

# GLOBALIS OPTIMALIZÁLÁSI ALGORITMUSOK PNS FELADATOK MEGOLDÁSÁRA

DOKTORI (PhD) ÉRTEKEZÉS

Nagy Ádám

Témavezető: Dr. Friedler Ferenc

Veszprémi Egyetem  
Műszaki Informatikai Kar  
Informatikai Tudományok Doktori Iskola

2004

# GLOBALIS OPTIMALIZÁLÁSI ALGORITMUSOK PNS FELADATOK MEGOLDÁSÁRA

Értekezés doktori (PhD) fokozat elnyerése érdekében

Írta: Nagy Ádám

Készült a Veszprémi Egyetem Informatikai Tudományok Doktori Iskolája keretében

Témavezető: Dr. Friedler Ferenc

Elfogadásra javaslom (igen / nem)

(aláírás)

A jelölt a doktori szigorlaton .....%-ot ért el

Veszprém,

.....

a Szigorlati Bizottság elnöke

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: ..... (igen / nem)

(aláírás)

Bíráló neve: ..... (igen / nem)

(aláírás)

A jelölt az értekezés nyilvános vitáján .....%-ot ért el

Veszprém,

.....

a Bíráló Bizottság elnöke

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

.....

Az EDT elnöke

# Tartalomjegyzék

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| Táblázatok jegyzéke                                                        | vi        |
| Ábrák jegyzéke                                                             | viii      |
| Kivonat                                                                    | x         |
| Abstract                                                                   | xii       |
| Abstrakt                                                                   | xiv       |
| Köszönetnyilvánítás                                                        | xvi       |
| <b>1. Bevezetés</b>                                                        | <b>1</b>  |
| <b>2. Folyamatszintézis: PNS feladatok</b>                                 | <b>4</b>  |
| 2.1. Előzmények, a szakirodalom áttekintése . . . . .                      | 4         |
| 2.2. Matematikai modell PNS feladat leírására . . . . .                    | 5         |
| 2.2.1. Folyamat (Process) gráf . . . . .                                   | 5         |
| 2.2.2. A folyamatszintézis általános modellje . . . . .                    | 6         |
| 2.2.3. PNS feladat lineáris modellje . . . . .                             | 9         |
| 2.3. Kombinatorikus algoritmusok PNS feladatok megoldásához . . . . .      | 18        |
| <b>3. Szeparábilis konkáv optimalizálás PNS feladatok megoldására</b>      | <b>21</b> |
| 3.1. Szeparábilis konkáv programozás szakirodalmának áttekintése . . . . . | 22        |
| 3.2. Általános algoritmus . . . . .                                        | 23        |
| 3.2.1. Részprobléma . . . . .                                              | 23        |
| 3.2.2. Korlátozás . . . . .                                                | 23        |
| 3.2.3. Szétválasztás . . . . .                                             | 27        |
| 3.2.4. A keretalgoritmus elemzése . . . . .                                | 28        |
| 3.3. "Csúsztatott" partícionálási szabály . . . . .                        | 30        |
| 3.3.1. $\bar{x}$ -partícionálás . . . . .                                  | 30        |
| 3.3.2. $\bar{x}$ konvergencia, végeesség . . . . .                         | 31        |
| 3.3.3. A módszer viselkedése . . . . .                                     | 31        |

|         |                                                                                             |    |
|---------|---------------------------------------------------------------------------------------------|----|
| 3.3.4.  | "Csúsztatott" vágási módszer . . . . .                                                      | 32 |
| 3.3.5.  | Konvergenca, végesség . . . . .                                                             | 35 |
| 3.3.6.  | Az eredmény rövid összefoglalása . . . . .                                                  | 35 |
| 3.4.    | Maximális rés partícionálás . . . . .                                                       | 35 |
| 3.4.1.  | Vágási stratégia . . . . .                                                                  | 36 |
| 3.4.2.  | Konvergenca . . . . .                                                                       | 36 |
| 3.4.3.  | Az eredmény rövid összefoglalása . . . . .                                                  | 42 |
| 3.5.    | Egy elégséges optimalitási kritérium szeparábilis konkáv minimalizálási feladatra . . . . . | 42 |
| 3.5.1.  | A relaxált lineáris programozási feladat . . . . .                                          | 42 |
| 3.5.2.  | A relaxált lineáris programozási feladat optimalitási kritériuma . . . . .                  | 44 |
| 3.5.3.  | Elégséges optimalitási kritérium . . . . .                                                  | 46 |
| 3.5.4.  | A $\mathcal{H}$ halmazról . . . . .                                                         | 50 |
| 3.5.5.  | Az eredmény rövid összefoglalása . . . . .                                                  | 55 |
| 3.6.    | Érzékenységi vizsgálaton alapuló vágási stratégia . . . . .                                 | 55 |
| 3.6.1.  | Egy bázisváltozó költsége módosul . . . . .                                                 | 56 |
| 3.6.2.  | Több bázisváltozó költsége módosul . . . . .                                                | 58 |
| 3.6.3.  | Szétválasztási stratégia . . . . .                                                          | 63 |
| 3.6.4.  | Az algoritmus helyességének bizonyítása . . . . .                                           | 64 |
| 3.6.5.  | Az algoritmus működésének elemzése . . . . .                                                | 64 |
| 3.6.6.  | Az eredmény rövid összefoglalása . . . . .                                                  | 65 |
| 3.7.    | Kombinatorikusan gyorsított algoritmus . . . . .                                            | 65 |
| 3.7.1.  | Bevezetés . . . . .                                                                         | 65 |
| 3.7.2.  | Részproblémák . . . . .                                                                     | 66 |
| 3.7.3.  | Kiterjesztés . . . . .                                                                      | 67 |
| 3.7.4.  | Algoritmus . . . . .                                                                        | 69 |
| 3.7.5.  | A helyesség bizonyítása . . . . .                                                           | 73 |
| 3.7.6.  | Az eredmény rövid összefoglalása . . . . .                                                  | 73 |
| 3.8.    | N-legjobb megoldás . . . . .                                                                | 74 |
| 3.8.1.  | Bevezetés . . . . .                                                                         | 74 |
| 3.8.2.  | Algoritmus . . . . .                                                                        | 77 |
| 3.8.3.  | A helyesség bizonyítás . . . . .                                                            | 82 |
| 3.8.4.  | Példa . . . . .                                                                             | 83 |
| 3.8.5.  | Az eredmény rövid összefoglalása . . . . .                                                  | 83 |
| 3.9.    | Gyakorlati tapasztalatok . . . . .                                                          | 86 |
| 3.9.1.  | Generált tesztfeladatok . . . . .                                                           | 86 |
| 3.9.2.  | Valós ipari feladatok . . . . .                                                             | 88 |
| 3.10.   | Alkalmazás: ipari hőellátó rendszer optimális tervezése . . . . .                           | 92 |
| 3.10.1. | Gőzhálózat, kiindulási feltételek . . . . .                                                 | 92 |
| 3.10.2. | Alternatív megoldási lehetőségek . . . . .                                                  | 93 |

|                                                                        |            |
|------------------------------------------------------------------------|------------|
| 3.10.3. Módszer . . . . .                                              | 94         |
| <b>4. A folyamat- és hőcserélőhálózat együttes szintézise</b>          | <b>99</b>  |
| 4.1. Bevezetés . . . . .                                               | 99         |
| 4.1.1. Hőcserélőhálózat szintézis . . . . .                            | 99         |
| 4.1.2. Az integrált folyamat- és hőcserélőhálózat szintézise . . . . . | 100        |
| 4.2. A szakirodalom áttekintése . . . . .                              | 101        |
| 4.2.1. Általános HENS módszerek . . . . .                              | 101        |
| 4.2.2. Integrált folyamat- és hőcserélőhálózat szintézise . . . . .    | 103        |
| 4.3. A hP-gráf . . . . .                                               | 105        |
| 4.3.1. Az anyagpont kiterjesztése . . . . .                            | 106        |
| 4.4. Kiindulási adatok, halmazok . . . . .                             | 106        |
| 4.4.1. Hideg és meleg hőáramok . . . . .                               | 106        |
| 4.4.2. Rejtett hő . . . . .                                            | 107        |
| 4.4.3. A hideg áramok eltolása . . . . .                               | 107        |
| 4.4.4. Az elemi hőáramok . . . . .                                     | 108        |
| 4.4.5. A részhőáramok . . . . .                                        | 108        |
| 4.5. A matematikai modell . . . . .                                    | 109        |
| 4.5.1. Anyagponthoz tartozó matematikai modell . . . . .               | 109        |
| 4.5.2. Potenciális hőcserék meghatározása . . . . .                    | 111        |
| 4.5.3. Hőegyensúlyi feltételek . . . . .                               | 112        |
| 4.5.4. Hőcserélők költsége . . . . .                                   | 114        |
| 4.5.5. Egyesített matematikai modell . . . . .                         | 115        |
| 4.6. Az integrált módszer leírása . . . . .                            | 116        |
| 4.6.1. A korlátozó LP feladat tulajdonsága . . . . .                   | 116        |
| 4.7. Szemléltető példa . . . . .                                       | 117        |
| 4.7.1. Általános leírás . . . . .                                      | 117        |
| 4.7.2. Az 1. csúcs . . . . .                                           | 121        |
| 4.7.3. Az 1.1.1.2 csúcs . . . . .                                      | 130        |
| 4.8. Alkalmazás: HDA folyamat . . . . .                                | 134        |
| 4.9. Az eredmény rövid összefoglalása . . . . .                        | 135        |
| <b>5. Új tudományos eredmények</b>                                     | <b>138</b> |
| 5.1. Az értekezés témaköréből készült publikációk . . . . .            | 140        |
| <b>Irodalomjegyzék</b>                                                 | <b>144</b> |

# Táblázatok jegyzéke

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| 2.1. Költségparaméterek a műveleti egységekre . . . . .                   | 16  |
| 2.2. Paraméterek az anyagokra . . . . .                                   | 16  |
| 3.1. Lokálisan optimális struktúrák számossága . . . . .                  | 75  |
| 3.2. Általános módszer: 20 műveleti egység . . . . .                      | 87  |
| 3.3. Általános módszer: 40 műveleti egység . . . . .                      | 87  |
| 3.4. Általános módszer: 60 műveleti egység . . . . .                      | 87  |
| 3.5. Általános módszer: 80 műveleti egység . . . . .                      | 87  |
| 3.6. Kombinatorikusan gyorsított módszer: 20 műveleti egység . . . . .    | 89  |
| 3.7. Kombinatorikusan gyorsított módszer: 40 műveleti egység . . . . .    | 89  |
| 3.8. Kombinatorikusan gyorsított módszer: 60 műveleti egység . . . . .    | 89  |
| 3.9. Kombinatorikusan gyorsított módszer: 80 műveleti egység . . . . .    | 89  |
| 3.10. Általános módszer: Alpha (41 műveleti egység) . . . . .             | 91  |
| 3.11. Kombinatorikusan gyorsított módszer: Alpha (41 műveleti egység) . . | 91  |
| 3.12. Általános módszer: Denmark (35 műveleti egység) . . . . .           | 91  |
| 3.13. Kombinatorikusan gyorsított módszer: Denmark (35 műveleti egység)   | 91  |
| 4.1. A lehetséges műveleti egységek . . . . .                             | 118 |
| 4.2. Költségparaméterek a műveleti egységekre . . . . .                   | 118 |
| 4.3. Nyersanyagok . . . . .                                               | 119 |
| 4.4. Külső hideg, meleg források . . . . .                                | 119 |
| 4.5. Műveleti egység osztályok . . . . .                                  | 121 |
| 4.6. Lehetséges hőáramok az 1. csúcsnál . . . . .                         | 122 |
| 4.7. Rejtett hőforrások az 1. csúcsnál . . . . .                          | 122 |
| 4.8. Lehetséges elemi hőáramok az 1. csúcsban . . . . .                   | 123 |

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| 4.9. Lehetséges részhőáramok az 1. csúcsban . . . . .                     | 125 |
| 4.10. Anyagok hőkapacitásai . . . . .                                     | 125 |
| 4.11. Hőcseréhez kapcsolódó változók az 1. csúcsban . . . . .             | 125 |
| 4.12. A külső hideg és meleg energiához kapcsolódó változók a 1. csúcsban | 127 |
| 4.13. Hőegyensúlyi feltételek együtthatói az 1. csúcsban . . . . .        | 129 |
| 4.14. A költségfüggvényhez tartozó paraméterek az 1. csúcsban . . . . .   | 131 |
| 4.15. Potenciális hideg és meleg áramok az 1.1.1.2. csúcsban . . . . .    | 131 |
| 4.16. Rejtett hő előfordulása az 1.1.1.2. csúcsban . . . . .              | 131 |
| 4.17. Meleg és hideg elemi hőáramok az 1.1.1.2. csúcsban . . . . .        | 132 |
| 4.18. Részhőáramok az 1.1.1.2. csúcsban . . . . .                         | 133 |

# Ábrák jegyzéke

|                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------|----|
| 2.1. P-gráf. . . . .                                                                                                      | 7  |
| 2.2. A szemléltető példa P-gráf ábrázolása. . . . .                                                                       | 13 |
| 3.1. Általános algoritmus. . . . .                                                                                        | 24 |
| 3.2. A korlátozási eljárás. . . . .                                                                                       | 26 |
| 3.3. A partícionálási eljárás. . . . .                                                                                    | 28 |
| 3.4. $\varepsilon$ -vágás. . . . .                                                                                        | 33 |
| 3.5. Maximális rés partícionálás. . . . .                                                                                 | 36 |
| 3.6. Az integrálkülönbség folytonos és nemfolytonos esetekre. . . . .                                                     | 37 |
| 3.7. Partícionálás az $l_j^0$ -tól különböző helyen. . . . .                                                              | 39 |
| 3.8. Az $f^+$ függvény relaxációja az $[l_j^0, u_j^q]$ intervallumon. . . . .                                             | 40 |
| 3.9. Kombinatorikusan gyorsított algoritmus. . . . .                                                                      | 70 |
| 3.10. A partícionálási eljárás a kombinatorikusan gyorsított algoritmusban. . . . .                                       | 71 |
| 3.11. Vágás egy még nem döntött változón. . . . .                                                                         | 73 |
| 3.12. $N$ -legjobb megoldást meghatározó algoritmus vázlata. . . . .                                                      | 78 |
| 3.13. $N$ -listát módosító eljárás. . . . .                                                                               | 80 |
| 3.14. A szemléltető példa lokálisan optimális struktúrái és a hozzá tartozó költség. . . . .                              | 84 |
| 3.15. Denmark feladat P-gráf ábrázolása. . . . .                                                                          | 90 |
| 3.16. Alpha feladat P-gráf ábrázolása. . . . .                                                                            | 90 |
| 3.17. Az eredeti hőellátó rendszer sematikus ábrázolása. . . . .                                                          | 93 |
| 3.18. (a) Időben változó hőigény trend. (b) A diszkretizálás eredményeként kapott kumulatív hőteljesítmény trend. . . . . | 94 |
| 3.19. Egy adott hőteljesítményhez tartozó maximális struktúra. . . . .                                                    | 96 |



|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| 4.1. Hőáramok reprezentálása. . . . .                                                  | 105 |
| 4.2. Rejtett hő reprezentálása. . . . .                                                | 105 |
| 4.3. Az anyag típusú pont kiterjesztése . . . . .                                      | 106 |
| 4.4. Egy mesterséges műveleti egység. . . . .                                          | 110 |
| 4.5. Folyamatábra a szemléltető feladathoz. . . . .                                    | 117 |
| 4.6. A szemléltető példa hP-gráfja. . . . .                                            | 120 |
| 4.7. Az ABB algoritmus által előállított leszámlálási fa (a legrosszabb eset). . . . . | 122 |
| 4.8. Kaszkád diagram a jellemző hőáramokról az 1. csúcsban. . . . .                    | 123 |
| 4.9. Hőegyensúly az elemi áramokra. . . . .                                            | 127 |
| 4.10. Az $SSH_1$ rész hőáram lehetséges párosításai az 1. csúcsban. . . . .            | 128 |
| 4.11. Az $SSH_3$ rész hőáram lehetséges párosításai az 1. csúcsban. . . . .            | 128 |
| 4.12. Megoldás struktúra az 1.1.1.2 csúcsban. . . . .                                  | 132 |
| 4.13. Kaszkád diagram a jellemző hőáramokról az 1.1.1.2. csúcsban. . . . .             | 133 |
| 4.14. Az optimális struktúrához tartozó hőátvitelek. . . . .                           | 134 |
| 4.15. HDA folyamat diagramja. . . . .                                                  | 134 |
| 4.16. A HDA folyamat maximális struktúrája. . . . .                                    | 136 |
| 4.17. Az optimális hálózatot tartalmazó folyamatábra. . . . .                          | 137 |

# Kivonat

## Globális optimalizálási algoritmusok PNS feladatok megoldására

A dolgozatban a folyamathálózat-szintézis feladat fontos osztályainak megoldására alkalmas módszereket vezetünk be. A vizsgált feladatok NP teljesek, az általános megoldó módszerekkel gyakorlati feladatok megoldása belátható időn belül nem lehetséges, ezért a célunk olyan specializált megoldó módszerek létrehozása, amely kihasználja a PNS feladatok tulajdonságait.

Először a PNS feladatosztály konkáv célfüggvénnyel kibővített lineáris modelljét vizsgáljuk. A kapcsolódó modell egy lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat, melyek megoldására a korlátozás és szétválasztás módszerét választottuk.

A megoldás során felmerülő szétválasztás lépésre hatékony és egyszerűen kiszámítható partícionálási stratégiákat mutatunk be, amelyeket összehasonlítottunk korábban bevezetett módszerekkel.

A korábbi partícionálási eljárásokkal ellentétben a korlátozási lépésben használt közelítő lineáris programozási feladat érzékenységi vizsgálatával figyelembe tudjuk venni a konvex poliéder és a célfüggvény viszonyát. Így egy olyan optimalizálási eljáráshoz jutunk, amely kihasználja a PNS feladatokhoz kapcsolódó feltételrendszer tulajdonságát.

Egy P-gráf jól reprezentálja a PNS feladathoz kapcsolódó modell változói között lévő függőségi kapcsolatokat. A kombinatorikusan gyorsított algoritmus a változók lehetséges értékeinek partícionálásával párhuzamosan egy P-gráfon végez műveleteket,

és így javítja a módszer hatékonyságát.

Valós rendszerek esetében sokszor az adott típusú matematikai modellben nem lehet kifejezni a rendszer összes tulajdonságát. Kíváncos lenne, hogy az optimális megoldáson túl az első  $N$  legjobb megoldást is meghatározzuk, amelyekből a felhasználó további megfontolások alapján ki tudja választani a megfelelő struktúrát. Az általános megközelítésben az ilyen szuboptimális megoldások nem értelmezhetők, viszont a kombinatorikus eszközök lehetővé teszik, hogy megfelelően definiáljuk, és algoritmikus módszerekkel generáljuk az ilyen optimális megoldáshoz közeli megoldásokat.

Integrált folyamat- és hőcserélőhálózat-szintézis során a teljes folyamat- és hőcserélőhálózat szintézise azonos időben történik, ellentétben a szekvenciális módszerekkel, ahol a különböző szintézis lépések egymás után történnek. Könnyen látható, hogy az ilyen stratégia nem vezet kielégítő megoldáshoz, hiszen pl. az optimális folyamathálózat meghatározásakor figyelmen kívül hagyják a hőcserével kapcsolatos információkat.

A dolgozat bemutat egy vegyes egész lineáris programozási modellen alapuló módszert, amely a Friedler és munkatársai által bevezetett ABB algoritmuson alapul, és megoldja az integrált szintézis feladatot.

# Abstract

## Global optimization algorithms for solving PNS problems

Algorithmic methods for solving important classes of Process Network Synthesis (PNS) problem have been elaborated. The examined problems are NP hard, solving industrial size problems with general solution methods is not possible within a reasonable time. Therefore our aim is to create specialized algorithmic methods exploit the nature of PNS problems.

The PNS problem with concave cost function can be considered as a separable concave programming problem. Efficient partition strategies have been introduced for solving the separable concave programming problem. By the help of the sensitivity analysis of the relaxed linear programming problem, the relationship between the convex polyhedron and cost function can be taken into account. As a result an efficient optimization method utilizing the characteristics of the PNS problems have been elaborated.

A combinatorially accelerated algorithm has also been proposed. In line with the partitioning the feasible domain it performs operations on the corresponding P-graph and improve the efficiency.

Solving practical problems the first N-best solution is to be generated beyond the optimal solution thus enabling the user to select the suitable structure under further consideration. New methodology has been introduced which enables us to determine adequately and generate suboptimal solutions close to the optimal one.

During the integrated synthesis of process and heat-exchanger networks, the process synthesis and heat-exchanger-network synthesis are performed simultaneously. The dissertation introduces a method, which is based on the algorithm ABB introduced by Friedler and colleagues and solves the integrated synthesis problem.

# Abstrakt

## Globale Optimierungsalgorithmen zur Lösung von PNS-Problemen

Algorithmische Methoden zum Lösen wichtiger Klassen von Problemen der Prozess-Netzwerk-Synthese (PNS) wurden entwickelt. Die untersuchten Probleme sind NP hart. Es ist unmöglich, Probleme im industriellen Maßstab mit allgemeinen Lösungsmethoden in absehbarer Zeit zu lösen. Daher ist es unser Ziel, spezielle algorithmische Methoden, die die Natur der PNS-Probleme ausnutzen, zu entwickeln.

Ein PNS-Problem mit konkaver Kostenfunktion kann als ein trennbares konkaves Programmierungsproblem betrachtet werden. Effiziente Partitionsstrategien werden vorgestellt, um das trennbare konkave Programmierungsproblem zu lösen. Mit Hilfe einer Sensitivitätsanalyse des vereinfachten linearen Programmierungsproblems kann die Beziehung zwischen dem konvexen Polyeder und der Kostenfunktion betrachtet werden. Darauf aufbauend wurde eine effiziente Optimierungsmethode erarbeitet, die die charakteristischen Eigenschaften des PNS-Problems ausnutzt.

Ein kombinatorisch beschleunigter Algorithmus wurde ebenso vorgeschlagen. Parallel zum Partitionieren der ausführbaren Domain, werden Operationen am damit verbundenen P-Graph durchgeführt und somit die Effizienz gesteigert.

Neben der optimalen Lösung muss die erste N-beste Lösung generiert werden, die es dem Verwender ermöglicht, die entsprechende Struktur gemäss weiterer Überlegung zu wählen. Eine neue Methode wurde entwickelt, die das Optimieren und Erzeugen von suboptimalen Lösungen in der Nähe des Optimums ermöglicht.

Bei der integrierten Synthese von Prozess- und Wärmetauscher-Netzwerken werden die Prozesssynthese und die Wärmetauscher-Netzwerk-Synthese simultan durchgeführt. Die Arbeit stellt eine Methode vor, die auf dem ABB-Algorithmus basiert. Dieser ABB-Algorithmus wurde von Friedler und Kollegen eingeführt und löst das integrierte Synthese-Problem.

# Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani mindazoknak, aki lehetővé tették, hogy ez a dolgozat elkészüljön.

Elsősorban témavezetőmnek, Dr. Friedler Ferenc professzor úrnak tartozom köszönettel, aki irányította munkámat, és értékes szakmai tanácsokkal látott el.

Dr. L. T. Fan professzor úrnak szakmai támogatásáért vagyok hálás.

Kollégáimnak a Veszprémi Egyetem Számítástudomány Alkalmazása Tanszéken, akik munkám elvégzéséhez támogatást nyújtottak.

Szüleimnek és bátyámnak a biztatásért.



# 1. fejezet

## Bevezetés

Egy feldolgozó rendszerben a rendszer az anyagok kémiai, fizikai és biológiai transzformációján keresztül állítja elő a kívánt termékeket a meglévő nyersanyagokból kiindulva. A rendszerben lévő transzformációkat a műveleti egységek végzik, melyek bemeneti anyagokat alakítanak át kimeneti anyagokká. A műveleti egységek és a közöttük lehetséges kapcsolatok egy hálózattal reprezentálhatók. A kívánt termékek előállítása gyakran az adott hálózat egy részhálózata által történik. Egy hálózatnak sok részhálózata van, mely képes az adott termékek előállítására. A melléktermék-kibocsátás, energia- és nyersanyagfogyasztás nagyban függ magától a részhálózat kiválasztásától, ezért az optimális hálózat vagy struktúra kiválasztása mind gazdasági, mind környezetvédelmi okokból igen fontos. A folyamathálózat-szintézis (Process Network Synthesis, PNS) célja ezen optimális struktúrák meghatározása.

A hőcserélőhálózatok (HENS) szintézise az egyik legfontosabb területe a folyamat-tervezés tudományának. Az utóbbi időben az egyik legintenzívebben kutatott területek közé tartozik, több száz publikáció jelent meg e témában az elmúlt évtizedekben. Fontossága annak is tulajdonítható, hogy a vegyipari rendszerek működési költségeinek jelentős része az energiaköltség, ezen belül is a hőenergia, amelynek a hasznosítása kiemelten fontos.

Integrált folyamat- és hőcserélőhálózat szintézis során a teljes folyamat- és hőcserélőhálózat szintézise azonos időben történik, szemben a szekvenciális módszerekkel, amikor először meghatározzák magát az optimális folyamathálózatot és utána az optimális hőcserélőhálózatot. Könnyen látható, hogy ez nem vezet optimális megoldáshoz, hiszen az optimális folyamathálózat meghatározásakor figyelmen kívül

hagyják a hőcserével kapcsolatos információkat.

Célom a meglévő általános megoldó módszereknél hatékonyabb módszerek kidolgozása volt, amelyet a vizsgált feladatosztály speciális tulajdonságainak kihasználásával értem el. Az általam kifejlesztett módszereket a PNS feladatok bizonyos típusainak megoldására dolgoztam ki, amelyek más feladatok megoldására is kedvezően viselkednek.

## Szakirodalmi áttekintés

A szakirodalomban a szintézis feladatot a bevezetésben megfogalmazott általánosságban nem vizsgálják, csak a fontos feladattípusokat külön-külön. Ennek megfelelően a szakirodalom áttekintését fejezetenként tárgyaljuk.

## Saját eredményeim kiemelése

A dolgozat tartalmi részében mindvégig többes szám első személyt használok. Annak érdekében, hogy a dolgozatban elkülönítsem mások szakirodalomból ismert eredményeitől sajátjaimat, másokéra a szerzők nevével hivatkozom, sajátjaimat pedig minden fejezet és a dolgozat összefoglalásában egyes szám első személyben egyértelműen megfogalmazom.

## Jelölésjegyzék

A dolgozatban kicsi (általában indexelt), latin (illetve időnként görög), dőlt betűkkel  $x_i$ ,  $y_j$ ,  $\gamma, \beta, \dots$  (valós) számokat jelölünk. Kivételt képeznek ez alól az  $f$ ,  $f_j$ ,  $\bar{f}$ ,  $f'_k$ ,  $g$ ,  $g_j, \dots$  betűk, amelyeket függvények jelölésére használunk. Az  $i$ ,  $j$ ,  $k$  és  $l$  indexekre utalnak. Az  $m$  a  $(P)$  feladat feltételeinek, míg az  $n$  a változóinak a számára utalnak. A dőlt, latin nagybetűk  $A$ ,  $B, \dots$  mátrixokat, míg a kalligrafikus, latin nagybetűk  $\mathcal{A}$ ,  $\mathcal{P}, \dots$  halmazokat jelölnek. Valós elemű halmazok pontjai, egyenlőtlenségrendszerek változói, korlátjai illetve mátrixok oszlopai (sorai) mind-mind vektorok, jelölésükre vastag latin kisbetűket  $\mathbf{x}$ ,  $\mathbf{b}$ ,  $\mathbf{l}$ ,  $\mathbf{u}$ ,  $\mathbf{0}$ ,  $\mathbf{a}_j, \dots$  használunk. A

valós számok halmazát  $\mathbb{R}$ , az  $n$ -dimenziós euklideszi teret  $\mathbb{R}^n$ , míg az  $m \times n$ -es valós mátrixok halmazát  $\mathbb{R}^{m \times n}$  jelöli. A csupa egyesből álló vektort jelölje  $\mathbf{e}$ .

## 2. fejezet

# Folyamatszintézis: PNS feladatok

Jelen fejezetben a folyamatszintézis feladatosztály alapvető definícióit ismertetjük. Áttekintetjük a szakirodalmat, továbbá bemutatjuk a Friedler és munkatársai által bevezetett kombinatorikus technikát.

### 2.1. Előzmények, a szakirodalom áttekintése

A hálózatszintézis egyik fő nehézségét annak kombinatorikus jellege okozza, hogy a lehetséges alternatívák nagy száma miatt optimális struktúra meghatározása igen számításigényes. A [22] dolgozat becslése szerint egy átlagos hálózatszintézis feladat  $10^5 - 10^{10}$  alternatívát tartalmazhat.

A korábban kidolgozott matematikai módszereken alapuló eljárások nagy része általános matematikai programozási módszereket alkalmaznak a folyamathálózat-szintézis feladat megoldására: [24], [33], [55], [58], [98], amelyek a folyamathálózat-szintézis kombinatorikus jellege miatt általában egy vegyes egész matematikai programozási feladat megoldását jelenti, például Benders dekompozíció [58], külső közelítés [23], [24]. A [30] összefoglalást nyújt a folyamathálózat-szintézisben használt globális optimalizálási módszerekről. A [72], [90] áttekintő írásokat ad a hálózatszintézis témaköréből.

Egy ipari méretű feladat megoldása óriási számításigényű, az általános módszerek nem használják ki a folyamathálózat-szintézis feladat strukturáltságát, ezért hatékonyságuk igen alacsony. A módszerek egy része heurisztikus szabályokat alkalmaz a számítások gyorsítása érdekében, ez viszont nem garantálja a globális optimum

megtalálását.

Kifejezetten folyamathálózat-szintézis feladatok megoldására a [83] szerzői dolgoztak ki egy módszert. A tárgyalt módszer a feladat kombinatorikus tulajdonságait leíró logikai összefüggéseket használva javítja a keresés hatékonyságát. A [8]-ban a szerzők bebizonyították, hogy a gráftechnikán alapuló kombinatorikus technika (lásd [37]) levezethető a [83]-ban vázolt logikai kifejezésekből. Kombinatorikus módszereket, részben a branch-and-bound technikát, részben dinamikus programozást alkalmaz: [34], [35]. Módszerük azonban nem teljesen általános, a feladatot részproblémákra bontó szétválasztó lépés nem független a feladattól.

Az evolúciós módszerek [45] alapvető tulajdonsága, hogy a megoldás menete során egy lehetséges megoldásból kiindulva, azt javítások sorozatával fejlesztik, mindig megtartva egy aktuális megoldást és így érik el az optimális vagy a közel optimális megoldást. A javító lépések alapján határozzák meg az aktuális struktúrából kiindulva egy lépéssel elérhető összes lehetséges struktúrát, majd azok közül kiválasztják a legjobbat.

## 2.2. Matematikai modell PNS feladat leírására

Fejezetünkben Friedler és munkatársai által kidolgozott PNS definíciót és a hozzá kapcsolódó matematikai modellt ismertetem [36].

### 2.2.1. Folyamat (Process) gráf

A hálózatszintézis feladatok reprezentálására az általános irányított gráfok alkalmatlanok. Az alkalmatlanság abból ered, hogy az egyszerű gráf nem tesz különbséget az anyagok és a műveleti egységek között, így a rendszer leírása sokszor nem egyértelmű.

Legyen  $\mathcal{M}$  objektumok véges halmaza, általában ezek különböző anyagok vagy anyagok fajtái, melyek transzformációin keresztül érjük el a kívánt célt. Egy transzformációt úgy értelmezünk, mint valamilyen hozzárendelést, amely az  $\mathcal{M}$  egy részhalmazához rendeli az  $\mathcal{M}$  egy másik részhalmazát. A műveleti egységek reprezentálják ezen transzformációkat. A műveleti egységek az anyagokon keresztül kapcsolódnak egymáshoz, ezen kapcsolatok egy irányított páros gráffal írhatók le.

**Definíció 2.2.1** Legyen  $\mathcal{M}$  véges halmaz,  $\mathcal{O} \subseteq \wp(\mathcal{M}) \times \wp(\mathcal{M})$  és  $\mathcal{M} \cap \mathcal{O} = \emptyset$ , ahol  $\wp(\mathcal{M})$  az  $\mathcal{M}$  halmaz hatványhalmaza. Az  $(\mathcal{M}, \mathcal{O})$  párt folyamat gráfnak (Process graph) vagy P-gráfnak nevezzük. A csúcsok halmaza  $\mathcal{M} \cup \mathcal{O}$  az élek halmaza  $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ , ahol

$$\mathcal{A}_1 = \{(x, Y) : Y = (\alpha, \beta) \in \mathcal{O}, x \in \alpha\},$$

$$\mathcal{A}_2 = \{(Y, z) : Y = (\alpha, \beta) \in \mathcal{O}, z \in \beta\}.$$

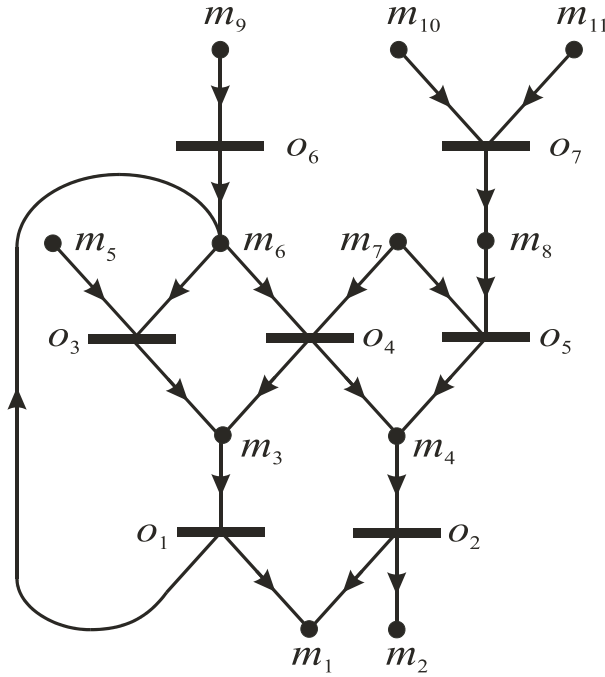
Az  $(\mathcal{M}', \mathcal{O}')$  P-gráf részgráfja az  $(\mathcal{M}, \mathcal{O})$ -nak azaz  $(\mathcal{M}', \mathcal{O}') \subseteq (\mathcal{M}, \mathcal{O})$ , ha  $\mathcal{M}' \subseteq \mathcal{M}$  és  $\mathcal{O}' \subseteq \mathcal{O}$ . Az  $(\mathcal{M}_1, \mathcal{O}_1)$  és  $(\mathcal{M}_2, \mathcal{O}_2)$  P-gráfok uniója azaz  $(\mathcal{M}_1, \mathcal{O}_1) \cup (\mathcal{M}_2, \mathcal{O}_2)$  legyen az  $(\mathcal{M}_1 \cup \mathcal{M}_2, \mathcal{O}_1 \cup \mathcal{O}_2)$  P-gráf. Ha  $(\alpha, \beta) \in \mathcal{O}$  akkor  $\alpha$  a bemeneti anyaghalmaz és  $\beta$  a kimeneti anyaghalmaz. Jelölje  $\omega^-(V)$ ,  $(\omega^+(V))$  a  $V$  csúcsba bemenő (kimenő) élek halmazát és  $\omega(V) = \omega^-(V) \cup \omega^+(V)$ . Legyen  $d^-(V) = |\omega^-(V)|$ ,  $d^+(V) = |\omega^+(V)|$  és  $d(V) = |\omega(V)|$ .

A 2.1 ábra egy P-gráfot ábrázol, az anyagpontokat  $(m_1, m_2, \dots, m_{11})$  körök, a műveleti egységeket  $(o_1, o_2, \dots, o_7)$  vízszintes vonal jelzi.

## 2.2.2. A folyamatszintézis általános modellje

Legyen  $\mathcal{P}$  az előállítandó anyagok (termékek) halmaza,  $\mathcal{R}$  a nyersanyagok halmaza, és  $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$  a rendelkezésre álló műveleti egységek halmaza. Továbbá legyen  $\mathcal{M} = \{m_1, m_2, \dots, m_l\}$  a műveleti egységekhez kapcsolódó anyagok véges halmaza. A következő feltételek teljesülnek:  $\mathcal{P} \cap \mathcal{R} = \emptyset$ ,  $\mathcal{P} \subseteq \mathcal{M}$ ,  $\mathcal{O} \subseteq \wp(\mathcal{M}) \times \wp(\mathcal{M})$  és  $\mathcal{M} \cap \mathcal{O} = \emptyset$ . Jelölje  $(\mathcal{M}, \mathcal{O})$  a problémához kapcsolódó P-gráfot.

Legyen adott az  $(\mathcal{M}, \mathcal{O})$  gráf egy részgráfja és legyen minden  $1 \leq j \leq n$ -re  $y_j = 1$ , ha a részgráf tartalmazza  $o_j$ -t és legyen  $y_j = 0$ , ha nem, így egy  $(y_1, y_2, \dots, y_n)$  bináris vektor egyértelműen meghatározza részgráfban lévő műveleti egységeket. Feltehetjük, hogy a részgráf nem tartalmaz izolált anyag típusú csúcsokat, így az  $\mathbf{y}$  indikátor vektor egyértelműen meghatározza a részgráfot.



2.1. ábra. P-gráf.

Vezessünk be a hálózatban lévő élekre és csúcsokra vonatkozó feltételeket. Legyen  $\mathcal{A} = \{a_1, a_2, \dots, a_r\}$  az élek halmaza és  $x_k$  ( $k = 1, 2, \dots, r$ ) az  $a_k$  élhez rendelt folytonos változó, amely az élen áthaladó anyag mennyiségét jelenti. A  $\varphi$  függvény rendelje hozzá az élhez, vagy az élek egy halmazához a megfelelő változók halmazát. A  $\varphi(a_{i_1}, a_{i_2}, \dots, a_{i_t}) = (x_{i_1}, x_{i_2}, \dots, x_{i_t})$  teljesül,  $\{a_{i_1}, a_{i_2}, \dots, a_{i_t}\} \subseteq \mathcal{A}$ -ra. Végül  $z_j$  jelölje az  $o_j$  ( $j = 1, 2, \dots, n$ ) műveleti egységhez rendelt változót, amely a műveleti egység méretét jellemzi.

Az  $o_j$  műveleti egységhez kapcsolódó feltétel illetve a költség a következő:

$$g_j(y_j, \varphi(\omega^-(o_j)), \varphi(\omega^+(o_j)), z_j) \leq 0, \quad j = 1, 2, \dots, n, \quad (2.2.1)$$

$$f_j(y_j, \varphi(\omega^-(o_j)), \varphi(\omega^+(o_j)), z_j), \quad j = 1, 2, \dots, n, \quad (2.2.2)$$

ahol  $f_j$  és  $g_j$  függvények általában differenciálhatók rögzített  $y_j$  értékre.

Hasonlóan az  $m_i$  anyagponthoz kapcsolódó feltételrendszer és költség a következő:

$$g'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) \leq 0, \quad i = 1, 2, \dots, l, \quad (2.2.3)$$

$$f'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))), \quad i = 1, 2, \dots, l.$$

A gyakorlatban  $g'$  és  $f'$  általában lineáris. A  $g'$  reprezentálja az anyagegyensúly feltételeket, illetve mennyiségi és minőségi követelményeket az adott anyagra. Az  $f'$  költségfüggvény lehet például a nyersanyagköltség stb.

### PNS feladat

Legyenek  $\mathcal{M} = \{m_1, m_2, \dots, m_l\}$  és  $(\mathcal{P}, \mathcal{R}, \mathcal{O})$  adottak, ahol  $\mathcal{P}$ ,  $\mathcal{R}$  és  $\mathcal{O}$  nemüres halmazok, továbbá  $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$ . Tegyük fel még, hogy  $\mathcal{P} \cap \mathcal{R} = \emptyset$ ,  $\mathcal{P} \subseteq \mathcal{M}$ ,  $\mathcal{R} \subseteq \mathcal{M}$ ,  $\mathcal{O} \subseteq \wp(\mathcal{M}) \times \wp(\mathcal{M})$ , és  $\mathcal{M} = \bigcup_{(\alpha, \beta) \in \mathcal{O}} (\alpha \cup \beta)$ . A PNS feladatot a következőképpen fogalmazhatjuk meg.

$$\begin{aligned} \min \quad & \sum_{j \in \{1, 2, \dots, n\}} f_j(y_j, \varphi(\omega^-(o_j)), \varphi(\omega^+(o_j)), z_j) + \\ & \sum_{i \in \{1, 2, \dots, l\}} f'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) \\ \text{feltéve, hogy} \quad & \end{aligned} \tag{2.2.4}$$

$$\begin{aligned} g_j(y_j, \varphi(\omega^-(o_j)), \varphi(\omega^+(o_j)), z_j) &\leq 0, \quad j = 1, 2, \dots, n \\ g'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) &\leq 0, \quad i = 1, 2, \dots, l \\ z_j &\leq My_j, \quad j = 1, 2, \dots, n \\ y_j \in \{0, 1\}, z_j &\geq 0, \quad j = 1, 2, \dots, n \end{aligned}$$

Itt  $M \in \mathbb{R}$  egy megfelelően választott nagy szám.

Az  $(\mathcal{M}, \mathcal{O})$  gráf egy részgráfja szorosan kapcsolódik a 2.2.4 modellt kielégítő megoldáshoz. A korábban említettek szerint a részgráf leírható egy  $(y_1, y_2, \dots, y_n)$  vektorral. Nyilvánvalóan nem minden  $(y_1, y_2, \dots, y_n)$ ,  $(y_i \in \{0, 1\}, i = 1, 2, \dots, n)$  vektor definiál valós folyamatot. A valódi folyamatot definiáló struktúrák rendelkeznek néhány közös kombinatorikus tulajdonsággal, amit explicite tartalmaz a 2.2.4 modell. Ezen tulajdonságokat figyelembe vételével az  $(\mathcal{M}, \mathcal{O})$  részgráfjainak halmaza redukálható a kombinatorikusan lehetséges megoldások halmazára. A redukálás mértékére jellemző, hogy például egy 35 műveleti egységből álló ipari feladatra a lehetséges részgráfok száma  $2^{35} \approx 3.4 \times 10^{10}$ , szemben a 3465 számú kombinatorikusan lehetséges struktúrák számával. A feladat részletes leírását a [39] tárgyalja.



**Definíció 2.2.2** Legyen adott a  $(\mathcal{P}, \mathcal{R}, \mathcal{O})$  hármas, továbbá legyen az  $(\mathcal{M}', \mathcal{O}')$   $P$ -gráf az  $(\mathcal{M}, \mathcal{O})$   $P$ -gráf részgráfja.  $(\mathcal{M}', \mathcal{O}')$  részgráf kombinatorikusan lehetséges struktúra (röviden lehetséges struktúra), ha a következő négy feltétel teljesül.

(S1)  $\mathcal{P} \subseteq \mathcal{M}'$ , azaz minden végtermék reprezentálva van a gráfban.

(S2)  $\forall x \in \mathcal{M}', d^-(x) = 0 \iff x \in \mathcal{R}$ , azaz egy anyag típusú csúcsnak pontosan akkor nincs bemenete, ha nyersanyagot reprezentál.

(S3)  $\forall u \in \mathcal{O}', \exists$  út  $[u, v]$ ,  $(\mathcal{M}', \mathcal{O}')$ -ban, ahol  $v \in \mathcal{P}$ , azaz minden műveleti egység típusú csúctól vezet út a terméket reprezentáló anyag típusú csúcsig.

(S4)  $\forall x \in \mathcal{M}', \exists(\alpha, \beta) \in \mathcal{O}'$  melyre  $x \in \alpha \cup \beta$ , azaz ha egy anyag típusú csúcs része a gráfnak, akkor kell lennie legalább egy bemenetének vagy legalább egy kimenetének egy műveleti egység típusú csúcs felől illetve felé.

A kombinatorikusan lehetséges struktúrák halmazát  $\mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ -val jelöljük.

Megemlítünk néhány összefüggést a megoldás struktúrákkal kapcsolatban, a bizonyítások [36]-ben megtalálhatóak.

**Definíció 2.2.3** Tegyük fel, hogy  $\mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O}) \neq \emptyset$ , akkor az összes kombinatorikusan lehetséges struktúra unióját jelöljük  $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O})$ -val. Azaz,

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

A  $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O})$  struktúrát maximális struktúrának nevezzük.

**Tétel 2.2.1** Az  $\mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$  halmaz zárt az unióra.

**Következmény 2.2.2**  $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O}) \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ .

### 2.2.3. PNS feladat lineáris modellje

Először a folyamatszintézis egyik alapmodelljét vezetjük be, ahol a feltételrendszer és a célfüggvény lineáris. Az alapmodell kiindulásnak tekinthető az általánosabb nemlineáris modell megoldása felé.

### Műveleti egység modell

Az előző fejezetben bemutatott általános PNS modell (2.2.4) egy-egy műveleti egység ( $o_k \in \mathcal{O}$ ) költsége a következő:

$$f_k(z_k, y_k) = a_k y_k + b_k z_k,$$

ahol a költségfüggvény működési és beruházási költségből tevődik össze:

$$a_k = Oa_k + \frac{Ia_k}{\text{megtérülési évek száma}}, \quad b_k = Ob_k + \frac{Ib_k}{\text{megtérülési évek száma}}$$

$Oa_k$  : a működési költség állandó része,

$Ob_k$  : a működési költség a műveleti egység méretétől függő része,

$Ia_k$  : a beruházási költség állandó része,

$Ib_k$  : a beruházási költség a műveleti egység méretétől függő része,

Mivel a költség független az illeszkedő élektől, így azt nem jelöltük a függvény paramétereinek között.

A (2.2.1) egyenletben szereplő  $g_k : \{0, 1\} \times \mathbb{R}^p \rightarrow \mathbb{R}^q$ -beli függvény, ahol  $p = d(o_k) + 1$ . A  $q$  természetesen függ a modelltől, jelen esetben  $q = d(o_k)$ .

Legyen  $a_s$  egy él, amely  $o_k$ -ra illeszkedik, azaz  $a_s = (m_i, o_k) \in \omega^-(o_k)$  (vagy  $a_s = (o_k, m_i) \in \omega^+(o_k)$ ),

$$g_{ks}(y_k, \varphi(\omega^-(o_k)), \varphi(\omega^+(o_k)), z_k) = 0, \quad a_s \in \omega(o_k)$$

A műveleti egység mérete és az illeszkedő élek folyamváltozói közötti kapcsolatot definiálja a következő képlet:

$$g_{ks}(y_k, \varphi(\omega^-(o_k)), \varphi(\omega^+(o_k)), z_k) = r_{ki}z_k - x_s, \quad a_s \in \omega(o_k) \quad (2.2.5)$$

Az  $r_{ki}$  a műveleti egységre jellemző paraméter. Nevezetesen  $r_{ki}$  az  $m_i$  anyag mennyisége, amelyet az  $o_k$  egységnyi kapacitással való működésekor fogyaszt  $a_s = (m_i, o_k) \in \omega^-(o_k)$ , illetve termel  $a_s = (o_k, m_i) \in \omega^+(o_k)$ .

### Anyagponthoz tartozó feltételek

A P-gráf általános modelljében egy anyaghoz rendelhető költség és feltétel. Legyen  $m_i \in \mathcal{M}$ , akkor az  $m_i$  anyagponthoz tartozó költség:

$$f'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) = \begin{cases} 0, & m_i \in \mathcal{M} \setminus \mathcal{R}, \\ C_i \sum_{a_s=(m_i, o_k) \in \omega^+(m_i)} x_s, & m_i \in \mathcal{R}, \end{cases}$$

ahol  $C_i$  jelenti az egységnyi tömegű  $m_i$  anyag költségét.

