BOOLE PROGRAMOZÁS GRÁFOK SEGÍTSÉGÉVEL[1]

NAGY BENEDEK
Debreceni Egyetem, Matematikai és Informatikai Intézet

Olyan speciális egészértékű programozási feladatokat vizsgálunk, amelyekben minden változó a $\{0,1\}$ értékhalmazból vehet fel értéket. Megismerkedünk az ún. alap illetve a módosított Boole-programozási feladattal. Az alap Boole-programozási feladatot speciális lineáris programozási feladattá alakítjuk, gráffal reprezentáljuk. Adunk egy algoritmust, amiben a feladatot gráfok segítségével oldjuk meg. A módosított feladat esetén ez a speciális lineáris programozási feladat az eredeti feladatnál gyengébb feltétel rendszert jelent. Ezt a gráfoknál az ún. releváns élek segítségével vesszük figyelembe. A módosított Boole-programozási feladatot az alapfeladathoz hasonló módszerrel oldhatjuk meg.

1 Bevezetés

Vannak olyan speciális egészértékű programozási feladatok, amelyekben a változók csak a 0, illetve az 1 értéket vehetik fel. Ilyenekkel találkozhatunk például a szakirodalomban (pl. [9]) hátizsák feladatokként ismert példákban akkor, amikor pl. egyes tárgyakról azt kell eldönteni, hogy benne legyenek-e egy kiválasztandó halmazban, vagy ne, valamint logikai feladatoknál is.

Ebben a cikkben bemutatjuk az ún. alap Boole-programozási feladatot, amit speciális lineáris programozási feladattá tudunk alakítani, illetve gráfok segítségével tudunk reprezentálni. A gráfokat átalakíthatjuk oly módon, hogy közben a megoldáshalmaz ne változzon. Felhasználva, hogy csak a $\{0,1\}$ értékhalmazból vehetnek fel értéket a változók, a gráf egyes lokális tulajdonságaiból következtethetünk az egyes csúcsok értékeire. (Ezeket az átalakítási, illetve csúcsmeghatározó lépéseket fogjuk lokális gráflépéseknek hívni.) Adunk egy algoritmust, aminek segítségével egy alap Boole-programozási feladat összes megengedett megoldása meghatározható. Ezután megismerkedünk a módosított feladattal, aminek feltételei közt egyenlőségek is előfordulhatnak. Ezt a feladatot nem tudjuk tisztán lineáris feltételekkel leírni, de tudunk hozzá olyan gráfreprezentációt készíteni, amely hűen tükrözi a feladatot. Az alap feladathoz tartozó algoritmust megfelelően kiegészítve olyan algoritmust kapunk, amely segítségével a módosított feladat megengedett megoldásait is meghatározhatjuk.

[1]Beérkezett: 2002. május 9. e-mail: nbenedek@math.klte.hu.

2 Boole-programozási feladatok

Amint azt a bevezetőben is írtuk, e cikkben olyan feladatokról lesz szó, amelyekben minden változó csak a $\{0,1\}$ értékhalmazból vehet fel értéket. Most pontosan definiáljuk, hogy milyen feltételrendszer esetén hogyan nevezzük a feladatot.

2.1. Jelölés. A változókat az ábécé nagybetűivel fogjuk jelölni, konkrét példákban más-más betűvel, egyébként az ábécé elejéről vett valamely betűt fogjuk indexelni.

2.1. Definíció. A (*) alakú feltételrendszerrel adott feladatokat alap Boole-programozási feladatoknak nevezzük, ha benne minden B_i változó legfeljebb egy (*) feltétel bal oldalán szerepel

$$(*) \qquad B_i \leq B_{r1} \cdot B_{r2} \cdot \ldots \cdot B_{rm} \cdot (1 - B_{p1}) \cdot (1 - B_{p2}) \cdot \ldots \cdot (1 - B_{pk}) \, .$$

A jobb oldali változók előfordulásaira nincs megkötés, azon kívül, hogy a szorzatok legalább egytagúak. Keressük azokat a lehetséges B_i értékeket, amelyekre a feltételrendszer teljesül.

2.2. Definíció. A következő alakú feltételekkel megadott feladatokat módosított Boole-programozási feladatoknak nevezzük:

$$(*) \qquad B_i \leq B_{r1} \cdot B_{r2} \cdot \ldots \cdot B_{rm} \cdot (1 - B_{p1}) \cdot (1 - B_{p2}) \cdot \ldots \cdot (1 - B_{pk})$$

vagy

$$(**) \qquad B_i = B_{r1} \cdot B_{r2} \cdot \ldots \cdot B_{rm} \cdot (1 - B_{p1}) \cdot (1 - B_{p2}) \cdot \ldots \cdot (1 - B_{pk}) \, ,$$

ahol minden B_i maximum egy feltétel bal oldalán szerpel. (A jobb oldali B változó előfordulásokra nincs megkötés, azon kívül, hogy a szorzat legalább egy tagú.)

Láthatjuk, hogy a módosított feladat abban különbözik az alap feladattól, hogy egyenlőség is szerepel a feltételek között.

