, ráadásul kutatási célra ingyenesen hozzáférhető (www.ibm.com/university), de nem forráskódon, hanem deklarációk és előre lefordított függvénykönyvtárak formájában. Az előre lefordított függvénykönyvtárak hátránya, hogy adott operációs rendszer és fejlesztői környezet adott verziójához kötődik. Azt a konkrét operációs rendszert és fejlesztői környezetet kell megvásárolnunk hozzá más gyártóktól, amely támogatott, és előfordulhat, hogy óvatlan frissítés után, a fejlesztést nem tudjuk folytatni.

9.2.2. Ritka mátrixok

Az LP könyvtárak használatának praktikus korlátja még a fentiek mellett, hogy a LP feladatot megadó változók kapcsolatait leíró együtthatókat, úgynevezett ritka mátrixokban tárolják. Egy ritka mátrixban nem szerepel az összes elem, hanem csak egy-egy elem és annak a sorszáma. Ennek oka, hogy gyakorlati hálózatokat leíró modellekben egyes elemek általában kevés másik elemmel vannak kapcsolatban. Például egy műveleti egység csak azon műveleti egységekkel, melyekkel legalább egy közös anyagot állítják elő vagy használják fel. Ez a szám P-gráfban átlagosan nem több mint tíz, függetlenül, a feladat méretétől. Tehát minden műveleti egység száz elemű hálózatban átlagosan csak minden tizedik, ezer elemű hálózatban minden századik elemmel van kapcsolatban. Így a kapcsolatok minőségét leíró vektorok nagyságrendekkel csökkenthetők ritka tárolással.

A ritka tárolás viszont kényelmetlenné teszi az LP együttható mátrixainak feltöltését, mert nem azt kell megmondanunk, hogy hány elem van a hálózatban, hogy hanem hogy mennyi az elem-elem kapcsolatok teljes száma, majd ezeket megoldó által definiált sorrendben felsorolni. Ha a méretezést elrontjuk a megoldó memóriakezelési hibával kilép. Ennek következménye, hogy a folytonos modelleket érdemes alaposan megtervezni, és a nem nulla együtthatók számának kiszámítását pontosan leírni és ellenőrizni, bármilyen LP megoldót is kívánunk a szoftverünkbe építeni. Ebben segítségünkre lehet, hogy a feltöltött modelleket általában fájlba is írhatjuk és egy modellező környezetben, olvasható formában is elemezhetjük.