1. fejezet

Bevezetés

1.1. Az egészértékű programozás tárgya

Az egészértékű programozás célja bizonyos optimalizálási feladatok megoldása, azaz az egészértékű programozás a feladatok egy csoportját jelenti. A vizsgálat tárgyát képezi egyben az összes olyan módszer is, amely ezen feladatok valamelyikének a megoldására alkalmas.

Az egészértékű programozás a

$$\begin{aligned} &\max f(x) \\ &\quad x \in S \end{aligned}$$

alakú feladatokkal foglalkozik, ahol S megszámlálható halmaz. Ebbe a nagyon általános formába beletartozik a kombinatorika számos problémája is. Ezekkel a kombinatorikus vagy diszkrét optimalizálás foglalkozik külön. Az egészértékű programozás tárgyát a kevésbé struktúrált, azaz általánosabb feladatok képezik. Az első nagyobb csoportot azok a feladatok alkotják, amelyek formailag nagyon hasonlítanak a lineáris programozás feladatához, de az algebrai feltételeken, azaz az egyenleteken és egyenlőtlenségeken túl a nemnegativitás mellett bizonyos egészértékűségi feltételeket is kirovunk, melyek három leggyakoribb alakja egy x változó esetén:

$$x = 0 \text{ vagy } 1, \tag{1.1}$$

$$0 \le x \le d, \quad x \text{ egész}, \tag{1.2}$$

$$0 \le x, \text{ egész}, \tag{1.3}$$

ahol d rögzített pozitív egész. A vizsgált feladatok zöme olyan, hogy valamennyi változóra azonos előírást teszünk. Az alkalmazások többsége szempontjából ez nem jelent megszorítást. Ha egy feladaton belül a típusok keverednek, a legnehezebben kezelhető típusnak megfelelően kell eljárni.

Az (1.1) típusba tartozókat döntési változóknak fogjuk nevezni, mert azt fejezik ki, hogy két lehetőség közül melyiket válasszuk, például megvalósítsunk-e egy beruházást vagy sem.

Azonnal látszik, hogy az (1.1) típus a (1.2) speciális esete. Azonban (1.2) is visszavezethető (1.3)-ra. Vezessünk be ugyanis x mellé egy y változót, valamint az

$$x + y = d \tag{1.4}$$

feltételt. Nyilvánvaló, hogy ha mind x-re, mind y-ra az (1.3) típusú feltételt rójuk ki, akkor egyikük értéke sem haladhatja meg d-t. Így érthető, hogy azok a módszerek, amelyeket az (1.3) típusra dolgoztak ki, az előző kettőnél is alkalmazhatók, míg fordítva nem, és a sokat vizsgált (1.1) típushoz sajátos módszereket kerestek és találtak.

Az eddigiek az ún. tiszta egészértékű feladatok, vagyis az olyanok, amelyekben valamennyi változóra van egészértékűségi követelmény. A feladatok másik nagy körét a vegyes feladatok képezik, amelyekben a változók egy része bizonyos határok között folytonosan változhat.

1.2. Feladatok és modellek

Az alábbiakban a legfontosabb feladatokat soroljuk fel. A sorrend némileg jelzi a feladatok bonyolultságát és számítástechnikai nehézségét. Az utóbbit gyakorlati szempontból azzal mérjük, hogy mi a még numerikusan megoldható feladatokban a változók száma. Természetesen ez függ az alkalmazható számítógép kapacitásától. Másfelől azonban a méret jelentősen befolyásolja a megoldható gyakorlati problémák körét. A feladatok alábbi felosztása azért keletkezett, mert valójában mind bizonyos mértékig eltérő módszereket igényelnek. Az általános módszerek természetesen valamennyi feladatot meg tudják oldani, de a speciálisakra sok olyant dolgoztak ki, amelyek ügyesen használják fel az adott feladat struktúráját, és így jelentősen kitolják a megoldhatóság határát.