2.1. Megjegyzés. Az alap Boole-programozási feladat feltételeit mindig kielégíti az azonosan 0 vektor (triviális megoldás). Ugyanez nem mondható el a módosított feladat esetén, aminek nem biztos, hogy van megoldása. (A legegyszerűbb példa az egyváltozós $A = 1 - A$, ami nem megoldható a 0,1 halmazon.)

2.1. Lemma. Az alap Boole-programozási feladatokat átírhatjuk a következő formába

$$(***) \qquad \begin{aligned} B_i &\leq B_{rj} \\ B_i &\leq 1 - B_{pk} \end{aligned}$$

Bizonyítás. A (*) formulában a jobb oldali szorzatot tényezőkre bontva láthatjuk, hogy minden tényező a $\{0,1\}$ halmazból vehet fel értéket. Ha

$B_i = 0$, akkor triviálisan a szorzat minden tagjára külön felírhatjuk a (***) összefüggések közül az aktuálisat. Ha $B_i = 1$, akkor viszont a jobb oldali szorzatnak is 1-nek kell lennie, tehát minden tagja 1, vagyis a megfelelő (***) összefüggések fennállnak. Az is világos, hogy amennyiben a (***) összefüggések egy adott B_i-re teljesülnek, akkor az ezekből összekombinált (*) is igaz. Tehát az alapfeladat feltételrendszerével ekvivalens feltételrendszert kaptunk.

A 2.1 lemma (***) feltételrendszere tulajdonképpen egy speciális lineáris 0-1 programozási feladat feltételrendszere. Az első típusú összefüggések két változó különbségének előjelét szabják meg, míg a második két változó összegének a maximumát korlátozza. Tehát az alap Boole-programozási feladat megoldásai éppen ennek a lineáris programozási feladatnak a megengedett megoldásai.

2.3. Definíció. Atomi feltételnek hívjuk a (***) feltételeit. Ezekben a reláció mindkét oldalán pontosan egy darab változó szerepel.

A módosított feladat esetén a megfelelő (***) összefüggéseknek ugyancsak fenn kell állniuk, de ezen feltételek csak szükségesek, nem adnak az eredeti feladattal ekvivalens feltételrendszert. Tehát itt csak közelíteni tudjuk az eredeti feladatot a lineáris feltételrendszerrel, de ennek ellenére a következő fejezetben a módosított feladatokra is egzakt megoldási módszert fogunk adni.

2.2. Megjegyzés. Ha olyan (*) feltételrendszerünk van, amelyben van olyan B_i, amely több feltétel bal oldalán is szerepel, akkor ezen feltételek jobb oldalait összeszorozva az eredetivel ekvivalens (ez a 2.1 lemmából következik), és a fenti definícióknak megfelelő Boole-programozási feladatot kapunk.

Az előbbi megjegyzés alapján tehát elég a 2.1 és 2.2 definíciókban szereplő Boole-programozási feladatok reprezentálására és összes megengedett megoldásának előállítására szorítkoznunk. A módszerünk működik azokban az esetekben is, ha a (*) alakú feltételrendszerre eredetileg nem teljesül az, hogy minden változó legfeljebb egyszer szerepel a bal oldalon, de a feltételrendszer átalakítható ilyenre.

2.4. Definíció. A P és a Q Boole-programozási feladatokat ekvivalensnek nevezünk, ha P feladat megoldása(i) ugyanaz(ok), mint a Q megoldása(i).

3 Boole-programozási feladatok gráfreprezentációja

Ebben a részben először definiáljuk azt az eszközrendszert, amit Boole-programozási feladatok megoldására fogunk használni. Ezután a megfelelő gráflépéseket alkalmazva alakítjuk a feladat gráfját vele ekvivalens feladatok gráfjává, haladva a megoldás felé. Először az alapfeladatot vizsgáljuk meg.

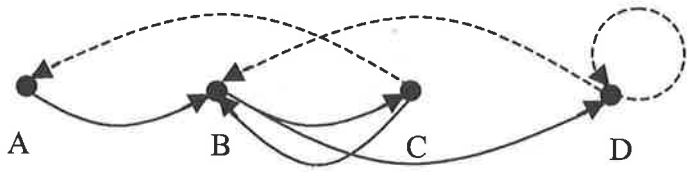
3.1 Gráfreprezentáció az alapfeladat megoldásához

3.1. Definíció. Egy Boole-programozási feladat gráfreprezentációján a következő irányított gráfot értjük. Vegyük a feladattal ekvivalens (***) alakú feltételrendszert. A gráf csúcsai legyenek a feladatban szereplő változók. Az éleket pedig kétféle nyíllal jelöljük: a B_i-t összekötjük a B_j-vel folytonos nyíllal, ha $B_i \leq B_j$ fennáll, és szaggatott nyíllal, ha $B_i \leq (1 - B_j)$. A nyíl a megfelelő összefüggés bal oldalán szereplő változótól a feltétel jobb oldalán szereplő felé mutat.

Tekintsünk egy példát:

3.1. Példa. (A változókat jelölje A, B, C és D.)
$A \leq B$
$B \leq C \cdot D$
$C \leq B \cdot (1 - A)$
$D \leq (1 - B) \cdot (1 - D)$.

A példa gráfja:



A példa megoldását az alfejezet végén mutatjuk be.