Az anyagokra vonatkozó feltételek (lásd korábban a (2.2.3) egyenletek) a köztes anyagokra meghatározzák az anyagegyensúlyt (azaz egy anyagból a fogyasztott mennyiség nem lehet több, mint a termelt mennyiség), továbbá a nyersanyagokra és termékekre vonatkozó korlátokat.

$$g'_i(\varphi(\omega^-(m_i)), g'_i(\varphi(\omega^+(m_i)))) = \sum_{a_j=(m_i, o_k) \in \omega^+(m_i)} x_j - \sum_{a_j=(o_k, m_i) \in \omega^-(m_i)} x_j + p_i - s_i$$

Felhasználva a műveleti egység modellt (lásd korábban a (2.2.5) egyenlet) a feltétel a következő formába írható:

$$g'_i(\varphi(\omega^-(m_i)), g'_i(\varphi(\omega^+(m_i)))) = \sum_{a_j=(m_i, o_k) \in \omega^+(m_i)} r_{ki} z_k - \sum_{a_j=(o_k, m_i) \in \omega^-(m_i)} r_{ki} z_k + p_i - s_i. \quad (2.2.6)$$

Az  $s_i > 0$  a rendelkezésre álló  $m_i \in \mathcal{R}$  nyersanyag mennyisége ( $s_i = 0$ , ha  $m_i \notin \mathcal{R}$ ) és  $p_i > 0$  az  $m_i \in \mathcal{P}$  termékekre vonatkozó legyártandó anyagmennyiség alsó korlátja ( $p_i = 0$ , ha  $m_i \notin \mathcal{P}$ ).

Hasonlóan felhasználva a műveleti egység modellt (lásd korábban a (2.2.5) egyenlet), a költség is a következő formába írható:

$$f'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) = \begin{cases} 0, & m_i \in \mathcal{M} \setminus \mathcal{R} \\ C_i \sum_{a_s=(m_i, o_k) \in \omega^+(m_i)} r_{ki} z_k, & m_i \in \mathcal{R}. \end{cases} \quad (2.2.7)$$

A PNS lineáris modelljéből kiküszöböltük az élekhez tartozó változókat. Összefoglaljuk modellünket:

$$\min \sum_{j \in \{1, 2, \dots, n\}} f_j(y_j, z_j) + \sum_{m_i \in \mathcal{R}} C_i \sum_{(m_i, o_k) \in \omega^+(m_i)} r_{ki} z_k$$

feltéve, hogy

$$\begin{aligned} \sum_{(m_i, o_k) \in \omega^+(m_i)} r_{ki} z_k - \sum_{(o_k, m_i) \in \omega^-(m_i)} r_{ki} z_k + p_i - s_i &\leq 0, \quad i = 1, 2, \dots, l \\ z_j &\leq M y_j, \quad j = 1, 2, \dots, n \\ y_j &\in \{0, 1\}, z_j \geq 0, \quad j = 1, 2, \dots, n \end{aligned} \quad (2.2.8)$$

Egy PNS feladatban a műveleti egységek az anyagpontokon keresztül kapcsolódnak egymáshoz, a kapcsolódás abban az értelemben lokális, hogy egy műveleti egység általában nincs kapcsolatban az összes többivel. Ez a tulajdonság nagyban kihat magára a matematikai modellre, hiszen a modellben a változók a műveleti egységekhez kapcsolódnak, a feltételek pedig a pontokhoz köthetők. Kijelenthetjük, hogy az együttható mátrix ritka, amely a megoldó módszerek szempontjából fontos.

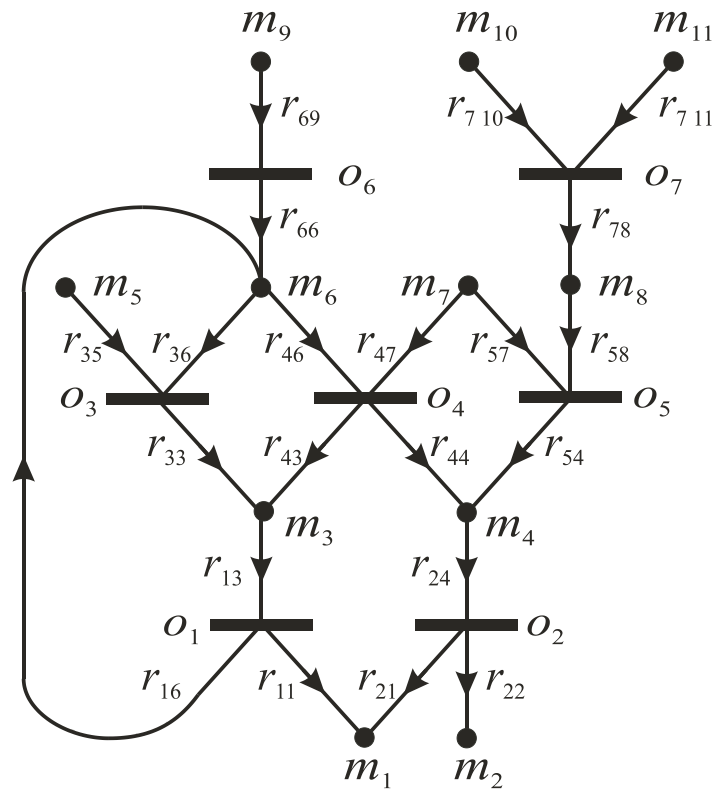
### Szemléltető példa

Tekintsük a következő példát.  $\mathcal{M} = \{m_1, \dots, m_{11}\}$ ,  $\mathcal{R} = \{m_5, m_7, m_9, m_{10}, m_{11}\}$ ,  $\mathcal{P} = \{m_1\}$ ,  $\mathcal{O} = \{o_1, o_2, \dots, o_7\}$ .

Műveleti egységek legyenek a következők:

$$\begin{aligned} o_1 &= (\{m_3\}, \{m_1, m_6\}), \quad o_2 = (\{m_4\}, \{m_1, m_2\}), \quad o_3 = (\{m_5, m_6\}, \{m_3\}), \\ o_4 &= (\{m_6, m_7\}, \{m_3, m_4\}), \quad o_5 = (\{m_7, m_8\}, \{m_4\}), \quad o_6 = (\{m_9\}, \{m_6\}), \\ o_7 &= (\{m_{10}, m_{11}\}, \{m_8\}). \end{aligned}$$

A P-gráf reprezentációt a 2.2 ábra mutatja. Az ábrán feltüntettük a műveleti egységekhez kapcsolódó  $r_{ij}$  paramétereket is.



2.2. ábra. A szemléltető példa P-gráf ábrázolása.

A 2.2.8 modell szerint a következő anyagegyensúly feltételeket kapjuk:

$$\begin{aligned}
-r_{11}z_1 - r_{21}z_2 + p_1 &\leq 0 & (m_1) \\
-r_{22}z_2 &\leq 0 & (m_2) \\
r_{13}z_1 - r_{33}z_3 - r_{43}z_4 &\leq 0 & (m_3) \\
r_{24}z_2 - r_{44}z_4 - r_{54}z_5 &\leq 0 & (m_4) \\
r_{35}z_3 - s_5 &\leq 0 & (m_5) \\
r_{36}z_3 + r_{46}z_4 - r_{16}z_1 - r_{66}z_6 &\leq 0 & (m_6) \\
r_{47}z_4 + r_{57}z_5 - s_7 &\leq 0 & (m_7) \\
r_{58}z_5 - r_{78}z_7 &\leq 0 & (m_8) \\
r_{69}z_6 - s_9 &\leq 0 & (m_9) \\
r_{710}z_7 - s_{10} &\leq 0 & (m_{10}) \\
r_{711}z_7 - s_{11} &\leq 0 & (m_{11})
\end{aligned}$$

Az egyenlőtlenségek mellett zárójelben feltüntettük, hogy mely anyagra vonatkoznak. Mielőtt értelmeznénk az egyenleteket, rendezzük át őket úgy, hogy a negatív előjelű tagokat vigyük át az egyenlőtlenség másik oldalára.

$$\begin{aligned}
p_1 &\leq r_{11}z_1 + r_{21}z_2 & (m_1) \\
0 &\leq r_{22}z_2 & (m_2) \\
r_{13}z_1 &\leq r_{33}z_3 + r_{43}z_4 & (m_3) \\
r_{24}z_2 &\leq r_{44}z_4 + r_{54}z_5 & (m_4) \\
r_{35}z_3 &\leq s_5 & (m_5) \\
r_{36}z_3 + r_{46}z_4 &\leq r_{16}z_1 + r_{66}z_6 & (m_6) \\
r_{47}z_4 + r_{57}z_5 &\leq s_7 & (m_7) \\
r_{58}z_5 &\leq r_{78}z_7 & (m_8) \\
r_{69}z_6 &\leq s_9 & (m_9) \\
r_{710}z_7 &\leq s_{10} & (m_{10}) \\
r_{711}z_7 &\leq s_{11} & (m_{11})
\end{aligned}$$

Vegyük észre, hogy egy  $r_{kj}z_k$  szorzat az  $o_k$  műveleti egység által egységnyi idő alatt fogyasztott vagy termelt anyagmennyiséget jelenti. Az egyenletek bal oldalán az adott anyagból elfogyasztott mennyiség, a jobb oldalon pedig a termelt mennyiség szerepel.

A paraméterek meghatározása során a dimenziókat megfelelően kell megválasztani. A mennyiségek egy időtartamra vonatkoznak, például a nyersanyagra vonatkozó felsőkorlát  $t/\text{év}$ -ben lehet megadva, vagy a műveleti egységekre a működési költség is egységnyi mérethez és egységnyi időtartamhoz van meghatározva.

Egy gráfban négyféle anyagtípusú csúcs létezik: nyersanyag, termék, köztes anyag, melléktermék. Csoportonként nézzük végig ez egyes egyenlőtlenség típusokat.

Melléktermékre ( $m_2$ ) vonatkozó egyenletek általában elhagyhatók, kivéve ha nincs valamilyen kikötés a kibocsájtására vonatkozóan. A melléktermékek halmaza könnyen meghatározható (olyan anyag típusú pontok, amelyek nem termékek és csak bemenő éleket tartalmaznak), így ezen egyenleteket a modellgenerálás folyamán figyelmen kívül hagyhatjuk. Látszik, hogy az ( $m_2$ ) egyenlet valóban elhagyható. Triviálisan teljesül, hiszen  $r_{22}$  és  $z_2$  is nemnegatív.

A termékekre vonatkozó egyenleteknél a bal oldalon szerepel az adott anyagból a műveleti egységek által felhasznált anyagmennyiség és a  $p_i$  legyártandó mennyiség. A  $p_i$ -t úgy tekinthetjük, mint egy fogyasztást, amit úgymond ki kell vinnünk a rendszerből. Megjegyezzük, hogy az axiómák (2.2.2 definíció) nem tiltják a termék fogyasztását. Például az ( $m_1$ ) egyenlet garantálja, hogy a  $o_1$  és  $o_2$  által termelt  $m_1$  anyag mennyisége nagyobb vagy egyenlő, mint a kívánt  $p_1$  mennyiség.

Köztes anyag esetén a feltétel szerint nem fogyasztunk többet egy anyagból, mint amennyit termelünk. Az ( $m_3$ ) egyenletben a baloldalon  $r_{13}z_1$  mennyiség az  $o_1$  által adott idő alatt  $m_3$ -ból fogyasztott mennyiséget jelenti. A jobb oldal az  $o_3$  és  $o_4$  által termelt  $m_3$  mennyiséget jelöli.

A nyersanyaghoz kapcsolódó egyenleteknél az egyenlőtlenség jobb oldalán jelenik meg az  $s$  mennyiség, ami egy külső forrást jelent. A korábban tárgyalt axiómák (2.2.2 definíció) szerint nyersanyagot nem termelünk, azaz a jobb oldalán csak a rendelkezésre álló  $s$  mennyiség állhat. A bal oldalán az adott anyagból a műveleti egységek által időegység alatt felhasznált anyagmennyiség szerepel. Például az ( $m_7$ ) feltétel kimondja, hogy az  $o_4$  és az  $o_5$  által fogyasztott  $m_7$  anyag  $r_{47}z_4 + r_{57}z_5$  mennyisége nem lehet több, mint a rendelkezésre álló  $s_7$ .

Rendeljünk értékeket a megfelelő paraméterekhez. Itt most év az időegység és a megtérülési évek száma 3. A 2.1 és 2.2 táblázatokban a paramétereket ismertetjük, a nem jelölt paramétereket tekintsük zérusnak.

2.1. táblázat. Költségparaméterek a műveleti egységekre

| Műveleti<br>egység | Beruházási költség |         | Működési költség |         | a    | b   |
|--------------------|--------------------|---------|------------------|---------|------|-----|
|                    | Állandó            | Változó | Állandó          | Változó |      |     |
| $o_1$              | 1500               | 210     | 250              | 100     | 750  | 170 |
| $o_2$              | 1800               | 270     | 1000             | 100     | 1600 | 190 |
| $o_3$              | 900                | 180     | 600              | 200     | 900  | 260 |
| $o_4$              | 3000               | 90      | 1500             | 120     | 2500 | 150 |
| $o_5$              | 900                | 570     | 800              | 200     | 1100 | 390 |
| $o_6$              | 750                | 120     | 500              | 130     | 750  | 170 |
| $o_7$              | 600                | 120     | 120              | 100     | 320  | 140 |

2.2. táblázat. Paraméterek az anyagokra

| Anyag    | $p$ | $s$   | $C$  |
|----------|-----|-------|------|
| $m_1$    | 100 | 0     | 0    |
| $m_5$    | 0   | 10000 | 700  |
| $m_7$    | 0   | 10000 | 1100 |
| $m_8$    | 0   | 10000 | 400  |
| $m_{10}$ | 0   | 10000 | 500  |
| $m_{11}$ | 0   | 10000 | 700  |



Az  $RK$  mátrix tartalmazza az  $r_{ij}$  ( $i = 1, \dots, 7$ ,  $j = 1, \dots, 11$ ) paramétereket.

$$RK = \begin{pmatrix} 2 & 0 & 3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0.3 & 1.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1.2 & 0.8 \end{pmatrix}$$

A következő programozási feladatot kapjuk:

$$\begin{aligned} \min & 750y_1 + 170z_1 + 1600y_2 + 190z_2 + 900y_3 + (260 + 700)z_3 + \\ & 2500y_4 + (150 + 1870)z_4 + 1100y_5 + (390 + 2200)z_5 + \\ & 750y_6 + (170 + 400)z_6 + 320y_7 + (140 + 1160)z_7 \end{aligned}$$

feltéve, hogy

$$\begin{pmatrix} -2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & -2 & -1 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & -1 & -3 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0.3 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1.7 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{pmatrix} \leq \begin{pmatrix} -100 \\ 0 \\ 0 \\ 10000 \\ 0 \\ 10000 \\ 0 \\ 10000 \\ 10000 \end{pmatrix}$$

$$z_i \leq My_i$$

$$z_i \geq 0, y_i \in \{1, 0\}, i = 1, \dots, 7$$

Megoldva a feladatot az optimális struktúra az  $o_1$ ,  $o_3$ ,  $o_6$  műveleti egységeket tartalmazza (azaz  $y_1 = 1$ ,  $y_3 = 1$ ,  $y_6 = 1$ ). A méretük,  $z_1 = 50.0$ ,  $z_3 = 75.0$ ,  $z_6 = 100$ .

### PNS mint konkáv szeparábilis programozási feladat

A lineáris modell általánosabb esete, amikor a műveleti egység költségét egy konkáv függvénnyel írjuk le. Gyakorlatban a műveleti egységek mérettől függő fajlagos költsége a méret növelésével csökken, ami egy a konkáv függvényekre jellemző tulajdonság. A konkáv függvények használatával modellünk jobban leírja a valós költségeket, mint a korábban bevezetett lineáris költségfüggvény:

$$f_k(z_k, y_k) = a_k y_k + b_k z_k^\alpha, \quad (2.2.9)$$

ahol  $\alpha \in \mathbb{R}$ ,  $0 \leq \alpha \leq 1$ ,  $a_k \in \mathbb{R}$ ,  $a_k \geq 0$ ,  $b_k \in \mathbb{R}$ ,  $b_k \geq 0$ . Gyakorlatban az  $\alpha = 0.6$  használatos. Későbbiekben ezen modell lesz vizsgálatunk egyik fő tárgya.

## 2.3. Kombinatorikus algoritmusok PNS feladatok megoldásához

Fejezetünkben Friedler és munkatársai által kidolgozott kombinatorikus alap algoritmusokat mutatjuk be.

### MSG algoritmus

A kombinatorikusan lehetséges struktúrák halmaza véges, és zárt az unióra (lásd 2.2.1 tétel), ha ez a halmaz nem üres, akkor létezik maximális struktúra, melynek minden kombinatorikusan lehetséges struktúra részhalmaza. A Friedler és munkatársai által kidolgozott MSG algoritmus [38] a maximális struktúrát,  $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O})$ -t generálja polinomiális időben.

### SSG algoritmus

Az SSG algoritmus [39] lehetővé teszi az összes kombinatorikusan lehetséges struktúra generálását. A kombinatorikusan lehetséges struktúrák generálásához alternatív döntéseket vagy döntések sorozatát kell végrehajtanunk. A döntések műveleti egységek bevételeiről vagy kizárásáról történnek.

Minden terméket legalább egy műveleti egységnek kell gyártani. Hasonlóan legalább egy műveleti egységnek kell gyártani egy olyan anyagot, amelyet egy korábbi döntés során bevett műveleti egység fogyaszt. A nyersanyagokra ez természetesen nem vonatkozik.

A döntések során vigyáznunk kell arra, hogy az egyszer már kizárt műveleti egységeket egy másik döntés során már nem választhatjuk be. Az inkonzisztens döntések elkerülése érdekében a műveleti egységeket három osztályba soroljuk: a beválasztott műveleti egységek halmaza, a kizárt műveleti egységek halmaza és a még nem döntött műveleti egységek halmaza.

Aktív halmaznak hívjuk az anyagok azon halmazát, amelyek előállításáról dönteni kell. Kezdetben az aktív halmaz a termékeket tartalmazza. Egy döntés során meghatározzuk az anyagot előállítani képes műveleti egységek közül azokat, amelyek az adott struktúrában előállítják az anyagot. Az így kiválasztott műveleti egységeket beválasztjuk a struktúrába, a többit pedig kizárjuk.

Döntés után frissítjük az aktív halmazt. Azokat a nem nyersanyagokat, amelyeket már beválasztott műveleti egység fogyaszt és még nem volt rajtuk döntés, hozzáadjuk az aktív halmazhoz.

Azokat az anyagokat, amelyek gyártásáról már döntöttünk, kivesszük az aktív halmazból. Amikor a konzisztens döntések eredményeként az aktív halmaz üressé válik, a beválasztott műveleti egységek reprezentálják megoldást.

A döntések sorozata ellentmondáshoz is vezethet, és előfordulhat, hogy egy aktív halmazban lévő anyagot előállító műveleti egységek közül korábban már mindet kizártuk.

A döntések összes lehetséges sorozatának leszámolásával minden megoldásstruktúra előállítható. A döntések összes lehetséges sorozatát úgy ábrázolhatjuk mint egy irányított fagráfot. A pontok az anyagokon végzett döntések, a kimenő élek pedig a lehetséges döntési alternatívák. A fa levélpontjai az inkonzisztens részproblémák és a kombinatorikusan lehetséges struktúrák.

### ABB algoritmus

Az ABB algoritmus [40] képes a kombinatorikusan lehetséges megoldásokból a költség szerinti optimális megoldást kiválasztani. Az eljárás az előző fejezetben ismertetett SSG módszeren alapszik. Az eljárás a korlátozás és szétválasztás (BB) keretalgoritmusra épül. A különbség az SSG-hez képest annyi, hogy egy döntés előtt egy korlátszámítási eljárást hajtunk végre, amely a részproblémához egy alsó korlátot rendel. Az alsókorlát alapján törölhetünk részproblémákat. Az ABB algoritmus az SSG -hez tartozó fagráf egy részét generálja ki. Az algoritmus annál hatékonyabb, minél kisebb az így bejárt fa, amit döntően a korlátozási eljárásban meghatározott alsó korlát élessége határoz meg.

### RSG algoritmus

Egy műveleti egység kizárása maga után vonhatja más műveleti egységek kizárását is. Az RSG algoritmus meghatározza és kizárja ezeket a műveleti egységeket. Leggyakoribb eset az, amikor egy műveleti egység kizárásával olyan diszjunkt részgráfok keletkeznek, ahonnan nem vezet irányított út a termékig (lásd a 2.2.2 definícióban az  $S3$  sérülése).

## 3. fejezet

# Szeeparábilis konkáv optimalizálás PNS feladatok megoldására

A PNS feladatosztály konkáv függvényekkel kibővített modellje egy lineáris feltételekkel adott szeeparábilis konkáv programozási feladat (lásd a (2.2.8) modellt és a (2.2.9) cél-függvényt). Egy PNS feladathoz kapcsolódó feltételrendszer magán hordozza a PNS feladat strukturáltságát, melyet az általános megoldók nem tudnak figyelembe venni. Célunk, hogy ezen tulajdonságok figyelembevételével a korábbinál hatékonyabb megoldó módszereket dolgozzunk ki.

Tekintsük a következő szeeparábilis, konkáv optimalizációs feladatot

$$\left. \begin{array}{l} \min \sum_{j=1}^n f_j(x_j) \\ A \mathbf{x} \leq \mathbf{b} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \quad (P) \quad (3.0.1)$$

ahol az  $A \in \mathbb{R}^{m \times n}$  mátrix, a  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$  adott vektorok,  $f_j : \mathbb{R} \rightarrow \mathbb{R}$  konkáv függvények. Legyen továbbá

$$\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : A \mathbf{x} \leq \mathbf{b}\} \quad \text{és} \quad \mathcal{T} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$$

halmazok, amely metszeteként előáll a  $(P)$  feladat megengedett megoldásainak a halmaza, azaz  $\mathcal{D} = \mathcal{A} \cap \mathcal{T}$ . Megjegyezzük, hogy korábban a PNS lineáris modelljében (lásd korábban (2.2.8)) a műveleti egységekhez tartozó  $\mathbf{z}$  változókat itt  $\mathbf{x}$ -el jelöljük.

A  $(P)$  feladat érdekességét az adja, hogy a legegyszerűbb nem konvex optimalizálási feladatosztályba tartozik. Elmondhatjuk, hogy a  $(P)$  feladat NP-teljes [75]. Fontos elméleti tulajdonsága az, hogy az optimális megoldása a  $\mathcal{D}$  poliédernek egy

csúcsában is felvétetik [69] sőt, ha az  $f_j$  függvények szigorúan konkávak, akkor az optimum csúcsban van.

**Állítás 3.0.1** *A (3.0.1) feladat optimauma a  $\mathcal{D}$  poliédernek egy extrémális pontjában is felvétetik.*

**Bizonyítás.** Legyen  $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ ,  $0 < \lambda < 1$ . Mivel  $f$  konkáv és nemlineáris, így a következő igaz:

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) > \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}) \geq \min\{f(\mathbf{x}), f(\mathbf{y})\}$$

Tehát, ha  $\bar{\mathbf{x}} \in \mathcal{D}$  nem extrémális pont, akkor létezik  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$ , hogy  $\bar{\mathbf{x}} = \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2$ , amiből az  $f(\bar{\mathbf{x}}) > \min\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}$  adódik.  $\square$

### 3.1. Szeparábilis konkáv programozás szakirodalmának áttekintése

A lineáris feltételrendszerrel adott szeparábilis konkáv minimalizálási problémához még számos gyakorlati kérdés vezet. A teljesség igénye nélkül megemlítünk néhány olyan műszaki tervezéssel kapcsolatos problémát, amely a  $(P)$  optimalizálási feladattal írható le: bizonyos irányításelméleti feladatok [3], konkáv hátizsák probléma [73], termelési és szállítási feladatok [60], termelési folyamatok tervezése [67], egyes hálózati folyamfeladatok [100], létesítmények optimális elhelyezése [92], stb.

A konkáv szeparábilis programozási feladat fontosságának megfelelően igen gazdag szakirodalma van. A szakirodalomban napjainkig ismertetett módszerek három fő csoportra oszthatók: extrémális pontok bejárása, metsző sík módszerek és korlátozás és szétválasztás (Branch-and-Bound, BB) módszerek. A BB módszereket tárgyalják a következő cikkek: [4], [11], [29], [61], [68], [81], [88], [89]. Csúcs leszámplálási eljárásokkal foglalkoznak például a [5], [27] és [26] dolgozatok. A metsző sík eljárások bemutatását a [9], [49], [82] és [95] munkákban találjuk meg. Előfordulnak még egyéb módszerek is, mint pl. a spline közelítés [59] vagy BB és metszősík módszer kombinálása [9].

A BB típusú algoritmusok egyik kritikus lépése a részfeladatok generálása, ez nagyban befolyásolja az optimális megoldás megtalálását és a módszer hatékonyságát. Az eljárások egy része hipertéglatestet használ a részfeladatok generálására. Különböző felosztási stratégiákat tárgyalnak a [16], [17], [84], [85], [86] közlemények. Az [50] szerzője egy a szimplexeken alapuló partícionálási stratégiát mutat be.

## 3.2. Általános algoritmus

Az algoritmus (3.1 ábra) egy Branch-and-Bound (BB) keretalgoritmusra ([1], [53], [63]) támaszkodik. Egy BB eljárás ismertetésekor beszélnünk kell a főbb lépésekről: ezek a részprobléma definíció, a korlátozási és szétválasztási lépések.

### 3.2.1. Részprobléma

Tekintsünk egy

$$\mathcal{T}^k = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l}^k \leq \mathbf{x} \leq \mathbf{u}^k\} \subseteq \mathcal{T}$$

hipertéglatestet és a hozzá tartozó

$$\mathcal{D}^k = \mathcal{A} \cap \mathcal{T}^k \subseteq \mathcal{D}$$

halmazt, ahol  $\mathbf{l} \leq \mathbf{l}^k < \mathbf{u}^k \leq \mathbf{u}$ . Ekkor a

$$\min_{\mathbf{x} \in \mathcal{D}^k} \sum_{j=1}^n f_j(x_j)$$

feladatot a  $(P^k)$  részproblémának nevezzük.

### 3.2.2. Korlátozás

Az alsókorlát meghatározása a korlátozás és szétválasztás módszerének (BB) az alapvető lépése. Az alsókorlát pontossága nagyban meghatározza az algoritmus konvergenciájának a sebességét.

Legyen  $(P^k)$  egy részprobléma a hozzátartozó  $\mathcal{T}^k$  hipertéglatesttel, és legyen

$$\mathcal{D}^k = \mathcal{A} \cap \mathcal{T}^k \neq \emptyset$$

---

Korlátozás és szétválasztás keretalgoritmus: a téglatest módszer

---

**Bemenő adatok:**

$m, n \in \mathbb{N}$   
 $A \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$  és  $\mathbf{l} \leq \mathbf{u}$   
 $f : \mathbb{R}^n \rightarrow \mathbb{R}$  konkáv függvény  
 $k = 0$ ,  $L = -\infty$ ,  $U = \infty$   
 $\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : A \mathbf{x} \leq \mathbf{b}\}$   
 $\mathcal{T}^0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$   
 $\mathcal{D}^0 = \mathcal{A} \cap \mathcal{T}^0$   
 $P^0 = (\mathcal{T}^0, \mathcal{D}^0)$   
 $\mathcal{S} = \{P^0\}$

**Kimenő adatok:**

a  $(P)$  feladat optimális megoldása  $\bar{\mathbf{x}}$   
 a  $(P)$  feladat optimum értéke  $U$

**Begin**

**while**  $(\mathcal{S} \neq \emptyset)$   
**begin**  
 $P^k = \text{Választ}(\mathcal{S});$   
 $(U, \bar{\mathbf{x}}, \beta^k) = \text{Korlátozás}(P^k, \bar{\mathbf{x}}, U, f);$   
 $L = \min_{P^j \in \mathcal{S}} \beta^j;$   
**if**  $U = L$  **then**  
 $\bar{\mathbf{x}}$  optimális megoldása a  $(P)$  feladatnak, *STOP*;  
 $\mathcal{S} = \text{Partícionálás}(\mathcal{A}, \mathcal{T}^k, f^k, \beta^k, \mathcal{S});$   
 $\mathcal{S} = \mathcal{S} \setminus \{P^k\};$   
**end**

**End.**


---

3.1. ábra. Általános algoritmus.



halmaz.

### Az $f_j$ konkáv függvények közelítése a $\mathcal{T}^k$ halmazon

Tekintsük az  $f_j : \mathbb{R} \rightarrow \mathbb{R}$  konkáv függvények lineáris relaxációját az  $[l_j^k, u_j^k]$  zárt intervallumon a következő módon:

$$F_j^k(x_j) = c_j^k x_j + d_j^k,$$

ahol

$$c_j^k = \frac{f_j(u_j^k) - f_j(l_j^k)}{u_j^k - l_j^k} \quad \text{és} \quad d_j^k = f_j(l_j^k) - \frac{f_j(u_j^k) - f_j(l_j^k)}{u_j^k - l_j^k} l_j^k = f_j(l_j^k) - c_j^k l_j^k,$$

azaz

$$F_j^k(x_j) = c_j^k x_j + d_j^k = c_j^k x_j + f_j(l_j^k) - c_j^k l_j^k.$$

Az  $f(\mathbf{x}) = \sum_{j=1}^n f_j(x_j)$  függvényt az

$$\begin{aligned} F^k(\mathbf{x}) &= \sum_{j=1}^n F_j^k(x_j) = \sum_{j=1}^n (c_j^k x_j + f_j(l_j^k) - c_j^k l_j^k) \\ &= (\mathbf{c}^k)^T \mathbf{x} + (f(\mathbf{l}^k) - (\mathbf{c}^k)^T \mathbf{l}^k) = (\mathbf{c}^k)^T \mathbf{x} + \delta^k \end{aligned}$$

alakú lineáris függvénnyel közelítjük a  $\mathcal{D}^k = \mathcal{A} \cap \mathcal{T}^k$  halmazon, ahol

$$\delta^k = f(\mathbf{l}^k) - (\mathbf{c}^k)^T \mathbf{l}^k.$$

Ekkor nyilván

$$f(\mathbf{x}) \geq F^k(\mathbf{x}) = (\mathbf{c}^k)^T \mathbf{x} + \delta^k$$

teljesül bármely  $\mathbf{x} \in \mathcal{D}^k$  esetén.

### Az alsókorlát meghatározása

Az alsókorlát kiszámítására a következő lineáris programozási feladatot használjuk:

$$\min_{\mathbf{x} \in \mathcal{D}^k} (\mathbf{c}^k)^T \mathbf{x} + \delta^k \quad (P_{LP}^k).$$

---

Korlátozás

---

**Bemenő adatok:**

$$P^k = (T^k, \mathcal{D}^k), \bar{\mathbf{x}}, U, f$$

**Kimenő adatok:**

$$U, \bar{\mathbf{x}}, \beta^k$$

**Begin**

számítsuk ki az  $F^k$  lineáris függvényt;

oldjuk meg a  $(P_{LP}^k)$  feladatot;

**if**  $\mathcal{D}^k = \emptyset$  **then**

**begin**

$$\beta^k = +\infty;$$

$$\alpha^k = +\infty;$$

**end**

**else begin**

legyen az  $\omega^k$  a  $(P_{LP}^k)$  feladat optimális megoldása;

$$\beta^k = F^k(\omega^k);$$

$$\alpha^k = f(\omega^k);$$

**end**

**if**  $\alpha^k < U$  **then**  $U = \alpha^k; \bar{\mathbf{x}} = \omega^k;$

**End.**

---

3.2. ábra. A korlátozási eljárás.

A  $(P_{LP}^k)$  feladat optimális megoldását jelölje  $\omega^k$ , és a hozzá tartozó célfüggvényérték legyen  $\beta^k = F^k(\omega^k) = (\mathbf{c}^k)^T \omega^k + \delta^k$ . Ekkor

$$\beta^k \leq (\mathbf{c}^k)^T \mathbf{x} + \delta^k \leq f(\mathbf{x}) \leq f(\omega^k) + (\nabla f(\omega^k))^T (\mathbf{x} - \omega^k)$$

teljesül bármely  $\mathbf{x} \in \mathcal{D}^k$  esetén, tehát alsókorlátot adtunk a  $(P^k)$  részprobléma optimális értékére. Míg a második egyenlőtlenség az  $f(\mathbf{x})$  függvény konkavitása miatt igaz, hiszen az

$$\bar{f}(\mathbf{x}) = f(\omega^k) + (\nabla f(\omega^k))^T (\mathbf{x} - \omega^k) \tag{3.2.1}$$

lineáris függvény az  $f$  konkáv függvénynek az  $\omega^k \in \mathcal{D}^k$  pontbeli érintője. A korlátozási eljárás a 3.2 ábrán látható.

### 3.2.3. Szétválasztás

A részproblémákat egy-egy  $\mathcal{T}^k$  hipertéglatesttel definiáltuk. A szétválasztás valamely adott  $\mathcal{T}^k$  téglatest ketté vágását jelenti.<sup>1</sup>

#### Szétválasztási szabály

A konvergencia és a végesség szempontjából kritikus dolog a  $i \in \mathcal{J}$  vágási változó indexének és a  $p \in [l_i^k, u_i^k]$  vágási pont meghatározása.<sup>2</sup>

#### Egy téglatest partícionálása

Legyen  $\mathcal{T}^k = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l}^k \leq \mathbf{x} \leq \mathbf{u}^k\}$  adott téglatest és legyen  $p \in \mathbb{R}$ , amelyre  $l_j^k \leq p \leq u_j^k$  teljesül, valamely  $j \in \mathcal{J}$  indexre.

Következőkben az  $x_j$  változót *vágási változónak* nevezzük, míg a  $p$  értéket *vágási pontnak*. A vágás a

$$\mathcal{T}^{(k,j,1)} = \{\mathbf{x} \in \mathcal{T}^k : l_j^k \leq x_j \leq p\} \quad \text{és} \quad \mathcal{T}^{(k,j,2)} = \{\mathbf{x} \in \mathcal{T}^k : p \leq x_j \leq u_j^k\} \quad (3.2.2)$$

halmazokat eredményezi, ahol  $j$  a vágási változó indexe,  $k$  az aktuális részprobléma indexe és 1. a jobb oldali, míg 2. a bal oldali téglatestre utal.<sup>3</sup>

A  $\mathcal{T}^{(k,j,1)}$  és  $\mathcal{T}^{(k,j,2)}$  hipertéglatestek definíciójából világos, hogy

$$\mathcal{T}^k = \mathcal{T}^{(k,j,1)} \cup \mathcal{T}^{(k,j,2)},$$

és a metszetük a két hipertéglatest  $(n-1)$ -dimenziós közös lapja lesz, azaz

$$\mathcal{T}^{(k,j,1)} \cap \mathcal{T}^{(k,j,2)} = \{\mathbf{x} \in \mathcal{T}^k : x_j = p\},$$

így a  $\mathcal{T}^k$  hipertéglatestnek egy felbontását kaptuk. A partícionálási eljárás a 3.3 ábrán látható.

<sup>1</sup>Esetünkben a BB eljárás fája, bináris fa lesz.

<sup>2</sup>Az általános módszert sokféle szétválasztási szabállyal működtethetjük, pl. az intervallumokat csak a célfüggvény alsó közelítésére használjuk, de a minimalizálást az eredeti megoldás halmazon végezzük. Ennek a változatnak előnye az, hogy az LP megoldása mindig eredeti csúcspontban van. Ezzel szemben a hátránya az, hogy időnként relaxált LP feladat optimális megoldása nem esik bele az intervallumba, így kapott megoldás nem lesz alsókorlát, hiszen a lineáris közelítő függvényünk csak az intervallumon belül lesz kisebb, mint az eredeti célfüggvény.

<sup>3</sup>Ha nem okoz félreértést, akkor egyszerűen  $\mathcal{T}^1$  és  $\mathcal{T}^2$  téglatestekről beszélünk.

---

*Particionálás*

---

**Bemenő adatok:**

$\mathcal{A}, T^k, f^k, \beta^k, \mathcal{S}$

**Kimenő adatok:**

$\mathcal{S}$

**Begin**

**if**  $U > \beta^k$  **then**

**begin**

      az  $x_j$  vágási változó meghatározása;

      a  $p$  vágási pont meghatározása;

$T^1$  és  $T^2$  téglatest meghatározása;

$\mathcal{S} = \mathcal{S} \cup \{(T^1, \mathcal{D}^1, \beta^k), (T^2, \mathcal{D}^2, \beta^k)\};$

**end**

**End.**

---

3.3. ábra. A particionálási eljárás.

### 3.2.4. A keretalgoritmus elemzése

A keretalgoritmusnak eddig még nem tárgyaltuk a *Választ* nevű eljárását. A *Választ* eljárás azt a  $P^k$  részproblémát választja ki, amelyre a  $\beta^k = L$  teljesül.

A *Korlátozás* eljárás során a kiválasztott részprobléma LP relaxáltját oldjuk meg. Ennek az optimumértékét jelöli  $\beta^k$ . Meghatározzuk az optimális megoldás helyén az eredeti célfüggvény  $\alpha^k$  értékét, amely a  $(P)$  feladat optimumértékére lesz felsőkorlát. Végül az aktuális  $L$  és  $U$  korlátokat összehasonlítjuk az  $\alpha^k$  és  $\beta^k$  számokkal és szükség esetén módosítjuk azokat.

Az algoritmus következő lépése a globális optimalitás vizsgálata. Ha az aktuális legjobb megoldás nem elégíti ki ezt a feltételt, akkor a vizsgált részfeladatot particionáljuk.

A *Particionálás* eljárásban először a részprobléma alsókorlátját hasonlítjuk össze a  $(P)$  feladat aktuális felsőkorlátjával. Ha  $\beta^k \geq U$  akkor az adott részproblémát eldobjuk anélkül, hogy particionálnánk, hiszen a részprobléma az eddig megtalált legjobb megoldásnál nem tartalmazhat jobb megoldásokat. A particionáláskor nyert részfeladatokat további elemzésnek vethetnénk alá.

Ha a  $\mathcal{D}^k = \mathcal{A} \cap T^k = \emptyset$ , akkor ezt a részfeladatot nem kellene hozzáadni a

feladatok halmazához. Ennek az eldöntése sajnos egy lineáris egyenlőtlenségrendszer megoldását jelenti, amely az LP relaxált megoldásával egyenértékű.

Másfelől, ha a  $\mathcal{D}^k$  nem tartalmazza a  $\mathcal{D}$  egyetlen egy extrémális pontját sem, akkor sem kellene hozzávenni a feladatok listájához. Sajnos ennek kiderítése sem egyszerű, hiszen a  $\mathcal{D}$  halmaz összes extrémális pontjának az ismeretéből a  $(P)$  feladat megoldása egyszerűen megkapható lenne.

Az eljárás során, ha a megengedett megoldás téglá, akkor igaz a következő állítás:

**Állítás 3.2.1** *Ha  $(P^k)$  részprobléma esetén  $\mathcal{D}^k = \mathcal{T}^k$ , akkor a  $(P_{LP}^k)$  feladat optimumértéke  $\beta^k$  nem lehet kisebb az aktuális legkisebb értéknél.*

**Bizonyítás.** Az eddig megtalált legkisebb célfüggvényértéket jelölje  $U$ .

A feltétel alapján

$$\mathcal{D}^k = \mathcal{A} \cap \mathcal{T}^k = \mathcal{T}^k.$$

A korlátozás lépésben a  $(P_{LP}^k)$  feladatot oldanánk meg, azaz egy  $n$ -dimenziós hipertéglatest felett optimalizálnánk az  $F^k(\mathbf{x})$  lineáris függvényt. Mivel az optimumérték valamely csúcsban is felvétetik, ezért van olyan optimális  $\omega^k$  megoldása a feladatnak, amely esetén

$$\omega_i^k = l_i^k \quad \text{vagy} \quad \omega_i^k = u_i^k$$

teljesül. Továbbá

$$f(\omega^k) = \sum_{j=1}^n f_j(\omega_j^k) = \sum_{j=1}^n F_j^k(\omega_j^k) = \beta^k \geq U$$

hiszen az intervallumok határpontjaiban a függvényértékek megegyeznek, azaz

$$F_j^k(l_j) = f_j(l_j) \quad \text{és} \quad F_j^k(u_j) = f_j(u_j).$$

Az egyenlőtlenség pedig azért teljesül, mert  $f(\omega^k) \geq U$ . □

Ilyen esetekben tehát a részprobléma nem kerül további partícionálásra.

### 3.3. "Csúsztatott" partícionálási szabály

A partícionálási stratégia döntően befolyásolja az algoritmus hatékonyságát. Jelen fejezetben, egy a szakirodalomban ismert [89] vágási szabályt ismertetünk, bemutatjuk ezen vágási stratégia korlátait, majd megadjuk a stratégia egy olyan módosítását, amely PNS feladatok megoldásakor hatékonyabban működik.

#### 3.3.1. $\bar{x}$ -partícionálás

A Shectman és munkatársai [89] által bevezetett partícionálási stratégiát vizsgáljuk meg. Vezessük be a következő függvényt:  $Level(P^k)$  függvény mutassa a **BB** fa azon szintjét, mely a  $(P^k)$  részproblémához tartozik, és legyen  $N$  egy pozitív egész szám.

```

if (  $Level(P^k) \bmod N = 0$  ) then
     $j = \operatorname{argmax}_{i \in \{1..n\}} (u_i^k - l_i^k)$ 
     $p = (u_j^k + l_j^k)/2.0$ 
else
     $j = \operatorname{argmax}_{i \in \{1..n\}} \{f_i(\omega_i^k) - F_i^k(\omega_i^k)\}$ 
    if (  $\bar{x} \in \mathcal{D}^k \wedge \bar{x}_j \in ]l_j^k, u_j^k[$  ) then
         $p = \bar{x}_j$ 
    else
         $p = (u_j^k + l_j^k)/2.0$ 
    endif
endif

```

Minden  $N$ -ik szinten lévő részprobléma vágásakor a leghosszabb él mentén vágjuk két egybevágó részbe a téglatestet, így a téglatestek oldalainak hossza zérushoz tart. Más esetekben a vágási változó legyen azon változó, ahol relaxációs távolság az LP aktuális megoldásnál  $(f_j(\omega_j^k) - F_j^k(\omega_j^k))$  maximális. Ha az eddigi legjobb megoldást tartalmazza az aktuális intervallum, és a vágásra kijelölt irányban pedig belső pontként tartalmazza, akkor vágási pont legyen az eddigi legjobb megoldás  $\bar{x}$ . Ha ez nem teljesül, felezzük az intervallumot a korábban kijelölt változó szerint.

### 3.3.2. $\bar{x}$ konvergencia, végeesség

Tekintsük a BB algoritmus által generált BB fát. Tegyük fel, hogy az algoritmus végtelen, tehát létezik egy végtelen  $\mathcal{T}^q$  sorozat, amelyre  $\mathcal{T}^{q+1} \subset \mathcal{T}^q$ , azaz a BB fában létezik egy a gyökből kiinduló végtelen út.

A vágási stratégiából könnyen levezethető a következő összefüggés [89]:

**Lemma 3.3.1**  $\lim_{q \rightarrow \infty} (u_j^q - l_j^q) = 0, \forall j \in \{1, \dots, n\}$ , azaz  $q \rightarrow \infty$  esetén a  $P^q$  részproblémát definiáló hipertéglatest oldalainak hossza tart a zérushoz.

Aminek a fontos következménye a 3.3.2 lemma.

**Lemma 3.3.2**  $\lim_{q \rightarrow \infty} (f(\omega^q) - F^q(\omega^q)) = 0$ , azaz  $q \rightarrow \infty$  esetén a  $P^q$  részprobléma alsó korlátja ( $F^q(\omega^q)$ ) tart a részprobléma felső korlátjához ( $f(\omega^q)$ ).

A 3.3.2 lemma biztosítja az algoritmus konvergenciáját, a 3.3.1 lemma segítségével viszont meg lehet mutatni, hogy az algoritmus véges időben azonosítja a globális optimumhelyet [89]. A optimális megoldás megtalálása után a program a további partícionálásokat már mindig az optimális megoldás mentén végzi. Egy változón egy adott részproblémában csak egyszer kell vágást végrehajtani, mert az olyan változókon, amelyekeken már történt vágás a megoldásban a lineáris közelítés eltérése zéró lesz, hiszen a vágás után a pont az intervallum szélére kerül. Így véges lépésben a relaxációs távolság zérussá változik, azaz a korábban megtalált megoldás optimalitása bizonyítottá válik.

### 3.3.3. A módszer viselkedése

Gyakorlati tapasztalatok azt mutatják, hogy a  $(P)$  feladat megoldása során az optimumhelyet a BB algoritmus elég korán azonosítja. A számítás nagy része arra fordítódik, hogy a nyitott részproblémákról belássuk, hogy azok nem tartalmaznak optimális megoldást, azaz ezen részproblémákat alsókorlát alapján el kell tudni vetni. A hangsúlyt tehát érdemes arra fektetni, hogy olyan vágási stratégiát dolgozzunk ki, amely az optimumot nem tartalmazó részproblémákat hatékonyan tudja kezelni.

Az optimális megoldásnál történő partícionáláskor az optimumhelyet mindkét keletkező részprobléma tartalmazni fogja. Ezért a  $\bar{x}$ -partícionálási stratégia nagy feladatok megoldása esetében nagy számítási és memória kapacitást igényel, amely nagyon megnehezíti ezen feladatok megoldását.

Miután a módszer megtalálta az optimális megoldást, kettévágja az aktuális részproblémát az optimális megoldásnál, majd ezen részproblémákat is kettévágja és ezt folytatja addig, amíg valamelyik szabály alapján el nem veti ezen részproblémákat. Alsó korlát alapján az ilyen részproblémákat nem lehet törölni, hiszen azok tartalmazzák az optimális megoldást. Az optimális megoldást tartalmazó részproblémák elvetése csak akkor történhet, ha a célfüggvény alsó közelítése a megoldásban pontossá válik. Egyéb gyorsítási módszerekkel sem érhetünk el javulást, hiszen az optimális megoldás garantálása érdekében alapkövetelmény, hogy az optimális megoldást tartalmazó részprobléma nem törölhető.

Tehát a módszer felépít egy teljes bináris fát. A fa mélysége függ az optimális megoldásban szereplő eredetileg nem korláton lévő nemlineáris változók számától. Ha  $k$  db nem korláton lévő nemlineáris változót tartalmaz az optimális megoldás, a hozzátartozó bináris fa  $k$  mélységű, azaz a  $2^{k+1} - 1$  db csúcspont van benne, ami a megoldott LP-k számát is jelzi. A nyitott részproblémák maximális számára is lehet becslést adni, ami a  $k$  mélységben lévő csúcsok száma, azaz  $2^k$ .

Nézzünk erre egy példát: egy gyakorlati feladat nemlineáris változóinak száma elérheti a több százat is. Tegyük fel, hogy az az optimumban mondjuk csak 50 nem korláton lévő változó van, ami  $2^{51} - 1 \approx 2 \times 10^{15}$  db LP megoldását teszi szükségessé. A részproblémák száma  $2^{50} \approx 10^{15}$  ( $\equiv$  millió  $\times$  milliárd), ha 1 részprobléma tárolásához kb 1 kbyte tár szükséges, akkor egymilliárd gigabyte memóriára lenne szükségünk. Az aktuális LP megoldás mentén történő vágás ( $\omega$  vágás [29]) esetében is hasonló viselkedésű lesz a módszer, mert a globális optimumhely meghatározása után a szaporodó részproblémák relaxált megoldása is a globális optimumhely lesz.

### 3.3.4. "Csúsztatott" vágási módszer

Az előző módszer fő gyengeségét próbáljuk elkerülni a pozitív tulajdonságok megtartásával. A relaxációs távolságot próbáljuk csökkenteni úgy, hogy a részproblémák

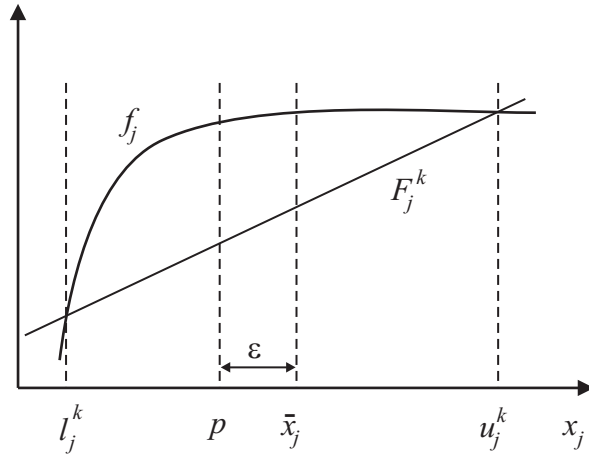


számát is tudjuk közben kezelni. Célunk az, hogy feleslegesen ne növeljük meg az optimumhelyet tartalmazó részproblémákat. Legyen a vágási stratégia a következő:

```

if (  $Level(P^k) \bmod N = 0$  ) then
     $j = \operatorname{argmax}_{i \in \{1, \dots, n\}} (u_i^k - l_i^k)$ 
     $p = (u_j^k + l_j^k)/2.0$ 
else
     $j = \operatorname{argmax}_{i \in \{1, \dots, n\}} \{f_i(\omega_i^k) - F_i^k(\omega_i^k)\}$ 
    if (  $\bar{\mathbf{x}} \in \mathcal{D}^k \wedge \bar{x}_j \in ]l_j^k + \varepsilon, u_j^k[$  ) then
         $p = \bar{x}_j - \varepsilon$ 
    else
         $p = (u_j^k + l_j^k)/2.0$ 
    endif
endif

```



3.4. ábra.  $\varepsilon$ -vágás.