1.2.1. A hátizsák feladat és variánsai

Operációkutatásban több feladatot is egy-egy történeten keresztül szokás bevezetni. Íme a hátizsák feladaté: Egy turista különböző tárgyakat vihet magával az útra. A tárgyak száma n. Adott minden j tárgy súlya, amit a_j-vel, és értéke, amit c_j-vel jelölünk. Érték alatt itt nem a tárgy árát, hanem azt a hasznot avagy örömöt értjük, amit a tárgy a túra alkalmával hajt, illetve okoz. A turista egy bizonyos, előre adott és b-vel jelölt súlykorlátnál többet nem akar magával vinni. Úgy akarja az elviendő tárgyakat kiválogatni, hogy azok együttes értéke maximális legyen, miközben együttes súlyuk a korlátot nem haladja meg. Feltételezzük, hogy *(i) a tárgyak értéke függetlenek egy-*

mástól, (ii) az összsúlyt, illetve összértéket úgy kapjuk, hogy a tárgyak súlyait, illetve értékeit összegezzük.

Minden tárgy esetében két választási lehetőségünk van: elvisszük vagy nem visszük el. Ezért minden j tárgyhoz bevezetünk egy x_j döntési változót, amelynek 1 értéke azt jelenti, hogy a tárgyat elvisszük, 0 értéke pedig azt, hogy nem.

Vegyük szemügyre az $a_j x_j$ szorzatot:

$$a_j x_j = \begin{cases} a_j & \text{ha } x_j = 1 \\ 0 & \text{ha } x_j = 0 \end{cases} = \begin{cases} a_j & \text{ha elvisszük a tárgyat,} \\ 0 & \text{ha nem visszük el a tárgyat.} \end{cases} \tag{1.5}$$

Tehát mindkét esetben a szorzat akkora, amekkora súllyal a túra alatt a tárgy a hátizsákban húzza a turista vállát. a $c_j x_j$ szorzat azt a hasznot fejezi ki, amit a tárgy a túra alkalmával hajt. Ezért a turistának a következő feladatot kell megoldania:

$$\begin{array}{rll} \max & \sum_{j=1}^{n} c_j x_j & \\ & \sum_{j=1}^{n} a_j x_j & \leq \quad b \\ x_j \in \{0,1\} & j = 1, ..., n. & \end{array} \tag{1.6}$$

Az eddigi megfogalmazásból nyilvánvaló, hogy hallgatólagosan feltettük, a tárgyak súlya és értéke pozitív. Azonban formálisan nem kötöttük ki, ugyanis könnyen belátható, hogy ez a feltételezés nem jelenti az általánosság megszorítását. Hatását. Ha $a_j = c_j = 0$, akkor teljesen mindegy, hogy a változó értéke 0 vagy 1. Ha $a_j \leq 0$ és $c_j \geq 0$, akkor van olyan $\mathbf{x}^*$ optimális megoldás, amelyben $x_j^* = 1$, és ha pedig $c_j \leq 0$, akkor olyan $\mathbf{x}^*$ optimális megoldás van, hogy $x_j^* = 0$. Egyetlen eset maradt, amikor $a_j, c_j < 0$. Vezessük be x_j helyett a komplementer változóját:

$$\bar{x}_j = 1 - x_j.$$

Ezt behelyettesítve a feladatba, $\bar{x}_j$ mindkét együtthatója pozitív lesz.

A hátizsák feladattal több gyakorlati probléma is modellezhető. Tegyük fel, hogy egy pénzintézet n befektetési lehetőség közül választhat. Ismert mindegyik esetben az igényelt pénzösszeg (a_j) és a befektetés által ígért haszon (c_j), valamint a felhasználható tőke nagysága (b). Feltéve, hogy a beruházások függetlenek egymástól, a maximális hasznot hozó befektetések kiválasztását éppen a fenti (1.6) feladat írja le. Ha a javasolt projektek bizonytalanok, de becslésünk van arról, hogy milyen valószínűséggel mekkora hasznot hoznak, akkor a célfüggvényben a haszon várható értékét kell szerepeltetni. Például a legegyszerűbb esetben egy projekt p_j valószínűséggel valósul meg, de akkor a teljes beígért hasznot hozza, míg $1 - p_j$ valószínűséggel a teljes befektetett tőke elvész, akkor a célfüggvényben x_j együtthatójának c_j helyett $p_j c_j$-nek kell lennie.

Teljesen hasonló a helyzet akkor is, ha pályázati pénzek szétosztásáról van szó. Ekkor is a szétosztható keret lesz b, az egyes pályázatok által igényelt összeg a_j, míg a célfüggvény-együttható a pályázat eszmei értéke, például a bírálók által adott pontszámok összege.