Nézzük meg, mit jelentenek a gráf élei. Ha az A csúcsból folytonos nyíl mutat a B csúcsba, akkor az $A \leq B$ relációnak kell teljesülnie, vagyis nem lehet az, hogy $A = 1$ és $B = 0$. Ha a gráfban szaggatott él mutat A-ból B-be, akkor ez $A \leq 1 - B$-t jelent. Ez esetben nem lehet $A = B = 1$. Egyszerű átalakítással láthatjuk, hogy két csúcs közti szaggatott nyíl jelentése független az iránytól. ($A \leq 1 - B$ ugyanakkor teljesül, amikor $A + B \leq 1$, illetve $B \leq 1 - A$.) Ezért az alapfeladat megoldásakor a gráfban a szaggatott éleket a továbbiakban nem tekintjük irányítottnak.

3.1. Megjegyzés. A gráf a feladat atomi feltételeit reprezentálja.

Most nézzük, hogyan aknázhatjuk ki a gráf egyes részeiben rejlő információt, melyek azok a lépések, amelyek segítségével a gráfban szereplő lokális információkat felhasználhatjuk új élek berajzolására, illetve a csúcsok értékének meghatározására.

3.2. Definíció. Élhozzáadó lépésnek nevezzük a gráf átalakítását oly módon, hogy két még nem ismert értékű változót reprezentáló csúcs közé új, eddig még a gráfban nem szereplő élt húzunk be, miközben az új gráf ekvivalens az eredetivel.

3.1. Lemma. a) Ha a gráfban vezet egy A csúcsból a C csúcsba folytonos élekből álló irányított út, akkor az $A \longrightarrow C$ él behúzása élhozzáadó lépés.

b) Ha a gráfban vezet egy A csúcsból egy B csúcsba folytonos élekből álló irányított út és a B csúcsot szaggatott él köti össze a C csúccsal, akkor az $A\text{- - -}C$ élhozzáadó lépés.

Bizonyítás. Ha (D_1, D_2), (D_2, D_3), ..., (D_{k-1}, D_k) a gráfban folytonos élekből álló irányított út, akkor a folytonos él definíciója szerint

$$D_1 \leq D_2 \leq \ldots \leq D_k \,,$$

$D_1 = A$ és $D_k = C$ mellett az a) állítás azonnal adódik. $D_1 = A$ és $D_k = B$ mellett $B \leq 1 - C$ miatt a b) állítás következik.

Most következzenek az ún. alapsémák. Ezek olyan élkombinációk, melyek segítségével meghatározható valamelyik benne szereplő csúcs értéke, anélkül, hogy bármelyik csúcs értékét tudtuk volna korábban.

3.2. Lemma („egyelemű alapséma"). Ha feladat tartalmaz szaggatott hurokélt, vagyis van olyan A csúcs, hogy $A\text{- - -}A$, akkor $A = 0$.

Bizonyítás. Feltéve, hogy $A = 1$ ellentmondást kapunk. $A = 0$-ra pedig teljesül az él által jelentett feltétel.

Ha egy feladat gráfjában egy csúcsra alkalmazhatjuk a fenti alapsémát (eredetileg volt benne ilyen él, vagy valamely élhozzáadó lépés során ilyen él került a gráfba), akkor az adott csúcshoz beírjuk a 0 értéket. Most mutatunk olyan alapsémákat, amelyekből hasonlóan nyerhetünk információt. Ezeket a megoldás gyorsítása érdekében használhatjuk, de élhozzáadó lépés használatával visszavezethetők az előző alapsémára.

3.3. Lemma („kételemű alapséma"). Ha a feladat gráfjában az A és a B csúcs közt van szaggatott él, és az A-ból a B-be vezet folytonos nyilakból álló irányított út, akkor az A értéke 0.

Bizonyítás. Élhozzáadásokkal (3.1 lemma a) az A és B közti irányított út miatt $A \longrightarrow B$ él behúzható, majd $A \longrightarrow B\text{- - -}A$-ra alkalmazva a b) élhozzáadást, szaggatott hurokélt kapunk A-nál, ami a 3.2 lemma alapján éppen az állításunkat jelenti.

3.4. Lemma („háromelemű alapséma"). Ha van olyan A, B és C csúcs, hogy A-ból B-be és C-be is vezet folytonos élekből álló irányított út, a B és a C között pedig szaggatott él van, akkor A értéke 0.

Bizonyítás. Először az irányított utakon alkalmazott élhozzáadásokkal megkapjuk az $A \longrightarrow B$ és $A \longrightarrow C$ éleket, majd élhozzáadás $A \longrightarrow B\text{- - -}C$, majd $A \longrightarrow C\text{- - -}A$-hoz. Ekkor az A-nál szaggatott hurokélt kapunk, ami a 3.2 lemma miatt éppen $A = 0$-t jelent.

Még egy lépést sorolunk fel itt, bár a következő sémából nem derül ki egyértelműen egyik csúcs értéke sem, használata mégis közelebb visz minket a megoldáshoz.

3.5. Lemma („egyenlősítő séma"). Ha egy gráfban két csúcs között (legyen ez A és B) mindkét irányba vezet folytonos nyíl, akkor értékük megegyezik.