Az alapötlet az, hogy az optimális megoldás mindig csak az egyik részintervallumban lesz benne, így alsókorlát alapján a másik eldobható.

Hasonló módszerrel találkozhatunk az intervallum aritmetikán alapuló optimalizálási eljárásokban az úgynevezett "clustering" probléma megoldására. A lapos helyi minimumok környékén a megállási kritérium után "fürtökben" lógnak az olyan részintervallumok, amelyek potenciálisan tartalmazhatják a megoldást. A javasolt módszer az volt, hogy a feladat megoldását több kezdőintervallummal kell megtenni,

így egy jól meghatározott  $\varepsilon$  értékkel megváltoztatott korlátok miatt sokkal kevesebb részintervallum fogja csak tartalmazni a helyi minimumokat [56], [57].

**Vágási pont:**  $(\bar{x}_j - \varepsilon)$

Tekintsük a 3.4 ábrát. Elemezzük, hogy mi történik az egyes részproblémákkal a particionálás után. Az  $[l_j^k, \bar{x}_j - \varepsilon]$  esetén az optimum már nem része a halmaznak ( $\bar{x}_j \notin [l_j^k, \bar{x}_j - \varepsilon]$ ). Azaz  $x_j$  változóhoz tartozó műveleti egység működése felülről korlátozódott úgy, hogy már nem képes kielégíteni az igényeket. A hiányzó igényeket vagy egy másik műveleti egység pótolhatja vagy egy teljesen más struktúra lesz az optimális. Ezek már szignifikáns változások lesznek az optimális megoldáshoz képest, és nagy valószínűséggel az alsókorlát alapján törlődik. Láthatjuk, hogy itt fontos szerepet kap az a tény, hogy a feltételrendszer egy PNS feladatot reprezentál.

### Az $\varepsilon$ meghatározása

Legyen  $\varepsilon > 0$  ( $\in \mathbb{R}$ ) olyan elegendően nagy mennyiség, mely szerint a bal oldali intervallumhoz ( $[l_j^k, \bar{x}_j - \varepsilon]$ ) tartozó részprobléma megoldáshalmazában nincs benne az optimális megoldás. Elméletileg természetesen nincs benne, de az LP megoldó gyakorlatban valamekkora toleranciával dolgozik, ennél a toleranciánál kell nagyobbak lennie az  $\varepsilon$ -nak.

$\varepsilon$  vágás esetében a vágási pont meghatározása az  $\bar{x}_j \in ]l_j^k + \varepsilon, u_j^k[$  feltétel figyelembevételével történik. A  $\bar{x}_j \in [l_j^k, l_j^k + \varepsilon]$  esetben úgy tekintjük, hogy a változó határon van, és rajta vágást nem hajtunk végre. Ha az összes nem korláton lévő változó az  $[l_i^k, l_i^k + \varepsilon]$  intervallumban van, akkor a részfeladatot megoldottnak tekintjük és elvetjük. Tehát egy  $\varepsilon$  élhosszúságú hiperkockát „hanyagoltunk” el. A lineáris feltételrendszer által meghatározott poliéderhez becsülhető a csúcsai közti legkisebb távolság. Ha az  $\varepsilon$  érték kisebb ennél a távolságnál, akkor a figyelmen kívül hagyott részben nem lehet más csúcs. Mivel korábban beláttuk, hogy az optimális megoldás a konvex poliéder egy csúcsában helyezkedik el, így az  $\varepsilon$  élhosszúságú hiperkocka elhanyagolható.

### 3.3.5. Konvergencia, végeesség

A konvergencia (lásd 3.3.2 lemma) fennáll itt is, mivel a 3.3.1 lemma az új vágási stratégiával is teljesül.

Az eredeti módszer végeessége abból adódik, hogy a globális optimumot véges időben azonosítja és az optimumhelyet tartalmazó részproblémákat az optimumnál vágja ketté. Az optimumnál történő vágás véges lépésben a relaxációs távolságot zérussá csökkenti. Az optimumot nem tartalmazó részproblémák lokális optima a globális optimumtól véges távolságra van. A konvergencia (3.3.2 lemma) eredményeként véges lépés alatt az ilyen típusú részproblémák alsó korlátjai az optimumnál nagyobbá válnak, és így ezen részproblémák véges lépés alatt törlődnek.

### 3.3.6. Az eredmény rövid összefoglalása

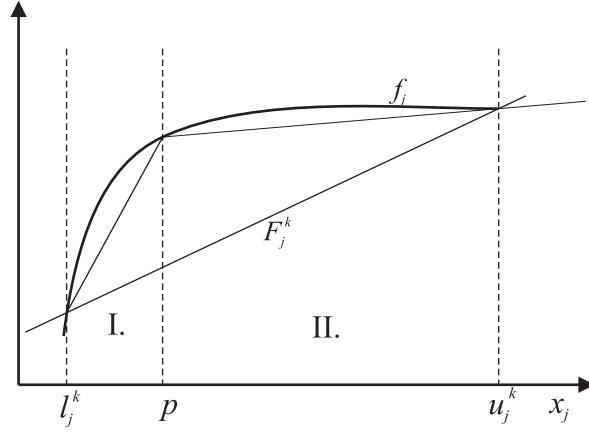
A 3.3 fejezet az 1a tézispontban megfogalmazott eredményeket tartalmazza.

A szakirodalomból ismert és széleskörűen alkalmazott (Shectman és munkatársai [89]) partícionálási stratégiát megvizsgálva bemutattam a partícionálási stratégia egyik kedvezőtlen tulajdonságát: a módszer feleslegesen sok olyan részproblémát generál, ami tartalmazza az optimális megoldást. Az, hogy az optimális megoldás sok aktív részproblémában szerepel, nagyban megnehezíti a megtalált megoldás optimalitásának bizonyítását. A bizonyítás így teljes bináris fa bejárását teszi szükségessé, amelynek a mélysége megegyezik az optimális hálózatban lévő csúcsok számával.

Ennek a kedvezőtlen tulajdonságnak a kiküszöbölésére dolgoztam ki az úgynevezett "csúsztatott" szétválasztási stratégiát, amelyben az optimális megoldást tartalmazó részproblémákat nem sokszorozzuk meg. A PNS feladatok megoldására ez különösen jól használható. A módszer helyességét bizonyítottam.

## 3.4. Maximális rés partícionálás

Egy BB algoritmus hatékonyságát nagyban befolyásolja az alsókorlát élessége. Fejezetünkben bevezetjük a maximális rés partícionálást (lásd 3.5 ábra), amely által meghatározott vágási pont minimalizálja a konkáv függvény és a lineáris közelítés



3.5. ábra. Maximális rés partícionálás.

integrálkülönbségét. Az integrálkülönbség minimalizálása garantálja az alsókorlát élességét és a konvergenciát is.

### 3.4.1. Vágási stratégia

Legyen  $P^k$  a kiválasztott részprobléma. A vágási változó kiválasztása legyen a következő:

$$j = \operatorname{argmax}_{i=1,\dots,n} \{f_i(\omega_i^k) - F_i^k(\omega_i^k)\}. \quad (3.4.1)$$

Legyen továbbá  $p$  az a vágási pont, amelyre

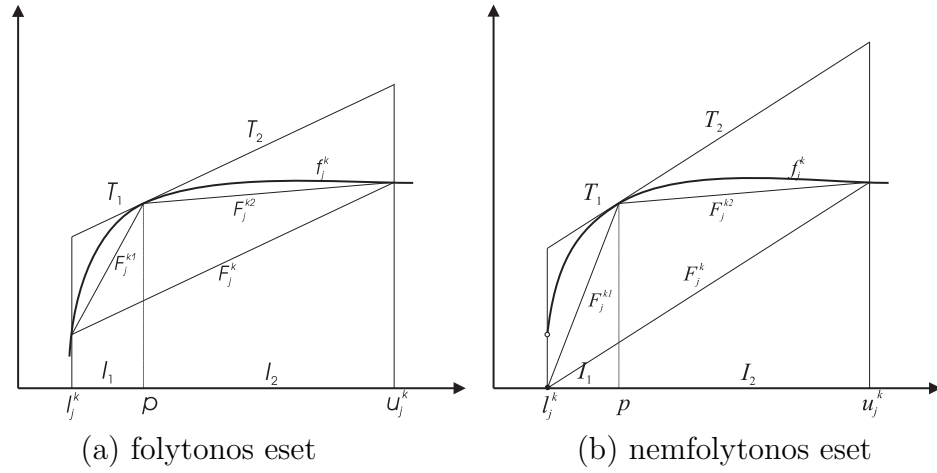
$$p = \operatorname{argsup}_{t \in [l_j^k, u_j^k]} \{f_j(t) - F_j^k(t)\}. \quad (3.4.2)$$

### 3.4.2. Konvergencia

Tekintsük a BB algoritmus által generált BB fát. Tegyük fel, hogy az algoritmus végtelen, tehát létezik egy végtelen  $\mathcal{T}^q$  sorozat, melyre  $\mathcal{T}^{q+1} \subset \mathcal{T}^q$ , azaz a BB fában létezik egy, a gyökérből kiinduló végtelen út.

Feltételezzük, hogy  $f$  az intervallum minden belső pontjában deriválható, és a végpontokban létezik a bal illetve a jobboldali derivált.

**Lemma 3.4.1** *Legyen  $P^k$  egy részprobléma, használjuk a (3.4.1) és (3.4.2) partícionálási szabályokat, és legyen  $P^{k_1}$  és  $P^{k_2}$  az ezután kapott részproblémák. Ekkor*



3.6. ábra. Az integrálkülönbség folytonos és nemfolytonos esetekre.

teljesülnek a következők:

$$\int_{l_j^k}^p f_j - F_j^{k1} < \frac{1}{2} \int_{l_j^k}^p f_j - F_j^k,$$

$$\int_p^{u_j^k} f_j - F_j^{k2} < \frac{1}{2} \int_p^{u_j^k} f_j - F_j^k.$$

**Bizonyítás.** Tekintsük a 3.6a ábrát. Legyen  $T_1$ ,  $T_2$  a megfelelő paralelogrammák területe, akkor a következők igazak:

$$\int_{l_j^k}^p f_j - F_j^k < T_1, \quad \int_p^{u_j^k} f_j - F_j^k < T_2.$$

Az egyenlőtlenségekből következik, hogy

$$\int_{l_j^k}^p f_j - F_j^{k1} = \int_{l_j^k}^p f_j - F_j^k - \frac{1}{2}T_1 < \int_{l_j^k}^p f_j - F_j^k - \frac{1}{2} \int_{l_j^k}^p f_j - F_j^k = \frac{1}{2} \int_{l_j^k}^p f_j - F_j^k,$$

és hasonlóan,

$$\int_p^{u_j^k} f_j - F_j^{k2} = \int_p^{u_j^k} f_j - F_j^k - \frac{1}{2}T_2 < \int_p^{u_j^k} f_j - F_j^k - \frac{1}{2} \int_p^{u_j^k} f_j - F_j^k = \frac{1}{2} \int_p^{u_j^k} f_j - F_j^k.$$

A nemfolytonos eset hasonlóan következik, lásd a 3.6b ábrát.  $\square$

**Tétel 3.4.2**  $\lim_{q \rightarrow \infty} f(\omega^q) - \beta^q = 0$ , azaz  $q \rightarrow \infty$  esetén a  $P^q$  részprobléma alsó korlátja ( $\beta^q$ ) tart a részprobléma felső korlátjához ( $f(\omega^q)$ ).

**Bizonyítás.** Két fő esetet vizsgálunk, amikor  $f$  folytonos  $\mathcal{T}^q$  felett, és amikor nem.

### 1. $f$ folytonos $\mathcal{T}^q$ felett

Tegyük fel, hogy létezik az indexeknek egy olyan  $\mathcal{N}_1 \subseteq \{1, \dots, n\}$  részhalmaza, hogy  $\forall q > K$  esetén nem történik vágás az  $\mathcal{N}_1$  indexhalmazba tartozó változókon. Továbbá feltehetjük azt is, hogy  $\mathcal{N}_1$  a legbővebb ilyen tulajdonságú indexhalmaz. Ezért  $\exists \varepsilon_1 > 0$ , hogy  $\forall q > K$  esetén

$$\forall j \in \mathcal{N}_1, \quad f_j(\omega_j^q) - F_j^q(\omega_j^q) \geq \varepsilon_1 > 0.$$

Az  $\{1, \dots, n\} \setminus \mathcal{N}_1$ -n viszont lesz vágás, tehát a korábbi állításunk szerint

$$\forall j \in \{1, \dots, n\} \setminus \mathcal{N}_1\text{-re}, \quad \int_{l_j^q}^{u_j^q} f_j - F_j^q \rightarrow 0, \text{ ahogy } q \rightarrow \infty.$$

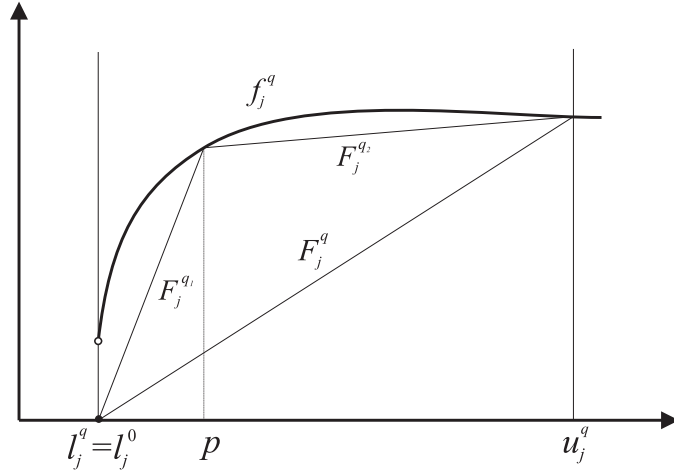
Mivel  $f$  folytonos  $\mathcal{T}^q$ -n, így  $\forall j \in \{1, \dots, n\} \setminus \mathcal{N}_1$ -re  $F_j^q \rightarrow f_j$  pontonként. Ezért  $\exists K_1 > K$ , hogy  $q > K_1$  esetén

$$\forall j \in \{1, \dots, n\} \setminus \mathcal{N}_1, \quad t \in [l_j^q, u_j^q], \quad f_j(t) - F_j^q(t) < \varepsilon_1.$$

Vagyis az algoritmus az  $\mathcal{N}_1$  halmazból fog választani vágási változót, de ez ellentmond annak, hogy  $\mathcal{N}_1$ -ben  $q > K$ -re nem történik vágás. Így következik, hogy az  $\mathcal{N}_1$  halmaz üres. Azaz

$$\forall j \in \{1, \dots, n\}\text{-re}, \quad \int_{l_j^q}^{u_j^q} f_j - F_j^q \rightarrow 0, \text{ ahogy } q \rightarrow \infty.$$

Így  $\forall j \in \{1, \dots, n\}$ -re  $F_j^q \rightarrow f_j$  teljesül pontonként. A folytonos esetre az állítást beláttuk.



3.7. ábra. Partícionálás az  $l_j^0$ -tól különböző helyen.

## 2. $f$ nem folytonos $\mathcal{T}^q$ felett

Ha az  $f$  nem folytonos  $\mathcal{T}^q$ -n, akkor a konkavitás miatt csak a határon lehet a szakadási pontja. Legyen  $\mathcal{D} \subseteq \{1, \dots, n\}$  a nem folytonos változók indexhalmaza. Az általánosság elvesztése nélkül feltehetjük, hogy  $f_j$  nem folytonos  $l_j^0$ -ban.

Vizsgáljuk ilyen esetben a vágások sorozatát. Különböztessük meg azt az esetet, mikor a vágás  $l_j^0$ -ban történik és azt amikor nem.

*Amikor vágás  $l_j^0$ -ban történt.*

Ekkor két intervallumot kaptunk:

Az egyik intervallum,  $([l_j^0, l_j^0])$  egy pontból áll, így az a függvény triviálisan folytonos.

A másik intervallumon  $([l_j^0, u_j^q])$  pedig a függvénynek megszüntetjük a szakadását azzal, hogy a

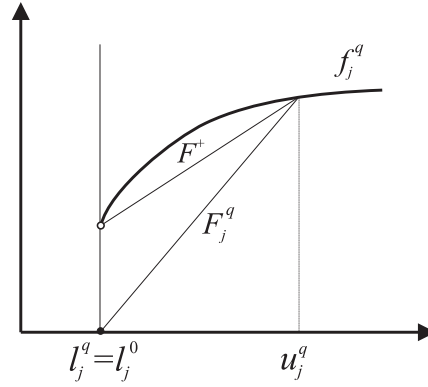
$$f_j^+ = \begin{cases} f_j(x_j), & x_j \in (l_j^0, u_j^q] \\ \lim_{t \rightarrow l_j^0+} f_j(t) & x_j = l_j^0 \end{cases}$$

függvényt definiáljuk, és így az  $f_j^+$  folytonos lesz  $[l_j^0, u_j^q]$ -n.

Az  $f_j^+$  különbözik az  $f_j$  -től az  $l_j^0 (= l_j^q)$  pontban, de a minimalizálás miatt ez nem okoz változást az optimumban. Így ez az eset is visszavezethető a folytonos esetre.

*Tegyük fel, hogy sosem vágunk az  $l_j^0$  pontban.*

A 3.7 ábrán látható, hogy a vágás után két új intervallumot kapunk, az egyiken  $[p, u_j^q]$ -n az  $f_j$  folytonos lesz – s így ezzel kész vagyunk – és a másik intervallumon



3.8. ábra. Az  $f^+$  függvény relaxációja az  $[l_j^0, u_j^q]$  intervallumon.

az  $f_j$ -nek szakadása van. Először megmutatjuk, hogy ez utóbbi intervallumok hossza tart a nullához, azaz,  $u_j^q \rightarrow l_j^0$  mialatt  $q \rightarrow \infty$ .

Indirekten bizonyítunk. Mivel mindig belső pontban vágunk, így az  $u_j^q$  szigorúan monoton módon csökken, tehát ezek sorozata konvergens. Tegyük fel, hogy  $\lim_{k \rightarrow \infty} u_j^q = u^+ (\neq l_j^0)$ . Legyen  $F^+$  a relaxációja az  $f_j^+$ -nak az  $[l_j^0, u_j^q]$  intervallumon (lásd 3.8 ábra).

Az  $F^+ > F_j^q$  teljesül kivéve az  $u_j^q$  pontot, ahol ezek egyenlők. A következő feltétel áll:

$$\int_{l_j^0}^{u_j^q} F^+ > \int_{l_j^0}^{u_j^q} F_j^q,$$

amiből

$$\int_{l_j^0}^{u_j^q} f_j^q - F_j^q = \int_{l_j^0}^{u_j^q} f_j^+ - F_j^q \geq \int_{l_j^0}^{u_j^q} F^+ - F_j^q > \frac{u^+ - l_j^0}{2} (f_j^+(l_j^0) - f_j(l_j^0)) > 0 \quad (3.4.3)$$

következik. Mivel (3.4.3) minden  $q$ -ra igaz, ezért ez ellentmond annak az állításunknak, hogy az integrálkülönbség tart a nullához.

Előzőekben beláttuk azt, hogy az intervallum hossza tart a nullához, vizsgáljuk meg hogyan viselkedik ekkor a lineáris közelítésünk.

Legyen  $\gamma_j = f_j^+(l_j^0) - f_j(l_j^0)$ . Mivel  $f_j^+$  folytonos  $[l_j^q, u_j^q]$ -n, ezért  $\forall \varepsilon > 0, \exists \delta$ , hogy  $u_j^q - l_j^q < \delta$ -ra a  $|f_j^+(u_j^q) - f_j^+(l_j^0)| < \varepsilon$  teljesül. Így  $\forall S \in ]0, \gamma_j[$ -re  $\exists K_1$ , hogy  $\forall q > K_1$  esetén

$$|f_j^+(u_j^q) - f_j^+(l_j^0)| < \gamma_j - S.$$



Azaz,

$$S - \gamma_j < f_j^+(u_j^q) - f_j^+(l_j^0) < \gamma_j - S. \quad (3.4.4)$$

A (3.4.4) egyenlőtlenségből,

$$f_j^+(u_j^q) > S + f_j^+(l_j^0) - \gamma_j = S + f_j(l_j^q). \quad (3.4.5)$$

Mivel  $f_j(u_j^q) = f_j^+(u_j^q)$ , így adódik az

$$f_j(u_j^q) - f_j^+(l_j^q) > S$$

összefüggés.

Mivel  $u_j^q \rightarrow l_j^0$ , így  $\forall S \in ]0, \gamma_j[$ -re,  $\forall M > 0$ -ra,  $\exists K_2$ , hogy  $\forall q > K_2$ , esetén  $u_j^q - l_j^q < \frac{S}{M}$ .

Tehát, ha  $q > \max(K_1, K_2)$ , akkor

$$\frac{f_j(u_j^q) - f_j(l_j^q)}{u_j^q - l_j^q} > M. \quad (3.4.6)$$

teljesül.

Állítjuk a következőt:  $\exists q$  hogy az algoritmus  $l_j^0$ -ban fog vágni.

Azaz  $\max_{t \in [l_j^0, u_j^q]} \{f_j(t) - F_j^q(t)\}$  az  $l_j^0$  pontban veszi fel az értéket. Legyen  $G = f_j^+(l_j^0) - F_j^q$  és  $A = \sup_{t \in [l_j^0, u_j^q]} f'(t)$ .  $\exists q$  melyre  $\frac{f_j(u_j^q) - f_j(l_j^0)}{u_j^q - l_j^0} > A$ . Legyen  $F_j^q(x_j) = Cx_j + B_j$ ,  $\Delta > 0$  és  $\Delta \leq u_j^q - l_j^0$ .

$$\begin{aligned} f_j(l_j^0 + \Delta) - F_j^q(l_j^0 + \Delta) &< f_j(l_j^0) + A\Delta - C_j\Delta - C_jl_j^0 - B_j \\ &= f_j(l_j^0) - F_j^q(l_j^0) + (A - C)\Delta \\ &< G. \end{aligned}$$

Hiszen korábban láttuk a (3.4.6) egyenletben, hogy a  $C$  tetszőlegesen nagy lehet. Ezzel tételünket beláttuk.  $\square$

**Tétel 3.4.3**  $\forall q$ -ra  $\mathcal{T}^q$  tartalmazza a globális optimumhelyet.

**Bizonyítás.** Ha a globális minimum  $\bar{\mathbf{x}} \notin \mathcal{D}^q = \mathcal{A} \cap \mathcal{T}^q$ , akkor  $f(\omega^q) > f(\bar{\mathbf{x}})$ , így  $\exists q$ , hogy  $F^q(\omega^q)$  elég közel van  $f(\omega^q)$ -hoz, azaz  $F^q(\omega^q) > f(\bar{\mathbf{x}})$ , és így nagyobb, mint bármelyik legkisebb  $L$  alsó korlát. Ez ellentmond annak a feltételnek, hogy mindig a legkisebb alsó korláttal rendelkező részproblémát választjuk.  $\square$

A módszer végeessége itt nem garantált, viszont hivatkozva a [89]-ban vázolt partícionálási szabályra, ezen vágási stratégia is kibővíthető úgy, hogy a végeesség garantálható legyen.

### 3.4.3. Az eredmény rövid összefoglalása

A 3.4 fejezet az 1b tézispontban megfogalmazott eredményeket tartalmazza.

Kidolgoztam egy új szétválasztási stratégiát, amely a célfüggvény és a relaxációs függvény integrálkülönbségét minimalizálja, ezáltal a relaxáció élességét maximálisra növeli. Bizonyítottam a módszer helyességét.

## 3.5. Egy elégséges optimalitási kritérium szeparábilis konkáv minimalizálási feladatra

Az eddigi szétválasztási eljárások a lehetséges megoldások halmazának partícionálásakor a célfüggvény és annak lineáris relaxációja alapján határozták meg a megfelelő vágási pontot. A relaxált lineáris programozási feladat vizsgálatával egy olyan stratégia lett kidolgozva, amely a partícionáláskor a konvex poliéder és a célfüggvény viszonyát figyelembe véve végzi a további partícionálást. A megfogalmazott algoritmus helyességét igazoljuk. Az elemzésekhez szükségünk lesz az LP feladatok vizsgálata során használatos néhány jelölésre.

### 3.5.1. A relaxált lineáris programozási feladat

A  $(P)$  feladat megoldása során a korlátozó lépés egy relaxált lineáris programozási feladat megoldása. A relaxált lineáris programozási feladat megoldásával és a megoldásnak az eredeti  $(P)$  feladat szempontjából történő érzékenységi vizsgálatával meg tudjuk adni a  $(P)$  feladat egy *elégséges optimalitási kritériumát*. Egész pontosan azt tudjuk eldönteni, hogy a relaxált lineáris programozási feladat optimális bázismegoldása egyben optimális megoldása-e a lineáris feltételes szeparábilis konkáv minimalizálási feladatnak is vagy sem.

A továbbiakban szükségünk lesz az egyváltozós konkáv függvények tulajdonságairól szóló jól ismert állításra (megtalálható pl. Császár Á., [18], 228. oldal).

**Állítás 3.5.1** *Legyen  $f$  egyváltozós függvény,  $I \subset D_f$  intervallum. A következő állítások egyenértékűek:*

- (a)  $f$  konkáv az  $I$  intervallumban;
- (b) az  $x, y \in I$ ,  $x \neq y$  esetre bevezetett

$$m(x, y) = \frac{f(y) - f(x)}{y - x}$$

jelöléssel, az  $a, b, c \in I$ ,  $a < b < c$  esetén

$$m(a, b) \geq m(a, c) \geq m(b, c);$$

- (c) bármely  $t \in I$  esetén  $m_t(x) = m(t, x)$  függvény fogyó az  $I \setminus \{t\}$  halmazon;
- (a) az  $a, b, c \in I$ ,  $a < b < c$  esetén

$$m(a, b) \geq m(b, c).$$

□

A konkáv függvények felsorolt tulajdonságainak a fontos következménye az alábbi állítás (megtalálható pl. Császár Á., [18], 232. oldal).

**Állítás 3.5.2** *Legyen  $f$  egyváltozós konkáv függvény az  $I \subset D_f$  nyílt intervallumban, ekkor:*

- (a) az  $f$  folytonos az  $I$  intervallumban;
- (b) az  $f$  bármely  $t \in I$  helyen jobbról és balról differenciálható és

$$f'_-(t) \geq f'_+(t);$$

(c) ha az  $a, b \in I$ ,  $a < b$  akkor

$$f'_+(a) \geq m(a, b) \geq f'_-(b),$$

sőt, ha az  $f$  szigorúan konkáv az  $I$  intervallumban, akkor

$$f'_+(a) > m(a, b) > f'_-(b).$$

□

### 3.5.2. A relaxált lineáris programozási feladat optimalitási kritériuma

Tekintettel arra, hogy a  $(P)$  feladat közelítését leíró lineáris programozási feladat célfüggvényében szereplő konstans tag nem befolyásolja azt, hogy az optimum hol vétetik fel, ezért a konstans tagot elhagyjuk a célfüggvényből és a  $(P_{LP})$  feladatot részletesen a következő alakban írhatjuk ki:

$$\left. \begin{array}{l} \min \mathbf{c}^T \mathbf{x} \\ A \mathbf{x} \leq \mathbf{b} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \quad (P_{LP}).$$

A  $(P_{LP})$  feladat megengedett megoldásainak a halmaza megegyezik a  $(P)$  feladatével. A  $(P_{LP})$  feladat optimális megoldásainak a halmazát jelölje

$$\mathcal{D}_c^* = \{\mathbf{x}^* \in \mathcal{D} : \mathbf{c}^T \mathbf{x}^* \leq \mathbf{c}^T \mathbf{x}, \mathbf{x} \in \mathcal{D}\}.$$

Az elemzéshez szükségünk lesz a lineáris programozási feladatok vizsgálata során használatos néhány jelölésre. Ezeket vezetjük most be.

Jelölje  $\mathcal{J}$  a  $(P_{LP})$  feladathoz tartozó változók és "slack" változók indexhalmazát, legyen  $\mathcal{J}_B \subset \mathcal{J}$  az optimális bázis változóinak az indexhalmaza és  $\mathcal{J}_N$  a bázison kívüli változók indexei. Ekkor az  $\{\mathbf{a}_j : j \in \mathcal{J}_B\}$  vektorok lineárisan függetlenek. Nyilván  $\mathcal{J} = \mathcal{J}_B \cup \mathcal{J}_N$ .

Jelölje  $\mathcal{J}_N^l \subset \mathcal{J}_N$  ( $\mathcal{J}_N^u \subset \mathcal{J}_N$ ) azon nem bázisban lévő változók indexhalmazát, amelyek alsó (felső) kortáton vannak. Természetesen  $\mathcal{J}_N^l \cup \mathcal{J}_N^u = \mathcal{J}_N$  (és  $\mathcal{J}_B \cap \mathcal{J}_N = \emptyset$ ).

Legyen  $\bar{A} = [B^{-1}A]$ , és jelölje a  $\mathbf{c}_B$  vektor, a  $\mathbf{c}$  vektornak a  $\mathcal{J}_B$  indexekre történő megszorítását.

Valamely  $\bar{\mathbf{x}} \in \mathcal{D}$  vektor esetén, ha  $\bar{\mathbf{x}}$  bázis megoldása a  $(P_{LP})$  feladatnak, akkor

$$\begin{aligned} l_i &\leq \bar{x}_i \leq u_i && \text{bármely } i \in \mathcal{J}_B, \\ \bar{x}_i &= l_i && \text{bármely } i \in \mathcal{J}_N^l, \\ \bar{x}_i &= u_i && \text{bármely } i \in \mathcal{J}_N^u. \end{aligned}$$

Ennek következtében, ha ismert a  $\mathcal{J}$  indexhalmaz  $(\mathcal{J}_B, \mathcal{J}_N^l, \mathcal{J}_N^u)$  partíciója, akkor a bázis változók értékeit az

$$\bar{\mathbf{x}}_B = B^{-1} \mathbf{b} - \sum_{j \in \mathcal{J}_N^l} l_j \bar{\mathbf{a}}_j - \sum_{j \in \mathcal{J}_N^u} u_j \bar{\mathbf{a}}_j$$

képlettel számíthatjuk ki, ahol az  $\bar{\mathbf{a}}_j$  vektor az  $\bar{A}$  mátrix  $j$ . oszlopvektora.

A  $(P_{LP})$  feladat optimalitási kritériumának a felírásához hasznos lesz a duál feladat megfogalmazása

$$\left. \begin{aligned} \max \quad & -\mathbf{b}^T \mathbf{y} + \mathbf{l}^T \mathbf{z} - \mathbf{u}^T \mathbf{s} \\ & -A^T \mathbf{y} + \mathbf{z} - \mathbf{s} = \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0} \end{aligned} \right\} \quad (D_{LP}),$$

és jelölje  $\bar{\mathcal{D}} = \{(\mathbf{y}, \mathbf{z}, \mathbf{s}) : -A^T \mathbf{y} + \mathbf{z} - \mathbf{s} = \mathbf{c}, \mathbf{y} \geq \mathbf{0}, \mathbf{z} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}\}$  a duál megengedett megoldások halmazát. Egyszerűen felírhatjuk a  $(P_{LP})$  és  $(D_{LP})$  feladatokhoz tartozó gyenge dualitás tételt.

**Állítás 3.5.3** Bármely  $\mathbf{x} \in \mathcal{D}$  és  $(\mathbf{y}, \mathbf{z}, \mathbf{s}) \in \bar{\mathcal{D}}$  vektorok esetén

$$\mathbf{c}^T \mathbf{x} \geq -\mathbf{b}^T \mathbf{y} + \mathbf{l}^T \mathbf{z} - \mathbf{u}^T \mathbf{s}$$

egyenlőtlenség teljesül és egyenlőség pontosan akkor áll fenn, ha

$$0 = \mathbf{c}^T \mathbf{x} + \mathbf{b}^T \mathbf{y} - \mathbf{l}^T \mathbf{z} + \mathbf{u}^T \mathbf{s} = \mathbf{y}^T (\mathbf{b} - A \mathbf{x}) + \mathbf{z}^T (\mathbf{x} - \mathbf{l}) + \mathbf{s}^T (\mathbf{u} - \mathbf{x}). \quad \square$$

Ezek után a  $(P_{LP})$  és  $(D_{LP})$  feladatokhoz tartozó (szükséges és elégséges) optimalitási kritériumot az alábbi módon adhatjuk meg

$$\begin{aligned} A \mathbf{x} &\leq \mathbf{b}, \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \\ -A^T \mathbf{y} + \mathbf{z} - \mathbf{s} &= \mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0} \\ \mathbf{y}^T (\mathbf{b} - A \mathbf{x}) &= 0, \quad \mathbf{z}^T (\mathbf{x} - \mathbf{l}) = 0, \quad \mathbf{s}^T (\mathbf{u} - \mathbf{x}) = 0. \end{aligned}$$

Feltéve az  $\mathbf{x}^* \in \mathcal{D}_c^*$  megoldásról azt is, hogy valamely  $B$  bázishoz tartozik, ekkor  $\mathbf{y}^* = \mathbf{c}_B^T B^{-1} \geq \mathbf{0}$ . Az optimalitási kritériumot felhasználva azt kapjuk, hogy

- $j \in \mathcal{J}_B$ ,  $l_j < x_j^* < u_j$  esetén  $z_j = 0$  és  $s_j = 0$  teljesül és így

$$-\mathbf{a}_j^T \mathbf{y} = c_j$$

- $j \in \mathcal{J}_N^l$ ,  $l_j = x_j^*$  esetén  $z_j \geq 0$  és  $s_j = 0$  teljesül, tehát

$$z_j = c_j + \mathbf{a}_j^T \mathbf{y} \geq 0$$

- $j \in \mathcal{J}_N^u$ ,  $u_j = x_j^*$  esetén  $z_j = 0$  és  $s_j \geq 0$  teljesül, tehát

$$-s_j = c_j + \mathbf{a}_j^T \mathbf{y} \leq 0$$

adódik. A fentiek alapján azt kapjuk, hogy az  $\mathbf{x}^* \in \mathcal{D}$  lehetséges bázismegoldás pontosan akkor optimális, ha

$$\mathbf{y}^* = \mathbf{c}_B^T B^{-1} \geq \mathbf{0}, \quad (3.5.1)$$

$$-\mathbf{c}_B^T B^{-1} \mathbf{a}_j \leq c_j \quad \text{bármely } j \in \mathcal{J}_N^l \text{ és} \quad (3.5.2)$$

$$-\mathbf{c}_B^T B^{-1} \mathbf{a}_j \geq c_j \quad \text{bármely } j \in \mathcal{J}_N^u \quad (3.5.3)$$

index esetén.

### 3.5.3. Elégséges optimalitási kritérium

Ebben a részben megfogalmazzuk és igazoljuk a  $(P)$  feladat *elégséges optimalitási kritériumát* a  $\mathcal{D}$  halmaz extrémális pontjára, bázismegoldására nézve.

Definiáljuk azt a  $\mathcal{H}$  halmazt, amely megadja a lineáris közelítő függvények valamilyen módon előállítható együtthatóit. A  $\mathcal{H} \subseteq \mathbb{R}^n$  halmaz később igen fontos szerepet kap vizsgálatunk során.

A  $\mathcal{H}$  halmaznak olyannak kell lennie, hogy ha elméletileg ismernénk a  $\mathcal{H}$  halmaz elemeihez tartozó összes lineáris programozási feladat megoldását, akkor az eredeti  $(P)$  feladatnak is meg kell tudjuk határozni az optimális megoldását. Ha ez nem

teljesülne, akkor eleve reménytelen lenne egy ilyen feladatot alsó lineáris közelítésen alapuló módszerekkel megoldanunk.

Először a  $\mathcal{H}$  halmazról általánosságban beszélünk és legfontosabb tulajdonságait használjuk, majd pedig később megadunk olyan – lehetőleg minél szűkebb – halmazokat, amelyek tartalmazzák a  $\mathcal{H}$  halmazt.

A következő lemmában igazoljuk, hogy a  $(P)$  feladat  $\hat{\mathbf{x}}$  optimális megoldásához tartozik egy  $\mathbf{h} \in \mathbb{R}^n$  vektor, amely esetén a relaxált lineáris programozási feladat optimális megoldáshalmazának az  $\hat{\mathbf{x}}$  eleme, azaz  $\hat{\mathbf{x}} \in \mathcal{D}_{\mathbf{h}}^*$ .

**Lemma 3.5.1** *Legyen adott a  $(P)$  feladat. Jelölje  $\hat{\mathbf{x}}$  a  $(P)$  optimális megoldását, azaz  $f(\hat{\mathbf{x}}) = \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})$ . Ekkor*

$$\bar{f}(\hat{\mathbf{x}}) = \min_{\mathbf{x} \in \mathcal{D}} \bar{f}(\mathbf{x}),$$

ahol az  $\bar{f}(\mathbf{x}) = (\nabla f(\hat{\mathbf{x}}))^T(\mathbf{x} - \hat{\mathbf{x}}) + f(\hat{\mathbf{x}})$ , a (3.2.1) képlettel definiált affin (lineáris) függvény.

**Bizonyítás.** A következő egyenlőtlenség az  $f$  függvény konkavitása miatt teljesül,

$$f(\mathbf{x}) \leq \bar{f}(\mathbf{x}) = (\nabla f(\hat{\mathbf{x}}))^T(\mathbf{x} - \hat{\mathbf{x}}) + f(\hat{\mathbf{x}}),$$

és egyenlőség csak a  $\hat{\mathbf{x}}$  pontba áll fenn, azaz  $f(\hat{\mathbf{x}}) = \bar{f}(\hat{\mathbf{x}})$ . Tekintsük azt a lineáris programozási relaxációt amelynél az  $\bar{f}(\mathbf{x})$  a feladat célfüggvénye. Ekkor

$$f(\hat{\mathbf{x}}) = \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) \leq \min_{\mathbf{x} \in \mathcal{D}} \bar{f}(\mathbf{x}) \leq \bar{f}(\hat{\mathbf{x}}) = f(\hat{\mathbf{x}})$$

amiből a

$$\min_{\mathbf{x} \in \mathcal{D}} \bar{f}(\mathbf{x}) = \bar{f}(\hat{\mathbf{x}})$$

adódik. □

A lemma feltételezi az  $f_j$  deriválhatóságát a  $[l_j, u_j]$  intervallumon. Könnyű megmondani, hogy az  $f_j$  nem deriválható pontjaiban a szuperderiváltak halmazának tetszőleges pontja is megfelelő az  $\bar{f}$  választásakor.

Vagyis megmutattuk, hogy létezik olyan  $\mathbf{h}$  célfüggvény vektor, amely esetén a relaxált lineáris programozási feladat optimális bázismegoldása, a  $(P)$  feladat optimális

megoldása is egyben. A  $\mathcal{H}$  halmaznak tehát tartalmaznia kell a lemmában használt  $\nabla f(\mathbf{x})$  vektorokat, ahol  $\mathbf{x} \in \mathcal{D}$ .

Bármely  $\bar{\mathbf{x}} \in \mathcal{D}$  lehetséges megoldáshoz elkészíthetjük a

$$\mathcal{C}_B = \{\mathbf{c} \in \mathbb{R}^n : \text{a } \mathbf{c} \text{ kielégíti a (3.5.1)–(3.5.3) feltételeket}\} \quad (3.5.4)$$

halmazt. A  $\mathcal{C}_B$ , az olyan célfüggvény együtthatókat tartalmazza, amelyek esetén a  $B$  bázissal adott  $\bar{\mathbf{x}}$  lehetséges megoldás optimális megoldása lesz a

$$\min_{\mathbf{x} \in \mathcal{D}} \mathbf{c}^T \mathbf{x} \Big\} \quad (P_c)$$

lineáris programozási feladatnak. Természetesen a  $\mathcal{C}_B$  halmaz nem üres. Könnyen igazolható a következő állítás, amely a szeparábilis konkáv minimalizálási feladat valamely lineáris programozási feladattal történő közelítéséről szól.

**Állítás 3.5.4** *Legyen adott az  $\bar{\mathbf{x}} \in \mathcal{D}$  lehetséges bázismegoldás, a  $B$  bázissal, és legyen  $\bar{\mathbf{h}} \in \mathcal{C}_B$ , akkor az  $\bar{\mathbf{x}}$  optimális bázismegoldása a*

$$\min_{\mathbf{x} \in \mathcal{D}} \bar{\mathbf{h}}^T \mathbf{x} \Big\} \quad (P_{\bar{\mathbf{h}}})$$

*lineáris programozási feladatnak, azaz  $\bar{\mathbf{x}} \in \mathcal{D}_{\bar{\mathbf{h}}}^*$ , ahol  $\mathcal{D}_{\bar{\mathbf{h}}}^*$  jelöli a  $(P_{\bar{\mathbf{h}}})$  feladat optimális megoldásainak a halmazát.* □

Ebből az is következik, hogy

$$\text{ha } \mathcal{H} \subseteq \mathcal{C}_B, \quad \text{akkor } \bar{\mathbf{x}} \in \mathcal{D}_{\bar{\mathbf{h}}}^* \quad (3.5.5)$$

teljesül, bármely  $\mathbf{h} \in \mathcal{H}$  esetén.

Készen állunk arra, hogy a  $(P)$  feladat *elégséges optimalitási feltételét* megfogalmazzuk és igazoljuk.

**Tétel 3.5.1** *Tekintsük a  $(P)$  lineáris feltételes szeparábilis konkáv minimalizálási feladatot és tegyük fel, hogy az  $f_j$  függvények szigorúan konkávak. Legyen  $\bar{\mathbf{x}} \in \mathcal{D}$  egy olyan  $B$  bázissal adott lehetséges bázismegoldás, amely esetén  $\mathcal{H} \subseteq \mathcal{C}_B$  teljesül. Ekkor  $\mathcal{D}^* = \{\bar{\mathbf{x}}\}$ .*



**Bizonyítás.** Mivel  $\mathcal{H} \subseteq \mathcal{C}_B$ , ezért  $\bar{\mathbf{x}} \in \mathcal{D}_h^*$  teljesül, bármely  $\mathbf{h} \in \mathcal{H}$  esetén. Másfelől tudjuk, hogy a  $(P)$  lineáris feltételes szeparábilis konkáv minimalizálási feladatnak létezik olyan globális  $\hat{\mathbf{x}}$  minimuma a  $\mathcal{D}$  halmazon, amely extrémális pontja a megengedett megoldások halmazának, vagyis bázismegoldása a feltételeknek. Tegyük fel, hogy  $\hat{\mathbf{x}} \neq \bar{\mathbf{x}}$ .

Legyen  $\hat{\mathbf{h}} = \nabla f(\hat{\mathbf{x}})$ . A 3.5.1 Lemma miatt  $\hat{\mathbf{x}} \in \mathcal{D}_{\hat{\mathbf{h}}}^*$ , másfelől  $\hat{\mathbf{h}} \in \mathcal{H} \subseteq \mathcal{C}_B$  miatt az  $\bar{\mathbf{x}} \in \mathcal{D}_{\hat{\mathbf{h}}}^*$  teljesül. Így fennáll a következő

$$f(\hat{\mathbf{x}}) = \bar{f}(\hat{\mathbf{x}}) = \bar{f}(\bar{\mathbf{x}}) > f(\bar{\mathbf{x}}). \quad (3.5.6)$$

Ami ellentmondáshoz vezet, vagyis  $\hat{\mathbf{x}} = \bar{\mathbf{x}}$ , amiből  $\mathcal{D}^* = \{\bar{\mathbf{x}}\}$  adódik.  $\square$

A szigorú egyenlőtlenség a szigorú konkavitási feltételből adódik. Ha a 3.5.1 Tétel feltételei közül elhagyjuk a szigorú konkavitási megköötéseket, akkor a (3.5.6) egyenlőtlenség a következő alakú lesz:

$$f(\bar{\mathbf{x}}) \geq f(\hat{\mathbf{x}}) = \bar{f}(\hat{\mathbf{x}}) = \bar{f}(\bar{\mathbf{x}}) \geq f(\bar{\mathbf{x}}),$$

és ekkor  $f(\bar{\mathbf{x}}) = f(\hat{\mathbf{x}})$  teljesül, tehát az  $\bar{\mathbf{x}} \in \mathcal{D}^*$ , de nem biztosítható a  $|\mathcal{D}^*| = 1$ .

Ezzel beláttuk, hogy a  $(P)$  feladat egy  $B$  bázishoz kapcsolódó  $\bar{\mathbf{x}} \in \mathcal{D}$  bázismegoldásának az *elégséges optimalitási kritériuma* a

$$\mathcal{H} \subseteq \mathcal{C}_B.$$

### Degenerált bázismegoldás

Jelölje  $\mathcal{C}_{\bar{\mathbf{x}}}$  azon célfüggvény együtthatók halmazát, amelyre az  $\bar{\mathbf{x}}$  optimális megoldása lesz a  $\min_{\mathbf{x} \in \mathcal{D}} \mathbf{c}^T \mathbf{x}$  lineáris programozási feladatnak.

A primál degenerált bázis esetén a meghatározott  $\mathcal{C}_B$  halmaz szűkebb lesz, mint a csúcsponthoz tartozó  $\mathcal{C}_{\bar{\mathbf{x}}}$  halmaz, így ha az optimalitási kritérium áll a  $\mathcal{C}_B$  halmazra, akkor ez igaz lesz egy  $\mathcal{C}_B$ -nél bővebb  $\mathcal{C}_{\bar{\mathbf{x}}}$  halmazra is.

Ha több információt szeretnénk összeszedni, akkor akár exponenciálisan sok, ugyanazt a csúcsot leíró bázissal kellene dolgoznunk, ami az amúgy is nehéz feladatot egy másik szempontból tenné nehezzé. Egy primál degenerált bázisból dolgozva az lehet a gond, hogy már optimális a megoldásunk, azaz  $\mathcal{H} \subseteq \mathcal{C}_{\bar{\mathbf{x}}}$ , de mivel mi csak  $\mathcal{C}_B$ -t

ismerjük, ezért nem tudunk az optimalitás következtetésére jutni. Ez összhangot mutat a lineáris programozási feladat elemzésekor kapott eredményekkel (lásd [10], [42], [74]).

### 3.5.4. A $\mathcal{H}$ halmazról

A  $\mathcal{H}$  halmaz (alsó) közelítéseket tartalmaz, ezért szoros kapcsolatban van az  $f$  függvény deriváltjaival (szuperderivált is lehet), hiszen a Lagrange középérték tétel miatt minden alsó közelítéshez létezik egy pont, ahol az alsó közelítés meredeksége a pontbeli derivált.

Az optimalitás vizsgálata a konvex  $\mathcal{C}_B$  poliéder és a  $\mathcal{H}$  halmaz tartalmazásának, illetve általában a két halmaz egymáshoz való viszonyának a vizsgálata. A  $\mathcal{H}$  halmaz meghatározásakor azt is figyelembe kell venni, hogy a viszony könnyen vizsgálható legyen.

A  $\mathcal{H}$  halmaz meghatározására tekintsük például az  $f$  függvény deriváltjainak (szuperderiváltaknak) az értékkészletét a  $\mathcal{D}$  halmaz felett. Ha  $f$  szigorúan konkáv, akkor  $f_j$  deriváltja szigorúan monoton csökkenő, így van neki  $g_j$  inverze. Ekkor az

$$\mathcal{F} = \{\mathbf{y} : A\mathbf{g}(\mathbf{y}) = \mathbf{b} \text{ és } \mathbf{l} \leq \mathbf{g}(\mathbf{y}) \leq \mathbf{u}\}$$

halmaz az  $f$  deriváltjának értékkészlete  $\mathcal{D}$  halmaz felett, ami jó  $\mathcal{H}$  halmaznak. Az  $\mathcal{F}$  bonyolult struktúrájú halmaz lehet, és a tartalmazás eldöntése hasonlóan nehéz feladat lenne, mint a  $(P)$  feladat megoldása.