A hátizsák feladatnak két fő jellemzője van: egyetlen algebrai feltétel szerepel benne, és minden együttható nemnegatív. Ezért általában valamilyen jelzővel ellátott hátizsák feladatról beszélnek akkor is, ha e két tulajdonság *egyike* teljesül.

Ilyen feladat a pénzváltási probléma, amelyről a 9. fejezetben lesz részletesen szó. Ekkor egy meghatározott pénzösszeget akarunk *pontosan* kifizetni egy adott pénzrendszer címleteivel úgy, hogy minimális számú pénzdarabot használunk fel. Ez a probléma merül fel például akkor, ha a dolgozók járandóságát készpénzben fizetjük ki, borítékolva kinek-kinek a magáét, vagy ha a pénztárgépnek automatikusan kell visszaadni az aprópénzt. Feltételezzük, hogy bármelyik címletből tetszőleges számú példányt felhasználhatunk. Legyen n a címletek száma, melyek értékei a_1, a_2, ..., a_n, a kifizetendő összeg pedig b. Ekkor a

$$\begin{array}{rlcl} \min & \sum_{j=1}^n x_j & & \\ & \sum_{j=1}^n a_j x_j & = & b \\ x_j \in \mathbb{Z}_+, & j = 1, ..., n & & \end{array} \tag{1.7}$$

speciális hátizsák feladatot kell megoldani.

A hátizsák feladat egy fontos változata az, amikor ún. általánosított felső korlátok vannak. A gyakorlatnak erre a változatra akkor van szüksége, amikor például egy szűk keresztmetszetet jelentő gépen n munkát kell elvégezni, és mindegyik munka esetében több technológia közül választhatunk, melyek a költségben és a végrehajtásukhoz szükséges időben különböznek egymástól. Legyen a j munka technológiáinak száma m_j, a j munka i technológiájának műveleti ideje, illetve költsége pedig a_{ij}, illetve c_{ij}. Legyen továbbá a gép teljes felhasználható kapacitása b. Ekkor tehát az összköltséget kell minimalizálni azon feltételek mellett, hogy *(a) a gép kapacitását nem léphetjük túl, és (b) minden munkához ki kell választani pontosan egy technológiai variánst.* Ezért x_{ij} változókat vezetünk be, ahol

$$x_{ij} = \begin{cases} 1 & \text{ha a } j \text{ munkát az } i \text{ technológia szerint gyártjuk le,} \\ 0 & \text{különben.} \end{cases}$$

Ekkor a megoldandó feladat:

$$\min \sum_{j=1}^{n} \sum_{i=1}^{m_i} c_{ij} x_{ij} \tag{1.8}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m_i} a_{ij}x_{ij} \;\leq\; b \tag{1.9}$$

$$\sum_{i=1}^{m_i} x_{ij} \;=\; 1 \quad j=1,...,n \tag{1.10}$$

$$x_{ij} \;\in\; \{\,0,\,1\,\} \quad j=1,...,n. \tag{1.11}$$

A (1.10) feltételek (1.11) segítségével biztosítják, hogy minden munkára pontosan egy technológiát válasszunk ki.

1.2.2. A halmazfedési feladat és társai

A feladat eredetileg az amerikai légitársaságoknál, a személyzet szolgálatra vezénylése kapcsán merült fel, de használható mozdonyvezetők esetében, vagy kórházi ügyeleti rend meghatározására, és még sok más helyen. Mi a feladatot a légitársaság példáján mutatjuk be.

Adott a menetrend, ami nem más, mint sok járat összessége. Egy járatot jellemző adatok a következők: az indulás és a megérkezés helye és időpontja. A társaság célja, hogy a bérköltségek minimalizálása érdekében a járatokat minél kevesebb személyzettel lássa el. Egy személyzet egy nap általában több járatot is kiszolgálhat. Azt, hogy melyek lehetnek ezek, néhány egyszerű praktikus és szakmai (jogi) szabály mondja meg. Az előbbire példa, hogy a személyzetnek azon a repülőtéren kell lenni, ahonnan a kiszolgálandó járat indul, az utóbbira pedig az, hogy a leszállás után egy bizonyos időt mindenképpen a földön kell tölteniük, nem repülhetnek azonnal tovább. Az utóbbira, a kötelező pihenésre, másutt is van példa. A mozdonyvezetők sem indulnak azonnal tovább, a kórházban is két 24 órás szolgálat között egy napnak el kell telnie.