Bizonyítás. A nyilak jelentéséből nyilvánvaló: $A \leq B$, $B \leq A$.

3.6. Lemma. Az egyenlősítő séma egy ekvivalenciarelációt jelent a gráf csúcsai között.

Bizonyítás. Evidens, hogy tetszőleges A csúcsra teljesül $A \leq A$ (a folytonos hurok-nyilat bármely csúcshoz felvehetnénk, ha segítene a megoldásban), a szimmetria az előző lemma (az egyenlősítés definíciója) miatt triviális. A tranzitivitás pedig: ha A és B közt minkét irányba vezet folytonos nyíl, a B és C közt ugyancsak, akkor élhozzáadásokkal A és C közti mindkét irányú nyilat megkaphatjuk.

3.1. Jelölés. Az egyenlősítő sémát úgy fogjuk felhasználni a gráfban, hogy a megfelelő csúcsokat egymás mellé, egy téglalapba írjuk, ezzel jelezve, hogy értékük megegyezik.

Most megnézzük, hogy ha egy él egyik végpontjában levő csúcs értéke már ismert, akkor ebből milyen további információt vonhatunk le, ezzel közeledve a megoldáshoz.

3.3. Definíció. Egy feladat gráfjában kiértékelő nyílnak hívunk egy élt akkor, ha valamelyik végpontjában levő változó értéke már ismert, és ez alapján meg tudjuk mondani a másik végpontban levő változó értékét is.

3.7. Lemma. Egy alap Boole-programozási feladat gráfjában a következő esetek kiértékelő nyilak.

a) $1 \longrightarrow A$ esetén $A = 1$;

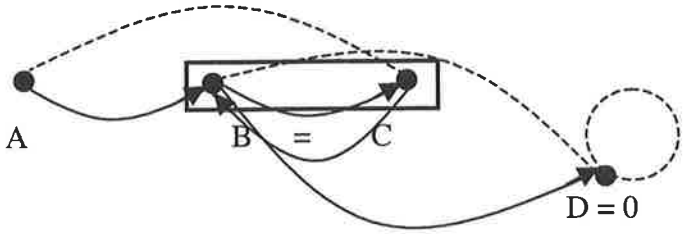
b) $1 \text{- - -} A$ esetén $A = 0$;

c) $A \longrightarrow 0$ esetén $A = 0$.

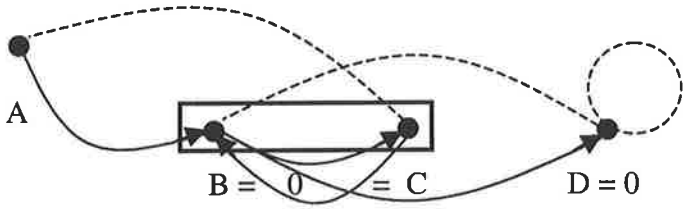
Bizonyítás. A nyilak jelentése alapján nyilvánvaló.

Tulajdonképpen ezzel felsoroltuk az összes lehetséges gráf-manipulációs lépést, amit egy alap Boole-programozási feladat megoldásához használhatunk. Nézzük most meg a 3.1 példa megoldását. (Technikailag pl. a gráf csúcsait egyvonalba rajzoljuk, a szaggatott éleket felül, a folytonosakat alul rajzoljuk be. Ha egy csúcs értéke 0, akkor ez alá a vonal alá rajzoljuk, ha értéke 1, akkor e vonal fölé. Az egy téglalapban levő csúcsokat együtt mozgathatjuk.)

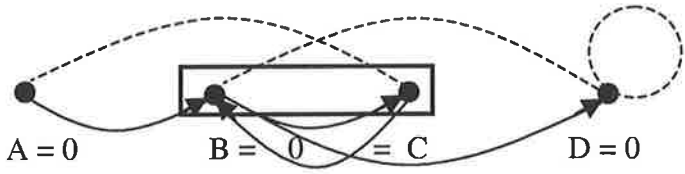
A példa gráfja tartalmazza az egyelemű alapsémát D-nél, a B és C csúcsok pedig egyenértékűek:



A $B \longrightarrow D = 0$ él kiértékelőnyíl:



Az $A \longrightarrow B = 0$ él kiértékelőnyíl:



Tehát a megoldás az, hogy $A = B = C = D = 0$.

Most előbb megnézzük, hogyan működik mindez a módosított feladat esetén, majd egy algoritmust adunk, amivel mindkétféle Boole-programozási feladat estén megkaphatjuk a megengedett megoldásokat.

3.2 Gráfmanipulációs lépések a módosított feladatnál

A módosított feladat esetén is a 3.1 definícióban adott gráfot fogjuk használni, egy kis különbséggel, ugyanis itt valahogy figyelembe kell vennünk azokat a feltételeket, amelyekben a $\leq$ reláció helyett $=$ szerepel, ezt az ún. releváns él fogalmának bevezetésével tesszük meg.