Megtehetjük, hogy az  $\mathcal{F}$  halmaznál bővebb halmazt választunk a  $\mathcal{H}$  halmaznak úgy, hogy struktúrája egyszerűbb lesz, mint az  $\mathcal{F}$  halmazé.

Nyilvánvaló, hogy a  $\mathcal{H}$  meghatározását az  $f$  függvény tulajdonságai (szigorú konkavitás, differenciálhatóság stb.) jelentősen befolyásolják. Másfelől, ha a  $\mathcal{H}$  halmaz struktúrája bonyolult (nem poliéder), akkor a  $\mathcal{H} \subseteq \mathcal{C}_B$  összefüggést ellenőrizni igen nehéz lehet. Ezért érdemes a  $\mathcal{H}$  halmazt tartalmazó, de egyszerű struktúrájú (pl. hipertégla) célfüggvény-együttható paraméterhalmazt meghatározni. Ha csak a  $(P)$  feladat adataira támaszkodunk, akkor a

$$\mathcal{H}_f = \{\mathbf{h} \in \mathbb{R}^n : h_j \in [f'_{j-}(u_j), f'_{j+}(l_j)]\}$$

halmazt tudjuk definiálni és nyilván  $\mathcal{H} \subseteq \mathcal{H}_f$  teljesül. Ha azonban valamely  $\bar{\mathbf{x}} \in \mathcal{D}$  lehetséges bázismegoldás esetén szeretnénk meghatározni a relaxált lineáris programozási feladatok szóba jövő célfüggvény-együtthatóit, akkor a

$$\mathcal{H}_{f,\bar{\mathbf{x}}} = \{\mathbf{h} \in \mathbb{R}^n : h_j \in [c_j^l, c_j^u]\}$$

halmazban gyűjthetjük össze a célfüggvény együtthatókat, ahol

$$c_j^u = \begin{cases} m(l_j, x_j^*), & x_j^* \neq l_j \\ f'_{j+}(l_j), & \text{különben} \end{cases} \quad \text{és} \quad c_j^l = \begin{cases} m(x_j^*, u_j), & x_j^* \neq u_j \\ f'_{j-}(u_j), & \text{különben} \end{cases}$$

A 3.5.1 és 3.5.2 állítások alapján

$$f'_{j-}(u_j) \leq c_j^l = m(\bar{x}_j, u_j) \leq m(l_j, \bar{x}_j) = c_j^u \leq f'_{j+}(l_j) \quad (3.5.7)$$

egyenlőtlenségek teljesülnek, így  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{H}_f$ , azaz az  $\bar{\mathbf{x}} \in \mathcal{D}$  bázismegoldásból nyerhető információt felhasználva a relaxált programozási feladatok célfüggvény együtthatóinak egy szűkebb halmazát tudtuk meghatározni. Philips és Rosen cikkükben [81] a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmazt vezették be.

A  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$  feltételből nem következik a  $\mathcal{H} \subseteq \mathcal{C}_B$ , ezért a  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$  mint elégséges optimalitási kritériumot a következő 3.5.2 tételben fogalmazom meg.

**Tétel 3.5.2** *Tekintsük a (P) lineáris feltételes szeparábilis konkáv minimalizálási feladatot és tegyük fel, hogy az  $f_j$  függvények konkávak. Legyen  $\bar{\mathbf{x}} \in \mathcal{D}$  egy olyan B bázissal adott bázismegoldás, amely esetén  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$  teljesül. Ekkor  $\bar{\mathbf{x}} \in \mathcal{D}^*$ .*

**Bizonyítás.** Legyen  $\hat{\mathbf{x}}$  a globális optimuma a (P) feladatnak, és tegyük fel, hogy  $\hat{\mathbf{x}} \neq \bar{\mathbf{x}}$  teljesül. Legyen

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : \min\{\bar{x}_i, \hat{x}_i\} \leq x_i \leq \max\{\bar{x}_i, \hat{x}_i\}, i = 1, \dots, n\}$$

hipertégla. Az  $\hat{\mathbf{x}} \in \mathcal{S}$  és  $\bar{\mathbf{x}} \in \mathcal{S}$  teljesül, továbbá elmondható, hogy mindkettő az  $\mathcal{S}$  hipertégla egy-egy csúcsa (extremális pontja). Legyen  $F(\mathbf{x}) = \hat{\mathbf{h}} \mathbf{x} + \mathbf{d}$  az  $f$  függvény relaxáltja az  $\mathcal{S}$  hipertégla felett ( $F(\mathbf{x}) \leq f(\mathbf{x})$ , minden  $\mathbf{x} \in \mathcal{S}$  esetén). Ha valamely  $j$  indexre  $\bar{x}_j = \hat{x}_j$  teljesül, akkor a pontbeli érintő legyen a lineáris relaxáció (nem

deriválható esetben valamely szupergradiens). Könnyen belátható, hogy  $\hat{\mathbf{h}} \in \mathcal{H}_{f,\bar{\mathbf{x}}}$  feltétel fennáll, amiből  $\bar{x} \in \mathcal{D}_{\mathbf{h}}^*$  következik. A következő egyenletek teljesülnek:

$$f(\hat{\mathbf{x}}) = \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathcal{S} \cap \mathcal{D}} f(\mathbf{x}), \quad \text{hiszen } \hat{\mathbf{x}} \in \mathcal{S}, \quad (3.5.8)$$

$$F(\bar{\mathbf{x}}) = \min_{\mathbf{x} \in \mathcal{D}} F(\mathbf{x}) = \min_{\mathbf{x} \in \mathcal{S} \cap \mathcal{D}} F(\mathbf{x}), \quad \text{hiszen } \bar{\mathbf{x}} \in \mathcal{S}. \quad (3.5.9)$$

A (3.5.8) és (3.5.9) egyenletekből kapjuk a kívánt összefüggést:

$$f(\bar{\mathbf{x}}) = F(\bar{\mathbf{x}}) = \min_{\mathbf{x} \in \mathcal{S} \cap \mathcal{D}} F(\mathbf{x}) \leq \min_{\mathbf{x} \in \mathcal{S} \cap \mathcal{D}} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}).$$

Az első egyenlőség annak a következménye, hogy  $\bar{\mathbf{x}}$  az  $\mathcal{S}$  hipertégla egy csúcsa, és így abban a pontban a relaxációs függvény megegyezik a célfüggvénnyel.  $\square$

Beláttuk, hogy az optimalitás következik a  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$  feltételből, viszont a  $|\mathcal{D}^*| = 1$  tulajdonság már nem garantálható.

A következőkben azt a kérdést szeretnénk megvizsgálni, hogy a

$$\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$$

tartalmazás ellenőrzése mennyi számolást igényel. Philips és Rosen [81] exponenciálisan sok lineáris programozási feladat megoldására vezette vissza a kérdést: ha ezeknek a lineáris programozási feladatoknak van közös optimális megoldása, akkor az egyben optimális megoldása a  $(P)$  feladatnak is. Könnyen belátható, hogy elegendő a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  hipertégla extrémális pontjairól eldönteni azt, hogy eleme-e a  $\mathcal{C}_B$  halmaznak vagy sem. Ezzel jelentős mennyiségű számítást takaríthatunk meg, de sajnos még mindig exponenciálisan sok pont ellenőrzéséről van szó. Ezt mi tesztpont segítségével jóval hatékonyabbá tesszük.

### Tesztpontok előállítása

A  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$  tartalmazás ellenőrzése helyett olyan tesztpontot szeretnénk előállítani bármely (3.5.1)–(3.5.3) feltételrendszerben szereplő egyenlőtlenséghez, amely lehetőleg megsérti az egyenlőtlenséget. A tesztpontot természetesen a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmazból választjuk ki. A (3.5.1)–(3.5.3) egyenlőtlenség-rendszerben a célfüggvény együtthatói a változók jelenleg.

A tesztpont elkészítését vizsgáljuk meg a  $j \in \mathcal{J}_N^l$  esetén, azaz a

$$-\mathbf{c}_B^T B^{-1} \mathbf{a}_j = -\mathbf{c}_B^T \bar{\mathbf{a}}_j \leq c_j$$

egyenlőtlenséget megsértő célfüggvény együtthatókat keressük a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmazból. Ez azt jelenti, hogy az egyenlőtlenség baloldalát szeretnénk minél nagyobbra, míg a jobboldalát a lehető legkisebbre választani. Ennek érdekében definiáljuk a  $\bar{\mathbf{h}}_j$  tesztpontot a következő módon.

$$\bar{h}_{ij} = \begin{cases} c_j^l, & i = j \\ c_j^l, & \bar{a}_{ij} > 0, i \in \mathcal{J}_B \\ c_j^u, & \bar{a}_{ij} < 0, i \in \mathcal{J}_B \\ h_{ij}, & i \notin (\mathcal{J}_B \setminus \{i : \bar{a}_{ij} = 0\}) \cup \{j\}, \text{ ahol } h_{ij} \in [c_i^l, c_i^u] \end{cases}$$

Ekkor nyilvánvaló, hogy  $\bar{\mathbf{h}}_j \in \mathcal{H}_{f,\bar{\mathbf{x}}}$  teljesül. A tesztpont konstrukciója alapján világos, hogy

$$\bar{\mathbf{h}}_B^T \bar{\mathbf{a}}_j + \bar{h}_{jj} \leq \mathbf{h}_B^T \bar{\mathbf{a}}_j + h_{jj}$$

teljesül bármely  $\mathbf{h} \in \mathcal{H}_{f,\bar{\mathbf{x}}}$  esetén, azaz

$$-\bar{\mathbf{h}}_B^T \bar{\mathbf{a}}_j - \bar{h}_{jj} \geq -\mathbf{h}_B^T \bar{\mathbf{a}}_j - h_{jj} \quad (3.5.10)$$

adódik. Amennyiben a tesztpont nem sérti meg a feltételt, azaz

$$0 \geq -\bar{\mathbf{h}}_B^T \bar{\mathbf{a}}_j - \bar{h}_{jj} \quad (3.5.11)$$

fennáll, akkor a (3.5.10) és (3.5.11) egyenlőtlenség alapján nincsen olyan pontja a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmaznak, amely a  $j \in \mathcal{J}_N^l$  feltételt megsértené. Általánosítva az előzőket, tetszőleges  $k \in \mathcal{J}_N^l \cup \mathcal{J}_N^u$  index esetén a  $\bar{\mathbf{h}}_k$  tesztpontot az alábbi módon definiálhatjuk a  $\mathcal{J}_i^+$  és a  $\mathcal{J}_i^-$ , ( $i \in \mathcal{J}_B$ ) halmazok segítségével, ahol

$$\mathcal{J}_i^+ = \{k \in \mathcal{J}_N^l : \bar{a}_{ik} < 0\} \cup \{k \in \mathcal{J}_N^u : \bar{a}_{ik} > 0\}, \text{ és} \quad (3.5.12)$$

$$\mathcal{J}_i^- = \{k \in \mathcal{J}_N^l : \bar{a}_{ik} > 0\} \cup \{k \in \mathcal{J}_N^u : \bar{a}_{ik} < 0\}, \quad (3.5.13)$$

az alábbi módon

$$\bar{h}_{ik} = \begin{cases} c_i^l, & k \in \mathcal{J}_i^-, i \in \mathcal{J}_B \\ c_i^u, & k \in \mathcal{J}_i^+, i \in \mathcal{J}_B \\ c_k^l, & i = k, \text{ és } k \in \mathcal{J}_N^l \\ c_k^u, & i = k, \text{ és } k \in \mathcal{J}_N^u \\ h_i, & i \notin (\mathcal{J}_B \setminus \{i : \bar{a}_{ik} = 0\}) \cup \{k\}, \text{ ahol } h_i \in [c_i^l, c_i^u]. \end{cases}$$

Összegezve, a következő állítást kapjuk.

**Tétel 3.5.3** *Ha a  $\bar{\mathbf{h}}_k$  tesztpont nem sérti meg a  $k \in \mathcal{J}_N^l \cup \mathcal{J}_N^u$  egyenlőséget, akkor tetszőleges  $\mathbf{h} \in \mathcal{H}_{f,\bar{\mathbf{x}}}$  vektor sem sérti meg.*  $\square$

Másfelől, ha valamely  $j \in \mathcal{J}_N^l$  ( $j \in \mathcal{J}_N^u$ ) esetén

$$-\bar{\mathbf{h}}_{B,j}^T \bar{\mathbf{a}}_j > c_j^l \quad (-\mathbf{h}_{B,j}^T \bar{\mathbf{a}}_j < c_j^u),$$

akkor a tesztpont megsérti a  $j$ . változóhoz tartozó optimalitási kritériumot.

Hasonló módon készíthetünk tesztpontot a

$$\bar{\mathbf{h}}_B^T B^{-1} \geq \mathbf{0}$$

tesztelésére is. Jelölje  $\bar{B} = B^{-1}$  mátrixot és ekkor  $\bar{\mathbf{b}}_i$  a  $\bar{B}$  mátrix  $i$ . oszlopa

$$\bar{h}_{ji} = \begin{cases} c_i^l, & b_{ji} > 0, j \in \mathcal{J}_B \\ c_i^u, & b_{ji} < 0, j \in \mathcal{J}_B \\ h_i, & j \in \mathcal{J}_N^l \cup \mathcal{J}_N^u \cup \{j \in \mathcal{J}_B : \bar{b}_{ji} = 0\} \text{ ahol } h_i \in [c_j^l, c_j^u] \end{cases}$$

Ekkor a  $\bar{\mathbf{h}}_{j,B}^T \bar{\mathbf{b}}_i \geq 0$  és bármelyik másik  $\mathbf{h} \in \mathcal{H}_{f,\bar{\mathbf{x}}}$  vektor is kielégíti az  $i$ . nemnegativitási feltételt.

Ez azt jelenti, hogy a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  hipertégla  $2^n$  csúcspontja helyett elegendő legfeljebb  $n$  tesztpont elkészítése annak érdekében, hogy az összes optimalitási feltételt leteszteljük.

Vezessük be a  $\hat{\mathcal{K}}$  indexhalmazt az alábbi módon

$$\hat{\mathcal{K}} = \{i : \bar{\mathbf{h}}_i \text{ tesztpont megsérti az } i. \text{ egyenlőtlenséget}\}.$$

Nyilván igaz, hogy a  $\hat{\mathcal{K}} = \emptyset$  esetén  $\mathcal{H}_{f,\bar{\mathbf{x}}} \subseteq \mathcal{C}_B$ , azaz az  $\bar{\mathbf{x}} \in \mathcal{D}^*$  teljesül. Ez azt jelenti, hogy valamely  $\bar{\mathbf{x}} \in \mathcal{D}$  pontról a következő módon dönthető el, hogy optimális megoldása-e a  $(P)$  feladatnak:

1. elkészítjük a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmazt;
2. figyelembe véve a  $B^{-1}$  és a  $B^{-1}A_N$  mátrix elemeit elkészítjük a  $\bar{\mathbf{h}}_j$  tesztpontot;

3. elvégezzük a tesztpontok ellenőrzését; ha nem találunk olyan  $j$  indexet, amelyre  $\bar{\mathbf{h}}_j$  megsérti a  $j$ . feltételt akkor az  $\bar{\mathbf{x}}_j$  optimális megoldása a feladatnak.

Ha azonban találunk olyan  $\bar{\mathbf{h}}_j$  tesztpontot, amely megsérti a  $j$ . feltételt, akkor abból sajnos nem vonhatjuk le azt a következtetést, hogy az  $\mathbf{x} \in \mathcal{D}$  nem optimális megoldás.

Mivel a  $\mathcal{H}$  és a  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmazok jelentősen függnak az  $l_j$  és  $u_j$  számoktól is, ezért várható, hogy a korlátozás és szétválasztás típusú algoritmusok hatékonyak lehetnek a  $(P)$  feladatok megoldására, ha  $\mathcal{H}_{f,\bar{\mathbf{x}}}$  halmaz átmérője gyorsan csökken.

### 3.5.5. Az eredmény rövid összefoglalása

A 3.5 fejezet a 2a tézispontban megfogalmazott eredményeket tartalmazza.

Megadtam a lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat egy elégséges optimalitási kritériumát.

## 3.6. Érzékenységi vizsgálaton alapuló vágási stratégia

Az elégséges optimalitási feltétel megfogalmazásakor említettük, hogy az itt megfogalmazott gondolatmenet alapul szolgálhat egy, a korlátozás és szétválasztás módszerén alapuló eljárás kidolgozásához.

Mielőtt rátérnénk a módszer bevezetésére a  $(P)$  feladatot módosítottuk: az egyenlőtlenségek helyett itt most egyenlőséget használunk. A módosítást azért vezettük be, mert a gyakorlatban használt LP megoldóknál általában a "slack" változókat nem kezelik külön, így azokat be kell venni az adatok feltöltésénél.

$$\left. \begin{array}{l} \min \sum_{j=1}^n f_j(x_j) \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} (P). \quad (3.6.1)$$

Bevezetünk egy partícionálási stratégiát, amely a célfüggvény együtthatóinak érzékenységi vizsgálatán alapul.

A módosítással a feladathoz tartozó  $P_{LP}$  relaxált feladat optimalitási kritériuma megváltozott, így a (3.5.1) feltétel már nem része a kritériumnak. Természetesen a  $\mathcal{C}_B$  halmaz is megfelelően módosult.

### 3.6.1. Egy bázisváltó költsége módosul

Tegyük fel, hogy adott az  $\bar{\mathbf{x}} \in \mathcal{D}_c^*$  megoldás a  $B$  bázissal. Legyen  $k \in \mathcal{J}_B$ , és

$$\mathbf{c}(k) = \mathbf{c} + \gamma_k \mathbf{e}_k = (c_1, c_2, \dots, c_k + \gamma_k, \dots, c_n),$$

ahol az  $\mathbf{e}_k$  az  $\mathbb{R}^n$  vektortér  $k$ . egységvektora. Az a kérdés, hogy meddig marad a  $B$  bázis optimális, azaz meddig teljesülnek a következő egyenlőtlenségek

$$c_j \geq -\mathbf{c}(k)^T B^{-1} \mathbf{a}_j = -\mathbf{c}_B^T B^{-1} \mathbf{a}_j - \gamma_k \bar{a}_{kj} \quad (3.6.2)$$

bármely  $j \in \mathcal{J}_N^l$ , illetve

$$c_j \leq -\mathbf{c}(k)^T B^{-1} \mathbf{a}_j = -\mathbf{c}_B^T B^{-1} \mathbf{a}_j - \gamma_k \bar{a}_{kj} \quad (3.6.3)$$

bármely  $j \in \mathcal{J}_N^u$  indexek esetén. A korábban bevezetett  $\mathcal{J}_k^+$ ,  $\mathcal{J}_k^-$  indexhalmazokat (lásd (3.5.12) és (3.5.13) egyenletek) használva definiáljunk olyan alsó és felső korlátot a  $\gamma_k$  számra, amely esetén a (3.6.2) és (3.6.3) egyenlőtlenségek fennállnak, azaz legyen

$$\gamma_k^- = \begin{cases} \max_{j \in \mathcal{J}_k^-} \frac{c_j + \mathbf{c}_B^T B^{-1} \mathbf{a}_j}{\bar{a}_{kj}} & \text{ha } \mathcal{J}_k^- \neq \emptyset \\ -\infty & \text{ha } \mathcal{J}_k^- = \emptyset \end{cases}$$

illetve

$$\gamma_k^+ = \begin{cases} \min_{j \in \mathcal{J}_k^+} \frac{c_j + \mathbf{c}_B^T B^{-1} \mathbf{a}_j}{\bar{a}_{kj}} & \text{ha } \mathcal{J}_k^+ \neq \emptyset \\ +\infty & \text{ha } \mathcal{J}_k^+ = \emptyset \end{cases}$$

és ekkor

$$\gamma_k^- \leq \gamma_k \leq \gamma_k^+$$

esetén a  $B$  bázis optimális marad. Nyilvánvaló, hogy a  $0 \in [\gamma_k^-, \gamma_k^+]$  és legalább az egyik érték a  $\gamma_k^-$  és  $\gamma_k^+$  közül véges.

A korlátok alapján számos esetben meghatározhatjuk a (rész)feladat további particionálását. Ennek elemeit dolgozzuk ki a következő részben.



### Particionálás

Célunk az, hogy lehetőleg egyszerű számítások segítségével olyan  $p$  vágási pontot határozzunk meg, amellyel valamely  $x_j$  változóra adott  $[l_j, u_j]$  intervallumot,  $[l_j, p]$  és  $[p, u_j]$  intervallumokra bontsuk úgy, hogy a relaxált LP optimális megoldásának a  $j$ . koordinátája az egyik intervallumba kerüljön. Felhasználva az  $f_j$  függvények konkavitását, definiáljuk a következő halmazokat:

$$\begin{aligned}\mathcal{K}_- &= \{j \in \mathcal{J}_B : c_j^l < c_j + \gamma_j^- < c_j^u\} \\ \mathcal{K}_+ &= \{j \in \mathcal{J}_B : c_j^l < c_j + \gamma_j^+ < c_j^u\}.\end{aligned}$$

Vezessük be továbbá a  $\mathcal{K} = \mathcal{K}_- \cup \mathcal{K}_+$  jelölést. Ekkor a  $\mathcal{K}$  azon indexek halmaza, amelyek esetén a célfüggvény  $j$ . együtthatója módosítható úgy, hogy az új LP feladatnak a felbontása után (legalább) az egyikben más legyen az optimális megoldása, mint eredetileg volt. Legyen

$$\tau = \max_{j \in \mathcal{K}} \{f_j(\bar{x}_j) - (c_j \bar{x}_j + d_j)\}, \quad (3.6.4)$$

ahol  $\mathbf{c}$  az (LP) feladat célfüggvény együtthatója. Legyen  $k$  index az, amelyre

$$\tau = f_k(\bar{x}_k) - (c_k \bar{x}_k + d_k).$$

Ha  $k \in \mathcal{K}_-$ , azaz

$$c_k^l < c_k + \gamma_k^- < c_k^u$$

teljesül, akkor létezik  $p \in (l_k, u_k)$ , amelyre

$$c_k + \gamma_k^- = f'_k(p) \quad (3.6.5)$$

Ha  $k \in \mathcal{K}_+$ , azaz

$$c_k^l < c_k + \gamma_k^+ < c_k^u$$

teljesül, akkor létezik  $p \in (l_k, u_k)$ , amelyre

$$c_k + \gamma_k^+ = f'_k(p). \quad (3.6.6)$$

A  $p$  vágási pont segítségével az aktuális megoldáshalmaz téglá részét a (3.2.2) összefüggésben leírt módon particionálhatjuk.

Abban az esetben ha  $\mathcal{K} = \emptyset$  akkor meg kell vizsgálni annak a lehetőségét, hogy egyszerre több bázisváltozó célfüggvény együtthatóját változtatjuk meg annak érdekében, hogy a megoldás optimalitási tulajdonsága megváltozzon.

### 3.6.2. Több bázisváltó költsége módosul

Ha egy célfüggvény együttható módosításával a bázis optimalitása nem változik meg, akkor azt az általános esetet kell tekintenünk, amikor az összes változó célfüggvény együtthatója változhat.

Feltehetjük, hogy a  $\mathcal{H}_{f,\bar{x}} \not\subseteq \mathcal{C}_B$ , ellenkező esetben a  $\bar{x}$  megoldás optimális megoldása lenne a részfeladatnak. Létezik egy  $\bar{\mathbf{h}}_j$  tesztpont  $j \in \mathcal{J}_N^l$  ( $j \in \mathcal{J}_N^u$ ), amelyre

$$-\bar{\mathbf{h}}_{B,j}^T \bar{\mathbf{a}}_j > c_j^l \quad (-\bar{\mathbf{h}}_{B,j}^T \bar{\mathbf{a}}_j < c_j^u).$$

Vezessük be a  $\hat{\mathcal{K}} = \hat{\mathcal{K}}_l \cup \hat{\mathcal{K}}_u$  indexhalmazt, ahol

$$\begin{aligned} \hat{\mathcal{K}}_l &= \{i \in \mathcal{J}_N^l : -\bar{\mathbf{h}}_{B,i}^T \bar{\mathbf{a}}_i > c_i^l\} \\ \hat{\mathcal{K}}_u &= \{i \in \mathcal{J}_N^u : -\bar{\mathbf{h}}_{B,i}^T \bar{\mathbf{a}}_i < c_i^u\}. \end{aligned}$$

A  $\hat{\mathcal{K}}$  azon indexeknek a halmaza, amelyeknél a  $(P_{LP})$  relaxált lineáris programozási feladat optimalitási kritériumai a tesztpont esetén megsérülnek. A korábbi feltevésből következik, hogy  $\hat{\mathcal{K}} \neq \emptyset$ .

#### Particionálás

Vezessük be a

$$\hat{\tau} = \max_{i \in \hat{\mathcal{K}}} |-\bar{\mathbf{h}}_{B,i}^T \bar{\mathbf{a}}_i - c_i^l|$$

számot. Legyen  $k \in \hat{\mathcal{K}}$  az az index, amelynél a  $\hat{\tau}$  érték felvétetik, azaz

$$\hat{\tau} = |-\bar{\mathbf{h}}_{B,k}^T \bar{\mathbf{a}}_k - c_k^l|.$$

Tekintettel arra, hogy  $\hat{\tau}$  méri az optimalitási kritérium maximális megsértését, ezért a  $k \in \hat{\mathcal{K}} \subseteq \mathcal{J}_N^l \cup \mathcal{J}_N^u$  az az index, amely esetén legjobban sérül az optimalitási kritérium. Tegyük fel, hogy  $k \in \hat{\mathcal{K}}_l$  és legyen

$$j' = \operatorname{argmax}_{i \in \{1,2,\dots,m\}} |-(\bar{h}_{B,ki} - c_{B,i}) \bar{a}_{ki}|. \quad (3.6.7)$$

A  $j'$  itt a  $B$  bázisra megszorított  $\bar{\mathbf{h}}_{B,k}$  vektor egy indexe, az egyszerűség kedvéért jelölje  $j$  az eredeti  $\bar{\mathbf{h}}_k$  vektorra vonatkozó megfelelő indexet. Induljunk ki a  $\hat{\mathcal{K}}_l$  elemeit definiáló egyenlőtlenségből, azaz

$$-\bar{\mathbf{h}}_{B,k}^T \bar{\mathbf{a}}_k > c_k^l,$$

amelyből egyszerű átalakítással kapjuk a

$$-\bar{h}_{kj} \bar{a}_{kj'} > c_k^l + \bar{\mathbf{h}}_{B,k}^T \bar{\mathbf{a}}_k - \bar{h}_{kj} \bar{a}_{kj'}$$

egyenlőtlenséget. Legyen

$$\gamma_j^* = \bar{h}_{kj} + \frac{c_k^l + \bar{\mathbf{h}}_{B,k}^T \bar{\mathbf{a}}_k}{-\bar{a}_{kj'}}. \quad (3.6.8)$$

Ha  $\gamma_j^* \notin (c_j^l, c_j^u)$  teljesül, akkor legyen

$$\gamma_j^* = t_k (\bar{h}_{kj} - c_j) + c_j, \quad (3.6.9)$$

ahol

$$t_k = \frac{c_k^l + \mathbf{c}_B^T \bar{\mathbf{a}}_k}{-(\bar{\mathbf{h}}_{B,k}^T - \mathbf{c}_B^T) \bar{\mathbf{a}}_k}.$$

A  $c_k^l + \mathbf{c}_B^T \bar{\mathbf{a}}_k < 0$  miatt  $0 < t < 1$ , így a  $\gamma_j^* \in (c_j^l, c_j^u)$ . Legyen  $\mathbf{g}_k = \bar{\mathbf{h}}_{B,k} - \mathbf{c}_B$  továbbá  $\bar{c}_k = c_k^l + \mathbf{c}_B^T \bar{\mathbf{a}}_k$ . A  $t_k$  a következő formába írható

$$t_k = \frac{\bar{c}_k}{-\mathbf{g}_k^T \bar{\mathbf{a}}_k},$$

továbbá

$$\gamma_j^* = t_k g_{kj'} + c_j.$$

Már csak a  $p$  vágási pont meghatározása maradt hátra ebben az esetben. Tekintsük az

$$f'(l_j) < \gamma_j^* < f'(u_j)$$

egyenlőtlenségrendszert. Ekkor létezik olyan  $p \in (l_j, u_j)$  szám, amelyre

$$\gamma_j^* = f'(p) \quad (3.6.10)$$

teljesül.

**A partíciónálás elemzése.** Az általánosság megszorítása nélkül feltehetjük, hogy  $k \in \hat{\mathcal{K}}_l$ , ahol legjobban sérül az optimalitási kritérium és  $j$  a vágási irány (azaz  $-\bar{g}_{kj'} \bar{a}_{kj'}$  a maximális), továbbá  $\bar{h}_{kj} = c_j^u$  (azaz  $\bar{a}_{kj'} < 0$  teljesül). A többi esetet teljesen analóg módon tudjuk kezelni.

A felosztás után két részproblémát kaptunk. Vizsgáljuk meg, hogy mi történik a  $k$  feltétellel a felosztás után. A partícionálás után marad a régi bázis, így csak a  $c_j^l$  és a  $c_j^u$  értéke fog változni.

A korábbiakhoz hasonlóan tekintsük a BB algoritmus által generált BB fát. Tegyük fel, hogy az algoritmus végtelen, tehát létezik a részproblémák egy végtelen  $P^q$  sorozata ( $P^{q+1}$  a  $P^q$  közvetlen utóda), melyre  $\mathcal{T}^{q+1} \subset \mathcal{T}^q$  teljesül.

**Állítás 3.6.1** *Az (3.6.8)-el definiált partícionálás véges sok esetben hajtódik végre.*

**Bizonyítás.** Legyen az aktuális részprobléma  $P^q$ . Jelölje  $(q)$  felső index hogy az adott érték mely részproblémára vonatkozik.

Tekintsük azt az esetet, amikor a  $P^{q+1}$  a  $P^q$  baloldali gyereke:  $[c_j^{l(q)}, \gamma_j^{*(q)}]$   
A  $\bar{h}_{kj}^{(q+1)}$  értéke  $\gamma_j^{*(q)}$  lesz.

$$\begin{aligned} -(\bar{\mathbf{h}}_{B,k}^{(q+1)})^T \bar{\mathbf{a}}_k &= - \sum_{i \in \{1, \dots, m\} \setminus \{j'\}} \bar{h}_{B,ki}^{(q)} \bar{a}_{ki} - (\gamma_j^{*(q)}) \bar{a}_{kj'} \\ &= - \sum_{i \in \{1, \dots, m\} \setminus \{j'\}} \bar{h}_{B,ki}^{(q)} \bar{a}_{ki} - \left( \bar{h}_{kj}^{(q)} + \frac{c_k^{l(q)} + (\bar{\mathbf{h}}_{B,k}^{(q)})^T \bar{\mathbf{a}}_k}{-\bar{a}_{kj'}} \right) \bar{a}_{kj'} \\ &= c_k^{l(q)} = c_k^{l(q+1)} \end{aligned}$$

Az  $k$  egyenlet már nem sérül. Mivel az egyenletek száma véges, így ez az eset csak véges sokszor fordulhat elő.

Tekintsük azt az esetet, amikor a  $P^{q+1}$  a  $P^q$  jobboldali gyereke:  $[\gamma_j^{*(q)}, c_j^{u(q)}]$   
A  $\bar{h}_{kj}^{(q+1)}$  értéke a  $c_j^{u(q)}$  lesz. Számoljuk ki ekkor az új  $\gamma_j^{*(q+1)}$  tagot.

$$\begin{aligned} \gamma_j^{*(q+1)} &= \bar{h}_{kj}^{(q+1)} + \frac{c_k^{l(q+1)} + (\bar{\mathbf{h}}_{B,k}^{(q+1)})^T \bar{\mathbf{a}}_k}{-\bar{a}_{kj'}} = \\ &= \bar{h}_{kj}^{(q)} + \frac{c_k^{l(q)} + (\bar{\mathbf{h}}_{B,k}^{(q)})^T \bar{\mathbf{a}}_k}{-\bar{a}_{kj'}} = \\ &= \gamma_j^{*(q+1)} = c_j^{l(q+1)} \end{aligned}$$

$\gamma_j^{*(q+1)} \notin (c_j^{l(q+1)}, c_j^{u(q+1)})$  miatt a  $j$  később már nem lehet vágási irány a (3.6.8) szabály szerint.

Megvizsgálva a keletkezett részproblémákat, arra a következtetésre jutunk, hogy a (3.6.8) szerinti partícionálási szabályt ismételten alkalmazva, véges lépésben eljutunk oda, hogy a részprobléma bázisa már nem változhat (azaz a részfeladat törölhető) vagy a (3.6.9) szabály szerint kell a felosztást elvégeznünk.  $\square$

**Állítás 3.6.2** *(3.6.9)-ban definiált partícionálás lépés véges végrehajtása után vagy az optimalitási feltétel teljesül, vagy egy változó célfüggvényének módosítása bázisváltozást eredményez.*

**Bizonyítás.** Hasonlóan az előző állítás bizonyításhoz itt is külön-külön vizsgáljuk a részproblémákat.

A  $\mathbf{c}$  az aktuálisan relaxált együtthatókat jelenti,  $x_j$  változó újrelaxálása után a  $c_j$  érték fog megváltozni. A változás mértéke legyen  $p = c_j^{(q+1)} - c_j^{(q)}$ , nyilvánvaló, hogy  $p \in (c_j^{l(q)} - c_j^{(q)}, c_j^{u(q)} - c_j^{(q)})$ .

*Baloldali részprobléma:*  $[c_j^{l(q)}, \gamma_j^{*(q)}]$ .

Becsüljük meg a  $k$  egyenlet mennyivel sérül a partícionálás után. Vezessük be az eredeti eltérésre következő jelölést:

$$d^{(q)} = -(\bar{\mathbf{h}}_{B,k}^{(q)})^T \bar{\mathbf{a}}_k - c_k^{l(q)} = -(\mathbf{g}_k^{(q)})^T \bar{\mathbf{a}}_k - \bar{c}_k^{(q)}.$$

Számítsuk ki  $d^{(q+1)}$ -t a partícionálás után.

$$\begin{aligned} d^{(q+1)} &= - \sum_{i \in \{1, \dots, m\} \setminus \{j'\}} g_{ki}^{(q+1)} \bar{a}_{ki} - (\gamma_j^{*(q)} - c_j^{(q+1)}) \bar{a}_{kj'} - \bar{c}_k^{(q+1)} = \\ &= - \sum_{i \in \{1, \dots, m\}} g_{ki}^{(q)} \bar{a}_{ki} + g_{kj'}^{(q)} \bar{a}_{kj'} - (\gamma_j^{*(q)} - c_j^{(q)} - p) \bar{a}_{kj'} - (\bar{c}_k^{(q)} + p \bar{a}_{kj'}) = \\ &= d^{(q)} + g_{kj'}^{(q)} \bar{a}_{kj'} - (\gamma_j^{*(q)} - c_j^{(q)}) \bar{a}_{kj'} \\ &= d^{(q)} + g_{kj'}^{(q)} \bar{a}_{kj'} - t_k g_{kj'}^{(q)} \bar{a}_{kj'} \\ &= d^{(q)} + g_{kj'}^{(q)} \bar{a}_{kj'} (1 - t_k) \end{aligned} \tag{3.6.11}$$

A  $-g_{kj'}^{(q)} \bar{a}_{kj'}$  szorzatról feltettük, hogy maximális – lásd (3.6.7) egyenlet – így a következő teljesül:

$$-g_{kj'}^{(q)} \bar{a}_{kj'} \geq \frac{\bar{c}_k^{(q)} + d^{(q)}}{m},$$

ahol  $m$  a feltételek száma, amely természetesen rögzített. Zárójelben megjegyezzük, hogy a becslésünk elég durva, a gyakorlatban a konvergencia gyorsabb. Visszatérve (3.6.11)-hez kapjuk, hogy:

$$\begin{aligned} d^{(q+1)} &= d^{(q)} - (-g_{kj'}^{(q)} \bar{a}_{kj'}) (1 - t_k) \\ &\leq d^{(q)} - \frac{c_k^{(q)} + d^{(q)}}{m} \frac{d^{(q)}}{\bar{c}_k^{(q)} + d^{(q)}} \\ &= d^{(q)} - \frac{d^{(q)}}{m} = d^{(q)} \left(1 - \frac{1}{m}\right). \end{aligned}$$

Azaz a vágások során a  $d^{(q)} \rightarrow 0$ . Mivel mindig a legjobban sértett feltételt választjuk, így az összes feltételre is igaz a konvergencia. Tehát bizonyos lépés után már nem sérül az egyenletrendszer.

*Jobboldali részprobléma:*  $[\gamma_j^{*(q)}, c_j^{u(q)}]$

Számoljuk ki az egyenletre vonatkozó  $\bar{c}_k^{(q+1)}$  és  $d_k^{(q+1)}$  értékeket.

$$\begin{aligned} \bar{c}_k^{(q+1)} &= \bar{c}_k^{(q)} - p(-\bar{a}_{kj'}) \\ &\leq \bar{c}_k^{(q)} + \gamma_j^{*(q)} \bar{a}_{kj'} \\ &= \bar{c}_k^{(q)} + g_{kj'}^{(q)} \bar{a}_{kj'} t_k \\ &= \bar{c}_k^{(q)} + g_{kj'}^{(q)} \bar{a}_{kj'} \frac{\bar{c}_k}{-\mathbf{g}_k^T \bar{\mathbf{a}}_k} \\ &\leq \bar{c}_k^{(q)} + g_{kj'}^{(q)} \bar{a}_{kj'} \frac{c_k^{(q)}}{-m g_{kj'}^{(q)} \bar{a}_{kj'}} \\ &= \bar{c}_k^{(q)} \left(1 - \frac{1}{m}\right) \end{aligned} \tag{3.6.12}$$

$$\begin{aligned} d^{(q+1)} &= - \sum_{i \in \{1, \dots, m\} \setminus \{j'\}} g_{ki}^{(q+1)} \bar{a}_{ki} - \gamma_j^{*(q)} \bar{a}_{kj'} - \bar{c}_k^{(q+1)} \\ &= - \sum_{i \in \{1, \dots, m\} \setminus \{j'\}} g_{ki}^{(q)} \bar{a}_{ki} - (g_{kj'}^{(q)} - p) \bar{a}_{kj'} - (\bar{c}_k^{(q)} - p(-\bar{a}_{kj'})) \\ &= - \sum_{i \in \{1, \dots, m\}} g_{ki}^{(q)} \bar{a}_{ki} - \bar{c}_k^{(q)} = d^{(q)} \end{aligned} \tag{3.6.13}$$

vagyis  $d$  nem változik.

Megfelelően nagy  $q$ -ra a következő teljesül:

$$-g_{kj'}^{(q)} \bar{a}_{kj'} \geq \frac{d^{(q)} + \bar{c}_k^{(q)}}{m} \geq \frac{d^{(q)}}{m} > \bar{c}_k^{(q)},$$

hiszen a (3.6.12) miatt  $\bar{c}_k^{(q)} \rightarrow 0$  és (3.6.13) miatt  $d^{(q)}$  nem változik. Továbbá

$$\begin{aligned} & -(\bar{h}_{kj}^{(q)} - c_j^{(q)}) \bar{a}_{kj'} > c_k^l + \mathbf{c}_B^T \bar{\mathbf{a}}_k \\ & -\bar{h}_{kj}^{(q)} \bar{a}_{kj'} - \sum_{i \in \{1, \dots, m\} \setminus \{j'\}} c_{B,i}^{(q)} \bar{a}_{ki} > c_k^l \end{aligned}$$

Ami pontosan azt jelenti, hogy vágások véges számú végrehajtásával el tudunk jutni egy olyan partícionáláshoz, ahol egy változó módosítása már bázisváltást eredményez és visszatérünk ahhoz az esethez, amikor egy változó célfüggvénye módosul.  $\square$

### 3.6.3. Szétválasztási stratégia

Fejezetünkben összefoglaljuk a korábbi fejezetekben kidolgozott partícionálási eljárásokat. A 3.3.1 fejezetben említett jelölést használva  $Level(P^k)$  függvény mutassa a BB fa azon szintjét, mely az  $(P^k)$  részproblémához tartozik és továbbá legyen  $N$  egy pozitív egész szám és  $\mathcal{K}$  a 3.6.1 fejezet szerint definiált indexhalmaz és  $\hat{\mathcal{K}}$  a 3.6.2 fejezet szerint definiált indexhalmaz.

**if** (  $Level(P^k) \bmod N = 0$  ) **then**

$$j = \operatorname{argmax}_{i \in \{1..n\}} (u_i^k - l_i^k)$$

$$p = (u_j^k + l_j^k) / 2.0$$

**else if**  $\mathcal{K} \neq \emptyset$  **then**

$$j = \operatorname{argmax}_{j \in \mathcal{K}} \{f_j(\bar{x}_j) - (c_j \bar{x}_j + d_j)\}$$

legyen  $p$  vágási pont a (3.6.5) és (3.6.6) képlet alapján meghatározva.

**else if**  $\hat{\mathcal{K}} \neq \emptyset$  **then**

$$j' = \operatorname{argmax}_{i \in \{1, 2, \dots, m\}} | -(\bar{h}_{B,ki} - c_{B,i}) \bar{a}_{ki} |.$$

legyen  $p$  vágási pont a (3.6.10) képlet alapján meghatározva.

**else**

$\hat{\mathcal{K}} = \emptyset$  esetén nincs tesztpont, amely megsértene valamely feltételt,

így a részprobléma optimális megoldása az aktuális bázismegoldás  
**endif**

### 3.6.4. Az algoritmus helyességének bizonyítása

Minden  $N$ -ik szinten lévő részproblémát a leghosszabb él mentén vágjuk a két egybevágó részbe, így a téglatestek oldalainak hossza zérushoz tart. A 3.3.1 lemma itt is teljesül, amiből a 3.3.2 lemma következik bizonyítva az algoritmus helyességét.

### 3.6.5. Az algoritmus működésének elemzése

A globális optimum megtalálását a BB fa minden  $N$ -ik szinten lévő részproblémájának partícionálása garantálja, viszont a hatékonyságát nem. Az relaxált LP vizsgálatával meghatározott vágási pontok azok, amelyek megalapozzák módszerünk hatékonyságát. Megjegyezzük, hogy érzésünk szerint – amit a tapasztalat is alátámaszt – az érzékenységi vizsgálatból kapott vágási pontok is garantálják az optimális megoldás megtalálását, de a bizonyítás még a jövő egyik feladata.

A vágási pontok meghatározását a következő alapgondolat vezette: mivel az optimális megoldás a poliéder egy csúcsában van, így érdemes olyan vágási pontokat generálni, amely esetében az új részproblémák közül az egyik relaxált feladatnak az optimális megoldása már egy másik lehetséges bázismegoldás legyen. Ez sokszor nem teljesíthető ( $\mathcal{K} = \emptyset$ ), ilyenkor az LP feladat érzékenységi vizsgálatával keresünk megfelelő vágási pontot. Ekkor célunk az, hogy minél kevesebb partícionálással olyan részfeladatokhoz jussunk, amelyek már vagy optimálisak (azaz  $\hat{\mathcal{K}} = \emptyset$ ) vagy bázisváltás lép fel.

A korábbi részben a partícionálás utáni részfeladatok viselkedését vizsgáltuk. Megmutattuk, hogy elegendően sok partícionálási lépés után egy részprobléma vagy terminálissá válik, vagy bázisváltás lép fel.

Az algoritmus szempontjából fontos a feltételrendszer által meghatározott poliéder szerkezete, azok az úgynevezett „szép” csúcsok (bázismegoldások), amelyek az érzékenységvizsgálat alapján eléggé stabilak a célfüggvény módosítására nézve. Például egy focilabda (amely 5 és 6 szögekből áll) csúcsai nem annyira „szépek” szemben egy



kocka csúcsaival. A PNS feladatok együtthatómátrixa ritka, így a feltételek által meghatározott síkok „szép” csúcsokat képezhetnek. A módszerünk jól kihasználja a PNS feladatok ilyen tulajdonságát.

Az érzékenységi vizsgálat segítségével a partícionálás utáni részproblémákról sok esetben előre tudjuk, hogy a korlátozási lépésben mi lesz az optimális bázis. Ilyen esetekben természetesen nem kell az LP feladatot megoldani.

### 3.6.6. Az eredmény rövid összefoglalása

A 3.6 fejezet a 2b tézispontban megfogalmazott eredményeket tartalmazza.

Új eljárást dolgoztam ki a lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat megoldása során felmerülő szétválasztás lépésre. Az eljárás a 2a tézispontban megfogalmazott optimalitási kritériumon alapulva végzi a részproblémák partícionálását, illetve a terminális részproblémák meghatározását. A megfogalmazott algoritmus helyességét igazoltam.

## 3.7. Kombinatorikusan gyorsított algoritmus

Egy PNS feladathoz tartozó konkáv szeparábilis feladat feltételrendszere magán hordozza a PNS feladat sajátosságait, melyek figyelembevételével új, hatékony módszer készíthető PNS feladatok optimalizálására.

### 3.7.1. Bevezetés

Egy P-gráf jól reprezentálja a PNS modellben lévő változók közötti függőségeket (lásd 2.2.3 fejezet). A matematikai modell egy lehetséges  $\mathbf{x}$  megoldását azonosíthatjuk a P-gráf egy  $s$  (struktúrával) P-részgráfiával :  $o_i \in s \Leftrightarrow x_i \neq 0$  (természetesen nem minden részgráfhoz létezik lehetséges  $\mathbf{x}$  megoldásvektor).

Az alap algoritmus (3.1 ábra) működése közben a lehetséges megoldások halmazát partícionálja. Az itt bevezetésre kerülő módszer abban különbözik a korábbi

eljárástól, hogy a partícionálással párhuzamosan az eljárás egy P-gráfon végez műveleteket. A gráf segítségével kiszűrhetünk olyan részproblémákat, amelyek nem teljesítik a korábban bevezetett axiómákat (2.2.2 definíció), ezáltal törölhetőek. Egy LP feladat megoldásának számításigénye jóval több egy P-gráfon végzett egyszerű műveletnél, viszont a P-gráf használata nélkül csak az alsókorlát számítás (egy LP feladat megoldása) után lenne lehetőség a részprobléma törlésére.

A módszer egy általánosabb feladatosztályt kezelését teszi lehetővé, amely félfolytonos változókat is tartalmazhat. Egy  $x_i \in \mathbb{R}$  változó félfolytonos, ha  $x_i = 0$  vagy  $0 \leq l_i \leq x_i \leq u_i$  teljesül ( $0 \notin [l_i, u_i]$ , különben folytonos változóval van dolgunk).

A félfolytonos változók gyakran előfordulnak valós ipari feladatok modellezésénél. Szemléletesen annyit jelent, hogy egy műveleti egység mérete nem lehet akármilyen kicsi.

A  $\mathcal{T}$  halmaz a következőképpen módosul:

$$\mathcal{T} = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq l_i \leq x_i \leq u_i \text{ vagy } x_i = 0, i = 1, \dots, n\}.$$

A félfolytonos változók miatt a  $\mathcal{T}$  halmaz nem konvex, így a kiindulási  $\mathcal{T}$  halmazt közelítenünk kell:

$$\mathcal{T}^0 = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_i \leq u_i, i = 1, \dots, n\}.$$

A műveleti egységhez kapcsolódó költségfüggvény (korábban lásd (2.2.9) egyenlet):

$$f_i(x_i) = \begin{cases} 0, & x_i = 0, \\ a_i + b_i x_i^\alpha, & x_i > 0, \end{cases} \quad (3.7.1)$$

ahol  $\alpha \in \mathbb{R}$ ,  $0 \leq \alpha \leq 1$ ,  $a_i \geq 0$ ,  $b_i \geq 0$ , ( $i = 1, \dots, n$ ).

### 3.7.2. Részproblémák

Korábbi tárgyalásunkban egy részproblémát egy  $\mathcal{T}^k$  hipertégla határozott meg (lásd 3.2.1 fejezet), a kombinatorikusan gyorsított módszer esetében a részproblémához egy P-gráfot is hozzárendelünk. A P-gráf leírását a műveleti egységek halmazának egy osztályozása adja meg:  $fix_0^k \subseteq \mathcal{O}$ ,  $fix_1^k \subseteq \mathcal{O}$ ,  $free^k \subseteq \mathcal{O}$  halmazok. A részprobléma meghatározásakor ezeket a halmazokat is definiálni kell.