Nevezzük tervnek a járatok egy olyan részhalmazát, amit egy személyzet ki tud szolgálni. Számos ilyen terv létezik. Egyes gyakorlati problémák esetében még a tervek számának nagyságrendjét is nehéz megbecsülni. Ha ismernénk az összes tervet, akkor a vállalat problémáját teljes egzaktsággal meg tudnánk oldani. Általában azonban csak arra van lehetőség, hogy néhány ezer vagy tízezer tervet állítsunk elő. Legyen $\mathcal{S}$ a járatok halmaza, n az összes ismert tervek száma, az egyes tervek pedig $\mathcal{T}_j$ $(j=1,...,n)$. Tudjuk tehát, hogy

$$\mathcal{T}_j \;\subset\; \mathcal{S}, \quad j=1,...,n.$$

A vállalat feladata ekkor így fogalmazható meg:

$$\begin{array}{c} \min\,|\,\mathcal{I}\,| \\ \mathcal{I} \;\subset\; \{\,1,\,2,\,\ldots\,,\,n\,\} \\ \bigcup_{j\in\mathcal{I}}\mathcal{T}_j \;=\; \mathcal{S}. \end{array} \tag{1.12}$$

Ez az alak azonban nem kényelmes optimalizáló eljárások számára, ezért a feladatot átírjuk egy ekvivalens alakba. Mivel a $\mathcal{T}_j$ terv az $\mathcal{S}$ halmaz részhalmaza, ezért egyértelműen jellemezhető az $\mathcal{S}$-re vonatkozó karakterisztikus vektorával. Az egyszerűség kedvéért tegyük fel, hogy $\mathcal{S} = \{1, ..., m\}$. Ekkor $\mathcal{T}_j$ karakterisztikus vektora $\mathbf{a}_j$, ahol

$$a_{ij} = \begin{cases} 1 & \text{ha } i \in \mathcal{T}_j, \\ 0 & \text{ha } i \notin \mathcal{T}_j. \end{cases}$$

Legyen továbbá x_j az a döntési változó, amelyik azt mondja meg, hogy a $\mathcal{T}_j$ tervet kiválasztjuk-e, vagy sem, azaz

$$x_j = \begin{cases} 1 & \text{ha a } \mathcal{T}_j \text{ tervet kiválasztjuk,} \\ 0 & \text{ha a } \mathcal{T}_j \text{ tervet nem választjuk ki.} \end{cases}$$

A kiválasztott tervek számát az x_j változók összege adja meg. Az (1.12) feladat ekvivalens alakja ekkor

$$\begin{array}{c} \min \sum_{j=1}^n x_j \\ \sum_{j=1}^n a_{ij} x_j \geq 1 \quad i = 1, ..., m \\ x_j \in \{0, 1\} \quad j = 1, ..., n. \end{array} \tag{1.13}$$

Az (1.13) problémát nevezzük *halmazfedési feladatnak*, hiszen ez épp (1.12)-vel ekvivalens, amely azt jelenti, hogy lefedünk egy véges alaphalmazt a részhalmazaival. Idővel ugyanezzel a névvel illették mindazon feladatokat, amelyekben mind a célfüggvényben, mind a jobb oldalon egynél nagyobb egészek is előfordulhatnak, de a balodalon minden együttható vagy 0, vagy 1.

Az (1.13) és vele együtt az (1.12) feladat az eredeti szolgálatra való vezénylés problémája felől egy ponton kritizálható. Mindkettő megengedi ugyanis, hogy az alaphalmaz egy elemét többszörösen is lefedjük. Egy járatot azonban csak egy személyzet szolgálhat ki. Erre a felvetésre kétféle válasz adható. Az egyik, hogy meghagyjuk a feltételek ilyen lazaságát, hogy az egyenlőtlenségek engedte nagyobb szabadságot kihasználva minél jobb optimumértéket érjünk el, és az esetleges többszörös fedés kérdését esetileg kezeljük. Például lehet, hogy elegendő az egyik tervből egyszerűen kihagyni a kérdéses járatot. A másik, hogy matematikai feltételként írjuk elő a pontos egyenlőséget, ami az (1.12) feladat nyelvén azt jelenti, hogy még megköveteljük:

$$\mathcal{T}_{j_1} \cap \mathcal{T}_{j_2} = \emptyset, \quad j_1, j_2 \in \mathcal{I}, \; j_1 \neq j_2.$$