Ehhez vizsgáljuk meg közelebbről, hogy az egyenlőségek a (**) feltételek között mit jelentenek. Ha a bal oldali változó értéke 1, akkor a jobb oldalon a szorzat minden tényezőjének 1-nek kell lennie, ez esetben a (***)-beli $\leq$ relációkkal az eredetivel ekvivalens a feltételrendszer. Ha a bal oldali változó

értéke 0, akkor egyenlőség esetén a jobb oldalon a szorzatban lennie kell legalább egy 0 értéknek. (A (***)-beli atomi feltételek megengednék azt az esetet is, amikor a szorzat minden tényezője 1.)

3.4. Definíció. Egy élt relevánsnak hívunk, ha elképzelhető, hogy 0 értékű csúcsból indul, és az eredeti (**) feltételrendszerből egyenlőséget fejez ki. (Vagyis, lehetséges, hogy $0 = 0$ alakú egyenlőséget fejez ki, olyan feltétel esetén, ahol eredetileg egyenlőség szerepelt a (**) alakban.)

Eredetileg, amikor megrajzoljuk egy módosított feladat gráfját, akkor benne minden olyan csúcsból induló élt relevánsnak tekintünk, amely eredetileg egyenlőség bal oldalán állt (**)-ban. Világos, hogy a megoldás során minden olyan, 0 értékű csúcsból indulnia kell mindig legalább egy releváns élnek, ahol az adott változó egy egyenlőség bal oldalán állt, ugyanis a jobb oldali szorzat pontosan akkor 0, ha van a tényezői közt 0, és a releváns él éppen ezt jelenti.

3.2. Megjegyzés. A módosított feladat esetén a relevancia miatt nem tekinthetjük a szaggatott éleket minden esetben irányítatlannak. A nem releváns szaggatott éleket továbbra is kezelhetjük irányítás nélkül, de a relevánsoknál az irány is fontos szerepet játszik.

3.2. Jelölés. A nem releváns éleket át fogjuk húzni, és !-lel jelöljük a csúcsnál, ha belőle egyetlen releváns él indul.

A releváns élekkel tudjuk a feltételek közt szereplő egyenlőségeket biztosítani az egyébként $\leq$ relációkat jelentő élek közt, amit a következő lemmában meg is mutatunk.

3.8. Lemma. Ha egy csúcsból egyetlen releváns él indul, akkor ez az él egyenlőséget fejez ki.

Bizonyítás. Tegyük fel, hogy A-ból egyetlen releváns él indul. Ekkor ez az egyetlen olyan él, amely az $A = 0$ esetben egyenlőséget fejezhet ki, tehát ha az eredeti feltételek teljesülnek, akkor ennek kötelező egyenlőséget jelentenie.

Ha ez az egyetlen releváns él az $A! \longrightarrow B$, és az A értéke 0, akkor a B értéke nem lehet 1. Ha ugyanis a B értéke 1 lenne, akkor ez az él egy $0 \leq 1$ atomi feltételt fejezne ki, ez esetben pedig a megfelelő (**) egyenlőségben a jobb oldalon minden tényező értéke 1 lenne (hiszen ez az egyetlen él volt releváns), ami azt jelenti, hogy az eredeti (**) alakú feltételt nem elégíti ki a kapott eredmény. Tehát $A = 0! \longrightarrow B$ esetén $B = 0$-nak kell teljesülnie. Ha viszont az $A = 1$, akkor a folytonos nyíl által jelentett atomi feltétel alapján $(A \leq B)$ a $B = 1$. Tehát az él jelentése: $A = B$.

$A!\text{- -}\!\rightarrow B$ esetén pedig, ha $A = 0$, a B nem lehet 0, ekkor ugyanis $A = 0 \leq (1 - B) = 1$ atomi feltételt fejezne ki, és mivel a többi él nem releváns, az eredeti (**)-beli egyenlőség nem állna fenn. Tehát ekkor $B = 1$. Ha viszont az $A = 1$, akkor a szaggatott nyíl által jelentett atomi feltétel miatt $B = 0$-nak kell teljesülnie. Tehát az $A!\text{- -}\!\rightarrow B$ típusú él jelentése: $A = (1 - B)$.

A megoldás során fontos lesz tehát, hogy hogyan válhat egy él irrelevánssá, ezt kiértékelő nyilak tárgyalása után pontosan leírjuk.

Most már tudjuk reprezentálni a gráf segítségével a módosított feladat egyenlőségeit. Lássuk tehát, a gráf-lépések hogyan módosulnak az alapfeladatban már leírtakhoz képest.

Először tekintsük az élhozzáadó lépéseket. Megmaradnak az eredetiek (3.1 lemma), és még ehhez jönnek hozzá újak.

3.9. Lemma. Ha egy gráfban tetszőleges A, B és C csúcsokra szerepelnek az alábbi táblázat valamelyik sorának bal oldali oszlopában található éle(i), és a megfelelő sor jobb oldali oszlopában található(k) még nem (sem relevánsként, sem irrelevánsként), akkor az(oka)t új nem releváns élként felvehetjük a gráfunkba. (A !-lel jelölt esetekben a megfelelő élek egyedüli relevánsok az adott csúcsból.)