Legyen  $P^k = (\mathcal{T}^k, \mathcal{D}^k, fix_0^k, fix_1^k, free^k)$  részprobléma. Legyen  $\mathcal{T}^k = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l}^k \leq \mathbf{x} \leq \mathbf{u}^k\} \subseteq \mathcal{T}^0$  és a  $fix_0^k, fix_1^k, free^k$  halmazok a korábban említett osztályozása a műveleti egységek halmazának. A halmazok között a következő összefüggések állnak fenn:

$$\begin{aligned} o_i \in fix_0^k &\Leftrightarrow l_i^k = u_i^k = 0, \\ o_i \in fix_1^k &\Leftrightarrow 0 \leq l_i \leq l_i^k \leq x_i \leq u_i^k \leq u_i, \\ o_i \in free^k &\Leftrightarrow 0 = l_i^k \leq x_i \leq u_i^k = u_i. \end{aligned}$$

A  $fix_0^k$  halmaz tartalmazza a kizárt műveleti egységeket,  $fix_1^k$  a beválasztott műveleti egységek halmaza és a  $free^k$  halmaz a még nem döntött műveleti egységeket tartalmazza.

### 3.7.3. Kiterjesztés

A kiterjesztési eljárás végrehajtásakor, a gráf már rögzített részét „építjük” tovább a korábban végzett döntések alapján. A kiterjesztési lépés a kombinatorikusan lehetséges megoldások tulajdonságait kihasználva végzi a döntéseket a még nem döntött műveleti egységek beveteléről vagy kizárásáról.

A kiterjesztés egy iteratív folyamat, a kiterjesztési lépéseket addig kell ismételni, amíg már nincs változás a gráfon. Ez véges sok iterációs lépést jelent, hiszen a műveleti egységek száma véges.

A kiterjesztés két irányban lehetséges: felfelé és lefelé. Megjegyezzük, hogy a kiterjesztések csak gyorsítják az algoritmus konvergenciáját, a konvergenciának nem feltételei.

#### Felfelé építkezés

Legyen  $m_i$  anyagpont, ahol potenciálisan kiterjesztést végezhetünk, továbbá kiszámoljuk  $\mathcal{U}_i$  halmazt, amely az  $m_i$ -t előállító műveleti egységek halmaza.

$$m_i \in \left( \bigcup_{(\alpha_j, \beta_j) \in fix_1^k} \alpha_j \right) \cup \mathcal{P} \setminus \mathcal{R}, \quad \mathcal{U}_i = \left( \bigcup_{(\alpha_j, \beta_j) \in \mathcal{O}, m_i \in \beta_j} o_j \right) \setminus fix_0^k.$$

Ha  $\mathcal{U}_i = \{(\alpha_i, \beta_i)\} \subseteq free^k$ , akkor  $free^k = free^k \setminus \{l\}$ ,  $fix_1^k = fix_1^k \cup \{l\}$ .

A szabály szerint, ha létezik egy  $m_i$  anyag, mely nem nyersanyag és azt valamely műveleti egység már fogyasztja, akkor azt legalább egy műveleti egységnek elő kell állítani. Ha az  $m_i$ -t előállító műveleti egységek halmaza egyelemű, akkor azt az elemet hozzá kell venni a már beválasztott műveleti egységek közé, különben az  $S2$  feltétel (2.2.2 definíció) sérülne.

### Lefelé építkezés

Legyen  $o_s = (\alpha_s, \beta_s) \in fix_1^k$ , melyre  $\beta_s \cap \mathcal{P} = \emptyset$ , legyen továbbá

$$\mathcal{U}_s = \left( \bigcup_{(\alpha_j, \beta_j) \in \mathcal{O}, \beta_s \cap \alpha_j \neq \emptyset} o_j \right) \setminus fix_0^k.$$

$\mathcal{U}_s$  tartalmazza az  $o_s$  által gyártott anyagokat fogyasztó műveleti egységeket. Ha  $\mathcal{U}_s = \{(\alpha_l, \beta_l)\} \subseteq free^k$ , akkor  $free^k = free^k \setminus \{l\}$ ,  $fix_1^k = fix_1^k \cup \{l\}$ .

Ha adott egy már beválasztott műveleti egység, mely nem gyárt terméket, és az általa gyártott anyagokat csak egy műveleti egység képes felhasználni, akkor azt hozzávesszük a beválasztott műveleti egységekhez. Ha az így meghatározott halmaz üres, akkor a részprobléma eldobható, hiszen sérül az  $S3$  feltétel (2.2.2 definíció).

A kiterjesztés eredményeként bizonyos műveleti egységek a  $free^k$  halmazból átkerülnek egy másik osztályba. Ilyenkor megváltoznak a műveleti egységekhez kapcsolódó változók korlátai. Jelölje most  $o_j \in free^k$  a műveleti egységet és  $x_j$  a hozzá tartozó változót ( $0 \leq x_j \leq u_j$  teljesül). Két eset van:

1. Az  $o_j$  műveleti egység a  $free^k$ -ból a  $fix_0^k$ -be kerül.

A műveleti egységet a kizártak közé helyeztük, így az  $x_j = 0$  teljesül ( $l_j^k = 0$ ,  $u_j^k = 0$ ).

2. Az  $o_j$  műveleti egység a  $free^k$ -ból a  $fix_1^k$ -be kerül.

A műveleti egységet a bevettek közé helyeztük. Az  $x_j = 0$  lehetőséget kizártuk, így  $l_j \leq x_j \leq u_j$  teljesül ( $l_j^k = l_j$ ,  $u_j^k = u_j$ ).

### 3.7.4. Algoritmus

Az algoritmus az általános szeparábilis konkáv minimalizációs eljárásra épül kiegészítve a P-gráf reprezentáción alapuló kombinatorikus részekkel. A korábban bevezetett (3.1 ábra) algoritmust módosítjuk és csak az új részeket részletezzük (lásd 3.9 ábra).

Csak akkor végezzük el a műveleteket a párhuzamosan épülő P-gráfon, ha a döntés előtt álló változón még nem volt döntés ( $j \in free^k$ ). Ha egy korábban döntött változón végzünk döntést, akkor a kapcsolódó művelet egységet már korábban beválasztottuk a struktúrába és így a P-gráf változatlan.

#### Korlátozás

A korlátozási lépés teljesen analóg a 3.2.2 fejezetben bevezetett eljárással. A félfolytonos változók miatt a  $\mathcal{T}$  halmazt kiterjesztettük  $\mathcal{T}^0$  halmazra. Az induló részproblémában definiált  $\mathcal{T}^0$  halmaz a  $\mathcal{T}$  halmaz kiterjesztése (3.9 ábra).

A kiterjesztés miatt az alaphalmazunk bővült. Ha valamely  $i \in \{1, \dots, n\}$ -re  $\omega_i^k \in ]0, l_i[$  teljesül, az  $\omega_i^k$  nem lehetséges megoldása az eredeti feladatnak. Az  $F^k(\omega)$  továbbra is alsó korlát, hiszen bővebb halmazon minimalizáltunk, az  $f(\omega^k)$  érték felső korlátként már nem használható.

#### Szétválasztás

A vágási változó kiválasztása legyen a következő szabályok szerint:

```

if (  $\exists i (\omega_i^k \in ]0, l_i[)$  ) then
     $j = \operatorname{argmax}_{\omega_i^k \in ]0, l_i[} \{f_i(\omega_i^k) - F_i^k(\omega_i^k)\}$ 
else if (  $\exists i (i \in free^k \wedge (f_i(\omega_j^k) - F_i^k(\omega_i^k)) > 0)$  ) then
     $j = \operatorname{argmax}_{i \in free^k} \{f_i(\omega_i^k) - F_i^k(\omega_i^k)\}$ 
else
     $j = \operatorname{argmax}_{i \in 1 \dots n} \{f_i(\omega_i^k) - F_i^k(\omega_i^k)\}$ 
endif

```

A korábbi vágási stratégiákhoz képest a különbség annyi, hogy először olyan  $x_j$  változón döntünk, amelyhez tartozó  $\omega_j^k$  megoldás értéke nem megengedett. Ha nincs

---

Kombinatorikusan gyorsított algoritmus

---

**Bemenő adatok:**

$m, n \in \mathbb{N}$   
 $A \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$  és  $\mathbf{0} \leq \mathbf{l} \leq \mathbf{u}$   
 $f : \mathbb{R}^n \rightarrow \mathbb{R}$  függvény  
 $k = 0$ ,  $L = -\infty$ ,  $U = \infty$ ,  
 $\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$   
 $\mathcal{T}^0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}$   
 $\mathcal{D}^0 = \mathcal{A} \cap \mathcal{T}^0$   
 $P^0 = (\mathcal{T}^0, \mathcal{D}^0, \emptyset, \emptyset, \mathcal{O})$   
 $\mathcal{S} = \{P^0\}$

**Kimenő adatok:**

a  $(P)$  feladat optimális megoldása  $\bar{\mathbf{x}}$   
a  $(P)$  feladat optimum értéke  $U$

**Begin**

```

while ( $\mathcal{S} \neq \emptyset$ )
begin
 $P^k = \text{Választ}(\mathcal{S})$ ;
 $(L, U, \bar{\mathbf{x}}) = \text{Korlátozás}(P^k, \bar{\mathbf{x}})$ ;
if  $U = L$  then
 $\bar{\mathbf{x}}$  optimális megoldása a  $(P)$  feladatnak, STOP;
 $(P^{k_1}, P^{k_2}, P^{k_3}, \iota) = \text{Partícionálás}(P^k, f^k, \beta^k)$ ;
if  $\iota = 1$  then
 $\mathcal{S} = \mathcal{S} \cup \{P^{k_1}, P^{k_2}\}$ ;
else if  $\iota = 2$  then
begin
 $RS\mathcal{G}(P^{k_1})$ ;
hajtsuk végre a kiterjesztési lépéseket a részproblémákra
 $\mathcal{S} = \mathcal{S} \cup \{P^{k_1}, P^{k_2}, P^{k_3}\}$ ;
end
end

```

**End.**


---

3.9. ábra. Kombinatorikusan gyorsított algoritmus.

---

*Particionálás*

---

**Bemenő adatok:**

$$P^k, f^k, \beta^k$$

**Kimenő adatok:**

$$P^{k_1}, P^{k_2}, P^{k_3}, \iota$$

**Begin**

**if**  $U \geq \beta^k$  **then**

**begin**

      a vágási változó  $x_j$  meghatározása;

      a vágási pont  $p$  meghatározása;

**if**  $j \in free^k$  **then**

**begin**

          kiszámoljuk az új halmazokat (lásd (3.7.4) és (3.7.5) képletek).

$$\iota = 2;$$

**end**

**else begin**

        kiszámoljuk az új halmazokat (lásd (3.7.2) és (3.7.3) képletek).

$$\iota = 1;$$

**end**

**end**

**else begin**

$$\iota = 0;$$

    a halmazok legyenek üresek;

**end**

$$\mathcal{S} = \mathcal{S} \setminus \{P^k\}$$

**End.**

---

3.10. ábra. A particionálási eljárás a kombinatorikusan gyorsított algoritmusban.

ilyen, akkor olyan változón döntünk, amelyen még nem volt döntés ( $j \in free^k$ ), különben a korábban már bevezetett módon döntünk.

A szétválasztási stratégia szerint lehetőleg a még nem döntött változón vágunk, és mivel csak az  $i \in free^k$ -ra lehetséges az  $\omega_i^k \in ]0, l_i[$  nem megengedett megoldás, így véges lépésben valódi megoldást kapunk.

A vágási pont meghatározása lehet a korábban a bemutatott "csúsztatott" vagy maximális rés partícionálásban használt választás.

Egy részfeladat partícionálása is módosul a korábbiakhoz képest. Legyen az aktuális részfeladat  $P^k$ ,  $j$  a vágási változó és  $p$  a vágási pont.

Ha  $j \in fix_1^k$ , akkor két részprobléma keletkezik ( $P^{k_1}, P^{k_2}$ ).

$$\begin{aligned} free^{k_1} &= free^k, fix_0^{k_1} = fix_0^k, fix_1^{k_1} = fix_1^k, \\ free^{k_2} &= free^k, fix_0^{k_2} = fix_0^k, fix_1^{k_2} = fix_1^k, \end{aligned} \quad (3.7.2)$$

$$\begin{aligned} \mathcal{T}^1 &= \{\mathbf{x} \in \mathcal{T}^k : l_j^k \leq x_j \leq p\} \\ \mathcal{T}^2 &= \{\mathbf{x} \in \mathcal{T}^k : p \leq x_j \leq u_j^k\} \end{aligned} \quad (3.7.3)$$

Ha  $j \in free^k$ , akkor három részprobléma keletkezik ( $P^{k_1}, P^{k_2}, P^{k_3}$ ). A 3.11 ábrán a I. II. III. jelöli a megfelelő részproblémákat.

$$\begin{aligned} free^{k_1} &= free^k \setminus \{j\}, fix_0^{k_1} = fix_0^k \cup \{j\}, fix_1^{k_1} = fix_1^k, \\ free^{k_2} &= free^k \setminus \{j\}, fix_0^{k_2} = fix_0^k, fix_1^{k_2} = fix_1^k \cup \{j\}, \\ free^{k_3} &= free^k \setminus \{j\}, fix_0^{k_3} = fix_0^k, fix_1^{k_3} = fix_1^k \cup \{j\}, \end{aligned} \quad (3.7.4)$$

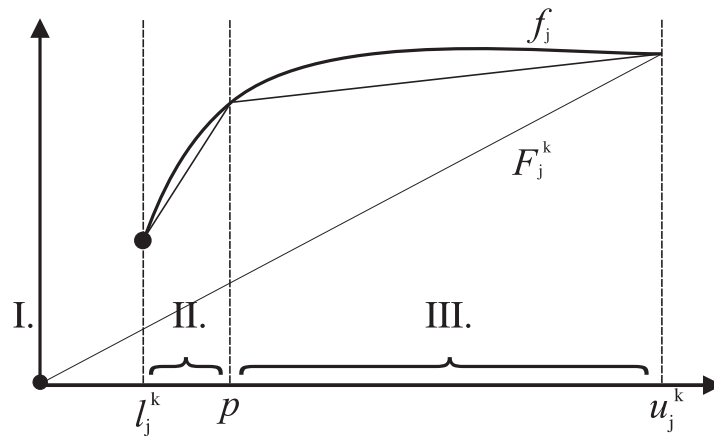
$$\begin{aligned} \mathcal{T}^{k_1} &= \{\mathbf{x} \in \mathcal{T}^k : x_j = 0\} \\ \mathcal{T}^{k_2} &= \{\mathbf{x} \in \mathcal{T}^k : l_j \leq x_j \leq p\} \\ \mathcal{T}^{k_3} &= \{\mathbf{x} \in \mathcal{T}^k : p \leq x_j \leq u_j\} \end{aligned} \quad (3.7.5)$$

A partícionálási eljárást a 3.10 ábra foglalja össze.

## RSG

Korábban már említett *RSG* algoritmus (2.3 fejezet) itt is használható változatlan formában.





3.11. ábra. Vágás egy még nem döntött változón.

### 3.7.5. A helyesség bizonyítása

Az eredeti algoritmushoz képest (lásd 3.1 ábra) fő eltérés a félfolytonos változók kezelése.

Egy már korábban döntött  $x_j$  változón ( $j \in \text{fix}_0^k \cup \text{fix}_1^k$ ) a partícionálás már az eredeti algoritmus szerint viselkedik. Egy még nem döntött változó ( $j \in \text{free}^k$ ) döntés után már döntötté válik, mivel kezdetben a nem döntött változók száma véges, ezért véges számú olyan döntés van, amely még nem döntött változón történik, azaz véges lépésben tér el módszerünk az eredeti algoritmustól. Ezek alapján mondhatjuk, ha az eredeti algoritmusunk konvergens, akkor a kombinatorikusan gyorsított algoritmus is konvergens lesz, és ez végességre is teljesül.

### 3.7.6. Az eredmény rövid összefoglalása

A 3.7 fejezet a 3a tézispontban megfogalmazott eredményeket tartalmazza.

A Friedler és munkatársa által kidolgozott P-gráf módszert felhasználva elkészítettem a lineáris feltételrendszerrel adott szeparábilis konkáv programozási feladatot megoldó algoritmus kombinatorikusan gyorsított változatát.

## 3.8. N-legjobb megoldás

Vegyipari rendszerek esetében a matematikai modell nem képes (vagy abban az esetben ha képes lenne, akkor túl bonyolulttá válna a modell) a rendszer minden tulajdonságát leírni, pl.: irányíthatóság, megvalósíthatóság stb. Ezért szükség van az optimális megoldáson túl szuboptimális megoldások meghatározására is, amelyek közül a felhasználó tudja kiválasztani a számára megfelelőt.

### 3.8.1. Bevezetés

Egy optimalizációs feladatban, amely tartalmaz folytonos változókat nincs értelme második legjobb megoldásról beszélni. A kombinatorikusan lehetséges megoldások definícióját (2.2.2 definíció) bevezetve, a szuboptimális megoldások jól definiáltakká válnak.

Definiáljuk  $\tilde{f}$  függvényt az  $f$  költségfüggvény (lásd (3.7.1) egyenlet) felhasználásával:

$$\tilde{f}_i(x_i) = \begin{cases} 0, & x_i = 0, \\ b_i x_i^\alpha, & x_i > 0, \end{cases} \quad (3.8.1)$$

ahol  $b_i \geq 0$  és  $0 \leq \alpha \leq 1$ , azaz a fix költségtől tekintettünk el.

Legyen  $s \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ , kombinatorikusan lehetséges struktúra. Jelölje  $\tilde{z}(s)$  érték az  $\tilde{f}$  költségfüggvénnyel vett optimumértéket az  $s$  struktúrára megszorítva ( $o_j \notin s$ -re  $x_j = 0$ ), és hasonlóan jelölje  $z(s)$  érték az  $f$  költségfüggvénnyel vett optimumértéket az  $s$  struktúrára megszorítva.

**Definíció 3.8.1** *A  $p \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$  kombinatorikusan lehetséges struktúra lokálisan optimális struktúra, ha*

$$\forall q \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O}) ((q \subset p) \Rightarrow (z(p) < z(q)))$$

*teljesül. A lokálisan optimális struktúrák halmazát jelölje  $\mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ .*

3.1. táblázat. Lokálisan optimális struktúrák számossága

| Feladat | Komb. lehetséges struktúrák | Lok. opt. struktúrák |
|---------|-----------------------------|----------------------|
| Denmark | 3465                        | 96                   |
| Alpha   | 423                         | 36                   |

$p \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$  kombinatorikusan lehetséges struktúra arányos költség szerinti lokálisan optimális struktúra, ha

$$\forall q \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O}) ((q \subset p) \Rightarrow (\tilde{z}(p) < \tilde{z}(q)))$$

teljesül. Az arányos költség szerinti lokálisan optimális struktúrák halmazát jelölje  $\tilde{\mathcal{L}}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ .

A lokálisan optimális struktúrák halmaza függ a modell paramétereitől, ellentétben a kombinatorikusan lehetséges megoldások halmazával, amely csak maximális a struktúrától függ.

A 3.1 táblázat tartalmazza két ipari feladat [39] lokálisan optimális megoldásainak a számát és a kombinatorikusan lehetséges struktúrák számát.

A lokálisan optimális struktúrák halmaza része a kombinatorikusan lehetséges struktúrák halmazának, továbbá egy lokálisan optimális struktúrához hozzátartozik egy  $\mathbf{x}$  megoldásvektor, amely megadja a struktúrában szereplő műveleti egységek méretét. Elképzelhető, hogy egy  $p$  struktúrához több optimális  $\mathbf{x}$  megoldásvektor tartozik, amely egy ritka eset, de nem zárható ki. Szemléletes jelentése az, hogy egy azonos struktúrához két optimális konfiguráció (megoldásvektor) tartozik. Ilyenkor mindkét megoldásvektor hasznos lehet, és az algoritmus azonosítani fogja ezeket. A következőkben a lokálisan optimális struktúrák néhány tulajdonságát mutatjuk meg.

**Állítás 3.8.1**  $\mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O}) \subseteq \tilde{\mathcal{L}}(\mathcal{P}, \mathcal{R}, \mathcal{O})$

**Bizonyítás.** Megmutatjuk, hogy ha  $s \in \mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ , akkor  $s \in \tilde{\mathcal{L}}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ .

Legyen  $\mathbf{x}^s \in \mathbb{R}^n$ , melyre  $z(s) = f(\mathbf{x}^s)$ ,  $\tilde{\mathbf{x}}^s \in \mathbb{R}^n$ , melyre  $\tilde{z}(s) = \tilde{f}(\tilde{\mathbf{x}}^s)$ .  $\tilde{z}(s) \leq z(s)$ , hiszen  $\tilde{f} \leq f$ .

Indirekten: Tfh.  $s \notin \tilde{\mathcal{L}}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ , akkor  $\exists q \subset s$  és  $\tilde{\mathbf{x}}^q$  vektor, melyre  $\tilde{z}(q) = \tilde{f}(\tilde{\mathbf{x}}^q) \leq \tilde{f}(\tilde{\mathbf{x}}^s) = \tilde{z}(s)$ . A következő teljesül:

$$\sum_{i:\tilde{x}_i^q \neq 0} a_i \leq \sum_{i:\tilde{x}_i^s \neq 0} a_i, \text{ hiszen, } q \subset s \text{ és } s \in \mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O}).$$

Az egyenlőtlenség felhasználásával kapjuk, hogy

$$f(\tilde{\mathbf{x}}^q) = \tilde{f}(\tilde{\mathbf{x}}^q) + \sum_{i:\tilde{x}_i^q \neq 0} a_i \leq \tilde{f}(\tilde{\mathbf{x}}^s) + \sum_{i:\tilde{x}_i^q \neq 0} a_i \leq \tilde{f}(\tilde{\mathbf{x}}^s) + \sum_{i:\tilde{x}_i^s \neq 0} a_i = f(\mathbf{x}^s).$$

Mivel  $q \subset s$ , ezért  $s \notin \mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O})$  következik.  $\square$

### Állítás 3.8.2 $\mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O}) \neq \tilde{\mathcal{L}}(\mathcal{P}, \mathcal{R}, \mathcal{O})$

**Bizonyítás.** Triviálisan adható egy ellenpélda.

$$\mathcal{O} = \{1, 2\}, o_1 = (\{m_2\}, \{m_1\}), o_2 = (\{m_3\}, \{m_1\})$$

*Modell paraméterek*

A műveleti egységekhez tartozó paraméterek:

$$l_1 = 0, l_2 = 0, u_1 = 100, u_2 = 100, a_1 = 1, a_2 = 1000, b_1 = 0, b_2 = 0, r_{11} = 1, r_{12} = 1, r_{21} = 1, r_{23} = 1.$$

Az anyagponthoz tartozó paraméterek:

$$C_1 = 0, C_2 = 100, C_3 = 1, p_1 = 10, p_2 = 0, p_3 = 0, s_1 = 0, s_2 = 100, s_3 = 5$$

Könnyen ellenőrizhető, hogy az  $\{1, 2\} \notin \mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ , viszont  $\{1, 2\} \in \tilde{\mathcal{L}}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ .  $\square$

Összefoglalva: a lokálisan optimális struktúrák nem tartalmazznak olyan műveleti egységeket, amelyek öncélúan vesznek részt a végtermékek gyártásában. Ellenkező esetben sok értelmetlen struktúrát lehetne létrehozni, amelyben az optimális megoldást kibővítjük fiktív műveleti egységekkel, és azokat valamilyen szinten működtetjük.

**Állítás 3.8.3** Legyen  $p \in \mathcal{L}(\mathcal{P}, \mathcal{R}, \mathcal{O})$  lokálisan optimális struktúra, és  $\mathbf{x}^* \in \mathbb{R}^n$  a hozzá tartozó optimális megoldásvektor, akkor  $\mathbf{x}^*$  a  $\mathcal{D}$  politop extrémális pontja (csúcsa).

**Bizonyítás.** A megoldásvektor úgy kapjuk, hogy a korábban említett módon  $p$  struktúrára megszorítjuk a  $\mathcal{D}$  halmazt, és azon keressük az optimális megoldását. A  $p$  struktúrára megszorított  $\mathcal{D}$  halmaz a következőképpen néz ki:

$$\mathcal{D} \cap \left( \bigcap_{o_i \notin p} \{x_i : x_i = 0\} \right). \quad (3.8.2)$$

Mivel  $x_i \geq 0$  teljesül minden  $i$ -re, így az új halmazban nincs új extrémális pont, és korábban már megmutattuk, hogy az optimális megoldás mindig extrémális pontban van (lásd 3.0.1 lemma).  $\square$

**Megjegyzés:** Visszafelé természetesen nem igaz, egy extrémális pont nem biztos, hogy kielégíti a lokálisan optimális struktúra definícióját. Tekintsünk a legegyszerűbb esetet: legyen két csúcs, amelyben a nemzéró változók halmaza megegyezik, és a csúcsokban számolt célfüggvény értéke különböző, a csúcsok közül csak az egyik lehet lokálisan optimális struktúra.

A csúcs leszámplálási eljárásokkal elvileg lehetséges  $N$ -legjobb megoldást keresni. A lokálisan optimális struktúrákhoz tartozó csúcsok még csak nem is szomszédosak, így az összes csúcs generálását tenné szükségessé a módszer. Továbbá minden csúcsra ki kell értékelni a függvényt és ellenőrizni, hogy lokálisan optimális struktúra-e?

Felmerül a kérdés, hogy miért különböztetjük meg a lokálisan optimális struktúrákat az arányos költség szerinti lokálisan optimális struktúráktól. A 2.3 fejezetben említett *ABB* algoritmust is lehet úgy módosítani, hogy az  $N$ -legjobb megoldást szolgáltatassa. Az *ABB* módszer a költség fix részét külön kezeli, így csak az arányos költség szerinti lokális optimális struktúrákat képes azonosítani.

### 3.8.2. Algoritmus

A módszer a kombinatorikusan gyorsított algoritmusra (lásd 3.9 ábra) épül. A módszer a lokálisan optimális struktúrák halmazán keresi meg az első  $N$ -legjobbat. Az algoritmus vázlatos leírását a 3.12 ábra tartalmazza, csak az új részeket részletezzük. Megjegyezzük, hogy az arányos költség szerinti lokálisan optimális struktúrák halmazán történő keresés csak kis mértékben tér el a bemutatott eljárástól, az arra történő módosítás megvalósítható.

---

N-legjobb megoldás algoritmus

---

**Bemenő adatok:**

$N, m, n \in \mathbb{N}$   
 $A \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$  és  $\mathbf{0} \leq \mathbf{l} \leq \mathbf{u}$   
 $f : \mathbb{R}^n \rightarrow \mathbb{R}$  konkáv függvény  
 $k = 0$ ,  $L = -\infty$ ,  $U = \infty$   
 $\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$   
 $\mathcal{T}^0 = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}$   
 $\mathcal{D}^0 = \mathcal{A} \cap \mathcal{T}^0$   
 $P^0 = (\mathcal{T}^0, \mathcal{D}^0)$   
 $\mathcal{S} = \{P^0\}$

**Kimenő adatok:**

a  $(P)$  feladat első  $N$  legjobb lokálisan optimális struktúrát tartalmazó  $\mathbf{s}^N$  lista

**Begin**

```

while ( $\mathcal{S} \neq \emptyset$ )
begin
   $P^k = \text{Választ}(\mathcal{S})$ ;
   $(L, U, \bar{\mathbf{x}}) = \text{Korlátozás}(P^k, \bar{\mathbf{x}})$ ;
   $N\text{-Lista Frissít}(\omega^k, \mathbf{s}^N)$ 
  if  $P^k$  nem terminális then
     $(P^{k_1}, P^{k_2}, P^{k_3}, \iota) = \text{Partícionálás}(P^k, f^k, \beta^k)$ ;
  else
     $\iota = 0$ 
  if  $\iota = 1$  then
     $\mathcal{S} = \mathcal{S} \cup \{P^{k_1}, P^{k_2}\}$ ;
  else if  $\iota = 2$  then
    begin
       $RSG(P^{k_1})$ ;
      hajtsuk végre a kiterjesztési lépéseket a részproblémákra
       $\mathcal{S} = \mathcal{S} \cup \{P^{k_1}, P^{k_2}, P^{k_3}\}$ ;
    end
  end
end

```

**End.**


---

3.12. ábra.  $N$ -legjobb megoldást meghatározó algoritmus vázlata.

A módszert a kombinatorikusan gyorsított algoritmusból (lásd 3.9 ábra) származtatjuk. A potenciális lokálisan optimális struktúrákat egy listába gyűjtjük ( $N$ -lista), amit az algoritmus folyamatosan frissít. A részproblémák terminalitásának eldöntése és a vágási stratégia is módosult.

## Particionálás

A particionálás megegyezik a Kombinatorikusan gyorsított algoritmusnál bevezetett eljárással (lásd 3.7.4 fejezet).

### Az $N$ -lista frissítése

Az eddig megtalált legjobb megoldásokat tartalmazza az  $N$ -lista. Jelölje rendre  $s_i^N \subseteq \mathcal{O}$ ,  $\mathbf{x}_i^N \in \mathbb{R}^n$ ,  $i \in \{1, \dots, N\}$ , az  $N$ -listában lévő  $i$ -edik struktúrát, és a hozzá tartozó megoldásvektort. Legyen  $\omega^k$  az aktuális részfeladatokhoz tartozó megoldásvektor,  $q^k \subseteq \mathcal{O}$  az  $\omega^k$  megoldásvektorhoz tartozó struktúra, amelyre  $o_j \in q^k \leftrightarrow \omega_j^k \neq 0$ ,  $(q^k \subseteq \text{fix}_1^k \cup \text{free}^k)$ , és  $\beta^k$  az alsókorlát. Az eljárás leírását a 3.13 ábra tartalmazza.

Vegyük sorra az eljárás egyes lépéseit:

1. Az első sor azt vizsgálja, hogy ha az aktuális megoldás rosszabb, mint a lista legrosszabb megoldása, akkor az eljárás befejeződik és nem történik semmi változás a listában.
2. A ciklus megvizsgálja az összes listában lévő elemet, és összehasonlítja az aktuális elemmel.
  - (a) A listában egy hasonló struktúrát találtunk.
    - i. Az aktuális  $q^k$  struktúrának kisebb a célfüggvény értéke, így a listában lévő  $s_i^N$  elemet töröljük a listából.
    - ii. Az aktuális struktúrának a célfüggvényértéke megegyezik a listában lévő struktúrával, és a megoldásvektorok különbözőek. Ebben az esetben mindkét megoldást megtartjuk és megnöveljük a listát, mert ilyen esetekben egy struktúra törlése maga után vonja az összes azonos struktúra törlését.

---

N-lista Frissít

---

**Bemenő adatok:**

$\omega^k$ ,  $\mathbf{s}^N$  lista

**Kimenő adatok:**

$\mathbf{s}^N$  lista

**Begin**

if ( $f(\omega^k) > f(\mathbf{x}_N^N)$ ) then

return;

for  $i \in \{1, \dots, N\}$  do

begin

if ( $q^k = s_i^N$ ) then

if ( $f(\omega^k) < f(\mathbf{x}_i^N)$ ) then

delete  $s_i^N, \mathbf{x}_i^N$ ;

else if ( $\omega^k \neq \mathbf{x}_i^N \wedge f(\omega^k) = f(\mathbf{x}_i^N)$ )

IncreaseList;

if ( $s_i^N \subset q^k \wedge f(\mathbf{x}_i^N) < f(\omega^k)$ ) then

return;

else if ( $q^k \subset s_i^N \wedge f(\omega^k) \leq f(\mathbf{x}_i^N)$ ) then

delete  $s_i^N, \mathbf{x}_i^N$ ;

else if ( $q^k \cap s_i^N \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$ ) then

IncreaseList;

end

$s_N^N = q^k; \mathbf{x}_N^N = \omega^k$ ;

ReorderList;

**End.**

---

3.13. ábra.  $N$ -listát módosító eljárás.



- (b) Létezik olyan struktúra a listában, amely részhalmaza az aktuális struktúrának és a célfüggvényértéke kisebb. Ez ellentmond a lokálisan optimális struktúra definíciójának, a listánk változatlan marad.
- (c) Az aktuális struktúra részhalmaza egy, a listában lévő struktúrának és a célfüggvényértéke kisebb. Ez ellentmond a lokálisan optimális struktúra definíciójának, ezért a listaelemet töröljük a listából, és az aktuális elemet hozzáadjuk a listához.
- (d) Ha két listaelemnek van közös része, amely maga is megoldásstruktúra, akkor elképzelhető, hogy egy új struktúra (ami része a metszetnek) több listaelem törléséhez vezet. Beszúrás esetén növeljük a lista méretét, így az új listaelem nem „szorít” ki egy, már korábban bevett elemet a listából. A  $q^k \cap s_i^N \in \mathcal{S}(\mathcal{P}, \mathcal{R}, \mathcal{O})$  feltétel eléggé általános, ez élesíthető egy LP feladat fizibilitás vizsgálatával, ami viszont számításigényes.

A lista mérete akkor csökken, amikor egy új potenciális listaelem több korábbi listaelem törléséhez vezetett. A lista elemszáma nem csökkenhet az eredetileg megadott  $N$  alá, viszont előfordulhat, hogy az algoritmus lefutása után az  $N$ -lista mérete nagyobb, mint amit paraméterként megadtunk, hiszen több elem együttes törlését nem tudjuk előre megjósolni.

### Terminalitás ellenőrzése

Egy részproblémát akkor tekintünk terminálisnak, ha már nem tartalmaz további lokálisan optimális struktúrát, mely eleme lehetne a listának és ezáltal az törölhető.

Hasonlóan, mint az  $N$ -lista bővítésénél jelölje  $s_i^N$ ,  $i = 1, \dots, N$  a  $N$ -listában lévő  $i$ -ik struktúrát és legyen  $\mathbf{x}_i^N$  a hozzá tartozó megoldásvektor. Legyen  $\omega^k, \beta^k$  az aktuális megoldásvektor és a hozzá kapcsolódó alsókorlát. Egy  $P^k$  részprobléma törölhető, ha a következő feltétel teljesül:

$$(f(\omega^k) - \beta^k < \varepsilon) \vee \beta^k > f(\mathbf{x}_N^N) \vee (\exists s_j^N \in \{N - lista\} s_j^N \subseteq fix_1^k \wedge \beta^k > f(\mathbf{x}_j^N)), \quad (3.8.3)$$

ahol  $\varepsilon > 0$  egy tolerancia paraméter.

### 3.8.3. A helyesség bizonyítás

Korábban már említettük, hogy az algoritmus a kombinatorikusan gyorsított algoritmusra épül, amelynek konvergenciája bizonyított. A helyesség bizonyításhoz azt kell megmutatnunk, hogy az  $N$ -lista a kívánt megoldásstruktúrákat tartalmazza, és a terminalitási kritérium helyes.

Tekintsük a BB algoritmus által generált BB fát. Tegyük fel, hogy az algoritmus végtelen, tehát létezik egy végtelen  $P^k$  sorozat melyre,  $P^{k+1}$  a  $P^k$  leszármazottja, azaz BB fában létezik egy, a gyökből kiinduló végtelen út.

**Lemma 3.8.1** *A (3.8.3) terminalitási feltétel elégséges egy részprobléma törléséhez.*

**Bizonyítás.** Az indoklás egyszerű, a  $(f(\omega^k) - \beta^k < \varepsilon)$  kifejezés igaz volta a részprobléma megoldását jelenti, és egy megoldott részprobléma nem tartalmazhat további lokálisan optimális struktúrákat, ezáltal törölhető. A  $\beta^k > f(\mathbf{x}_N^N)$  teljesülése szerint, a részprobléma nem tartalmazhat olyan lokálisan optimális struktúrát, amely eleme lehetne a listának. Ha a  $(\exists s_j^N \in \{N - lista\} s_j^N \subseteq fix_1^k \wedge \beta^k > f(\mathbf{x}_j^N))$  teljesül, akkor a részprobléma nem tartalmazhat további lokálisan optimális struktúrákat hiszen, ha tartalmazna, akkor a struktúrában lévő műveleti egységek a  $fix_1^k$  halmaz bővítései lennének, és a költség alsó korlátja is rosszabb, mint a listában lévő elem költsége.  $\square$

**Lemma 3.8.2** *A (3.8.3) terminalitási feltétel valamely  $k$ -ra teljesül.*

**Bizonyítás.** A konvergencia miatt biztos, hogy véges lépés után a  $P^k$  részfeladat törlődik, hiszen  $f(\omega^k) - \beta^k < \varepsilon$  feltétel teljesülni fog, ami elegendő az állításunk bizonyításához, viszont az ezen feltétel alapján való döntés lassú futást eredményezne. Nem véletlenül nem csak ezt a feltételt tartalmazza a (3.8.3) terminalitási feltételünk.

Nézzük meg azt az esetet, amikor nem tartalmaz lokálisan optimális struktúrát a részfeladat. Azt kell megmutatni, hogy létezik olyan  $k$ , amelyre a terminalitási feltételünk igazzá válik. A  $\beta^k < f(x_N^N)$  feltehető, különben a feladat egyszerűen törölhető. Mivel feltettük, hogy a részprobléma nem tartalmaz lokálisan optimális struktúrát és  $\beta^k < f(x_N^N)$  is teljesül, ezért valamely  $k$ -ra léteznie kell a listában egy

lokálisan optimális  $s_j^N$  struktúrának, amelyre  $q^k \subset s_j^N$  és  $f(x^k) > f(x_j^N)$ . Továbbá a konvergencia miatt a  $\beta^k > f(x_j^N)$  teljesülni fog, azaz a  $(\exists s_j^N \in \{N - lista\} \ s_j^N \subseteq fix_1^k \wedge \beta^k > f(x_j^N))$  feltétel igazgá válik.

Tegyük fel, hogy a  $P^k$  sorozatban egy lokálisan optimális struktúrát tartalmaz a részprobléma. A  $\beta^k < f(x_N^N)$  itt is feltehető. Ez az eset hasonlít a kombinatorikus gyorsított algoritmusban az optimális megoldást tartalmazó részfeladat viselkedéséhez, hiszen alsó korlát alapján nem lehet eldobni a részfeladatot. Tehát a részfeladat a  $f(\omega^k) - \beta^k < \varepsilon$  feltétel alapján fog terminálissá válni.  $\square$

**Tétel 3.8.3** *A (3.8.3) terminalitási feltétel helyes.*

**Bizonyítás.** A 3.8.1 és 3.8.2 lemmákból következik.  $\square$

A 3.8.3 tétel a terminalitási kritérium helyességét bizonyítja, az algoritmus helyességéhez még az  $N$ -lista korrekt voltát kell megmutatni.

**Tétel 3.8.4** *Az  $N$ -lista korrekt.*

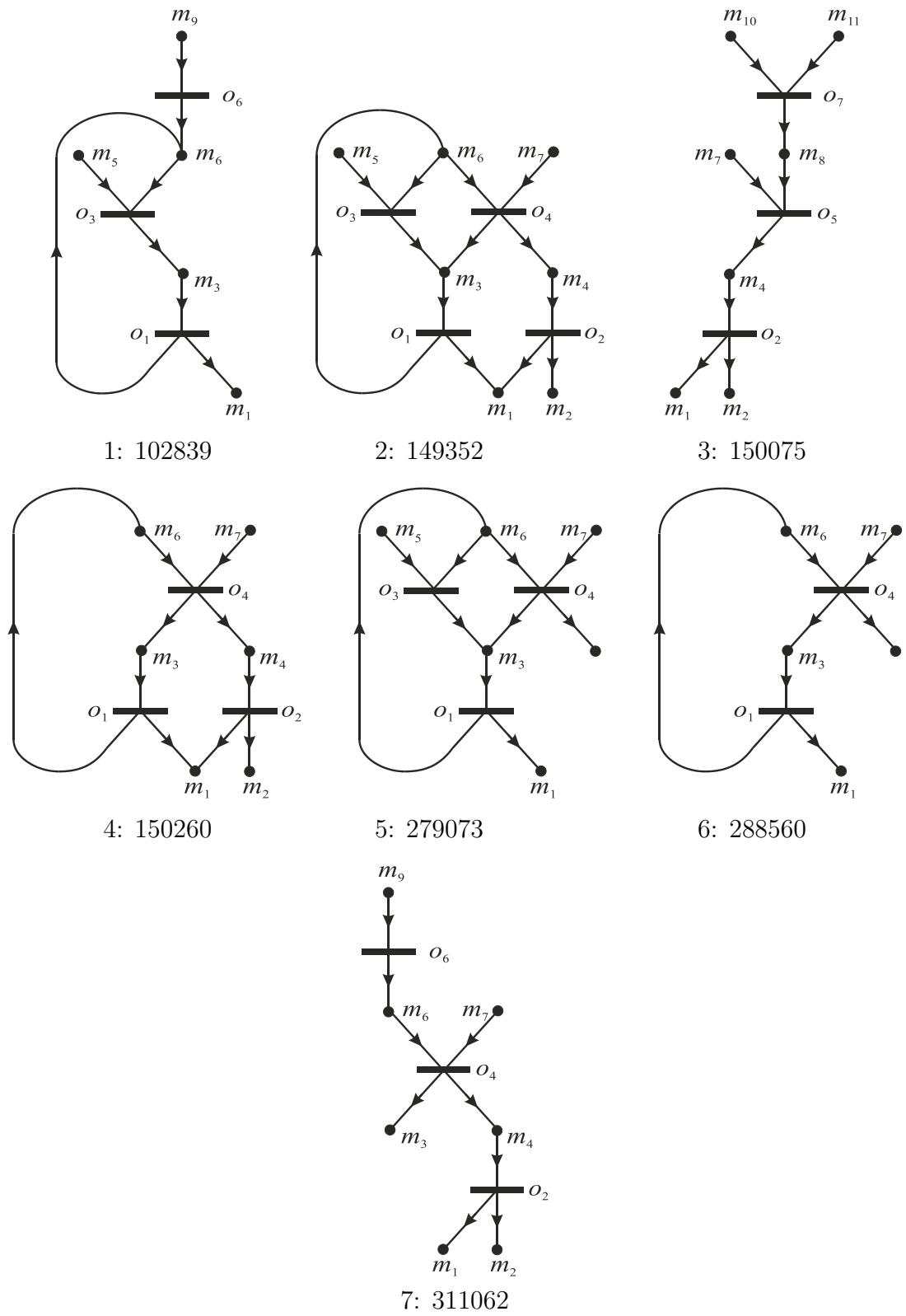
**Bizonyítás.** A 3.8.1 lemma szerint csak olyan részproblémát töröl az algoritmus, amely nem tartalmaz további lokálisan optimális struktúrákat. Így minden lokálisan optimális struktúrát az „ $N$ -lista Frissít” algoritmus (3.13 ábra) megpróbálja beszúrni a listába. A beszúrást korábban már részleteztük (3.8.2 fejezet), a lokálisan optimális struktúra definíciójának értelmében minden esetet megvizsgáltunk.  $\square$

### 3.8.4. Példa

A 2.2.3 fejezetben ismertetett feladatra futtatjuk le a módszert, annyi módosítással, hogy célfüggvény itt konkáv. Keressük a összes lokálisan optimális struktúrát (kis méretű feladatról lévén szó itt ez megtehető). A 3.14 ábra mutatja a megoldásokat, és a megoldáshoz tartozó költséget.

### 3.8.5. Az eredmény rövid összefoglalása

A 3.8 fejezet a 3b tézispontban megfogalmazott eredményeket tartalmazza.



3.14. ábra. A szemléltető példa lokálisan optimális struktúrái és a hozzá tartozó költség.

PNS feladatokra bevezettem a lokálisan optimális struktúrák fogalmát, amely lehetővé teszi az optimális megoldások mellett szuboptimális megoldások meghatározását. Kidolgoztam a kombinatorikusan gyorsított algoritmus egy változatát, amely képes ezen szuboptimális megoldások generálására.

## 3.9. Gyakorlati tapasztalatok

Az eddig tárgyalt a szeparábilis konkáv minimalizáláshoz kapcsolódó algoritmusokat megvalósítottam C++ nyelven. LP feladat megoldóként a Fábián Csaba által kidolgozott és kutatási célokra forráskódon rendelkezésünkre bocsátott LINX megoldót [28] használtuk.

### 3.9.1. Generált tesztfeladatok

Minden feladatmérethez 200 generált PNS feladatot [6] oldottunk meg a különböző megoldó módszerekkel.

#### Általános módszer

A korábban bevezetett partícionálási módszereket vizsgáljuk. A 3.2 – 3.5 táblázatok tartalmazzák a különböző méretű tesztfeladatokon végrehajtott futási eredményeket.

*Néhány megjegyzés az oszlopokhoz*

$\bar{x}$  jelöli a Shectman és munkatársai [89] által bevezetett partícionálást.

$\varepsilon$  jelöli a "csúsztatott" partícionálást.

Max jelöli a maximális rés partícionálást.

Érz. jelöli az érzékenységi vizsgálaton alapuló vágási stratégiát.

<sup>1</sup> Átlagolt megoldási idő. (mp)

<sup>2</sup> Megoldott LP-k átlagos száma.

<sup>3</sup> A felhasznált memória átlagos nagysága. (MB)

<sup>4</sup> Az optimális megoldást hányadik LP megoldása után lett azonosítva.

<sup>5</sup> Az időkorlát ( $10^6$  db LP megoldása) alapján nem megoldott feladatok száma.

<sup>6</sup> A memória korlát (300MB) alapján nem megoldott feladatok száma.

3.2. táblázat. Általános módszer: 20 műveleti egység

| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 0.19             | 639.83           | 0.04              | 49.5                     | 0                   | 0                    |
| $\varepsilon$ | 0.27             | 1007.32          | 0.01              | 72.9                     | 0                   | 0                    |
| Max           | 0.84             | 2541.55          | 0.1               | 27.9                     | 0                   | 0                    |
| Érz.          | 0.01             | 16.21            | 0.002             | 4.1                      | 0                   | 0                    |

3.3. táblázat. Általános módszer: 40 műveleti egység

| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 22.48            | 9171             | 1.858             | 645.2                    | 0                   | 0                    |
| $\varepsilon$ | 7.12             | 11131            | 0.291             | 2269.5                   | 0                   | 0                    |
| Max           | 11.51            | 16182            | 0.761             | 919.1                    | 5                   | 1                    |
| Érz.          | 0.13             | 49               | 0.011             | 8.5                      | 0                   | 0                    |

3.4. táblázat. Általános módszer: 60 műveleti egység

| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 139.07           | 49638            | 10.013            | 2400.7                   | 16                  | 1                    |
| $\varepsilon$ | 84.25            | 70621            | 2.894             | 2419.9                   | 12                  | 0                    |
| Max           | 121.92           | 44667            | 5.329             | 265.9                    | 23                  | 8                    |
| Érz.          | 2.07             | 394              | 0.069             | 8.1                      | 0                   | 2                    |

3.5. táblázat. Általános módszer: 80 műveleti egység

| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 181.42           | 93845            | 10.291            | 17088                    | 44                  | 12                   |
| $\varepsilon$ | 168.58           | 108925           | 4.608             | 3397                     | 43                  | 0                    |
| Max           | 100.99           | 42020            | 4.441             | 3863                     | 24                  | 10                   |
| Érz.          | 13.122           | 1172             | 0.309             | 8                        | 2                   | 4                    |

### **Kombinatorikusan gyorsított módszer**

A kombinatorikusan gyorsított eljárást (lásd 3.7 fejezet) teszteltük a különböző partícionálási stratégiákkal. Az érzékenységi vizsgálaton alapuló vágási módszer direkt nem építhető be a kombinatorikusan gyorsított keretalgoritmusba a félfolytonos változók miatt, így az nem szerepel a listában. Ezen stratégia kibővítése a jövő egyik feladata. A 3.6 – 3.9 táblázatok tartalmazzák a különböző méretű tesztfeladatokon végrehajtott futási eredményeket.

### **Elemzés**

Az átlagolt eredmények számításakor csak az összes megoldó módszer által megoldott feladatokat vettük figyelembe. Ezzel sajnos a feladatokat jól megoldó módszereket „büntetjük”, hiszen az általuk megoldott, de egy rosszabb módszer által nem megoldott feladatokat figyelmen kívül hagyjuk, ezért a nem megoldott feladatok számát külön kiemeltük.