Ez csak annyi változást okoz az (1.13) feladatban, hogy az egyenlőtlenséget egyenlőségre kell cserélni, azaz a feladat alakja

$$\begin{array}{c} \min \sum_{j=1}^n x_j \\ \sum_{j=1}^n a_{ij} x_j = 1 \quad i = 1, ..., m \\ x_j \in \{0, 1\} \quad j = 1, ..., n. \end{array} \tag{1.14}$$

Itt valójában nem lefedjük az alaphalmazt, hanem részekre bontjuk, ezért (1.14)-et *halmazfelbontási feladatnak* hívjuk.

A feladatcsalád harmadik tagja a *halmazkitöltési feladat*, amelynek esetében egy véges alaphalmazba egy előre adott készletből minél több részhalmazát akarjuk beletenni. Ha mind a halmazfedési, mind a halmazkitöltési feladat esetében az általánosabb esetet tekintjük, amikor az egyes elemek többszörös felhasználása is megengedett, akkor a két feladat ekvivalens egymással, ami egyszerűen úgy látható be, hogy bármelyik feladatból indulunk is ki, véve az összes változó komplementerét, ami a feladat egy ekvivalens átalakítása, a másik feladathoz jutunk.

1.2.3. Általános lineáris egészértékű programozási feladatok

Egyes esetekben előfordulhat, hogy a vizsgált probléma természetéből adódóan az együtthatók között negatívak is vannak, vagy nem indokolt valamennyi változó esetében az egészértékűség feltételezése. Az első esetben általános tiszta egészértékű feladatról beszélünk. Alakja a következő:

$$\max \sum_{j=1}^{n} c_j x_j \tag{1.15}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, ..., m \tag{1.16}$$

$$x_j \ \ (1.1) \text{ vagy } (1.2) \text{ vagy } (1.3) \text{ típusú } \ j = 1, ..., n.$$

Itt már az együtthatók előjelére és értékére semmiféle megszorítást nem teszünk. Ilyen feladathoz jutunk az (1.1) típusú változók mellett, ha egymással összefüggő beruházások sorsáról kell dönteni. Ekkor az x_j változó 1 értéke a j beruházás megvalósítását, 0 értéke elejtését írja elő. A beruházások együtt is csak korlátos mennyiséget használhatnak az erőforrásokból. Az utóbbiakból rendelkezésre álló mennyiségeket a b_i számok adják meg. A j beruházás igénye az i erőforrásból a_{ij}. Ha $a_{ij} < 0$, akkor a j beruházás termeli az i erőforrást.

Egy másik probléma, ami ilyen feladatra vezet, egy rövid sorozatokat gyártó üzem optimális termékösszetételének meghatározása. Itt egy változó egy termékre mondja meg, hogy hány darabot kell belőle gyártani, így (1.2) típusú lesz. A feltételek ismét az erőforrásokra vonatkoznak. (Itt erőforrás a fontosabb berendezések kapacitása, a beszerezhető alapanyag mennyisége stb.)

Ha bizonyos változók tetszőleges nemnegatív értéket vehetnek fel, akkor a feladat alakja:

$$\begin{array}{lllll}
\max & \sum_{j=1}^{n} c_j x_j & + & \sum_{k=1}^{p} d_k y_k & & \\
 & \sum_{j=1}^{n} a_{ij} x_j & + & \sum_{k=1}^{p} h_{ik} y_k & \leq \; b_i & i = 1, ..., m \\
 & y_k \geq 0 \quad k = 1, ..., p & & & & \\
 & x_j \;\; \text{(1.1) vagy (1.2) vagy (1.3) típusú} \;\; j = 1, ..., n, & & & &
\end{array}$$

ahol tehát az x_j-kkel ellentétben az y_k-k folytonosan változnak. Az ilyen problémákat nevezik vegyes egészértékű feladatoknak.