	Ha előfordul	akkor felvehető
1)	A- -→ B	$A \leftarrow\!\!\!/$ - B
2)	A! ⟶ B	$A \leftarrow\!\!\!/\!\!-\!\!\!/ B$
a)	$A \longrightarrow B \longrightarrow C$	$A \not\rightarrow C$
b)	$A \longrightarrow B$- - - C	A- $\not$- C
c)	A- - - B! ⟶ C	A- $\not$- C
d)	A- - - B!- -→ C	$A \not\rightarrow C$
e)	A- - - $B \leftarrow$ - -!C	$A \not\rightarrow C$

Bizonyítás. A megoldásban minden élnek megfelelő atomi feltételnek igaznak kell lennie, valamint a 0 típusú csúcsokból indulnia kell releváns élnek, ha az adott változó egyenlőség bal oldalán állt. Mivel releváns élből nem lesz irreleváns e lépések során, ezért ez a rész triviálisan teljesül. Nézzük most esetekre bontva, hogy hogyan teljesülnek az atomi feltételek.

1) Már az alapesetnél a szaggatott nyíl jelentésénél tárgyaltuk: $A \leq 1-B$ ugyanakkor teljesül, mint $B \leq 1-A$.

2) A 3.8 lemma alapján $A = B$, tehát $A \leq B$ mellett $B \leq A$ is teljesül.

a)-b) ugyanaz, mint az alap feladatnál.

c) ez tulajdonképpen következménye az előzőknek: a 2)-t alkalmazva BC-re, majd a b)-t CBA-ra, egyébként $A \leq 1-B$ és 3.8 lemma alapján $B = C$, tehát $A \leq 1-C$ teljesül, ami ekvivalens a $C \leq 1-A$ -val.

d)-e) $A \leq 1-B$, és a 3.8 lemma, illetve 1) eset alapján $C = 1-B$, így az $A \leq C$ is igaz.

Az új éleket azért irrelevánsan vesszük fel, mert az eredeti feladat atomi feltételei között nem szerepeltek, tehát nem fejezhetnek ki az eredeti (**) feltételben szereplő egyenlőséget.

Az alapsémáink ugyanazok lesznek mint az alapfeladat esetén voltak. Tulajdonképpen itt is elegendő az egyelemű alapséma használata (3.2 lemma), illetve fontos az egyenlősítő séma (3.5 lemma) is.

3.3. Megjegyzés. A 2) élhozzáadó lépés tulajdonképpen azt jelenti, hogy ha két csúcs (A és B) közt A! ⟶ B kapcsolat van, akkor őket is „egyenlősíthetjük".

Nézzük a kiértékelő nyilakat! A már ismertetettek mellett kapunk néhány újabbat is, a relevancia felhasználásával. (Ahol a ! jelzi, ott csak abban az

esetben hajtható végre a kiértékelés, ha ez az egyetlen induló releváns él abból a csúcsból, a többi esetben nincs semmiféle megkötés az adott él relevanciájára, vagy a releváns élek számára.)

3.10. Lemma. Egy módosított feladat gráfjában a következő élek kiértékelő nyilak

a) $1 \longrightarrow A$, ekkor $A = 1$;
b) $1 \text{- - -} A$, ekkor $A = 0$;
c) $0! \longrightarrow A$, ekkor $A = 0$;
d) $A! \longrightarrow 1$, ekkor $A = 1$;
e) $A \longrightarrow 0$, ekkor $A = 0$;
f) $A! \text{- -} \rightarrow 0$, vagy $A \leftarrow \text{- -} !0$, ekkor $A = 1$;

Bizonyítás. A 3.7, 3.8 lemmák és a 3.3 megjegyzés alapján triviális.

Most nézzük meg, hogy hogyan változik az élek releváns tulajdonsága a felhasználásuk közben. (Ez egy fontos kérdés, hiszen mint láttuk, igazán csak akkor tudunk egy releváns élt „kihasználni", ha egyedüli relevánsként indul az adott csúcsból.)

3.11. Lemma. A kiértékelés után az 1 értékű csúcsból induló nyilak mind irrelevánssá válnak. Ezen kívül a következő nyilak irrelevánsok, ha az adott csúcsból nem egyetlen releváns él indul (ha ez az egyetlen releváns, akkor a fenti d), illetve f) kiértékelést kell előbb elvégeznünk):

g) $A \longrightarrow 1$;
h) $A \text{- -} \rightarrow 0$.

Bizonyítás. A releváns él definíciója (3.4 definíció) miatt nyilvánvaló, hogy már ismert, 1 értékű csúcsból nem indulhat. g) - h) esetek: Az él csak akkor lehetne releváns, ha $A = 0$ értéket venne fel, ez esetben viszont egyik él sem egyenlőséget fejez ki, hanem $A = 0 < 1$ relációt, tehát ezek az élek nem lehetnek relevánsok.

Még egy helyzetben fogjuk használni az irrelevánssá tételt, ez technikai jellegű, ugyanis így tudjuk figyelembe venni az „egyenlősítés" hatását. (A gráfunk ez esetben is ekvivalens marad az eredetivel.)

3.12. Lemma. Ha egy A csúcsból több ugyanolyan típusú (folytonos, illetve szaggatott) releváns nyíl is vezet ugyanabba a téglalapba (egymással egyenlősített csúcsokba), akkor ezek közül egyet hagyunk meg relevánsnak, a többit áthúzzuk. Az így kapott gráf ekvivalens a korábbival.