#### **3.9.2. Valós ipari feladatok**

Két valós ipari feladaton teszteltük a módszereket (lásd 3.15 és 3.16 ábrák). A 3.10 – 3.13 táblázatok tartalmazzák az eredményeket. A feladatok leírását a [39] tartalmazza.



3.6. táblázat. Kombinatorikusan gyorsított módszer: 20 műveleti egység

| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 0.11             | 309.2            | 0.044             | 53.2                     | 0                   | 0                    |
| $\varepsilon$ | 0.08             | 227.5            | 0.022             | 15.6                     | 0                   | 0                    |
| Max           | 0.04             | 124.1            | 0.003             | 7.3                      | 0                   | 0                    |

3.7. táblázat. Kombinatorikusan gyorsított módszer: 40 műveleti egység

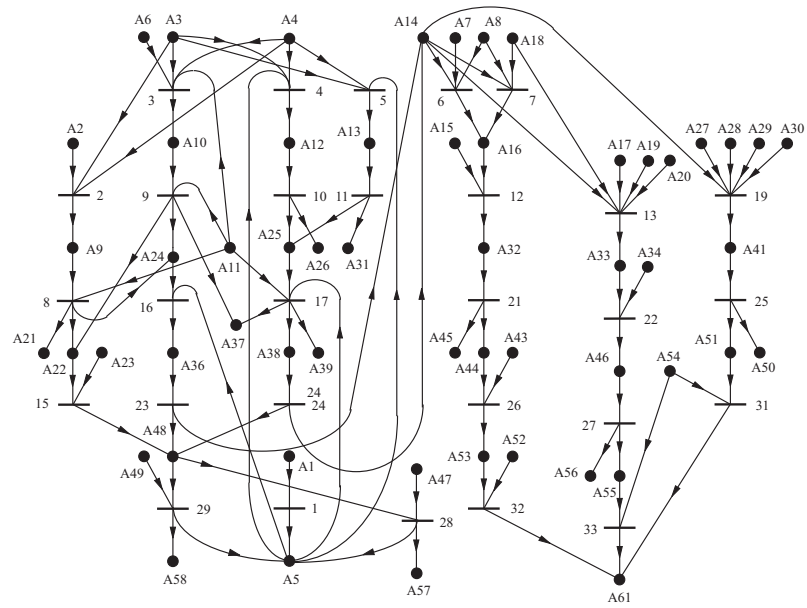
| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 22.23            | 8063             | 1.897             | 552                      | 0                   | 0                    |
| $\varepsilon$ | 6.22             | 4401             | 0.875             | 133                      | 0                   | 0                    |
| Max           | 0.181            | 246              | 0.015             | 12                       | 0                   | 0                    |

3.8. táblázat. Kombinatorikusan gyorsított módszer: 60 műveleti egység

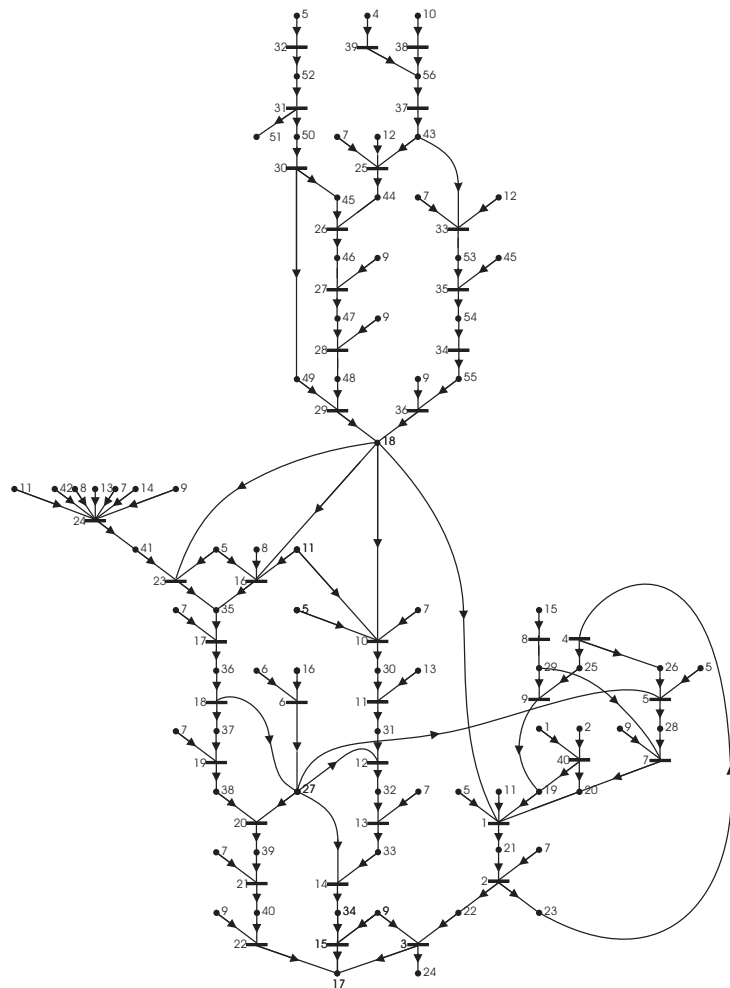
| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 160.72           | 68927            | 10.616            | 1816                     | 6                   | 1                    |
| $\varepsilon$ | 139.46           | 36848            | 9.984             | 651                      | 0                   | 4                    |
| Max           | 2.15             | 1952             | 0.146             | 146                      | 0                   | 0                    |

3.9. táblázat. Kombinatorikusan gyorsított módszer: 80 műveleti egység

| Stratégia     | Idő <sup>1</sup> | #LP <sup>2</sup> | Mem. <sup>3</sup> | Opt.meghat. <sup>4</sup> | Lim.LP <sup>5</sup> | Lim.Mem <sup>6</sup> |
|---------------|------------------|------------------|-------------------|--------------------------|---------------------|----------------------|
| $\bar{x}$     | 438.65           | 109955           | 31.273            | 3423                     | 8                   | 25                   |
| $\varepsilon$ | 351.08           | 87853            | 31.756            | 9402                     | 0                   | 25                   |
| Max           | 6.22             | 3685             | 0.497             | 437                      | 0                   | 0                    |



3.15. ábra. Denmark feladat P-gráf ábrázolása.



3.16. ábra. Alpha feladat P-gráf ábrázolása.

3.10. táblázat. Általános módszer: Alpha (41 műveleti egység)

| Stratégia     | Idő(mp) | #LP   | Memória (MB) | Opt.meghat. |
|---------------|---------|-------|--------------|-------------|
| $\bar{x}$     | 41.40   | 44765 | 10.63        | 295         |
| $\varepsilon$ | 9.48    | 16639 | 0.30         | 359         |
| Max           | 1.48    | 2592  | 0.04         | 793         |
| Érz.          | 0.65    | 225   | 0.03         | 7           |

3.11. táblázat. Kombinatorikusan gyorsított módszer: Alpha (41 műveleti egység)

| Stratégia     | Idő(mp) | #LP   | Memória (MB) | Opt.meghat. |
|---------------|---------|-------|--------------|-------------|
| $\bar{x}$     | 25.59   | 33779 | 11.63        | 21          |
| $\varepsilon$ | 0.42    | 679   | 0.02         | 21          |
| Max           | 0.57    | 1043  | 0.01         | 18          |

3.12. táblázat. Általános módszer: Denmark (35 műveleti egység)

| Stratégia     | Idő(mp) | #LP  | Memória (MB) | Opt.meghat. |
|---------------|---------|------|--------------|-------------|
| $\bar{x}$     | 0.32    | 1022 | 0.285        | 1           |
| $\varepsilon$ | 1.17    | 3521 | 0.051        | 1           |
| Max           | 0.09    | 181  | 0.002        | 1           |
| Érz.          | 0.01    | 1    | 0.001        | 1           |

3.13. táblázat. Kombinatorikusan gyorsított módszer: Denmark (35 műveleti egység)

| Stratégia     | Idő(mp) | #LP  | Memória (MB) | Opt.meghat. |
|---------------|---------|------|--------------|-------------|
| $\bar{x}$     | 0.51    | 1322 | 0.47         | 1           |
| $\varepsilon$ | 0.51    | 1042 | 0.17         | 1           |
| Max           | 0.09    | 217  | 0.01         | 1           |

### 3.10. Alkalmazás: ipari hőellátó rendszer optimális tervezése

Vegyipari rendszerek jellemzője a nagy hőenergiaigény. A hőenergia szükségletet az esetek jelentős részében vízgőz kondenzációjával biztosítják. A gőzt kazánházakban állítják elő vagy ellennyomásos erőműből szolgáltatják. A gőz előállításának és felhasználási helyének távolsága akár több kilométer is lehet, így a gőzellátó hálózat paraméterei (pl. csővezeték keresztmetszete, szigetelése) a hálózat beruházási- és üzemeltetési-költségét is jelentősen befolyásolják.

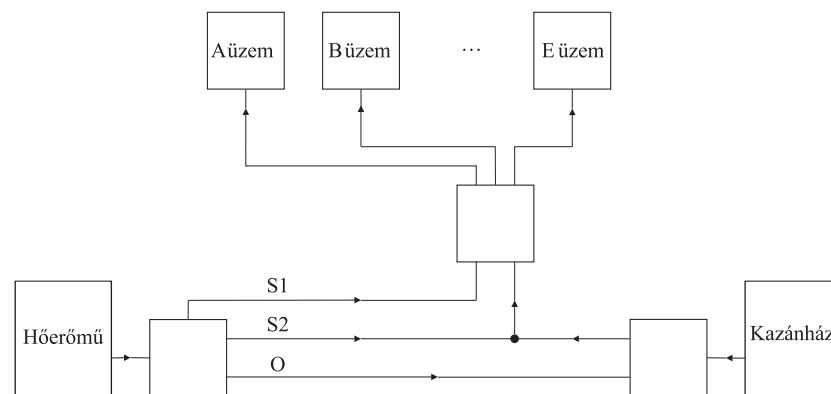
Összetett vegyipari rendszerekben a szakaszos üzemű és az időszakosan működte-tett technológiáknak köszönhetően a felhasználás helyén fellépő gőzigény jelentős ingadozást mutathat. A gőzigény időbeni változása tovább nehezíti az optimális gőzellátó rendszer meghatározását, hiszen a gőzellátó rendszernek az időben változó gőzigény összességét tekintve kell optimálisan működnie, miközben eléggé rugalmasnak kell lennie ahhoz, hogy a fellépő igénymaximumokat is biztosítani tudja.

Számos közlemény foglalkozik a változó igényű hőellátó hálózat optimalizálásával. Például egy multiperiodikus optimalizálás eljárást mutat be a [78], [79]; egy kombinatorikus optimalizálási módszert ismertet a [76]; csak a gőznyomás optimalizálását végzi a [70].

A fejezetben ismertetett eljárás természetesen tetszőleges változó igényű PNS feladatokra is használható.

#### 3.10.1. Gőzhálózat, kiindulási feltételek

A gőzellátó rendszer eredeti struktúráját a 3.17 ábra szemlélteti. Az erőmű és a kazánház gőzelosztókon keresztül csatlakozik az  $S1$ ,  $S2$ ,  $O$  jelű vezetékhez. Az  $A, \dots, E$  üzemek az  $S1$  és  $S2$  csatlakozása után  $S2$ -ből leágazásokon keresztül kapják a gőzt. Az üzemi leágazások viszonylag közel vannak egymáshoz, mintegy 20-50m-re, az erőmű és az első leágazás közötti csőhossz körülbelül 1800m, a kazánház és az erőmű közötti csőhossz pedig 2 km. Az  $O$  jelű vezeték eredetileg használaton kívül van, csak az  $S1$  és  $S2$  vezetékeket használják a gőzigény ellátására.

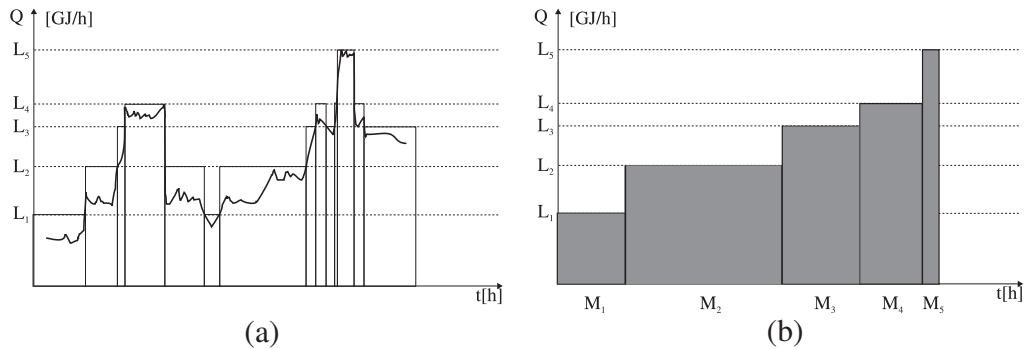


3.17. ábra. Az eredeti hőellátó rendszer sematikus ábrázolása.

### 3.10.2. Alternatív megoldási lehetőségek

Az előbbieket figyelembevételével a következő gőzellátási lehetőségeket fogalmazzuk meg:

1. Megszüntetjük a kazánházat (hőellátás csak a hőerőmű által történik).
  - (a) Az áramlási keresztmetszet növelése. Ez az *O* jelű vezeték üzembe helyezésével és további csővezetékek kiépítésével valósítható meg.
  - (b) A nyomáscsökkenés növelése.
    - i. Megnöveljük a gőz induló nyomását.
    - ii. A csőben a gőzelvételi helyen a nyomás csökkentése.
2. A kazánház további üzemeltetése.
  - (a) A tápvízellátás megváltoztatásával.
  - (b) A kazánház időszakos üzemeltetésével.



3.18. ábra. (a) Időben változó hőigény trend. (b) A diszkrétizálás eredményeként kapott kumulatív hőteljesítmény trend.

### 3.10.3. Módszer

A változó hőigény miatt a korábban bemutatott hálózatszintézis módszerek közvetlenül nem használhatók. A gőzfogyasztó egységek összesített hőteljesítmény-igényének az időbeni alakulását kapjuk meg bemenetként. Hogy az optimalizáló módszert alkalmazni tudjuk az alábbiak szerint diszkrétizáljuk, alakítjuk át a hőteljesítmény-igény trendet.

1. Megkeressük a minimum és maximum hőteljesítmény-igény értékeket. A nullától a maximum értékig terjedő tartományt felosztjuk  $n$  részre úgy, hogy a legalsó teljesítménysáv felső határa a talált minimum érték fölött legyen (diszkrétizálás).
2. A trenden végighaladva az egyes hőteljesítmény-igény értékekhez tartozó időtartamokat teljesítménysávonként összegezzük.

Ezzel az átalakítással a hőteljesítmény-igény trendből  $n$  darab (öt) teljesítményszintet kapunk a hozzá tartozó időtartamokkal együtt, amit a hőteljesítmény-igény trendhez hasonlóan teljesítmény - idő koordináta-rendszerben ábrázolhatunk (3.18 ábra).

Ennek alapján már egyszerűen megfogalmazható a feladat: határozzuk meg azt a gőzellátó rendszert, amely képes az  $n$  (öt) féle eltérő teljesítmény egyenkénti biztosítására úgy, hogy az egyes teljesítményszintekhez tartozó időtartamokkal súlyozott

költségösszeg minimális legyen. Az alternatív megoldásokból létrehozuk a maximális struktúrát (3.19 ábra).

Egy termékhez (hőigény szinthez) tartozó hálózatot annyi példányban kell vennünk, ahány termékünk van. Egy termékhez tartozó hálózaton belül az ismétlődő, többször előforduló műveleti egységek közül a megoldásstruktúrában csak egy szerepelhet, hiszen egy adott időpillanatban az adott műveleti egység nem dolgozhat párhuzamosan önmagával. A különböző termékek különböző időpontokban történő gőzigényt elégítenek ki, így a különböző termékekhez tartozó hálózatok egyidejűleg is tartalmazhatják ugyanazt a műveleti egységet, esetleg megegyező vagy eltérő diszkrét állapotban. Lényeges azonban, hogy minden műveleti egység esetleges beruházási költsége csak egyszer szerepelhet a költségfüggvényben.

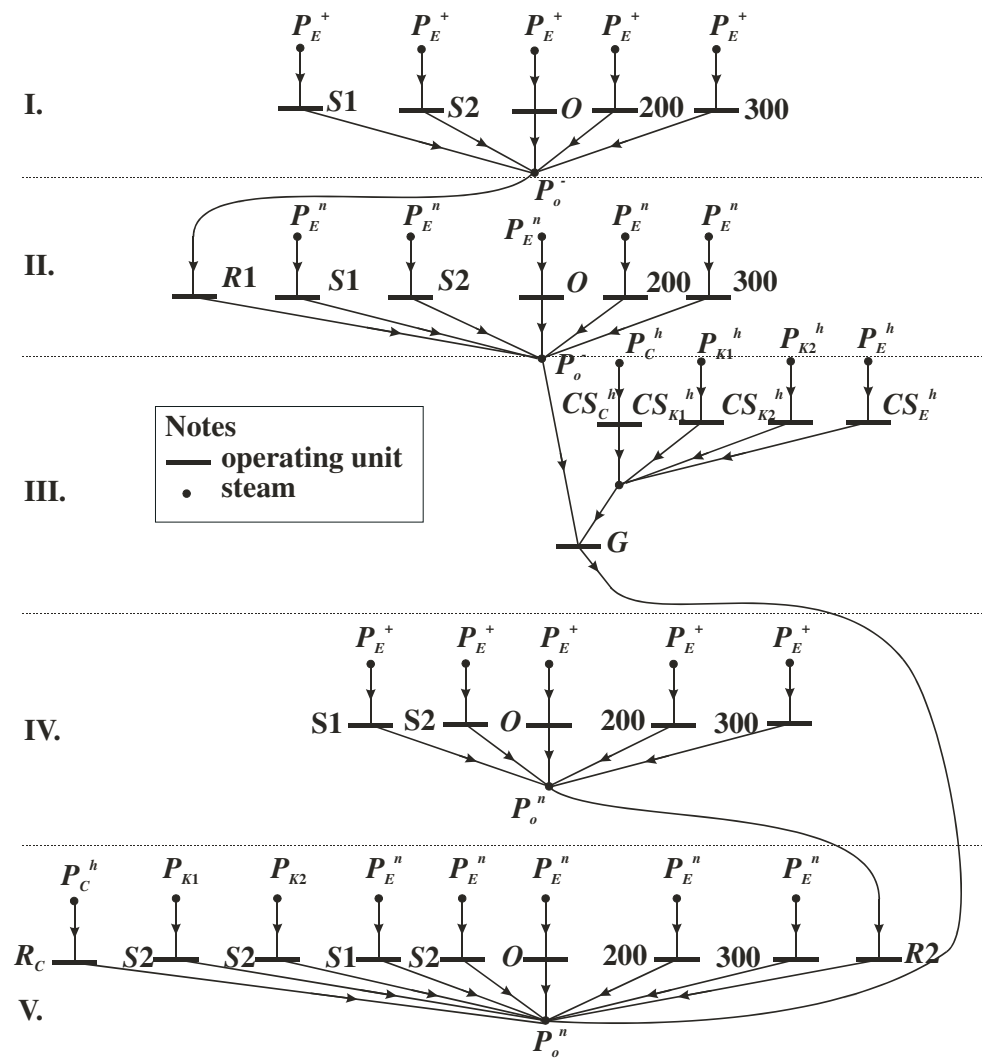
### Matematikai modell

$$\begin{aligned}
 \sum_{j=1}^{n_i} y_{ij}^k &\leq 1, \quad k \in P, \quad i = 1, 2, \dots, N \\
 z_{ij} &\leq \sum_{k \in P} y_{ij}^k, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n_i \\
 y_{ij}^k &\leq z_{ij}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n_i, \quad k \in P \\
 x_{ij}^k &\leq y_{ij}^k u_{ij}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n_i, \quad k \in P \\
 \sum_{i=1}^N \sum_{j=1}^{n_i} s_{il} x_{ij}^k &= 0, \quad l \in M \setminus R \setminus P, \quad k \in P \\
 \sum_{i=1}^N \sum_{j=1}^{n_i} s_{ik} x_{ij}^k &\geq p_k, \quad k \in P \\
 \sum_{i=1}^N \sum_{j=1}^{n_i} -s_{il} x_{ij}^k &\leq q_l, \quad k \in P, \quad l \in R
 \end{aligned}$$

*Éves költség:*

$$\sum_{i=1}^N \sum_{j=1}^{n_i} \left( \frac{a_{ij}}{E} z_{ij} + \sum_{k \in P} (c_{ij} t^k + d_{ij} x_{ij}^k t^k) \right) + \sum_{l \in R} \left( r_l \sum_{k \in P} \sum_{i=1}^N \sum_{j=1}^{n_i} -s_{il} t^k x_{ij}^k \right),$$

ahol



3.19. ábra. Egy adott hőteljesítményhez tartozó maximális struktúra.



- $y_{ij}^k$  : bináris változó az  $i$  műveleti egység  $j$  diszkrétizált esetének létezése az adott struktúrában, a  $k$  termék előállításánál,
- $x_{ij}^k$  : változó ( $x_{ij}$ ) az  $i$  műveleti egység  $j$  diszkrétizált esetének kapacitása  $k$  termék előállításában, ( $x_{ij}^k \geq 0$ ),
- $z_{ij}$  : bináris változó, az  $i$  műveleti egység  $j$  diszkrétizált esetének a fixköltség számításánál,
- $u_{ij}$  : az  $i$  műveleti egység  $j$  diszkrétizált esetének kapacitás korlátja,
- $n_i$  : az  $i$  műveleti egység diszkrétizált eseteinek a száma,
- $a_{ij}$  : az  $i$  műveleti egység  $j$  diszkrétizált esetének a beruházási költségének állandó része,
- $c_{ij}$  : az  $i$  műveleti egység  $j$  diszkrétizált esetének a működtetési költségének állandó része,
- $d_{ij}$  : az  $i$  műveleti egység  $j$  diszkrétizált esetének a kapacitással arányos működtetési költsége,
- $r_l$  :  $l$  nyersanyag egységnyi mennyiségére vonatkozó költsége,  $l \in \mathcal{R}$ ,
- $q_l$  : az  $l$  nyersanyagra megadott felsőkorlát,  $l \in \mathcal{R}$ ,
- $p_k$  : a  $k$  termékre megadott teljesítményigény,  $k \in \mathcal{P}$ ,  $t/hour$ ,
- $t_k$  :  $k$  terméknek megfelelő teljesítményszint biztosítása órákban,  $k \in \mathcal{P}, hour$ ,
- $s_{il}$  : az  $l \in \mathcal{M}$  anyagnak az  $i$  műveleti egység által szolgáltatott ( $s_{il} > 0$ ) ill. felhasznált ( $s_{il} < 0$ ) gőzteljesítménye (tonna/óra) egységnyi kapacitás mellett ( $s_{il} = 0$  esetén értelemszerűen az  $i$  műveleti egységnek nincs kapcsolódása az  $l$  anyaghoz),
- $E$  : megtérülési idő években.

A modellünk 530 bináris változót tartalmaz; az eljárásunk 550.41 másodperc alatt azonosította az optimális hálózatot egy PC-n (Athlon 1.33GHz, 1852 MFLOPS, 3687 MIPS), a megoldott LP feladatok száma 115596 volt. Az éves költség 143,905,000 HUF, amely 8 százalékkal kevesebb, mint a jelenlegi költség.

## 4. fejezet

# A folyamat- és hőcserélőhálózat együttes szintézise

### 4.1. Bevezetés

A folyamathálózat szintéziséről korábban már a 2. fejezetben részletesen írtunk. Bevezetésünkben a hőcserélőhálózatokról teszünk említést.

#### 4.1.1. Hőcserélőhálózat szintézis

A hőcserélőhálózatok (HENS) szintézise az egyik legfontosabb területe a folyamattervezés tudományának. Az utóbbi időben az egyik legintenzívebben kutatott területek közé tartozik, több száz publikáció jelent meg e témában az elmúlt évtizedekben. A fontossága annak is tulajdonítható, hogy a vegyipari rendszerek költségeinek jelentős része az energiaköltség, ezen belül is a hőenergiáé, aminek hasznosítása kiemelten fontos.

Az első publikáció, amely a hőcserélőhálózatok tervezésével foglalkozik 1944-ben jelent meg [93], az első teljes hőcserélőhálózat-tervezést 1965-ben publikálták [52], 1969-ben jelent meg az első szigorú matematikai leírás [71].

Az általános HENS problémát a következőképpen definiáljuk:

Adottak:

- a meleg anyagáramok halmaza ( $\mathcal{F}^H$ ), melyeket a bemenő hőmérsékletről kell

lehűteni a kimenő hőmérsékletre;

- a hideg anyagáramok halmaza ( $\mathcal{F}^C$ ), melyeket a bemenő hőmérsékletről kell felmelegíteni a kimenő hőmérsékletre;
- az anyagáram hőkapacitása és az anyagáram nagysága;
- a rendelkezésre álló külső meleg és hideg források, a megfelelő hőmérsékletekkel együtt, és ezen források költsége;
- hőcserélők költsége.

Cél: egy költségoptimalis hőcserélőhálózat meghatározása, amely kielégíti a fenti igényeket. A feladat nem könnyű, az általános hőcserélőhálózatok szintézise NP nehéz probléma [41].

#### 4.1.2. Az integrált folyamat- és hőcserélőhálózat szintézise

Integrált folyamat- és hőcserélőhálózat szintézise során a teljes folyamat- és hőcserélőhálózat meghatározása azonos időben történik. A szekvenciális módszerek esetében először meghatározzák magát az optimális folyamathálózatot és utána határozzák meg az optimális hőcserélő hálózatot. Könnyen látható, hogy az utóbbi nem vezet optimális megoldáshoz, hiszen az optimális folyamathálózat meghatározásakor figyelmen kívül hagyják a hőcserével kapcsolatos információkat.

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

#### Anyagáramhoz kapcsolódó hőáram

A kiterjesztett feladatban az anyagáramhoz tartozhat hőmérséklet is. Ha egy műveleti egység kimenő anyagárama közvetlenül kapcsolódik egy másik műveleti egység bemenő anyagáramára és a definiált hőmérsékletek különbözőek, akkor hőbevitel, illetve hőelvitel válik szükségessé, az itt fellépő hő mennyisége az anyagárammal arányos.

## Rejtett hő

Rejtett hőnek nevezzük azokat a hőigényeket, amikor a műveleti egységgel úgy valósítunk meg hőforgalmat, hogy közben a műveleti egység hőmérséklete állandó marad. Ilyen lehet például a halmazállapot-változást és a kémiai átalakulást kísérő entalpiaváltozás. Egyes műveleti egység típusoknál a művelet során betáplált, illetve elvont rejtett hő mennyisége többszöröse lehet annak a hőforgalomnak, amit a műveleti egységbe belépő, illetve kilépő anyagáramok képviselnek. Erre jó példa a rektifikáló kondenzátora és kiforralója, vagy egy erősen exoterm illetve endoterm hőszínezetű reaktor. Ezen okok miatt a folyamatszintézis hőintegrációval való kiterjesztésekor szükséges a rejtett hő figyelembevétele, és a matematikai modellbe való beépítése. A rejtett hőt típusától függetlenül úgy definiáljuk a modellben, hogy minden egyes esetben megadjuk a hőmérsékletet és azt a hőmennyiséget is, ami a rejtett hő forrásához tartozik a műveleti egység egységnyi kapacitása mellett.

## 4.2. A szakirodalom áttekintése

Először az általános HENS módszerek szakirodalmát tekintjük át, majd az integrált módszerekről szólunk.

### 4.2.1. Általános HENS módszerek

A HENS módszereket két fő csoportra oszthatjuk: szekvenciális módszerek és a teljes HEN szintézis.

#### Szekvenciális szintézis

A szekvenciális szintézis során a feladatot olyan részfeladatokra bontjuk, amelyek különböző cél szerint oldják meg a problémát. A célok között egy sorrendet állítunk fel, amely általában valamilyen heurisztikán alapszik. A megfogalmazott részfeladatokat a célok szerint csökkenő sorrendben oldjuk meg és az előző feladat eredményét alkalmazzuk a következő feladat megoldásakor. A hőcserélőhálózatok szintézise folyamán általában a következő három célt szokták használni:

1. a külső meleg és hideg források használatának minimalizálása,
2. a hőcserélők számának minimalizálása,
3. a hőcserélő felület minimalizálása.

Így azt a megoldást kapjuk, amely minimális külső forrást használ ezen belül a legkevesebb hőcserélővel és ezen belül a legkisebb hőcserélő felületet használja [7].

A szekvenciális módszereket két fő csoportra oszthatjuk:

1. Evolúciós módszerek: pinch eljárás (PDM) [2], [64], [65]; duális hőmérséklet [13], [94]; és pszeudó-pinch eljárás [87], [94], [99].
2. Matematikai programozási módszerek: vegyes egész lineáris egyenletek megoldásának sorozatára épülnek a [12], [80], illetve nemlineáris optimalizálási feladatok megoldására épül a [32].

A pinch módszer egy olyan grafikus eljárás, amely a hőmérséklet intervallumokat felhasználva számolja ki a minimálisan felhasznált külső energia mennyiségét. Az eljárás közben a rendszer szűk keresztmetszeteit is megkapjuk, ezeket nevezzük pinch pontoknak. A pinch pontok hőmérsékleti pontok, amelyeken keresztül nem történik hőátadás. A feladat a pinch pontok mentén felbontható részfeladatokra. A duális hőmérséklet módszer megengedi a hőcserét a pinch pontokon keresztül, ezáltal a kapott hálózat kevesebb hőcserélőt tartalmaz, illetve a hálózat struktúrája egyszerűsödik. A pszeudó-pinch tervezés szintén lazítja a pinch feltételt. A pinch pontok mentén történő partícionálási stratégián alapszik a vertikális hőcsere elv [47], [48].

A matematikai programozási módszeren alapuló eljárás a korábban említett három részfeladatot oldja meg. Egy részfeladat megoldásértéke paraméterként szolgál a következő részfeladat számára. A minimális külső hőforrás meghatározására lineáris programozási modellt írnak fel a [12], [80] szerzői, vegyes egész lineáris feladatot (MILP) és vegyes egész nemlineáris feladatot (MINLP) használ [43], [44], amelyek már strukturális megszorításokat is tartalmaznak. A minimális külső hőforrás meghatározása mellett hőcserét is meghatározza a [12], [80].

## Teljes szintézis

A szekvenciális szintézissel ellentétben itt az a cél, hogy a feladat dekomponálása nélkül határozzuk meg az optimális hálózatot. Általában ezek vegyes egész nem-lineáris programozási feladatot (MINLP) fogalmazznak meg a feltételektől függően. Az egyik legkorábban publikált HENS modell a [103], aminek a hátránya, hogy nem engedélyezi a hőáramok megosztását. Egy másik MINLP feladat a [31]-ben található, amely a hőmérsékleti intervallumok partícionálásán alapszik. A [101]-ben publikált modell feltételezi, hogy egy megosztott hőáram csak egy hőcserélőn megy keresztül, így a feltételrendszer lineáris lesz. [20] szerzői bevezettek egy módszert, amely képes alsó és felső korlát meghatározására egy HENS feladatnál.

### 4.2.2. Integrált folyamat- és hőcserélőhálózat szintézise

A HENS feladatot megoldó módszerek közvetlenül nem használhatók, mivel az anyagáramok nagysága nem adott (lásd a HENS feladat definícióját), így a hőáramok hőtartalma ismeretlen. Az integrált PNS-HENS módszereket a folyamatszintézis típusa szerint különböztethetjük úgy meg, mint szakaszos és folytonos. A jelenlegi munkánk a folytonos típushoz tartozik, de röviden kitérek a szakaszos esetre is.

### Folytonos PNS-HENS integrált módszerek

Az integrált hőcserélő- és folyamatszintézis módszerek általában már meglévő HENS és PNS eljárások módosítását használják. Az itt említett eljárásokat a HENS rész szerint tárgyaljuk.

Pinch eljáráson alapuló módszerek: az [54] szerzői a hőmérséklet-entalpia diagramot terjesztik ki; a [66] a folyamattervezés feladatot vizsgálja a pinch módszer felhasználásával. A [104] dolgozatban úgy valósítanak meg HENS retrofit tervezést, hogy a kapcsolódó folyamatban a folyamértékek megváltozhatnak. Az [51] is Pinch technológiát alkalmaz, több különböző folyamat közös külső hideg és meleg forrásainak optimalizálására.

Nemlineáris folytonos (NLP) modelleket vezet be a [25], ahol a modell egyszerre

optimalizálja a folyamatot, minimális külső hideg vagy meleg forrást és a hőmérsékleteket. A szerzők tapasztalataik alapján megállapítják a korábban már említett észrevételt, hogy jelentős eltérés van a költségekben a szimultán optimalizálás és a szekvenciálisan végrehajtott PNS-HENS között.

Vegyes egész lineáris modellt alkalmaznak a következő munkák. A [77]-ben bemutatott módszer két fő lépésből áll: egy belső lépés a hőintegrációt valósítja meg, és közben egy külső lépésben pedig a hálózatot optimalizálja. A [19] cikk a szintézisfeladathoz kapcsolódó keretalgoritmust mutat be, amely a rendszer részeit megfelelő esetekben egyesíti, illetve dekomponálja. Az eljárást egy desztilláló rendszerre alkalmazza, maga a MILP modell azonos a [20]-ban leírt modellel. A módszer képes kiszűrni a lehetséges alternatívák egy olyan részhalmazát, amely már nem lehet optimális.

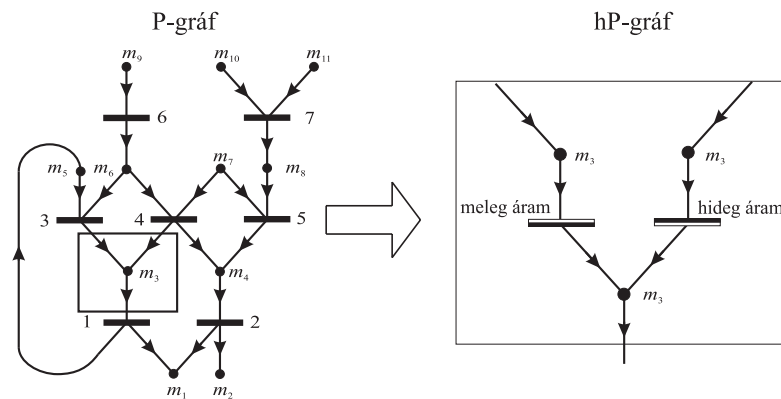
Vegyes egész nemlineáris modellt (MINLP) alkalmaznak: a [102] feltételezi, hogy csak egy meleg és hideg külső hőforrás áll rendelkezésre; a [46] szerzői elemzik a hőintegráció nehézségeit, majd a [25]-ben szereplő MINLP modellt analizálva jutnak el a feladat egy új MINLP megfogalmazásához. A [21] egy kereteljárást ad a folyamat hierarchikus dekompozíciójára, az optimalizációs lépések a dekompozíció által meghatározott szintenként történnek, ellentétben az eddigi módszerekkel, amelyek egy nagy MINLP feladatot definiáltak. Itt sok kis MINLP feladatot kell megoldani, így kérdéses, hogy a globális optimumot mennyire tudja garantálni az eljárás.

Egyéb eljárások: a [62] szekvenciális folyamatszintézis módszert használ, amely egy interfészen keresztül kapcsolódik a HENS megoldóhoz. A [91] szerzői áttekintést adnak a legújabb eredményekről a folyamat integrációban, a munkában külön fejezet foglalkozik a hőcserélő hálózatokkal.

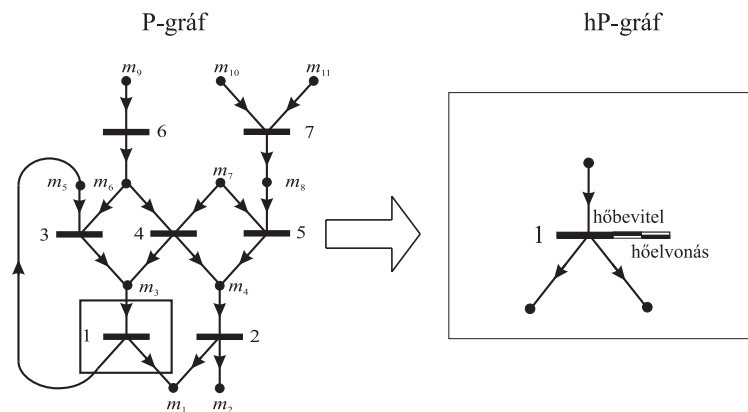
### **Szakaszos PNS-HENS integrált módszerek**

A szakaszos folyamatok esetében nehezíti a feladatot, hogy egyben ütemezési problémákat is meg kell oldanunk. MILP modell felírásával jutnak el a megoldásig a [96], [105], [106] munkák. Heurisztikával keres megoldást a [97], majd egy MINLP modell felírásával javítja a korábban megtalált megoldást. Tisztán heurisztikus megközelítést alkalmaznak a [14] és [15] dolgozatok.





4.1. ábra. Hőáramok reprezentálása.

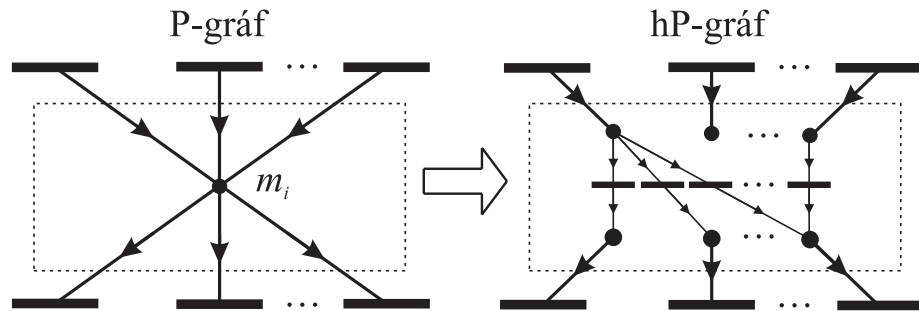


4.2. ábra. Rejtett hő reprezentálása.

### 4.3. A hP-gráf

A korábban bevezetett P-gráf reprezentációt bővítjük ki úgy, hogy képes legyen a lehetséges hőbevitel és hőelvonások reprezentálására.

Az anyagáramhoz kapcsolódó hőforgalom hP-gráf reprezentációját mutatja be a 4.1 ábra. A műveleti egységhez kapcsolódó rejtett hő hP-gráf reprezentációját mutatja be a 4.2 ábra.



4.3. ábra. Az anyag típusú pont kiterjesztése

#### 4.3.1. Az anyagpont kiterjesztése

Olyan esetekben, ahol  $m_i$  anyagot több műveleti egység is gyártja, illetve fogyasztja; elképzelhető, hogy azt különböző hőmérsékleten végzik. A P-gráf figyelmen kívül hagyja az anyagok hőmérsékleti paramétereit, a több műveleti egység által termelt anyagokat összekeveri, így a hőmérsékletre vonatkozó paraméterek torzulnának. Ilyen esetekben a hP-gráfban minden termelő-fogyasztó műveleti egység párra külön kell meghatározni a két műveleti egység között átáramlott anyagmennyiséget. Ennek érdekében az anyagpontot felbontjuk mesterséges anyagpontokká és mesterséges műveleti egységekké. A kiterjesztésnek az általános ábráját mutatja be a 4.3 ábra.

### 4.4. Kiindulási adatok, halmazok

#### 4.4.1. Hideg és meleg hőáramok

Jelölje  $prod(m_i)$  az  $m_i$  anyagot előállító és  $feed(m_i)$  az anyagot fogyasztó műveleti egységek halmazát, azaz

$$\begin{aligned} prod(m_i) &= \{o_k : o_k = (\alpha, \beta) \in \mathcal{O}, i \in \beta\}, \\ feed(m_i) &= \{o_k : o_k = (\alpha, \beta) \in \mathcal{O}, i \in \alpha\}. \end{aligned}$$

Egy hőáramot egy hármassal jellemezhetünk attól függően, hogy melyik anyagról van szó, és mely műveleti egység bemenetén illetve kimenetén fordul elő. Figyelembe

véve az anyagok hőmérsékleti paramétereit, a lehetséges hideg és meleg áramok:

$$\begin{aligned}
\mathcal{F}^H &= \{(i, k, l) : t_{ik}^{out} > t_{il}^{in}, y_k \neq 0, y_l \neq 0, m_i \in \mathcal{M}, o_k \in \text{prod}(m_i), o_l \in \text{feed}(m_i)\} \\
&= \{FH_1, FH_2, \dots, FH_{n_{FH}}\}, \\
\mathcal{F}^C &= \{(i, k, l) : t_{ik}^{out} < t_{il}^{in}, y_k \neq 0, y_l \neq 0, m_i \in \mathcal{M}, o_k \in \text{prod}(m_i), o_l \in \text{feed}(m_i)\} \\
&= \{FC_1, FC_2, \dots, FC_{n_{FC}}\}.
\end{aligned}$$

A hőáramokat definiáló indexek alapján meg tudjuk mondani egy hőáram kezdő- és véghőmérsékletét, például  $FC_j = (i, k, l) \in \mathcal{F}^C$ -re  $t_{ik}^{out}$ ,  $t_{il}^{in}$  kezdő- illetve véghőmérséklete. A könnyebb megértés céljából ezen kezdő- és véghőmérsékleteket jelölje  $t_0^{(i,k,l)}$  és  $t_1^{(i,k,l)}$ . Hasonlóan  $FH_j = (i, k, l) \in \mathcal{F}^H$ -ra azzal a kikötéssel, hogy  $t_0^{(i,k,l)}$  jelentse mindig a kisebb hőmérsékletet. (Meleg áramoknál a kezdő hőmérséklet a magasabb, és ezt fordítsuk meg a későbbi könnyebb jelölés miatt.)

#### 4.4.2. Rejtett hő

A feladat definiálásakor szükséges megadni a rejtett hő paramétereit. Legyen  $o_k \in \mathcal{O}$ , és jelölje  $l_k$  az  $o_k$ -hoz tartozó rejtett hőforrások számát. Jelölje  $h_{ki}$  az  $i$ . rejtett hőforráshoz tartozó paramétert, amely megmondja az  $o_k$  egységnyi működésekor az időegységre eső betáplálandó, illetve elvonandó hő mennyiségét. Legyen  $TM_{ki}$  az  $i$ . rejtett hő hőmérsékleti paramétere ( $i = 1, \dots, l_k$ ).

A műveleti egységekhez kapcsolódó rejtett hőt két indexszel jellemezhetjük. Az első index megmondja, hogy mely műveleti egységhez tartozik a rejtett hő, a második pedig a műveleti egységen belüli sorszámát jelöli. A lehetséges meleg és hideg rejtett hőforrások halmaza legyen  $(\mathcal{L}^H, \mathcal{L}^C)$ .

$$\begin{aligned}
\mathcal{L}^H &= \{(k, j) : h_{kj} > 0, y_k \neq 0, o_k \in \mathcal{O}\} = \{LH_1, \dots, LH_{n_{LH}}\}, \\
\mathcal{L}^C &= \{(k, j) : h_{kj} < 0, y_k \neq 0, o_k \in \mathcal{O}\} = \{LC_1, \dots, LC_{n_{LC}}\}.
\end{aligned}$$

Egy  $(j, k) \in \mathcal{L}^H \cup \mathcal{L}^C$ -ra legyen  $t^{(k,j)}$  a rejtett hőforráshoz kapcsolódó hőmérséklet.

#### 4.4.3. A hideg áramok eltolása

Minden hideg hőáram, illetve hideg külső forrás hőmérsékleti paramétereit el kell tolni pozitív irányba minimális megközelítési távolsággal ( $MT$ ). A továbbiakban, amikor a

hideg áramok, rejtett hőforrások hőmérsékleteiről beszélünk, akkor már a módosított értékekkel dolgozunk.  $(i, k, l) \in \mathcal{F}^C$ -ra a  $t_0^{(i,k,l)} + MT$ ,  $(k, j) \in \mathcal{L}^C$ -re a  $t^{(k,j)} + MT$  értékekkel számolunk.

#### 4.4.4. Az elemi hőáramok

Megadjuk az elemi hőmérsékleti intervallumok fogalmát.

##### *Elemi hőmérsékleti intervallumok*

Definiáljuk a következő sorozatot:  $t_1, t_2, \dots, t_{n_e}$ , ahol a sorozat elemei az  $(i, k, l) \in \mathcal{F}^H \cup \mathcal{F}^C$  hőáramhoz tartozó kezdő- és véghőmérsékletek, illetve  $(k, j) \in \mathcal{L}^C \cup \mathcal{L}^H$  a rejtett hőforráshoz tartozó hőmérsékleti értékek egyszeres előfordulással. Legyen a sorozat rendezve ( $i > j \Rightarrow t_i > t_j$ ). A hőintervallumokat a  $t_p$  sorozat szerint elemi intervallumokra bontjuk. Legyen  $E_p = [t_p, t_{p+1}]$ ,  $p \in [1, \dots, n_e]$  egy elemi intervallum.

Az elemi hőintervallumok alapján úgynevezett elemi hőáramokká ( $\mathcal{E}^H, \mathcal{E}^C$ ) bontjuk fel az áramokat:

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

#### 4.4.5. A részhőáramok

Az elemi hőintervallumok felhasználásával a hőáramokat felbontjuk részhőáramokra. Az  $(i, k, l) \in \mathcal{F}^C \cup \mathcal{F}^H$ ,  $[t_0^{(i,k,l)}, t_1^{(i,k,l)}]$  hőmérséklet intervallum felírható  $[t_p, t_{p+d}]$  alakban, ahol  $t_p = t_0^{(i,k,l)}$ ,  $t_{p+d} = t_1^{(i,k,l)}$ .  $[t_p, t_{p+d}]$  felbontható  $I_{qs} = [t_q, t_{s+1}]$  részintervallumokra, ahol  $p \leq q \leq s \leq p + d - 1$  (ez összesen  $\binom{d+1}{2}$  intervallum).

Az előbb vázolt intervallumfelbontásokat a hőáramokra használva részhőáramokat

kapunk:

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

## 4.5. A matematikai modell

Fejezetünkben a PNS modellt terjesztjük ki a HENS-hez kapcsolódó részekkel. Korábban már definiáltuk a PNS lineáris modelljét (2.2.3 fejezet). A bevezetett HENS modell független magától a PNS részhez kapcsolódó modelltől, így a megértés könnyítése céljából használjuk a PNS lineáris modelljét.

### 4.5.1. Anyagponthoz tartozó matematikai modell

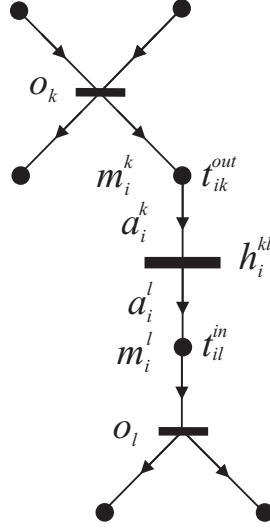
Fejezetünkben a kiterjesztett anyagpontokhoz tartozó feltételeket részletezzük. Legyen  $\mathcal{T}$  azon anyagok halmaza, melyre létezik specifikált hőmérséklet. Legyen  $m_i \in \mathcal{T}$ ,  $(o_k, m_i) \in \omega^-(m_i)$  és  $(m_i, o_l) \in \omega^+(m_i)$ . Jelölje  $t_{ik}^{in}$  az  $m_i$  anyag mint  $o_k$  műveleti egység kimeneti anyagának hőmérsékletét. Hasonlóan jelölje  $t_{il}^{out}$  az  $m_i$  anyag mint  $o_l$  műveleti egység bemeneti anyagának hőmérsékletét. Jelöljük  $h_i^{kl}$ -vel az  $o_k$  műveleti egységből  $o_l$  műveleti egységbe áramló  $m_i$  anyaghoz tartozó mesterséges műveleti egységet a rajta átáramló anyag mennyiségét pedig jelölje  $w_i^{kl}$ . Jelölje továbbá  $m_i^k$  az  $o_k$  műveleti egységhez tartozó új „mesterséges anyagpontot” (lásd 4.4 ábra).