Például az energiarendszer irányításakor a meghatározandó mennyiségek közé tartozik egy-egy erőművi berendezés által előállított energia. Egy gép vagy egyáltalán nem termel áramot, vagy az általa előállított energia egy pozitív alsó és felső korlát közé esik, vagyis a termelt energia

$$\{0\} \cup \{l, u\}$$

halmazba esik, ahol $0 < l < u$. Ennek a viszonynak a leírására két változóra van szükségünk. Az első, x, egy $0 - 1$ változó, amely azt mondja meg, hogy egyáltalán működik-e a berendezés, a második, y pedig azt, hogy mennyi az általa termelt energia. Ezt a következő feltételekkel érhetjük el:

$$y \geq lx, \quad y \leq ux, \quad x \in \{0, 1\}.$$

Mind a tiszta, mind a vegyes egészértékű programozási feladat előfordul egyenlőség típusú feltételek mellett is.

1.2.4. Az utazó ügynök feladat és alkalmazásai

Adott n város. Jelöljön ezek közül tetszőleges két különbözőt i és j. Ismert ezek c_{ij} távolsága. Egy kereskedelmi ügynöknek kell egy adott városból kiindulva valamennyit bejárni úgy, hogy mindegyiket pontosan egyszer keresse fel, és útját az induló pontban fejezze be, a megtett távolság pedig minimális legyen.

A feladatnak rendkívül különböző területeken számos alkalmazása ismert. Elsőként kell említeni a közvetlen szállítási problémákat. Ha egy járműnek több címet is fel kell keresnie, akkor optimális útvonalának meghatározását éppen az utazó ügynök feladata írja le.

Ha egy numerikusan vezérelt megmunkáló gép több lyukat is fúr ugyanabba a munkadarabba, akkor a gép lyukak közti mozgásának idejét, ami bizonyos értelemben veszteségidő, célszerű minimalizálni. Hasonlóképp a chipgyártásban a szilíciumdarabkát lézersugárral munkálják meg (égetik ki). Egy lapocskán akár több millió pont is lehet, amit a sugárnak fel kell keresnie. Nem mindegy, hogy ezt mekkora távolság megtételével teszi meg, mert ettől függ a gyártás hatékonysága.

Egy még absztraktabb, de a gyakorlatban gyakran előforduló alkalmazás a következő. Egy gépen különböző munkadarabokat gyártunk. Az egyes munkadarabok között a gépet át kell állítani az új típus gyártására. Az ehhez szükséges időt, illetve költséget nevezik átszerszámozási időnek, illetve költségnek. Az átszerszámozási idő ugyanúgy veszteség, mint a fúrógép áthaladása egyik pontról a másikra. Ezért az idők összegét minimalizálni kell.

A fenti példák rávilágítanak arra, hogy az alkalmazások során nem mindegy, hogy a távolságot miben mérjük. Vegyük először a gépkocsi esetét. Tegyük fel, hogy az valamilyen terméket szállít egy nagyvárosban. Vannak olyan dolgok, amiket hajnalban szállítanunk, amikor az utcák még üresek. Ilyen például az újság. Ekkor két pontnak a városi úthálózat megszabta geometriai távolsága és az ezen út megtételéhez szükséges idő igen erős korrelációban van egymással, gyakorlatilag mindegy, hogy miben mérünk. Más termékek szállítása azonban szükségképpen napközben történik ezért a zsúfolt szakaszok megtételéhez szükséges idő nem arányos a geometriai távolsággal.

Más összefüggésre világít rá a chipgyártás és a fúrógép összehasonlítása. A lézersugár esetében két pont között a távolság az a szög, amennyit a sugárnak el kell fordulnia ahhoz, hogy az egyik pontból a másikba jusson. Ez a fajta távolság rendelkezik a megszokott geometriai tulajdonságokkal, mint a szimmetricitás és a háromszög-egyenlőtlenség. Az utóbbi nem feltétlenül igaz a fúrógép esetében. Itt ugyanis az egyik pontból a másik pontba való átmenet három részből áll: először a frissen elkészült lyukból ki kell húzni függőlegesen a fúrót, méghozzá olyan magasra, hogy a fúró a felület fölött biztonságosan elmehessen, ezután megy a másik pont fölé, majd ott leereszkedik. Mivel a felület gépalkatrészek esetében nem egyetlen sík, ezért a függőleges mozgások pontpárról pontpárra változnak, ami elrontja a háromszög-egyenlőtlenséget. Az átszerszámozási idők esetében még a szimmetricitás sem lesz igaz. Például, ha egy A munkadarabhoz szükséges valamennyi szerszám kell a B munkadarabhoz is, de fordítva nem, akkor a BA átszerszámozási idő 0 lesz, míg az AB pozitív. A feladat számítástechnikai nehézsége függ attól, hogy az ismert geometriai tulajdonságok fennállnak-e.