Bizonyítás. Tegyük fel, hogy A-ból B-be és a vele egy téglalapban levő C-be is ugyanolyan típusú releváns nyíl vezet. Mivel B és C értéke meg kell hogy egyezzen az eredeti gráfhoz tartozó megoldásokban is, ezért a két releváns él ugyanazt a relációt fejezi ki. Tehát a megoldásban vagy mindkettő releváns lesz, vagy egyik sem. Nekünk csak az a fontos, hogy a (**)-beli egyenlőséghez legyen legalább egy releváns él, tehát bármelyiket irrelevánssá tehetjük, ha közben a másik releváns marad.

Az itt ismertetett esetekben az addig relevánsként szereplő éleket áthúzzuk a gráfban, ezzel jelezve, hogy irrelevánssá váltak.

3.4. Megjegyzés. Amelyik él releváns egy megoldásban, az az él relevánsan szerepel az eredeti gráfban is.

3.5. Megjegyzés. Ezekkel a gráf-transzformációs lépésekkel minden lehetséges atomi feltételszerű relációt behozhatunk a gráfba, ami a gráfban levő élkombináció lokális jelentésén alapul.

A következő részben ezen gráf-lépéseket felhasználva készítünk egy algoritmust, amely segítségével mind az alap, mind a módosított Boole-programozási feladatok lehetséges megoldásait meghatározhatjuk.

3.3 Univerzális algoritmus a tárgyalt Boole-programozási feladatok megoldására

Ebben a részben az algoritmust ismertetjük, aminek segítségével előállíthatunk egy, illetve az összes megengedett megoldást. Ezután megnézzük az algoritmus működését két újabb példán.

3.1. Algoritmus.

1. Rajzoljuk meg a feladat gráfját, ügyelve, hogy csak az egyenlőséggel adott eredeti feltételeknek megfelelő élek legyenek relevánsok.

2. Implicit leszámlálás módszere:

2.1 Alkalmazzuk a lehetséges élhozzáadó lépéseket.

2.2 Alkalmazzuk a lehetséges alapsémákat csúcsok értékeinek meghatározására.

2.3 Az egyenlősítő séma használatával tegyük közös téglalapba a megfelelő csúcsokat.

2.4 Ha van már ismert értékű csúcs, akkor próbáljuk a kiértékelő nyilak segítségével más csúcsok értékét is meghatározni.

2.5 Ellenőrizzük, mely nyilak váltak irrelelvánssá.

2.6 Ha van megoldás, akkor tároljuk, és menjünk 2.10-re (ha csak egy megoldás kell, stop).

2.7 Ha nincs megoldás, és nem változott a gráf, egy változót rögzítsünk, és menjünk 2.1-re.

2.8 Ha nincs megoldás, de változott a gráf, akkor menjünk 2.1-re.

2.9 Ha ellentmondásra jutottunk, menjünk 2.10-re.

2.10 (Ágcsere) Ha lehetséges ágcsere, akkor ágcsere, különben stop.

Megoldást akkor kaptunk, ha minden csúcshoz pontosan egy érték van rendelve a $\{0,1\}$ halmazból, és egyik csúcshoz sem kell a másik értéket is hozzárendelnünk valamely él (alapséma vagy kiértékelőnyíl) jelentése alapján. Illetve minden olyan 0 értékű csúcsból indul releváns él, amelyből eredetileg indult (a (**) feltételek bal oldali változói). Ekkor minden élnek megfelelő atomi feltétel teljesül, és az egyenlőségek is fennállnak.

Ellentmondásra akkor jutunk, ha a lépések során egy csúcshoz az 1 és a 0 érték is hozzárendelésre kerül.

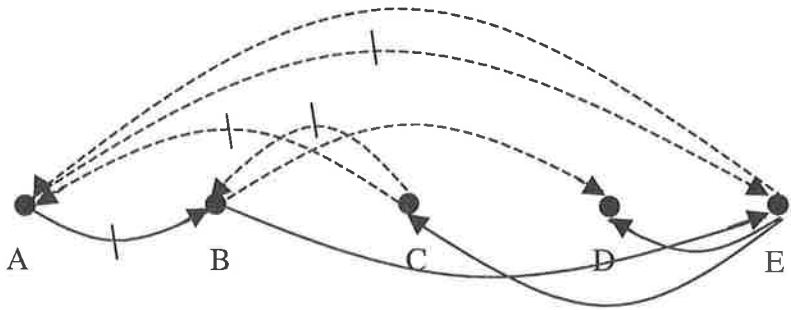
Ágcserének nevezzük azt, amikor a legutóbbi önkényes értékadás (2.7) előtti gráfot vesszük vizsgálatunk tárgyává, ha ennél az értékadásnál még csak az adott csúcs egyik lehetséges értékével próbálkoztunk, akkor most a másikat fogjuk megpróbálni. Ha már mind a két lehetséges értéket vizsgáltuk, akkor visszalépünk az ez előtti legutolsó 2.7 lépésre. Amennyiben a legelőször végrehajtott 2.7 lépésnél is visszalépünk (vagyis ott is próbáltuk mindkét lehetséges értéket), akkor az algoritmus véget ér, az addig talált megoldások a megengedett megoldásai az adott Boole-programozási feladatnak.