A korábban adott  $m_i$  anyagponthoz kapcsolódó feltételt új feltételekkel fogjuk helyettesíteni. Jelölje  $\mathcal{K}_i$  az  $m_i$  anyagpont kiterjesztésével létrehozott új mesterséges műveleti egységek halmazát, és  $\mathcal{M}_i$  az új anyagpontok halmazát. A mesterséges műveleti egységek egy input, illetve output anyaggal rendelkeznek. Így két feltétel tartozik egy műveleti egységhez.

A mesterséges műveleti egységekre vonatkozó költség és feltételrendszer:

*Költség:*

$$f_i^{kl}() = 0, \quad h_i^{kl} \in \mathcal{K}_i, \quad i \in \mathcal{T}$$



4.4. ábra. Egy mesterséges műveleti egység.

*Feltételek:*

$$\begin{aligned} g_i^{kl_1}(x_i^k, w_i^{kl}) &= 0, & h_i^{kl} \in \mathcal{K}_i, i \in \mathcal{T}, \\ g_i^{kl_2}(x_i^k, w_i^{kl}) &= 0, & h_i^{kl} \in \mathcal{K}_i, i \in \mathcal{T}. \end{aligned}$$

Itt legyen  $a_i^k = (m_i^k, h_i^{kl}) \in \omega^-(h_i^{kl})$ ,  $a_i^l = (h_i^{kl}, m_i^l) \in \omega^+(h_i^{kl})$ , és jelölje  $x_i^k = \varphi(a_i^k)$ ,  $x_i^l = \varphi(a_i^l)$  az élekhez tartozó változókat.

$$\begin{aligned} g_i^{kl_1}(y_k, y_l, x_i^k, h_i^{kl}) &= w_i^{kl} - x_i^k, & h_i^{kl} \in \mathcal{K}_i, i \in \mathcal{T} \\ g_i^{kl_2}(y_k, y_l, x_i^k, h_i^{kl}) &= w_i^{kl} - x_i^l, & h_i^{kl} \in \mathcal{K}_i, i \in \mathcal{T} \end{aligned}$$

A mesterséges anyagpontokhoz tartozó költség és feltételrendszer:

$$f_i^k(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) = 0, \quad m_i^k \in \mathcal{M}_i,$$

$$\begin{aligned} g_i^k(\varphi(\omega^-(m_i^k)), \varphi(\omega^+(m_i^k))) &\leq 0, & m_i^k \in \mathcal{M}_i, o_k \in \text{prod}(m_i), \\ g_i^l(\varphi(\omega^-(m_i^l)), \varphi(\omega^+(m_i^l))) &\leq 0, & m_i^l \in \mathcal{M}_i, o_l \in \text{feed}(m_i). \end{aligned}$$

Legyen  $o_k \in \text{prod}(m_i)$  és  $a_s = (o_k, m_i^k) \in \omega^+(o_k)$ ,

$$g_i^k(\varphi(\omega^-(m_i^k)), \varphi(\omega^+(m_i^k))) = \sum_{o_l \in \text{feed}(m_i)} w_i^{kl} - x_s. \quad (4.5.1)$$

Hasonlóan  $o_l \in \text{feed}(m_i)$  és  $a_s = (m_i^l, o_l) \in \omega^+(o_k)$ ,

$$g_i'^l(\varphi(\omega^-(m_i^l)), \varphi(\omega^+(m_i^l))) = x_s - \sum_{o_k \in \text{prod}(m_i)} w_i^{kl}. \quad (4.5.2)$$

Felhasználva a műveleti egységekhez tartozó (2.2.5) feltételeket, a (4.5.1) és (4.5.2) formulák rendre a következő formában írhatók:

$$\begin{aligned} g_i'^k(\varphi(\omega^-(m_i^k)), \varphi(\omega^+(m_i^k))) &= \sum_{o_l \in \text{feed}(m_i)} w_i^{kl} - r_{ki} z_k, \\ g_i'^l(\varphi(\omega^-(m_i^l)), \varphi(\omega^+(m_i^l))) &= r_{li} z_l - \sum_{o_k \in \text{prod}(m_i)} w_i^{kl}. \end{aligned}$$

Modellünkben kiküszöböltük az élekhez tartozó  $\mathbf{x}$  változókat a  $\mathbf{z}$  és  $\mathbf{w}$  változók segítségével.

#### 4.5.2. Potenciális hőcserék meghatározása

Minden áramhoz definiáljuk az áramok egy részhalmazát, melyeknek képes hőt átadni.

A meleg részáramokhoz rendeljük a vele párosítható hideg részáramokat.

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

Minden elemi áramhoz, rejtett meleg és hideg forráshoz hozzárendeljük a megfelelő külső forrást.

$$\begin{aligned} JFU(FSC_j) &= \{u \in \mathcal{U}^H : UT_u \geq t_{p+1}\}, FSC_j = (i, k, l, p) \in \mathcal{E}^C \\ JFU(FSH_j) &= \{u \in \mathcal{U}^C : UT_u \leq t_p\}, FSH_j = (i, k, l, p) \in \mathcal{E}^H \\ JLU(LC_j) &= \{u \in \mathcal{U}^H : UT_u \geq T_{kj}\}, LC_j = (k, j) \in \mathcal{L}^C \\ JLU(LH_j) &= \{u \in \mathcal{U}^C : UT_u \leq T_{kj}\}, LH_j = (k, j) \in \mathcal{L}^H \end{aligned}$$

### A nem párosítható hőáramok

Lehetnek olyan áramok, melyek valami oknál fogva nem párosíthatók. A tiltott hőátadásokat kivesszük a megfelelő halmazokból.

### 4.5.3. Hőegyensúlyi feltételek

A hőegyensúlyi egyenleteket az elemi hőáramokra írjuk fel, a hőcserék definiálásakor a részhőáramokat, valamint a rejtett hideg és meleg forrásokat használjuk.

A hőegyensúlyi feltételek definiálásakor meg kell határoznunk az elvonandó illetve betáplálendő hő mennyiségét.

$$\begin{aligned} QFH_j &= c_i w_i^{kl} (t_{p+1} - t_p), FSH_j = (i, k, l, p) \in \mathcal{E}^H \\ QFC_j &= c_i w_i^{kl} (t_{p+1} - t_p), FSC_j = (i, k, l, p) \in \mathcal{E}^C \\ QLH_j &= h_{ki} z_k, LH_j = (k, i) \in \mathcal{L}^H \\ QLC_j &= h_{ki} z_k, LC_j = (k, i) \in \mathcal{L}^C \end{aligned}$$

### Hőcserékhez tartozó változók

A változók az egymáshoz rendelt hideg és meleg részhőáram vagy a rejtett hőáram közötti átvitt hőmennyiséget jelölik. A változókat a hozzárendelések típusainak megfelelően különböztetjük meg. A változóknak két fő indexük van: az első mindig a meleg részhőáram azonosítója, a második pedig a hozzárendelt hideg részhőáram azonosítója.

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

A külső hőforrásokat az elemi hőáramokhoz rendeljük. Az ide vonatkozó változók esetében is az egyik index a hőáramra vonatkozik, a másik a megfelelő külső forrásra.



Az indexek sorrendje a hőelvonás irányát mutatja.

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

A hőegyensúlyi feltételek:

$$\begin{aligned}
H(FSH_i) &= 0, FSH_i \in \mathcal{E}^H \\
H(FSC_i) &= 0, FSC_i \in \mathcal{E}^C \\
H(LH_i) &= 0, LH_i \in \mathcal{L}^H \\
H(LC_i) &= 0, LC_i \in \mathcal{L}^C
\end{aligned}$$

Részletezzük az egyenletek bal oldalát:

$$\begin{aligned}
H(FSH_i) &= QFH_i - \sum_{SSH_j=(i,k,l,q,s) \in \mathcal{I}^H, s \leq p \leq q} \frac{t_{p+1} - t_p}{t_{q+1} - t_s} \\
&\quad \left[ \sum_{SSC_{j'} \in JFF(SSH_j)} QFF_{jj'} + \sum_{LC_{j'} \in JFL(SSH_j)} QFL_{jj'} \right] - \\
&\quad \sum_{u \in JFU(FSH_i)} QFU_{iu}, \\
FSH_i &= (i, k, l, p) \in \mathcal{E}^H,
\end{aligned}$$

$$\begin{aligned}
H(FSC_{i'}) &= QFC_{i'} + \sum_{SSC_{j'}=(i',k',l',q',s') \in \mathcal{I}^C, s' \leq p' \leq q'} \frac{t_{p'+1} - t_{p'}}{t_{q'+1} - t_{s'}} \\
&\quad \left[ \sum_{SSC_{j'} \in JFF(SSH_j)} QFF_{jj'} + \sum_{SSC_{j'} \in JLF(LH_j)} QLF_{jj'} \right] + \\
&\quad \sum_{u \in JFU(FSC_{i'})} QUF_{ui'}, \\
FSC_{i'} &= (i', k', l', p') \in \mathcal{E}^C,
\end{aligned}$$

$$\begin{aligned}
H(LH_i) &= QLH_i - \sum_{SSC_{i'} \in JLF(LH_i)} QLF_{ii'} - \\
&\quad \sum_{LC_{i'} \in JLL(LH_i)} QLL_{ii'} - \sum_{u \in JLU(LH_i)} QLU_{iu}, \\
&\quad LH_i \in \mathcal{L}^H,
\end{aligned}$$

$$\begin{aligned}
H(LC_{i'}) &= QLC_{i'} + \sum_{LC_{i'} \in JFL(SSH_i)} QFL_{ii'} - \\
&\quad \sum_{LC_{i'} \in JLL(LH_i)} QLL_{ii'} + \sum_{u \in JLU(LC_{i'})} QUL_{ui'}, \\
&\quad LC_j \in \mathcal{L}^C.
\end{aligned}$$

#### 4.5.4. Hőcserélők költsége

A modellünkben a hőcserélő költség a felülettel arányos. Az arányossági tényezőket adjuk most meg a különböző típusú hőcserék esetén.

$$\begin{aligned}
CF_{F_{jj'}} &= A_{ii'} \frac{1}{U_{ii'} LMTD(t_s, t_{q+1}, t_{s'}, t_{q'+1})}, \\
SSH_j &= (i, k, l, q, s) \in \mathcal{I}^H, \\
SSC_{j'} &= (i', k', l', q', s') \in JFF(SSH_j),
\end{aligned}$$

ahol  $A_{ii'}$  az  $m_i$  és  $m_{i'}$  anyagok közötti egységnyi felületű hőcserélő költsége és  $U_{ii'}$  az anyagok közötti fajlagos hőátadási tényező ( $\frac{W}{m^2 \cdot K}$ ). Az  $LMTD$  jelöli a részhőáramok közötti logaritmikus középhőmérsékletet.

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

A rejtett források esetében a hőátvitel valamilyen köztes anyagon keresztül történik. Ilyenkor erre az anyagra vonatkoznak az anyagtól függő paraméterek. Jelölje most

ezt az anyagot  $m$ .

$$\begin{aligned}
CFL_{jj'} &= A_{im} \frac{1}{U_{im} LMTD(t_s, t_{q+1}, T_{k'i'}, T_{k'i'})}, \\
SSH_j &= (i, k, l, q, s) \in \mathcal{I}^H, LC_{j'} = (k', i') \\
CLF_{jj'} &= A_{mi'} \frac{1}{U_{mi'} LMTD(T_{ki}, T_{ki}, t_{s'}, t_{q'+1})} \\
LH_j &= (k, i) \in \mathcal{L}^H, SSC_{j'} = (i', k', l', q', s') \in JLF(LH_j), \\
CLL_{jj'} &= A_{mm} \frac{1}{U_{mm}} \frac{2}{(T_{ki} + T_{k'i'})} \\
LH_j &= (k, i) \in \mathcal{L}^H, LC_{j'} = (k', i') \in JLL(LH_j)
\end{aligned}$$

### Külső hő energia költsége

A külső meleg és hideg energia költsége az időegység alatt elvitt hő lineáris függvénye.  
 $u \in \mathcal{U}^H \cup \mathcal{U}^C$  -re a  $UC_u$  jelentse a lineáris költséghez tartozó együtthatót.

### 4.5.5. Egyesített matematikai modell

Összefoglalva az eddigieket a matematikai modell a következőképpen néz ki:

Célfüggvény:

$$\begin{aligned}
\min \quad & \sum_{m_i \in R} f'_i + \sum_{o_j \in O} f_j + \\
& \sum_{QFF_{jj'}} CFF_{jj'} QFF_{jj'} + \sum_{QFL_{jj'}} CFL_{jj'} QFL_{jj'} + \\
& \sum_{QLF_{jj'}} CLF_{jj'} QLF_{jj'} + \sum_{QFU_{iu}} UC_u QFU_{iu} + \\
& \sum_{QLU_{iu}} UC_u QLU_{iu}
\end{aligned} \tag{4.5.3}$$

*Feltételrendszer:*

$$\begin{aligned}
g'_i(\varphi(\omega^-(m_i)), \varphi(\omega^+(m_i))) &\leq 0, \quad m_i \in \mathcal{M} \setminus \mathcal{T} \\
g'_i{}^k(\varphi(\omega^-(m_i^k)), \varphi(\omega^+(m_i^k))) &\leq 0, \quad m_i^k \in \mathcal{M}_i, o_k \in \text{prod}(m_i) \quad i \in \mathcal{T} \\
g'_i{}^l(\varphi(\omega^-(m_i^l)), \varphi(\omega^+(m_i^l))) &\leq 0, \quad m_i^l \in \mathcal{M}_i, o_l \in \text{feed}(m_i) \quad i \in \mathcal{T} \\
H(FSH_i) &= 0, \quad FSH_i \in \mathcal{E}^H \\
H(FSC_i) &= 0, \quad FSC_i \in \mathcal{E}^C \\
H(LH_i) &= 0, \quad LH_i \in \mathcal{L}^H \\
H(LC_i) &= 0, \quad LC_i \in \mathcal{L}^C
\end{aligned} \tag{4.5.4}$$

## 4.6. Az integrált módszer leírása

A módszer az ABB algoritmus [40] módosítottása. 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 ( $y_k = 1$ ) és kizárásáról ( $y_k = 0$ ). A modell felírásakor a fixen kizárt ( $y_k = 0$ ) műveleti egységek jelentenek változásokat, az ahhoz kapcsolódó hőáramok, rejtett meleg és hideg források nem kerülnek bele a modellbe.

### 4.6.1. A korlátozó LP feladat tulajdonsága

A korlátozás lépés során egy LP feladatot kell megoldanunk. A feltételrendszer az anyagokra és az elemi hőáramokra vonatkoznak, a változók a műveleti egységekhez és a hőcserélőkhöz vannak hozzárendelve. A potenciális hőcserélők száma az egymáshoz rendelhető részhőáramok és rejtett források kombinációitól függ, ezért már közepes feladatok esetében is nagyszámú potenciális hőcserélőt kapunk. Ez több nagyságrendben is eltérhet maguktól az elemi áramok számától. Ez azt jelenti, hogy az LP feladat oszlopainak a száma jóval nagyobb a sorok számánál. Gyakorlati példánál nagyméretű LP feladatok merülhetnek fel, amelyek kevés számú feltételt

és nagyszámú változót tartalmaznak. A SIMPLEX módszer ilyen típusú LP feladatokra igen hatékony. Az optimális megoldásban az összes hőcserélő számához képest elenyésző az optimális megoldáshoz tartozó hőátvitel szám, így a megoldásban szereplő nemzéró változók száma is jóval kevesebb mint a modellhez tartozó változósám.

## 4.7. Szemléltető példa

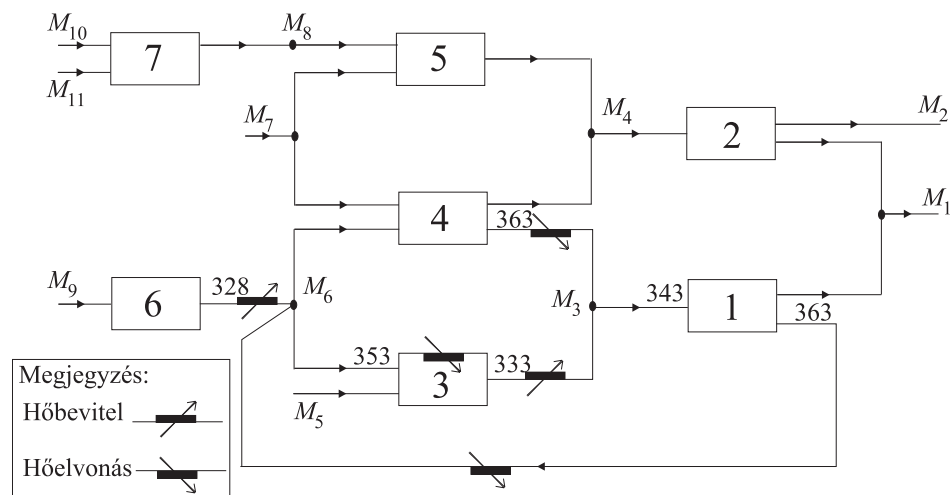
Egy szemléltető példa alapján fogjuk bemutatni algoritmusunk működését.

### 4.7.1. Általános leírás

A feladat maximális struktúrájának reprezentációját a 4.5 ábra mutatja. A téglalapok a műveleti egységeket, az irányított élek az anyagáramokat ábrázolják. Az esetlegesen szükséges hőigényeket is feltüntettük.

Célunk egy olyan költségoptimalis részstruktúra megtalálása, amely kielégíti a feltételeket. A feltételek lehetnek: termékekre megfogalmazott korlát, hőigények kielégítése, anyagegyensúly feltételek.

A termék az  $M_1$  anyag, melyből 100 t/év mennyiséget kell gyártani. A műveleti



4.5. ábra. Folyamatábra a szemléltető feladathoz.

4.1. táblázat. A lehetséges műveleti egységek

| Műveleti<br>egység | Rejtett hő |            | Bemeneti                   | Kimeneti              |
|--------------------|------------|------------|----------------------------|-----------------------|
|                    | Hőm. (K)   | Forrás eü. | áramok                     | áramok                |
| $o_1$              | —          | —          | $M_3(3, 343)$              | $M_1(2), M_6(1, 363)$ |
| $o_2$              | —          | —          | $M_4(1.5)$                 | $M_1(1), M_2(0.5)$    |
| $o_3$              | 353        | 20         | $M_5(1), M_6(1, 353)$      | $M_3(2, 333)$         |
| $o_4$              | —          | —          | $M_6(0.3), M_7(1.7)$       | $M_3(1, 363), M_4(1)$ |
| $o_5$              | —          | —          | $M_7(2), M_8(1)$           | $M_4(3)$              |
| $o_6$              | —          | —          | $M_9(1)$                   | $M_6(1, 328)$         |
| $o_7$              | —          | —          | $M_{10}(1.2), M_{11}(0.8)$ | $M_8(2)$              |

egység modelljét leíró paramétereket a 4.1 táblázat tartalmazza, a kapcsolódó költségparaméterek a 4.2 táblázatban találhatók. A műveleti egységekkel kapcsolatosan felmerülő rejtett hő mennyiségét a forrás együttható és a méret szorzata adja meg (lásd a 4.1 táblázat megfelelő oszlopai). Egy műveleti egység egy bemeneti vagy kimeneti anyagáramának paraméterét az anyagnév utáni zárójelbe tett mennyiség jellemzi, a második érték a hőmérsékletre vonatkozik. A költségparamétereket a 4.2 táblázat tartalmazza. A példánkban a megtérülési évek száma 5.

4.2. táblázat. Költségparaméterek a műveleti egységekre

| Műveleti<br>egység | Beruházási költség |         | Működési költség |         |
|--------------------|--------------------|---------|------------------|---------|
|                    | Állandó            | Változó | Állandó          | Változó |
| $o_1$              | 7500               | 1200    | 500              | 160     |
| $o_2$              | 3800               | 1000    | 140              | 250     |
| $o_3$              | 8000               | 1000    | 400              | 170     |
| $o_4$              | 15000              | 1500    | 500              | 100     |
| $o_5$              | 10000              | 1500    | 900              | 300     |
| $o_6$              | 3000               | 750     | 200              | 100     |
| $o_7$              | 5000               | 800     | 700              | 160     |

4.3. táblázat. Nyersanyagok

| Név      | Költség [USD/t] | Limit [t/év] |
|----------|-----------------|--------------|
| $M_5$    | 140             | Nincs limit  |
| $M_7$    | 200             | Nincs limit  |
| $M_9$    | 250             | Nincs limit  |
| $M_{10}$ | 50              | Nincs limit  |
| $M_{11}$ | 70              | Nincs limit  |

4.4. táblázat. Külső hideg, meleg források

| Külső forrás | Típus | Hőmérséklet (K) | Költség (USD/MJ) |
|--------------|-------|-----------------|------------------|
| $H_1$        | Meleg | 373.0           | 2.0              |
| $C_1$        | Hideg | 293.0           | 3.0              |

A nyersanyagok felsorolását és a hozzá megfelelő értékek a 4.3 táblázat tartalmazza.

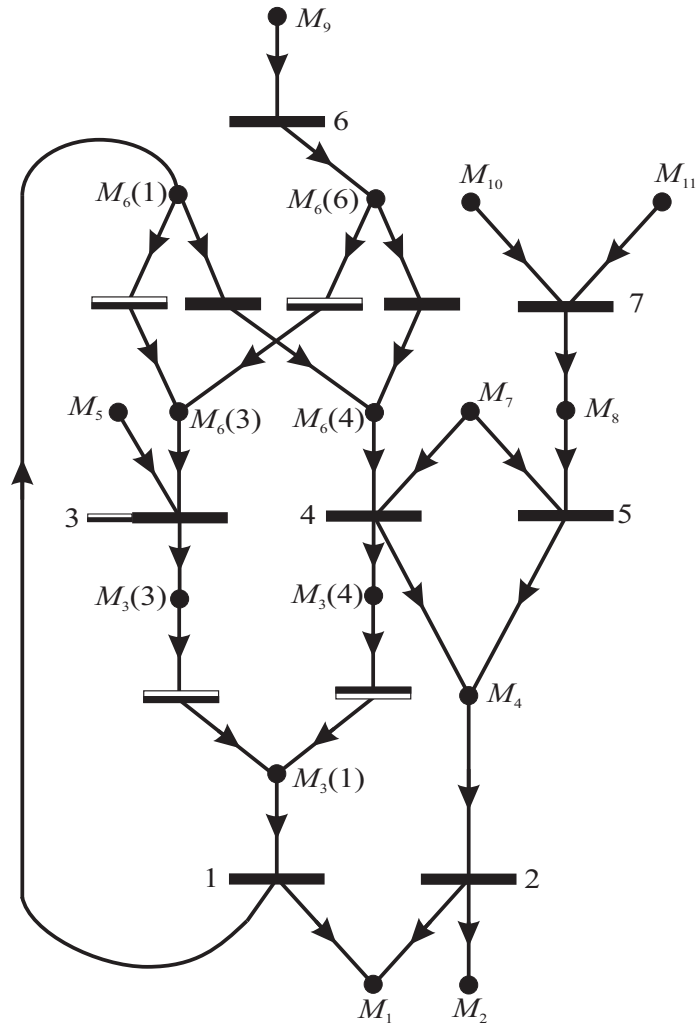
A felhasználható külső hideg és meleg forrásokat 4.4 tábla tartalmazza a rendelkezésre álló hőmérséklet és a költségadatokkal együtt. A hőcserélő költségét anyagpáronként lehet definiálni, a példánkban minden anyagpárra azonos költséget adunk meg: 5.0 USD/m<sup>2</sup>; hasonlóan a hőátadási tényező is egy anyagpárra vonatkozik, most itt minden párra 1.0 MJ/(h K m<sup>2</sup>).

## hP-Gráf

A feladat hP-gráf reprezentációját mutatja a 4.6 ábra, ahol már a lehetséges hőcseréket is feltüntettük.

## ABB algoritmus

A feladatot a 2.3 fejezetben említett ABB módszerrel lett megoldva, az algoritmus által bejárt BB keresőfa a 4.7 ábrán látható. A BB fa minden pontjához tartozik a műveleti egységek egy osztályozása: kizárt, beválasztott, nem döntött



4.6. ábra. A szemléltető példa hP-gráfja.



4.5. táblázat. Műveleti egység osztályok

| Csúcs   | Műveleti egység |              |           |
|---------|-----------------|--------------|-----------|
|         | Nem döntött     | Beválasztott | Kizárt    |
| 1       | 1,2,3,4,5,6     | —            | —         |
| 1.1     | 3,4,6           | 1            | 2,5,7     |
| 1.1.1   | 6               | 1,3          | 2,4,5,7   |
| 1.1.1.1 | —               | 1,3          | 2,4,5,6,7 |
| 1.1.1.2 | —               | 1,3,6        | 2,4,5,7   |
| 1.1.2   | 6               | 1,4          | 2,3,5,7   |
| 1.1.3   | 6               | 1,3,4        | 2,5,7     |
| 1.2     | 4,5,6,7         | 2            | 1,3       |
| 1.3     | 3,4,5,6,7       | 1,2          | —         |
| 1.3.1   | 6               | 1,2,3,5,7    | 4         |
| 1.3.2   | 5,6,7           | 1,2,4        | 3         |
| 1.3.3   | 5,6,7           | 1,2,3,4      | —         |

műveleti egységek. A 4.5 táblázat tartalmazza a BB fában a csúcsokhoz tartozó osztályozásokat.

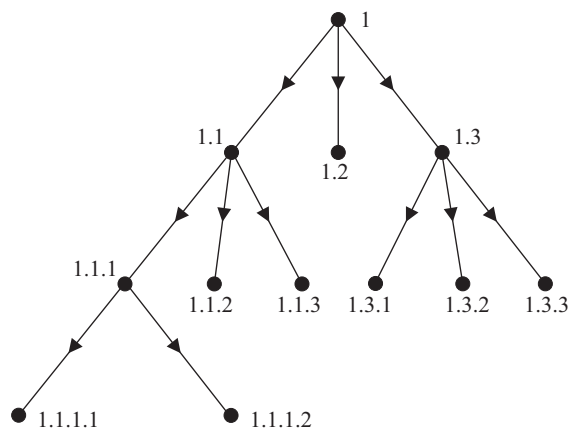
Továbbiakban két csúcsra, 1 (gyökerpont) és 1.1.1.2 (levélpont), részletezzük a modell felírását.

#### 4.7.2. Az 1. csúcs

Az 1. csúcs a gyöker csúcsot reprezentálja, még nem történt döntés, így minden műveleti egység a nem döntött osztályban található. Potenciálisan két meleg és két hideg áram van, ezeket a 4.6 táblázat tartalmazza, továbbá rejtett hő a 3. műveleti egységhez tartozik, ennek a paramétereit a 4.7 táblázat tartalmazza.

A meleg és hideg áramokat kaszkád diagrammal ábrázolhatjuk (lásd 4.8 ábra), ahol a hideg áramokat a minimális megközelítési távolsággal (10 K) már eltoltuk. A diagramban az  $I_1, I_2, \dots, I_5$  jelöli a hőmérsékleti intervallumokat.

A hőmérsékleti intervallumok ( $I_1, I_2, \dots, I_5$ ) a meleg és hideg áramokat elemi hő-áramokká partícionálják (lásd a 4.8 táblázat).



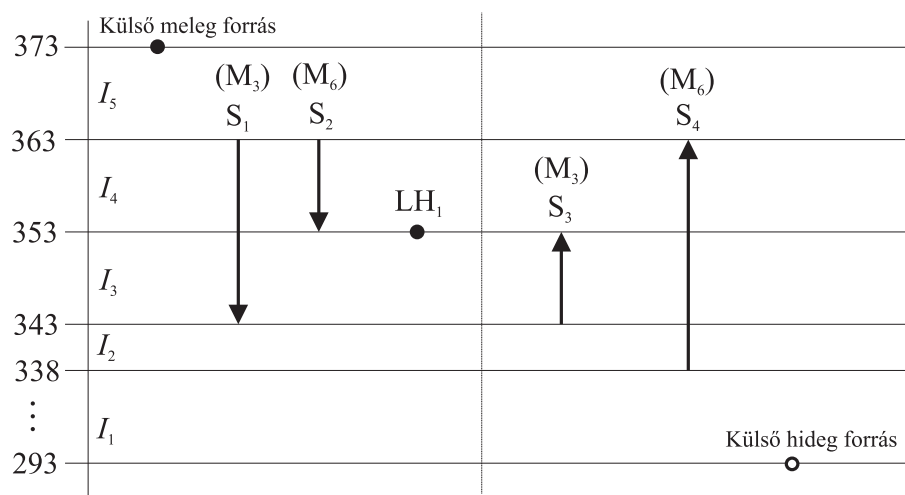
4.7. ábra. Az ABB algoritmus által előállított leszámlálási fa (a legrosszabb eset).

4.6. táblázat. Lehetséges hőáramok az 1. csúcsnál

| Áram  | Típus | Anyag | Kezdő hőm. (K) | Vég hőm. (K) |
|-------|-------|-------|----------------|--------------|
| $S_1$ | meleg | $M_3$ | 363.0          | 343.0        |
| $S_2$ | meleg | $M_6$ | 363.0          | 353.0        |
| $S_3$ | hideg | $M_3$ | 333.0          | 343.0        |
| $S_4$ | hideg | $M_6$ | 328.0          | 353.0        |

4.7. táblázat. Rejtett hőforrások az 1. csúcsnál

| Áram   | Típus | Műveleti egység | Hőmérséklet (K) |
|--------|-------|-----------------|-----------------|
| $LH_1$ | meleg | 3               | 353.0           |



4.8. ábra. Kaskád diagram a jellemző hőáramokról az 1. csúcsban.

| 4.8. táblázat. Lehetséges elemi hőáramok az 1. csúcsban |       |       |                |              |
|---------------------------------------------------------|-------|-------|----------------|--------------|
| Elemi hőáramok                                          | Típus | Anyag | Kezdő hőm. (K) | Vég hőm. (K) |
| $FSH_1$                                                 | meleg | $M_3$ | 363.0          | 353.0        |
| $FSH_2$                                                 | meleg | $M_3$ | 353.0          | 343.0        |
| $FSH_3$                                                 | meleg | $M_6$ | 363.0          | 353.0        |
| $FSC_1$                                                 | hideg | $M_3$ | 343.0          | 353.0        |
| $FSC_2$                                                 | hideg | $M_6$ | 338.0          | 343.0        |
| $FSC_3$                                                 | hideg | $M_6$ | 343.0          | 353.0        |
| $FSC_4$                                                 | hideg | $M_6$ | 353.0          | 363.0        |

Részhőáramokat az elemi hőáramok folytonos intervallumot alkotó kombinációi határozzák meg. A részhőáramok listája a 4.9 táblázatban látható.

Egy hőcserélő egy (hideg-meleg) részhőáram párral adható meg, a  $JSS(SSH_i)$  ( $i = 1, 2, 3, 4$ ) halmazok tartalmazzák a  $SSH_i$ -vel potenciálisan párosítható részhőáramokat, melyeket az alábbiakban részletezzük:

$$\begin{aligned} JSS(SSH_1) &= \{SSC_1, SSC_2, SSC_3, SSC_5\}, \\ JSS(SSH_2) &= \{SSC_1, SSC_2, SSC_3, SSC_4, SSC_5, SSC_6, SSC_7\}, \\ JSS(SSH_3) &= \{SSC_1, SSC_2, SSC_3, SSC_5, SSC_6, SSC_7\}, \\ JSS(SSH_4) &= \{SSC_1, SSC_2, SSC_3, SSC_4, SSC_5, SSC_6, SSC_7\}. \end{aligned}$$

A rejtett hő úgy kezelhető, mint egy részhőáram amelynek a kezdő- és vég hőmérséklete ugyanaz:

$$JLS(LH_1) = \{SSC_1, SSC_2, SSC_3, SSC_5\}$$

A külső meleg, hideg forrásokat minden elemi hőáramhoz hozzá kell rendelnünk:

$$\begin{aligned} JSU(FSH_1) &= \{C_1\}, JSU(FSH_2) = \{C_1\}, JSU(FSH_3) = \{C_1\}, \\ JSU(FSH_4) &= \{C_1\}, JSU(FSC_1) = \{H_1\}, JSU(FSC_2) = \{H_1\}, \\ JSU(FSC_3) &= \{H_1\}. \end{aligned}$$

Itt  $C_1$  a külső hideg forrás és  $H_1$  a külső meleg forrás. Hasonlóan a rejtett hőre:

$$JLU(LH_1) = \{C_1\}.$$

Az anyagok hőkapacitásait a 4.10 táblázat tartalmazza.

Egy elemi hőáram számára betáplálendő illetve elvonandó hő mennyiségét a hőkapacitás az áram és a hőmérsékleti intervallum szorzata adja ( $QFC_i$ ,  $i = 1, 2, \dots, 7$ ). A rejtett hő mennyisége a kapcsolódó műveleti egység méretének és rejtett hő paraméterének szorzata ( $QLH_1$ ). Meleg hőáramokra  $QFH_i > 0$ , hideg hőáramokra  $QFC_i < 0$  teljesül ez természetesen a rejtett hőre is igaz.

A hőátvitelhez kapcsolódó változókat a megfelelő részáramok vagy rejtett hőáramok indexeivel azonosítjuk, az első index a meleg áramra, a második index a hideg áramra vonatkozik (lásd a 4.11 táblázatot).

Rejtett és elemi hőáramokhoz rendeljük a külső meleg és hideg energiaforrásokat, a külső energiához kapcsolódó változókat a 4.12 tábla tartalmazza.

A hőcserék részhőáramok között mennek végbe, viszont a hőegyensúlyi feltételeket

4.9. táblázat. Lehetséges részhoáramok az 1. csúcsban

| Részhoáramok | Típus | Anyag | Intervallum     | Kezdő hőm. (K) | Vég hőm.(K) |
|--------------|-------|-------|-----------------|----------------|-------------|
| $SSH_1$      | meleg | $M_3$ | $I_3$           | 353.0          | 343.0       |
| $SSH_2$      | meleg | $M_3$ | $I_4$           | 363.0          | 353.0       |
| $SSH_3$      | meleg | $M_3$ | $I_3, I_4$      | 363.0          | 343.0       |
| $SSH_4$      | meleg | $M_6$ | $I_4$           | 363.0          | 353.0       |
| $SSC_1$      | hideg | $M_3$ | $I_3$           | 338.0          | 353.0       |
| $SSC_2$      | hideg | $M_6$ | $I_2$           | 343.0          | 343.0       |
| $SSC_3$      | hideg | $M_6$ | $I_3$           | 338.0          | 353.0       |
| $SSC_4$      | hideg | $M_6$ | $I_4$           | 343.0          | 363.0       |
| $SSC_5$      | hideg | $M_6$ | $I_2, I_3$      | 338.0          | 353.0       |
| $SSC_6$      | hideg | $M_6$ | $I_3, I_4$      | 343.0          | 363.0       |
| $SSC_7$      | hideg | $M_6$ | $I_2, I_3, I_4$ | 338.0          | 363.0       |

4.10. táblázat. Anyagok hőkapacitásai

| Anyagnév | érték |
|----------|-------|
| $M_3$    | 0.4   |
| $M_4$    | 1.0   |
| $M_6$    | 1.0   |

4.11. táblázat. Hőcseréhez kapcsolódó változók az 1. csúcsban

|         | $SSC_1$    | $SSC_2$    | $SSC_3$    | $SSC_4$    | $SSC_5$    | $SSC_6$    | $SSC_7$  |
|---------|------------|------------|------------|------------|------------|------------|----------|
| $SSH_1$ | $QFF_{11}$ | $QFF_{12}$ | $QFF_{13}$ | —          | $QFF_{15}$ | —          | —        |
| $SSH_2$ | $QFF_{21}$ | $QFF_{22}$ | $QFF_{23}$ | $QFF_{24}$ | $QFF_{25}$ | $QFF_{26}$ | $Q_{27}$ |
| $SSH_3$ | $QFF_{31}$ | $QFF_{32}$ | $QFF_{33}$ | —          | $QFF_{35}$ | $QFF_{36}$ | $Q_{37}$ |
| $SSH_4$ | $QFF_{41}$ | $QFF_{42}$ | $QFF_{43}$ | $QFF_{44}$ | $QFF_{45}$ | $QFF_{46}$ | $Q_{47}$ |
| $LH_1$  | $QLF_{11}$ | $QLF_{12}$ | $QLF_{13}$ | —          | $QLF_{15}$ | —          | —        |

az elemi hőáramokra számítjuk, ezért fontos a részhőáramból elvont hő elemi hőáramra eső részének a meghatározása (lásd 4.9 ábra). A hőáram egy téglalapnak tekinthető, a horizontális mérete a fajhő és az anyagáram szorzata, a vertikális hossz pedig a kezdő- és véghőmérséklet különbsége. A két oldal szorzata adja meg az időegység alatt elvonandó illetve betáplálendő hőmennyiséget.

Tekintsük az  $FSH_k$  ( $EFGH$  téglalap) elemi hőáramot a hozzá kapcsolódó hőmérsékleti intervallummal  $[T_2, T_3]$ . Jelölje a hozzá tartozó téglalap területét  $QF_k$ . Legyen  $SSH_i$  ( $ABCD$  téglalap) a részhőáram és  $SSC_j$  a hőcserében résztvevő másik részhőáram. A  $[T_1, T_4]$  az  $SSH_i$ -hoz tartozó hőmérsékleti intervallum és legyen az átvitt hő mennyisége  $QFF_{ij}$ . Az  $EFGH$  és  $ABCD$  téglalapok metszete jelzi az  $FSH_k$  -ről ténylegesen elvitt hőmennyiséget, mely a következőképpen számolható:

$$\frac{T_3 - T_2}{T_4 - T_1} QFF_{ij} \quad (4.7.1)$$

A megfelelő hőintervallumok aránya határozza meg, hogy a  $QFF_{ij}$  mennyiségű elvitt hő mekkora hányada származik a kérdéses elemi részhőáramról.

Tekintsük az  $FSH_2$  elemi hőáramot. Az  $FSH_2$ -re vonatkozó hőegyensúlyi feltétel meghatározásában a rá illeszkedő részhőáramokat kell figyelembe vennünk, ezek most az  $SSH_1$  és az  $SSH_3$ . Az  $SSH_1$  részhőáram lehetséges párosításait a 4.10 ábrán mutatjuk be. Hasonlóan a  $SSH_3$  részhőáram lehetséges párosításait a 4.11 ábra mutatja. Az  $FSH_2$ -re vonatkozó hőegyensúlyi feltétel a következőképpen néz ki:

$$\begin{aligned} 0 = & QFH_2 - QFF_{11} - 0.5 QFF_{31} - QFF_{12} - 0.5 QFF_{32} - QFF_{13} \\ & - 0.5 QFF_{33} - QFF_{15} - 0.5 QFF_{35} - 0.5 QFF_{36} - 0.5 QFF_{37} - QFU_{21}. \end{aligned}$$

Hasonlóan minden elemi hőáramra és rejtett hőre felírható a hőegyensúlyi feltétel, ezek együtthatóit a 4.13 táblázat tartalmazza.

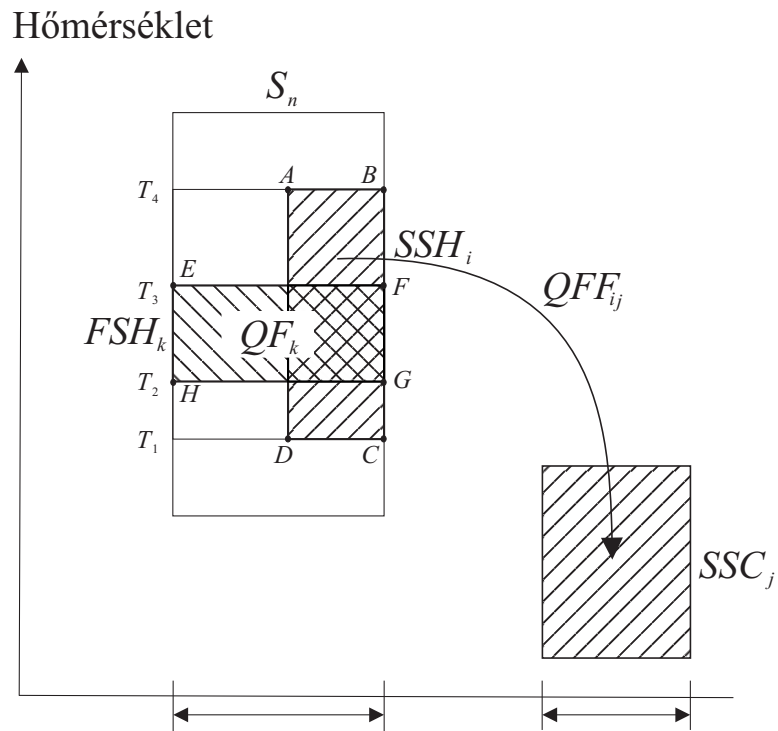
A hőcserélő költsége a felülettel arányos, ennek a kiszámításához tekintsünk egy hőcserélőt, amely az  $SSH_i$  és  $SSC_j$  részhőáramok között van. A hőcseréhez tartozó költség a következő:

$$c_{ij} QFF_{ij} = a_{ij} \frac{1}{U_{ij} LMTD_{ij}} QFF_{ij},$$

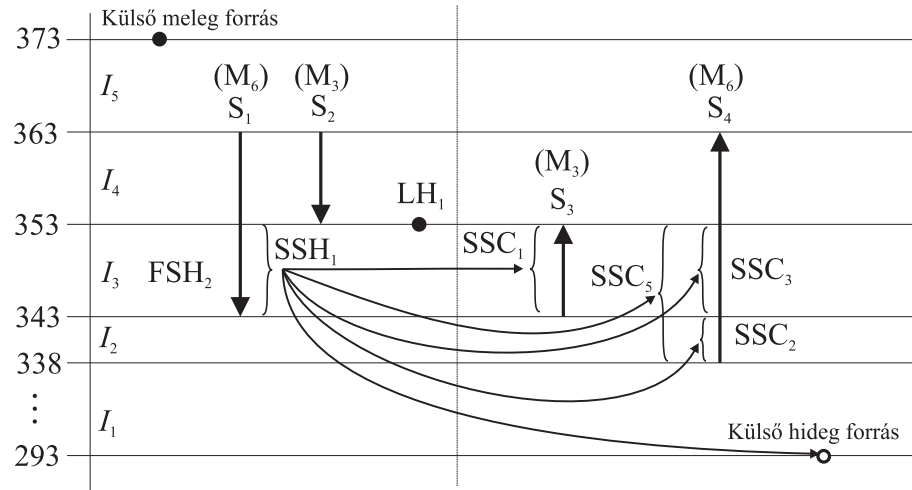
ahol  $QFF_{ij}$  az átvitt hőmennyiség. A 4.14 táblázat tartalmazza ezen  $c_{ij}$  együtthatókat.

4.12. táblázat. A külső hideg és meleg energiához kapcsolódó változók a 1. csúcsban

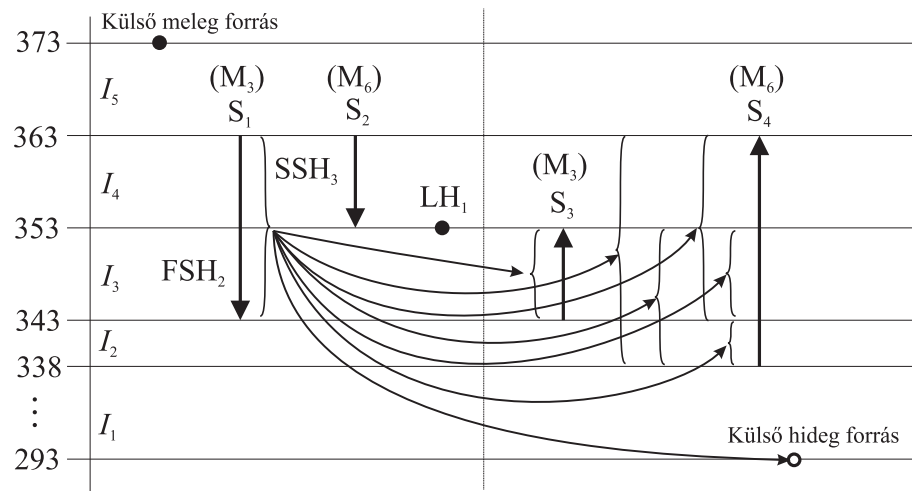
|       | $FSH_1$    | $FSH_2$    | $FSH_3$    | $FSC_1$    | $FSC_2$    | $FSC_3$    | $FSC_4$    | $LH_1$     |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|
| $C_1$ | $QFU_{11}$ | $QFU_{21}$ | $QFU_{31}$ | —          | —          | —          | —          | $QLU_{11}$ |
| $H_1$ | —          | —          | —          | $QUF_{41}$ | $QUF_{51}$ | $QUF_{61}$ | $QUF_{71}$ | —          |



4.9. ábra. Hőegyensúly az elemi áramokra.



4.10. ábra. Az  $SSH_1$  részáram lehetséges párosításai az 1. csúcsban.



4.11. ábra. Az  $SSH_3$  részáram lehetséges párosításai az 1. csúcsban.



4.13. táblázat. Hőegyensúlyi feltételek együtthatói az 1. csúcsban

| Változók   | $QFH_1$ | $QFH_2$ | $QFH_3$ | $QFC_1$ | $QFC_2$ | $QFC_3$ | $QFC_4$ | $QLH_1$ |
|------------|---------|---------|---------|---------|---------|---------|---------|---------|
| $QFF_{11}$ |         | -1      |         | 1       |         |         |         |         |
| $QFF_{21}$ | -1      |         |         | 1       |         |         |         |         |
| $QFF_{31}$ | -1/2    | -1/2    |         | 1       |         |         |         |         |
| $QFF_{41}$ |         |         | -1      | 1       |         |         |         |         |
| $QFF_{12}$ |         | -1      |         |         | 1       |         |         |         |
| $QFF_{22}$ | -1      |         |         |         | 1       |         |         |         |
| $QFF_{32}$ | -1/2    | -1/2    |         |         | 1       |         |         |         |
| $QFF_{42}$ |         |         | -1      |         | 1       |         |         |         |
| $QFF_{13}$ |         | -1      |         |         |         | 1       |         |         |
| $QFF_{23}$ | -1      |         |         |         |         | 1       |         |         |
| $QFF_{33}$ | -1/2    | -1/2    |         |         |         | 1       |         |         |
| $QFF_{43}$ |         |         | -1      |         |         | 1       |         |         |
| $QFF_{24}$ | -1      |         |         |         |         |         | 1       |         |
| $QFF_{44}$ |         |         | -1      |         |         |         | 1       |         |
| $QFF_{15}$ |         | -1      |         |         | 1/3     | 2/3     |         |         |
| $QFF_{25}$ | -1      |         |         |         | 1/3     | 2/3     |         |         |
| $QFF_{35}$ | -1/2    | -1/2    |         |         | 1/3     | 2/3     |         |         |
| $QFF_{45}$ |         |         | -1      |         | 1/3     | 2/3     |         |         |
| $QFF_{26}$ | -1      |         |         |         |         | 1/2     | 1/2     |         |
| $QFF_{36}$ | -1/2    | -1/2    |         |         |         | 1/2     | 1/2     |         |
| $QFF_{46}$ |         |         | -1      |         |         | 1/2     | 1/2     |         |
| $QFF_{27}$ | -1      |         |         |         | 1/5     | 2/5     | 2/5     |         |
| $QFF_{37}$ | -1/2    | -1/2    |         |         | 1/5     | 2/5     | 2/5     |         |
| $QFF_{47}$ |         |         | -1      |         | 1/5     | 2/5     | 2/5     |         |
| $QLF_{11}$ |         |         |         | 1       |         |         |         | -1      |
| $QLF_{12}$ |         |         |         |         | 1       |         |         | -1      |
| $QLF_{13}$ |         |         |         |         |         | 1       |         | -1      |
| $QLF_{15}$ |         |         |         |         | 1/3     | 2/3     |         | -1      |
| $QFU_{11}$ | -1      |         |         |         |         |         |         |         |
| $QFU_{21}$ |         | -1      |         |         |         |         |         |         |
| $QFU_{31}$ |         |         | -1      |         |         |         |         |         |
| $QUF_{11}$ |         |         |         | 1       |         |         |         |         |
| $QUF_{21}$ |         |         |         |         | 1       |         |         |         |
| $QUF_{31}$ |         |         |         |         |         | 1       |         |         |
| $QUF_{41}$ |         |         |         |         |         |         | 1       |         |
| $QLU_{11}$ |         |         |         |         |         |         |         | -1      |

A felírt matematikai programozási modell lineáris, a megoldás egy alsó költségkorlátot ad a részproblémára nézve.