Az alapmodellt több különböző módon is kiegészítették további feltételekkel.

A gyakorlatban fontos eset az, amikor egyetlen telephelyről szállítunk ki nagyobb mennyiségű árut. A nagyobb mennyiség azt jelenti, hogy az összes szállítandó mennyiség meghaladja a használt gépkocsi kapacitását. Fel kell tehát osztani a műveletet kis körutakra. Legyen g a gépkocsi kapacitása, az érintendő pontok, vagyis a városok $\{0, 1, ..., n\}$, ahol 0 az a telephely, ahonnan a kiszállítás történik, q_j a j városba szállítandó mennyiség, m a körjáratok száma, $\mathcal{P}_1, ..., \mathcal{P}_m$ pedig a körjáratokhoz tartozó városok halmaza. Végül jelölje a $\mathcal{P}_i$ halmazon az utazó ügynök feladat optimális megoldásának értékét $t(\mathcal{P}_i)$. Feltételezve, hogy a szokásos geometriai tulajdonságok teljesülnek, és minden j esetén az oda szállítandó áru mennyisége nem haladja meg a gépkocsi kapacitását, azaz

$$q_j \;\le\; g \quad j = 1, ..., n,$$

minden várost elegendő egyszer felkeresni. Ezért megkövetelhetjük a

$$\forall\, 1 \le i < k \le m: \quad \mathcal{P}_i \cap \mathcal{P}_k \;=\; \{\, 0 \,\} \tag{1.17}$$

feltételt. Minden várost fel kell keresnünk, tehát teljesülnie kell az

$$\bigcup_{i=1}^{m} \mathcal{P}_i \;=\; \{\, 0,\, 1,\, \ldots\,, m \,\} \tag{1.18}$$

egyenlőségnek. A megoldandó feladat tehát az (1.17) és (1.18) feltétel mellett a

$$\min \sum_{i=1}^{m} t(\mathcal{P}_i) \tag{1.19}$$

célfüggvény optimalizálása.

1.3. A feladatok osztályozása

A fentiekben a lineáris tiszta egészértékű feladatok hat alapesetét különböztettük meg részint a változók három, részint a feltételek két típusa (egyenlet és egyenlőtlenség) szerint. Egész együtthatókat feltételezve leírjuk a hat feladat feladat kapcsolatát.

Egy

$$\mathbf{a}^T \mathbf{x} \;=\; b$$

alakú egyenlet a változók típusától és a többi feltételtől függetlenül mindig helyettesíthető a következő két egyenlőtlenséggel:

$$\begin{array}{rcr} \mathbf{a}^T\mathbf{x} & \le & b \\ -\mathbf{a}^T\mathbf{x} & \ge & -b. \end{array}$$

Az (1.1) típusú változók az (1.2) típusúak speciális esetei. Érdekes azonban az, hogy az utóbbiak is visszavezethetők az előbbiekre. Ha egy x változótól azt követeljük meg, hogy 0 és $d > 0$ közé essék, akkor x helyettesíthető néhány bináris változóval. Legyen ugyanis

$$k \;=\; \lfloor \log_2 d \rfloor.$$

Ekkor írjuk fel x-et kettes számrendszerben, és a számjegyei legyenek az ismeretlenek, tehát az

$$x = \sum_{p=0}^{k} 2^p y_p$$

helyettesítést végezzük el. Meg kell követelni természetesen a

$$\sum_{p=0}^{k} 2^p \tag{1.20}$$

feltételeket.