#### **4.7.3. Az 1.1.1.2 csúcs**

Az aktuális csúcs egy levélpont a BB fában. Az 1, 3 és 6 műveleti egységek a beválasztott halmazban vannak, a többi műveleti egység pedig a kizártakat tartalmazó halmazban (a nem döntött műveleti egységek halmaza üres). A potenciális meleg és hideg áramokat a 4.15 táblázat, a rejtett hőket a 4.16 táblázat tartalmazza. A 4.13 ábrán látható a részfeladathoz kapcsolódó kaszkád diagram.

A 4.17 táblázat tartalmazza a meleg és hideg elemi hőáramok listáját, a 4.18 tábla pedig a részhőáramokat tartalmazza.

Megoldva az LP feladatot, 51534 USD lett az éves költség. Egy levélpontban voltunk, így ez valódi költséget jelent. Az ABB algoritmus miután bejárta a BB fa egy részét, azonosítja az optimális megoldást, amely az 1.1.1.2 csúcsban megtalált struktúra (4.12 ábra). Az optimális struktúra tartalmazza az 1, 3, és 6 műveleti egységeket; méretük rendre 50, 75, and 25. Az hőcseréket az 4.14 ábra mutatja.

4.14. táblázat. A költségfüggvényhez tartozó paraméterek az 1. csúcsban

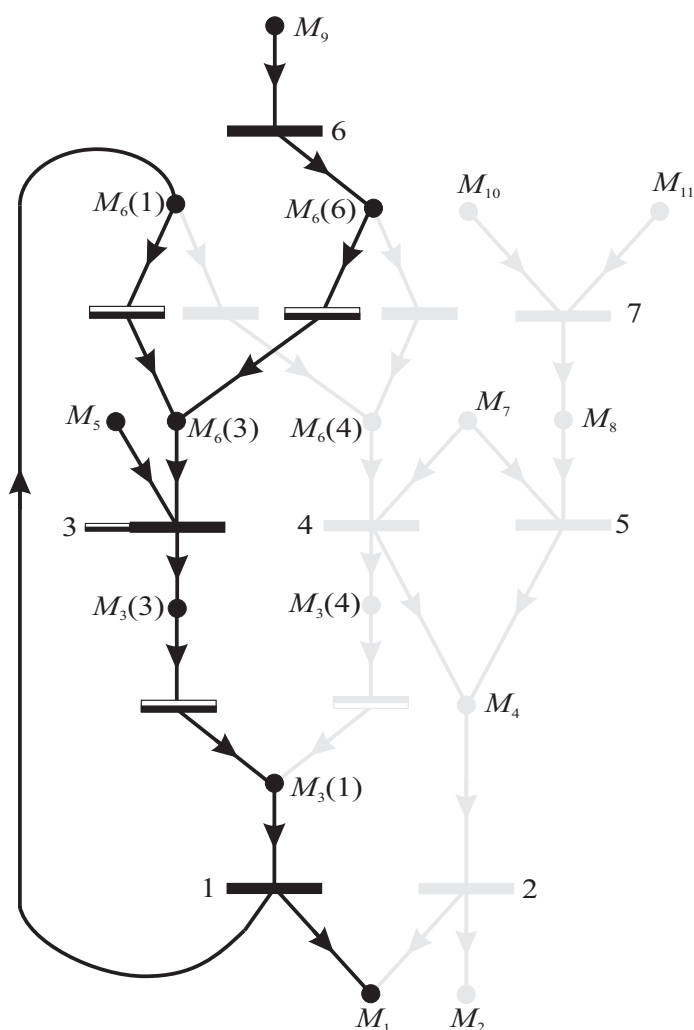
| Változó  | Együttható $c_{ij}$ | Változó    | Együttható $c_{ij}$ |
|----------|---------------------|------------|---------------------|
| $Q_{11}$ | 0.5000              | $QFF_{15}$ | 0.4055              |
| $Q_{21}$ | 0.2500              | $QFF_{25}$ | 0.2231              |
| $Q_{31}$ | 0.3465              | $QFF_{35}$ | 0.2877              |
| $Q_{41}$ | 0.2500              | $QFF_{45}$ | 0.2231              |
| $Q_{12}$ | 0.2876              | $QFF_{26}$ | 0.3466              |
| $Q_{22}$ | 0.1823              | $QFF_{36}$ | 0.5000              |
| $Q_{32}$ | 0.2310              | $QFF_{46}$ | 0.3466              |
| $Q_{42}$ | 0.1823              | $QFF_{27}$ | 0.3054              |
| $Q_{13}$ | 0.5000              | $QFF_{37}$ | 0.4055              |
| $Q_{23}$ | 0.2500              | $QFF_{47}$ | 0.3054              |
| $Q_{33}$ | 0.3466              | $QLF_{11}$ | 0.3466              |
| $Q_{43}$ | 0.2500              | $QLF_{12}$ | 0.2231              |
| $Q_{24}$ | 0.5000              | $QLF_{13}$ | 0.3466              |
| $Q_{44}$ | 0.5000              | $QLF_{15}$ | 0.3054              |

4.15. táblázat. Potenciális hideg és meleg áramok az 1.1.1.2. csúcsban

| Áram  | Típus | Anyag | Kezdő hőm. (K) | Vég hőm. (K) |
|-------|-------|-------|----------------|--------------|
| $S_1$ | meleg | $M_6$ | 363.0          | 353.0        |
| $S_2$ | hideg | $M_3$ | 363.0          | 343.0        |
| $S_3$ | hideg | $M_6$ | 328.0          | 353.0        |

4.16. táblázat. Rejtett hő előfordulása az 1.1.1.2. csúcsban

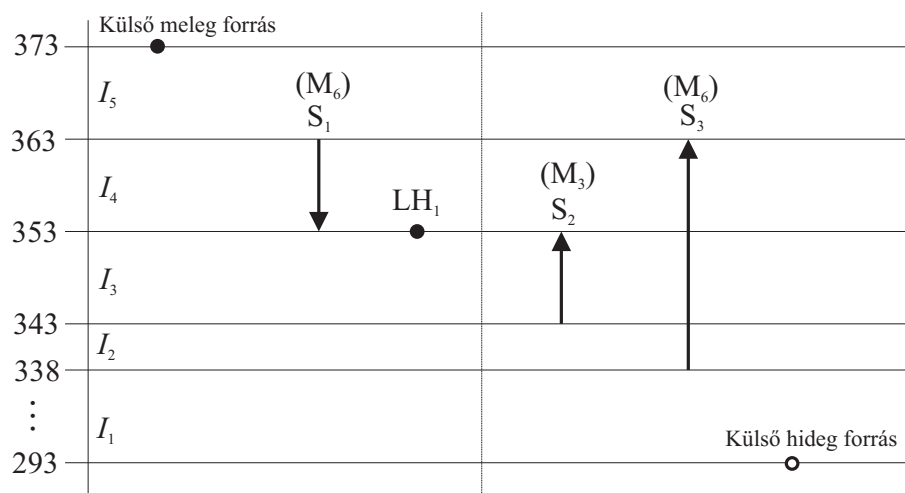
| Áram   | Típus | Művelet egység | Hőm. (K) |
|--------|-------|----------------|----------|
| $LH_1$ | meleg | 3              | 363.0    |



4.12. ábra. Megoldás struktúra az 1.1.1.2 csúcsban.

4.17. táblázat. Meleg és hideg elemi hőáramok az 1.1.1.2. csúcsban

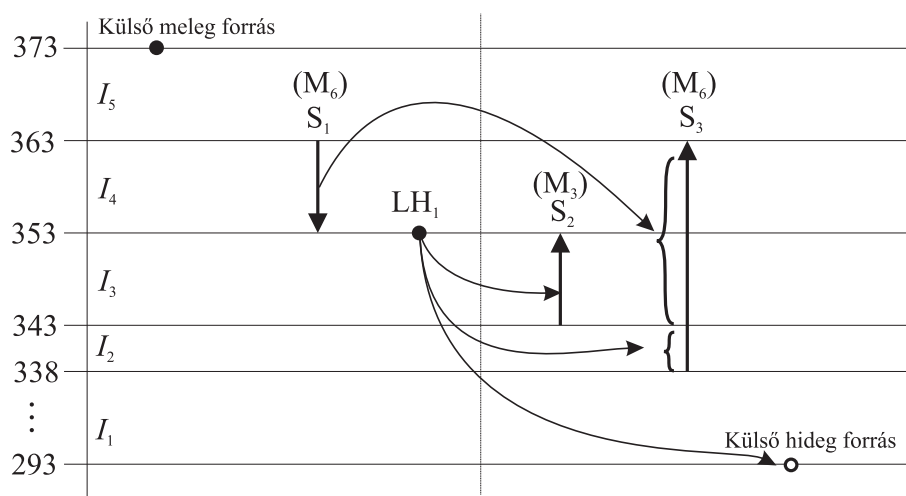
| Elemi hőáramok | Típus | Anyag | Kezdő hőm. (K) | Vég hőm. (K) |
|----------------|-------|-------|----------------|--------------|
| $FSH_1$        | meleg | $M_6$ | 363.0          | 353.0        |
| $FSC_1$        | hideg | $M_3$ | 333.0          | 343.0        |
| $FSC_2$        | hideg | $M_6$ | 338.0          | 343.0        |
| $FSC_3$        | hideg | $M_6$ | 343.0          | 353.0        |
| $FSC_4$        | hideg | $M_6$ | 353.0          | 363.0        |



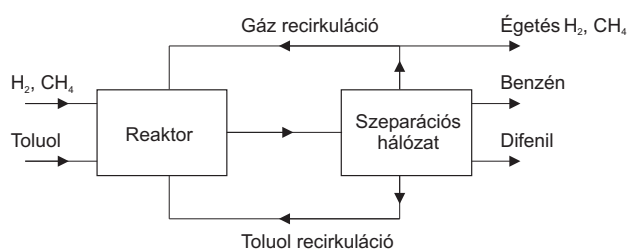
4.13. ábra. Kaszkád diagram a jellemző hőáramokról az 1.1.1.2. csúcsban.

4.18. táblázat. Rész hőáramok az 1.1.1.2. csúcsban

| Rész hőáramok | Típus | Anyag | Intervallum     | Kezdő hőm. (K) | Vég hőm.(K) |
|---------------|-------|-------|-----------------|----------------|-------------|
| $SSH_1$       | meleg | $M_6$ | $I_4$           | 363.0          | 353.0       |
| $SSC_1$       | hideg | $M_3$ | $I_2$           | 338.0          | 343.0       |
| $SSC_2$       | hideg | $M_6$ | $I_3$           | 343.0          | 353.0       |
| $SSC_3$       | hideg | $M_6$ | $I_4$           | 353.0          | 363.0       |
| $SSC_4$       | hideg | $M_6$ | $I_2, I_3$      | 338.0          | 353.0       |
| $SSC_5$       | hideg | $M_6$ | $I_3, I_4$      | 343.0          | 363.0       |
| $SSC_6$       | hideg | $M_6$ | $I_2, I_3, I_4$ | 338.0          | 363.0       |



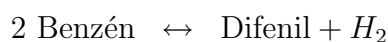
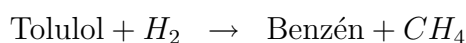
4.14. ábra. Az optimális struktúrához tartozó hőátvitelek.



4.15. ábra. HDA folyamat diagramja.

## 4.8. Alkalmazás: HDA folyamat

A szakirodalomban jól ismert probléma kiterjesztésére alkalmazzuk az eljárásunkat. A HDA (toluene-hydrodealkylation) [22] folyamat magába foglal egy reaktort és egy szeparációs hálózatot (lásd 4.15 ábra). A reakciók a következők:



A nyersanyagok a toluol és hidrogén, amelyeket melegítés után a visszavezetett toluollal összekeverve vezetik be a reaktorba. A reaktorból kijövő anyagáram hidrogént, metánt, benzént, toluolt és a nem kívánt difenilt tartalmaz. A hidrogént és a metán

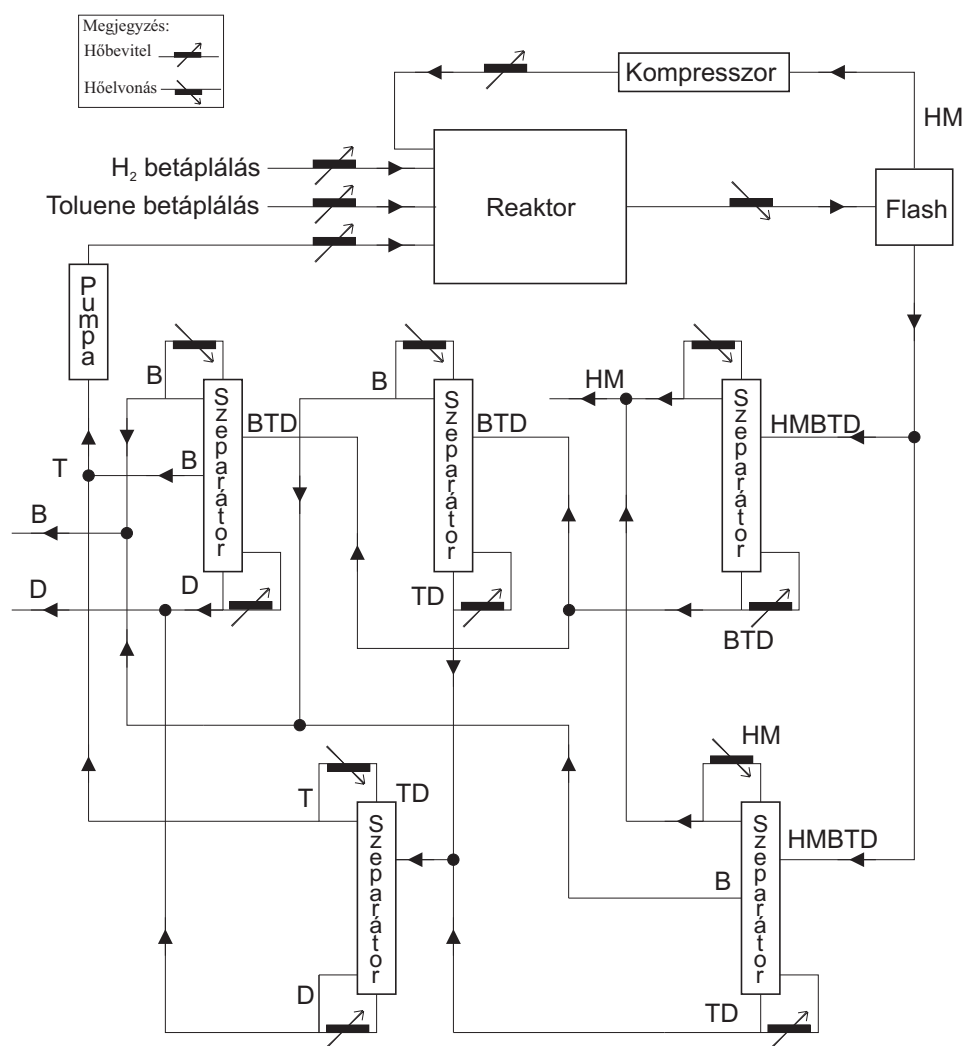
legnagyobb részét kondenzálással kivonják. Az így kivont hidrogén csak egy részét vezetik vissza, mert szennyezőanyagként metánt is tartalmaz és a feldúsulás elkerülése miatt a gáz egy részét elégetik. A termék a benzén, amit a szeparációs rész választ el a nem kívánt difeniltől és a toluoltól, amit visszavezetnek. A szeparációs rész két egyszerű (egy bemenet-két kimenet) és két összetett (egy bemenet-három kimenet) szeparátort tartalmaz (lásd 4.16 ábra).

A reaktor működési hőmérséklete  $895K$  így a bemenő anyagokat erre a hőmérsékletre kell melegíteni. A reaktort elhagyó anyagáramot hűteni kell a kondenzáció miatt. Minden szeparátorhoz tartozik egy kiforraló és egy kondenzáló rész, amely szintén hőbevitelt vagy hőelvitelt tesz szükségessé. A reaktor exoterm hőszínezetű, így hűtést igényel. A rendszer tartalmaz 4 meleg és 4 hideg hőáramot továbbá 6 meleg és 5 hideg rejtett hőt. A meleg elemi hőáramok száma 14, a hideg elemi hőáramok száma pedig 48. Ez  $48+12$  lineáris feltételt jelent a HENS modellben. 105 hideg és 333 meleg részhőáram keletkezett, amely 10227 db potenciális hőcserélőt eredményezett. Az eljárásunk 96.07 másodperc alatt azonosította az optimális hálózatot egy PC-n (Celeron 400MHz). Az optimális hálózatot a 4.17 ábra mutatja, amely 7 műveleti egységet és 18 hőcserélőt tartalmaz.

## 4.9. Az eredmény rövid összefoglalása

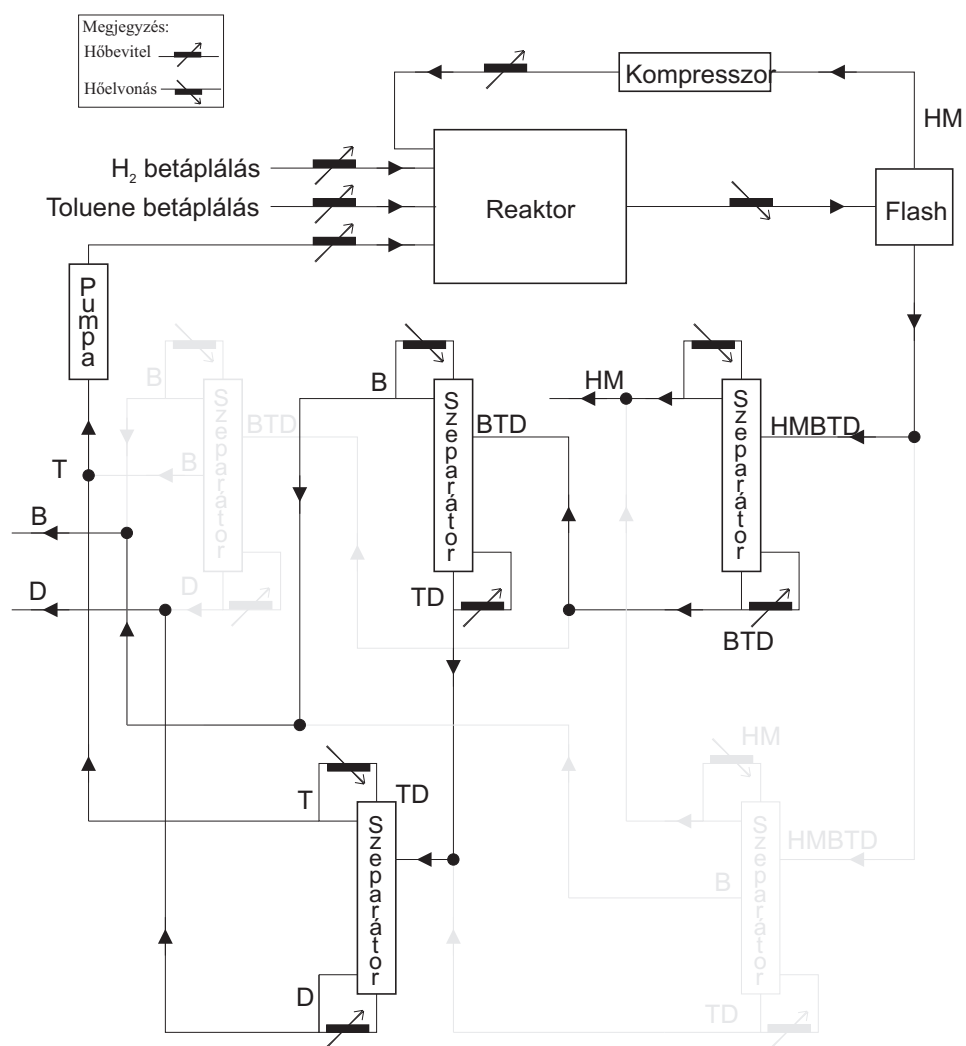
A 4. fejezet a 4. tézispontban megfogalmazott eredményeket tartalmazza.

Az integrált folyamathálózat- és hőcserélőhálózat-szintézis feladat megoldásához a korlátozás és szétválasztás algoritmus részproblémáinak matematikai modelljében a hőcserélőhálózat nemlineáris feltételeit kombinatorikus eszközökkel lineárisra transzformáltam, ezáltal a feladat megoldhatóvá vált a MILP alapú PNS keretalgoritmussal.



4.16. ábra. A HDA folyamat maximális struktúrája.





4.17. ábra. Az optimális hálózatot tartalmazó folyamatábra.

## 5. fejezet

# Új tudományos eredmények

Az értekezés új tudományos eredményeinek tézisszerű összefoglalása.

1. A lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat megoldása során felmerülő szétválasztás lépésre két új partícionálási stratégiát dolgoztam ki, melyek számításigénye alacsony, és PNS feladatokon kedvező tulajdonságokkal bírhatnak nagyméretű feladatok esetén. A két módszer viselkedését gyakorlati példákön illusztráltam.
  - (a) A szakirodalomból ismert és széleskörűen alkalmazott (Shectman és munkatársai [89]) partícionálási stratégiát megvizsgálva bemutattam a partícionálási stratégia egyik kedvezőtlen tulajdonságát: a módszer feleslegesen sok olyan részproblémát generál, ami tartalmazza az optimális megoldást. Az, hogy az optimális megoldás sok aktív részproblémában szerepel, nagyban megnehezíti a megtalált megoldás optimalitásának bizonyítását. A bizonyítás így teljes bináris fa bejárását teszi szükségessé, amelynek a mélysége megegyezik az optimális hálózatban lévő csúcsok számával. Ennek a kedvezőtlen tulajdonságnak a kiküszöbölésére dolgoztam ki az úgynevezett "csúsztatott" szétválasztási stratégiát, amelyben az optimális megoldást tartalmazó részproblémákat nem sokszorozzuk meg. A PNS feladatok megoldására ez különösen jól használható. A módszer helyességét bizonyítottam.
  - (b) Kidolgoztam egy új szétválasztási stratégiát, amely a célfüggvény és a relaxációs függvény integrálkülönbségét minimalizálja, ezáltal a relaxáció

élességét maximálisra növeli. Bizonyítottam a módszer helyességét.

2. A lineáris programozási feladat érzékenységi vizsgálatát felhasználva új eljárást dolgoztam ki a lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat megoldására.
  - (a) Megadtam a lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat egy elégséges optimalitási kritériumát.
  - (b) Új eljárást dolgoztam ki a lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat megoldása során felmerülő szétválasztás lépésre. Az eljárás a 2a tézispontban megfogalmazott optimalitási kritériumon alapulva végzi a részproblémák partícionálását, illetve a terminális részproblémák meghatározását. A megfogalmazott algoritmus helyességét igazoltam.
3. A lineáris feltételrendszerrel adott, változóiban szétválasztható konkáv programozási feladat megoldására kidolgoztam a szakirodalomban ismert korlátozás és szétválasztás alapalgoritmus kombinatorikusan gyorsított változatát. A kombinatorikus gyorsítások elsősorban a PNS feladatok megoldására hatékonyak, de ritka mátrixszal adott, jól struktúrált feltételek esetében is jól alkalmazhatók.
  - (a) A Friedler és munkatársa által kidolgozott P-gráf módszert felhasználva elkészítettem a lineáris feltételrendszerrel adott szeparábilis konkáv programozási feladatot megoldó algoritmus kombinatorikusan gyorsított változatát.
  - (b) PNS feladatokra bevezettem a lokálisan optimális struktúrák fogalmát, amely lehetővé teszi az optimális megoldások mellett szuboptimális megoldások meghatározását. Kidolgoztam a kombinatorikusan gyorsított algoritmus egy változatát, amely képes ezen szuboptimális megoldások generálására.

4. Az integrált folyamathálózat- és hőcserélőhálózat-szintézis feladat megoldásához a korlátozás és szétválasztás algoritmus részproblémáinak matematikai modelljében a hőcserélőhálózat nemlineáris feltételeit kombinatorikus eszközökkel lineárisra transzformáltam, ezáltal a feladat megoldhatóvá vált a MILP alapú PNS keretalgoritmussal.

## 5.1. Az értekezés témaköréből készült publikációk

### Referált folyóiratcikkek nemzetközi folyóiratban

- [1] Nagy, A. B., R. Adonyi, L. Halasz, F. Friedler, and L. T. Fan, *Integrated Synthesis of Process and Heat Exchanger Networks: Algorithmic Approach*, Applied Thermal Engineering, **21** (2001), 1407-1427.
- [2] Halasz, L., A. B. Nagy, T. Ivicz, F. Friedler, L. T. Fan, *Optimal Retrofit Design and Operation of the Steam-Supply System of a Chemical Complex*, Applied Thermal Engineering **22** (2002), 939 -947.
- [3] Hertwig, T. A., A. Xu, A. B. Nagy, R. W. Pike, J. R. Hopper, and C. L. Yaws, *A Prototype System for Economic, Environmental and Sustainable Optimization of a Chemical Complex*, Clean Techn. Environ. Policy, **3** (2002), 363-370.
- [4] Illes, T., A. B. Nagy, *New and elementary proof of a sufficient optimality criteria for constrained, concave minimization problems*, közlésre elfogadva Journal of Optimization Theory and Applications (2004).

### Konferencia-kiadványokban megjelent közlemények

- [5] Hertwig, T. A., A. Xu, A. B. Nagy, R. W. Pike, J. R. Hopper, and C. L. Yaws, *A Prototype System for Economic, Environmental and Sustainable Optimization of*

*a Chemical Complex*, Computer-Aided Chemical Engineering, **9** R. Gani, S.B. Jorgensen (Eds.), Elsevier (2001), 1017-1022.

- [6] 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*, appeared in the proceedings of the PRES'01 (4th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Florence, Italy, May 20-23, 2001, pp.1-6
- [7] Halasz, L., A. B. Nagy, T. Ivicz, F. Friedler, and L. T. Fan, *Optimal Operation of the Steam Supply Network of a Complex Chemical Processing System*, appeared in the proceedings of the PRES'01 (4th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Florence, Italy, May 20-23, 2001, pp.331-336
- [8] T. A. Hertwig, A. B. Nagy, and R. W. Pike, *An Advanced System for Optimizing the design of a Chemical-Production Complex*, appeared in Proceedings of IFA Technical Conference, New Orleans, Louisiana, United States, October 1-4, 2000.

### Nemzetközi konferencia előadások

- [9] Holenda, B., A. B. Nagy, A. Dallos, F. Friedler, and L. T. Fan, *A Combinatorial Approach for Generating Environmentally Benign Solvents and Separation Agents*, presented at ICheaP-6, Pisa, Italy, June 8-11, 2003.
- [10] Halasz, L., A. B. Nagy, M. Narodoslawsky, F. Friedler, *Synthesis Method for Tankage Treatment*, presented at the PRES'02 (5th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Praha, Czech Republic, August 25-29, 2002.
- [11] Hertwig, T. A., A. B. Nagy, R. W. Pike, J. R. Hopper, and C. L. Yaws, *An Advanced System for Optimizing the Design of a Chemical Production Complex*,

presented at ESCAPE-11 (Eleventh European Symposium on Computer Aided Process Engineering), Kolding, Denmark, May 27-30, 2001.

- [12] 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*, Plenary lecture, presented at the PRES'01 (4th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Florence, Italy, May 20-23, 2001.
- [13] Halasz, L., A. B. Nagy, T. Ivicz, F. Friedler, and L. T. Fan, *Optimal Operation of the Steam Supply Network of a Complex Chemical Processing System*, presented at PRES '01 (4th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Florence, Italy, May 20-23, 2001.
- [14] Pike, R. W., T. A. Hertwig, and A. B. Nagy, *An advanced system for optimizing the design of a chemical-production complex*, presented at IFA Technical Conference, New Orleans, Louisiana, United States, October 1-4, 2000
- [15] Nagy, A. B., G. Biros, F. Friedler, and L. T. Fan, *Integrated Synthesis of Combined Process and Heat Exchanger Networks*, presented at the CHISA 2000 (14th International Congress of Chemical and Process Engineering), Praha, Czech Republic, August 27-31, 2000.
- [16] Nagy, A. B., F. Friedler, and L. T. Fan, *Integrated Synthesis of Combined Process and Heat Exchanger Networks*, presented at the 20th Workshop on Chemical Engineering Mathematics, Veszprem, Hungary, July 26-29, 2000.
- [17] Nagy, A. B., L. Halasz, F. Friedler, and L. T. Fan, *Integrated Synthesis of Process and Heat Exchanger Networks: Algorithmic Approach*, presented at the PRES'99 (2nd Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction, Budapest, Hungary, May 31-June 2, 1999.
- [18] Nagy, A. B., F. Friedler, and L. T. Fan, *Combinatorial Acceleration of Separable*

*Concave Programming for Process Synthesis*, presented at the AIChE Annual Meeting, Miami Beach, FL, U.S.A., November 15-20, 1998.

- [19] Fülöp, J., F. Friedler, and A. B. Nagy, *Integration of Combinatorial and Global Optimization Approaches for Solving Process Network Synthesis Problems*, presented at the INFORMS (Institute for Operations Research and the Management Sciences) Conference, Seattle, WA, U.S.A., October 25-28, 1998.
- [20] Nagy, A. B., F. Friedler, and L. T. Fan, *Algorithmic Generation of Multiple Solutions for Process Synthesis*, presented at the CHISA '98 (13th International Congress of Chemical and Process Engineering), Praha, Czech Republic, August 23-28, 1998.
- [21] Fülöp, J., F. Friedler, and A. B. Nagy, *A Global Optimization for Solving Process Network Synthesis Problems*, presented at the ISMP'97 (International Symposium on Mathematical Programming), EPFL, Lausanne, Switzerland, August 24-29, 1997.

### **Hazai konferencia előadások**

- [22] Nagy Ádám, Friedler Ferenc, *Inside-out algoritmus PNS feladatok megoldására*, XXV. Magyar Operációkutatási Konferencia, Debrecen, 2001. október 17-20.
- [23] Nagy Ádám és Friedler Ferenc, *A folyamatszintézis globális optimalizálási eljárásainak összehasonlítása*, 26. Műszaki Kémiai Napok, Veszprém, 1998. április 15-17.

# Irodalomjegyzék

- [1] N. Agin, *Optimum seeking with branch and bound*, Management Sci. **13** (1966), 176–185.
- [2] S. Ahmad, B. Linnhoff, and R. Smith, *Cost optimum heat exchanger networks 2. targets and design for detailed capital cost models*, Comp. Chem. Eng. **14** (1990), 751–767.
- [3] P. Apkarian and H. D. Tuan, *Concave programming in control theory*, J. Global Optim. **15** (1999), 343–370.
- [4] H. P. Benson, *Separable concave minimization via partial outer approximation and branch and bound*, Oper. Res. Lett. **9** (1990), 389–394.
- [5] H. P. Benson and S. Sayin, *A finite concave minimization algorithm using branch and bound and neighbor generation*, J. Global Optim. **5** (1994), 1–14.
- [6] B. Bertok, F. Friedler, and L. T. Fan, *Generation of process synthesis problems for testing evaluating and comparing synthesis methods*, PRES’99 (Second Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction) Budapest, 1999.
- [7] L. T. Biegler, I. E. Grossmann, and A. W. Westerberg, *Systematic methods of chemical process design*, Prentice Hall: Upper Saddle River, New Jersey, 1997.
- [8] M. H. Brendel, F. Friedler, and L. T. Fan, *Combinatorial foundation for logical formulation in process network synthesis*, Comp. Chem. Eng. **24** (2000), 1859–1864.



- [9] K. M. Bretthauer and A. V. Cabot, *A composite branch and bound, cutting plane algorithm for concave minimization over a polyhedron*, Comput. Oper. Res. **21** (1994), 777–785.
- [10] T. Terlaky C. Roos and J.P. Vial, *Theory and algorithms for linear optimization: An interior point approach*, John Wiley and Sons, 1997.
- [11] A. V. Cabot and S. S. Erenguc, *A branch and bound algorithm for solving a class of nonlinear integer programming problems*, Naval Res. Logist. Quart. **33** (1986), 559–567.
- [12] J. Cerda, A. W. Westerberg, D. Mason, and B. Linnhoff, *Minimum utility usage in heat exchanger network synthesis*, Chem. Eng. Sci. **38** (1983), 373–387.
- [13] R. W. Colbert, *Industrial heat exchange networks*, Chem. Eng. Prog. **78** (1982), 47–54.
- [14] J. Corominas, A. Espuna, and L. Puigjaner, *A new look at energy integration in multiproduct batch processes*, Comp. Chem. Eng. **17** (1993), S15–S20.
- [15] ———, *Method to incorporate energy integration considerations in multiproduct batch processes*, Comp. Chem. Eng. **18** (1994), 1043–1055.
- [16] A. Csallner, T. Csendes, and M. Markót, *Multisection in interval branch-and-bound methods for global optimization I. theoretical results*, J. Global Optim. **16** (2000), 219–228.
- [17] T. Csendes and D. Ratz, *Subdivision direction selection in interval methods for global optimization*, SIAM Journal of Numerical Analysis **34** (1997), 922–938.
- [18] Á. Császár, *Valós analízis I.*, Tankönyvkiadó, Budapest, 1983.
- [19] M. M. Daichendt. and I. E. Grossmann, *Preliminary screening procedure for the minlp synthesis of process systems I. aggregation and decomposition*, Comp. Chem. Eng. **18** (1994), 663–677.
- [20] ———, *Preliminary screening procedure for the MINLP synthesis of process systems II. heat exchanger networks*, Comp. Chem. Eng. **18** (1994), 679–709.

- [21] ———, *Integration of hierarchical decomposition and mathematical programming for the synthesis of process flowsheets*, Comp. Chem. Eng. **22** (1998), 147–175.
- [22] J. M. Douglas, *Conceptual design of chemical processes*, McGraw-Hill, New York, 1988.
- [23] V. Dua and E. N. Pistikopoulos, *An outer-approximation algorithm for the solution of multiparametric MINLP problems*, Comp. Chem. Eng. **22** (1998), S955–S958.
- [24] M. A. Duran and I. E. Grossmann, *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Math. Programming **36** (1986), 307–339.
- [25] ———, *Simultaneous optimization and heat integration of chemical processes*, AIChE J. **32** (1986), 123–138.
- [26] M. E. Dyer, *The complexity of vertex enumeration methods*, Math. Oper. Res. **8** (1983), 381–402.
- [27] M. E. Dyer and L. G. Proll, *An algorithm for determining all extreme points of a convex polytope*, Math. Programming **12** (1977), 81–96.
- [28] Cs. Fabian, *Linux, an interactive linear programming library*, 1992.
- [29] J. E. Falk and R. M. Soland, *An algorithm for separable nonconvex programming problems*, Management Sci. **15** (1969), 550–569.
- [30] C. A. Floudas, *Recent advances in global optimization for process synthesis, design and control: Enclosure of all solutions*, Comp. Chem. Eng. **23** (1999), S963–S973.
- [31] C. A. Floudas and A. R. Ciric, *Strategies for overcoming uncertainties in heat exchanger network synthesis*, Comp. Chem. Eng. **13** (1989), 1133–1152.
- [32] C. A. Floudas, A. R. Ciric, and I. E. Grossmann, *Automatic synthesis of optimum heat exchanger network configurations.*, AIChE J. **32** (1986), 276–290.
- [33] C. A. Floudas and V. Visweswaran, *A global optimization algorithm (GOP) for certain classes of nonconvex NLPs - I. Theory*, Comp. Chem. Eng. **14** (1990), 1397–1417.

- [34] E. S. Fraga, *The automated synthesis of complex reaction separation processes using dynamic programming*, Chem. Eng. Res. Des. **74** (1996), 249–260.
- [35] E. S. Fraga and K. I. M. McKinnon, *The use of dynamic programming with parallel computers for process synthesis*, Computers chem. Engng **18** (1994), 1–13.
- [36] F. Friedler, L. T. Fan, and B. Imreh, *Process network synthesis: problem definition*, Networks **31** (1998), 119–124.
- [37] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan, *Graph-theoretic approach to process synthesis: Axioms and theorems*, Chem. Engng Sci. **47** (1992), 1973–1988.
- [38] ———, *Graph-theoretical approach to process synthesis: Polynomial algorithm for maximal structure generation*, Comput. Chem. Eng. **17** (1993), 929–942.
- [39] F. Friedler, J. B. Varga, and L. T. Fan, *Decision-mapping: A tool for consistent and complete decisions in process synthesis*, Chem. Engng Sci. **50** (1995), 1755–1768.
- [40] F. Friedler, J. B. Varga, E. Fehér, and L. T. Fan, *Combinatorial accelerated branch-and-bound method for solving the MIP model of process network synthesis*, Nonconvex Optimization and its Applications (Eds: C. A. Floudas and P. M. Pardalos), pp. 609–626, Kluwer Academic Publishers, Norwell, MA, U.S.A., 1996.
- [41] K. C. Furman and N.V. Sahinidis, *Computational complexity of heat exchanger network synthesis*, Comput. Chem. Eng. **25** (2001), 1371–1390.
- [42] T. Gal and H. J. Greenberg (eds.), *Advances in sensitivity analysis and parametric programming*, Kluwer Academic Publishers, 1997.
- [43] M. R. Galli and J. Cerda, *Synthesis of structural-constrained heat exchanger networks I. series networks*, Comp. Chem. Eng. **22** (1998), 819–839.
- [44] ———, *Synthesis of structural-constrained heat exchanger networks II. split networks*, Comp. Chem. Eng. **22** (1998), 1017–1035.
- [45] B. Gross and P. Roosen, *Total process optimization in chemical engineering with evolutionary algorithms*, Comp. Chem. Eng. **22** (1998), S229–S236.

- [46] I. E. Grossmann, H. Yeomans, and Z. Kravanja, *A rigorous disjunctive optimisation model for simultaneous flowsheet optimisation and heat integration*, Comp. Chem. Eng. **22** (1998), S157–S164.
- [47] T. Gundersen, S. Duvold, and Hashemi-Ahmady A., *An extended vertical MILP model for heat exchanger network synthesis*, Comp. Chem. Eng. **20** (1996), S97–S102.
- [48] T. Gundersen and I. E. Grossmann, *Improved optimization strategies for automated heat exchanger network synthesis through physical insights*, Comp. Chem. Eng. **14** (1990), 925–944.
- [49] K. L. Hoffman, *A method for globally minimizing concave functions over convex sets*, Math. Programming **20** (1981), 22–32.
- [50] R. Horst, *On generalized bisection of  $n$ -simplices*, Math. Comp. **66** (1997), 691–698.
- [51] C. W. Hui and S. Ahmad, *Total site integration using the utility system*, Comp. Chem. Eng. **18** (1994), 729–742.
- [52] C. S. Hwa, *Mathematical formulation and optimization of heat exchanger networks using separable programming*, AIChE-ICHEME Symposium Series No. 4 (1965), 101–106.
- [53] T. Ibaraki, *Theoretical comparisons of search strategies in branch-and-bound algorithms*, Internat. J. Comput. Information Sci. **5** (1976), 315–344.
- [54] J. Itoh, K. Shiroko, and T. Umeda, *Extensive applications of the  $t$ - $q$  diagram to heat integrated system synthesis*, Comp. Chem. Eng. **10** (1986), 59–66.
- [55] J. Kallrath, *Mixed integer optimization in the chemical process industry - experience, potential and future perspectives*, Chem. Eng. Res. Des. **78** (2000), 809–822.
- [56] B. Kearfott and K. Du, *The cluster problem in global optimization: The univariate case*, Computing (Suppl.) **9** (1992), 117–127.
- [57] ———, *The cluster problem in multivariate global optimization*, J. Global Optim. **5** (1994), 253–265.

- [58] G. R. Kocis and I. E. Grossmann, *A modeling and decomposition strategy for the MINLP optimization of process flowsheets*, Comp. Chem. Eng. **13** (1989), 797–819.
- [59] S. Kontogiorgis, *Practical piecewise-linear approximation for monotropic optimization*, INFORMS J. Comput. **12** (2000), 324–340.
- [60] T. Kuno and T. Utsunomiya, *A Lagrangian based branch-and-bound algorithm for production-transportation problems*, J. Global Optim. **18** (2000), 59–73.
- [61] B. W. Lamar, *Nonconvex optimization over a polytope using generalized capacity improvement*, J. Global Optim. **7** (1995), 127–142.
- [62] Y. D. Lang, L. T. Biegler, and I. E. Grossmann, *Simultaneous optimisation and heat integration with process simulators*, Comp. Chem. Eng. **12** (1988), 311–327.
- [63] E. L. Lawler and D. E. Wood, *Branch-and-bound methods: A survey*, Operations Res. **14** (1966), 699–719.
- [64] B. Linnhoff and S. Ahmad, *Cost optimum heat exchanger networks 1. minimum energy and capital using simple models for capital cost*, Comp. Chem. Eng. **14** (1990), 729–750.
- [65] B. Linnhoff and E. Hindmarsh, *The pinch design method for heat exchanger networks*, Chem. Eng. Sci. **38** (1983), 745–763.
- [66] B. Linnhoff, R. Smith, and J. D. Williams, *The optimization of process changes and utility selection in heat integrated processes*, Chem. Eng. Res. Des. **68** (1990), 221–236.
- [67] M. Liu, N. V. Sahinidis, and J. P. Sheetman, *Planning of chemical process networks via global concave minimization*, Global optimization in engineering design, Kluwer Acad. Publ., Dordrecht, 1996, pp. 195–230.
- [68] M. Locatelli and N. V. Thoai, *Finite exact branch-and-bound algorithms for concave minimization over polytopes*, J. Global Optim. **18** (2000), 107–128.
- [69] D. G. Luenberger, *Intoduction to linear and nonlinear programming*, Addison-Wesley Publishing Co., Massachusetts, 1973.

- [70] F. Marechal and B. Kalitventzeff, *Identification of the optimal pressure levels in steam networks using integrated combined heat and power method*, Chem. Eng. Sci. **52** (1997), 2977–2989.
- [71] A. H. Masso and D. F. Rudd, *The synthesis of system designs II. heuristic structuring*, AIChE J. **15** (1969), 10–17.
- [72] M. Minoux, *Network synthesis and optimum network design problems: models, solution methods and applications*, Networks **19** (1989), 313–360.
- [73] J. J. Moré and S. A. Vavasis, *On the solution of concave knapsack problems*, Math. Programming **49** (1990/91), 397–441.
- [74] K. G. Murty, *Linear programming*, John Wiley and Sons, 1983.
- [75] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Programming **39** (1987), 117–129.
- [76] L. Otavio, A. Maia, and R.Y. Qassim, *Synthesis of utility systems with variable demands using simulated annealing*, Comp. Chem. Eng. **21** (1997), 947–950.
- [77] K. P. Papalexandri and E. N. Pistikopoulos, *A decomposition-based approach for process optimization and simultaneous heat integration*, Chem. Eng. Res. Des. **76** (1998), 273–286.
- [78] K. P. Papalexandri, E.N. Pistikopoulos, and B. Kalitventzeff, *Modelling and optimization aspects in energy management and plant operation with variable energy demands-application to industrial problems*, Comp. Chem. Eng. **22** (1998), 1319–1333.
- [79] K. P. Papalexandri, E.N. Pistikopoulos, B. Kalitventzeff, M.N. Dumont, K. Urmann, and J. Gorschluter, *Operation of a steam production network with variable demands modelling and optimization under uncertainty*, Comp. Chem. Eng. **20** (1996), S763–S768.
- [80] S. A. Papoulias and I. E. Grossmann, *A structural optimization approach in process synthesis II. heat recovery networks.*, Comp. Chem. Eng. **7** (1983), 707–721.

- [81] A. T. Phillips and J. B. Rosen, *Sufficient conditions for solving linearly constrained separable concave global minimization problems*, J. Global Optim. **3** (1993), 79–74.
- [82] M. Porembski, *Cone adaptation strategies for a finite and exact cutting plane algorithm for concave minimization*, J. Global Optim. **24** (2002), 89–107.
- [83] R. Raman and I. E. Grossmann, *Symbolic integration of logic in MILP branch and bound methods for the synthesis of process networks*, Annals of Operations Research **42** (1993), 169–191.
- [84] D. Ratz, *Boxsplitting strategies for the interval gaussidel step in a global optimization method*, Computing **53** (1994), 1–16.
- [85] ———, *On branching rules in second-order branch-and-bound methods for global optimization*, Scientific Computing and Validated Numerics (Berlin) (G. Alefeld, A. Frommer, and B. Lang, eds.), Akademie Verlag, 1996, pp. 221–227.
- [86] D. Ratz and T. Csendes, *On the selection of subdivision directions in interval branchandbound methods for global optimization*, J. Global Optim. **7** (1995), 183–207.
- [87] E. Rev and Z. Fonyo, *Hidden and pseudo pinch phenomena and relaxation in the synthesis of heat-exchange networks*, Comp. Chem. Eng **10** (1986), 601–607.
- [88] J. B. Rosen, *Global minimization of a linearly constrained concave function by partition of feasible domain*, Math. Oper. Res. **8** (1983), 215–230.
- [89] J. P. Shectman and N. V. Sahinidis, *A finite algorithm for global minimization of separable concave programs*, J. Global Optim. **12** (1998), 1–35.
- [90] R. Smith, *Chemical process design*, McGraw-Hill, New York, 1994.
- [91] ———, *State of the art in process integration*, Applied Thermal Engineering **20** (2000), 1337–1345.
- [92] R. M. Soland, *Optimal facility location with concave costs*, Operations Res. **22** (1974), 373–382.
- [93] H. Ten Broeck, *Economic selection of exchanger sizes*, Ind. & Eng. Chem. **36** (1944), 64–67.

- [94] K. K. Trivedi, B. K. O'Neill, and J. R. Roach, *Synthesis of heat exchanger networks featuring multiple pinch points*, Comp. Chem. Eng. **13** (1989), 291–294.
- [95] H. Tuy, T. V. Thieu, and Ng. Q. Thai, *A conical algorithm for globally minimizing a concave function over a closed convex set*, Math. Oper. Res. **10** (1985), 498–514.
- [96] N. Vakkeliya-Bancheva, B. B. Ivanov, N. Shah, and C. C. Pantelides, *Heat exchanger network design for multipurpose batch plants*, Comp. Chem. Eng. **20** (1996), 989–1001.
- [97] J. A. Vaselenak, I. E. Grossmann, and A. W. Westerberg, *Heat integration in batch processing*, Ind. Eng. Chem. Proc. Des. Dev. **25** (1988), 357–366.
- [98] V. Visweswaran and C. A. Floudas, *A global optimization algorithm (GOP) for certain classes of nonconvex NLPs - II. application of theory and test problems*, Comp. Chem. Eng. **14** (1990), 1419–1434.
- [99] R. M. Wood, K. Suaysompol, B. K. O'Neill, J. R. Roach, and K. K. Trivedi, *A new option for heat exchanger network design.*, Chem. Eng. Prog. **87** (1991), 38–43.
- [100] Y. Yajima and H. Konno, *An algorithm for a concave production cost network flow problem*, Japan J. Indust. Appl. Math. **16** (1999), 243–256.
- [101] T. F. Yee and I. E. Grossmann, *Simultaneous optimization models for heat integration II. heat exchanger network synthesis*, Comp. Chem. Eng. **14** (1990), 1165–1184.
- [102] T. F. Yee, I. E. Grossmann, and Z. Kravanja, *Simultaneous optimization models for heat integration III. process and heat exchanger network optimization*, Comp. Chem. Eng. (1990), 1185–1200.
- [103] X. Yuan, L. Pibouleau, and Domenech S., *Experiments in process synthesis via mixed-integer programming*, Chem. Eng. Process. **25** (1989), 99–116.
- [104] J. Zhang and X. X. Zhu, *Simultaneous optimization approach for heat exchanger network retrofit with process changes*, Ind. Eng. Chem. Res. **39** (2000), 4963–4973.



- [105] X. G. Zhao, B. K. O'Neill, J. R. Roach, and R. M Wood, *Heat integration for batch processes part 1: Process scheduling based on cascade analysis*, Chem. Eng. Res. Des. **76(A)** (1998), 685–699.
- [106] ———, *Heat integration for batch processes part 2: Heat exchanger network design*, Chem. Eng. Res. Des. **76(A)** (1998), 700–710.