Most már könnyen beláthatjuk, hogy az egyenlőtlenségek visszavezethetők egyenletekre. Tekintsük az

$$\mathbf{a}^T \mathbf{x} \le b$$

feltételt. Egy új y nemnegatív egész változó bevezetésével nyerjük az

$$\mathbf{a}^T \mathbf{x} + y = b \tag{1.21}$$

egyenletet. Ezzel az (1.3) esetben a visszavezetés meg is történt. Az (1.2) esetben ehhez azt kell még észrevenni, hogy az $\mathbf{a}^T \mathbf{x}$ lehetséges értékei korlátos halmazt alkotnak, így az (1.21) megoldásaiban y értékei is szükségképpen korlátosak. Az (1.1) esetben is hasonlóan járunk el, csak az így bevezetett y változót még bináris változókkal ki kell váltani.

Tehát azt kaptuk, hogy bármely nemkorlátos változókat tartalmazó feladatra visszavezethető mind a négy korlátos változókat tartalmazó feladat. Az utóbbiak pedig egymás közt tetszőlegesen transzformálhatók. A visszavezetés értelme, hogy ha van módszerünk egy olyan feladat megoldására, amire a visszavezetés történt, akkor ez a módszer felhasználható a visszavezetés előtti feladat megoldására is.

A gyakorlatban általában ezeket a transzformációkat nem érdemes elvégezni. Ennek két oka van. Az egyik, hogy a feladat mérete jelentősen megnőhet, ami megnehezíti mind a feladat tárolását, mind a megoldását. Egy extrém példa: induljunk ki (1.2) típusú változókat tartalmazó, egyenletekkel korlátozott feladatból, és vezessük vissza bináris változós, egyenlőtlenségekkel leírt feladattá. Az első lépésben a feltételek száma megduplázódik, mert minden egyenletet két egyenlőtlenséggel helyettesítünk. Ezután be kell vezetnünk a további (1.20) alakú feltételeket és az ezekben szereplő új változókat. Ha eredetileg m feltételünk és n változónk volt és minden változó korlátja csak 2 vagy 3, akkor is a transzformált feladat mérete

$$(2m + n) \times 2n.$$

A másik ok, ami miatt a visszavezetéseket nem célszerű végrehajtani, hogy az egyenlőtlenségekkel, illetve egyenletekkel leírt, a bináris, illetve nemcsak bináris vektorokat tartalmazó halmazoknak más és más matematikai tulajdonságaik vannak. Ezen eltérések leírására a következő fejezetben részletesen visszatérünk. Az egyes alapfeladatokra kifejlesztett módszerek – többnyire implicit módon – felhasználják a stratégiájuk meghatározásánál ezeket a tulajdonságokat abban az értelemben, hogy bizonyos helyzetek hatékony kezelésére felkészülnek. A transzformált feladatok esetében azonban éppen az eltérő jellegzetességek miatt ez a hatékonyság nem tud érvényesülni.

1.4. Megjegyzések és irodalom

Az egészértékű programozásban a szóhasználat eltérően alakult a lineáris programozáshoz viszonyítva. Itt *megoldás* alatt azokat a vektorokat értjük, melyek a megfelelő egészértékűségi követelménynek eleget tesznek, függetlenül attól, hogy az algebrai feltételeket kielégítik-e vagy sem. Ennek megfelelően az (1.1), illetve az (1.2) típusú változóknál az összes megoldások száma

$$2^n, \quad \text{illetve} \quad \prod_{j=1}^{n}(d_j+1).$$

A *megengedett megoldás* kifejezéssel azokat a vektorokat illetjük, amelyek mind az egészértékűségi, mind az algebrai követelményeket kielégítik.

Az egészértékű programozás irodalma beláthatatlanul széles. Dantzig, a szimplex módszer megtalálója, már 1957-ben írt ilyen feladatokról [24]. A hazai irodalomban az első két öszefoglaló mű [35] és [56]. Az ütemezéselméletet is tárgyalja [93]. Az utazó ügynök feladatának önmagában is igen nagy irodalma van. Jó összefoglalás [59]. Egyetlen árunak több körjárattal történő kiszállítása először a [12] dolgozatban merült fel a Dagens Nyheter svéd napilap stockholmi kiszállítása kapcsán. Később ugyanez a feladat a budapesti sörszállítás optimalizálása során is előjött. Az utazó ügynök feladat ilyen irányú felhasználása annyira elterjedt, hogy már a nem operációkutatási szakirodalom is ezt ajánlja (ld. pl. [47]). A berlini mozgássérültek szállításának optimalizálásában is felhasználták a feladatot.