3.6. Megjegyzés. Ha a 2.7 lépés változórögzítése nélkül megvan egy megoldás, akkor ez az egyetlen, ha viszont ellentmondásra jutunk, mielőtt változót rögzítenénk, akkor nincs megoldása a feladatnak.

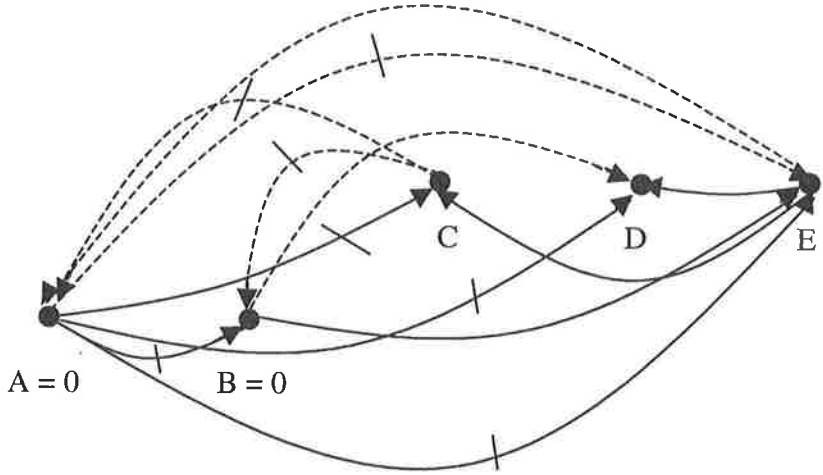
Nézzünk most egy példát.

3.2. Példa.
$A \leq B \cdot (1 - E)$
$B = (1 - D) \cdot E$
$C \leq (1 - A) \cdot (1 - B)$
$E = (1 - A) \cdot C \cdot D$

A példa gráfja:



Azok az élek relevánsok, amelyek az eredeti feltételrendszer egyenlőségeinek atomi feltételei. Az a) élhozzáadást használhatjuk az ABE, majd az AEC és AED csúcsok közt, így az A értéke 0 lesz, köszönhetően pl. az A és C, vagy A és E közti kételemű alapsémának. A BED csúcsokra alkalmazva az a) élhozzáadást a B és a D közt kapunk alapsémát, tehát a B is 0.



Élhozzáadásokkal még megkaphatjuk a következő éleket: folytonos nyilat BC közt, illetve szaggatott éleket: BA, AB, AC, DB, AD, DA, BC, BE, EB és természetesen az AA, illetve a BB hurokélt. Illetve az EA szaggatott nyíl irrelevánssá válik. Viszont innen tovább nincs használható lokális lépés, jön a 2.7 alapján a változórögzítés. Tegyük fel, hogy a $C = 0$, ekkor az EC folytonos nyíl kiértékelővé válik, $E = 0$. Megint odáig jutunk, hogy nincs használható gráf-lépés. Válasszuk a $D = 0$ esetet. Ez egy megoldás. Ágcsere: a $D = 1$ eset következik, ez is egy megoldás. Most visszalépünk egészen a C értékadásáig. Nézzük a $C = 1$ esetet. Ekkor az EC folytonos él irreleváns lesz. Megint nem tudunk tovább haladni, válasszuk a D-t 0-nak, ekkor az ED folytonos nyíl kiértékelhető: $E = 0$. Ez is egy megoldás. Ágcsere, és $D = 1$ esetet vizsgáljuk. Ekkor mivel az E-ből az egyetlen releváns él a D-be vezet, ezt kiértékelhetjük, $E = 1$, ami újabb megoldás. Visszalépünk, és nincs több választási lehetőség, mindet végignéztük. Tehát a lehetséges megoldások a következő (A, B, C, D, E) vektorok: $(0,0,0,0,0)$, $(0,0,0,1,0)$, $(0,0,1,0,0)$ és $(0,0,1,1,1)$.

3.7. Megjegyzés. Ha a megengedett megoldások nem egyforma jók számunkra, akkor az algoritmus 2.7 lépésében tudjuk a keresést a megfelelő irányba terelni. Pl. ha egy változónak még nincs értéke, és jobban szeretnénk, hogy 1 legyen, akkor ebben a lépésben válasszuk azt 1-nek, így ha van olyan megoldás amiben ez a változó 1 értékű, akkor hamarabb találjuk azt meg, mint egyébként.

3.8. Megjegyzés. Mint a 2.1. megjegyzésben jeleztük, előfordulhat, hogy egy módosított Boole-programozási feladatnak nincs megengedett megoldása. Ez előfordul akkor, ha a változók között van olyan irányított kör, amely csupa egyetlen releváns élből áll (vagyis minden csúcsánál ott a ! jel, és a releváns élen megyünk tovább), és ez a kör páratlan számú szaggatott élt tartalmaz.

(Ugyanígy ellentmondásos a feltételrendszer, ha a gráfmanipulációs lépésekkel el tudjuk érni ilyen kör létezését, mint a következő példa is mutatja.)

3.3. Példa.

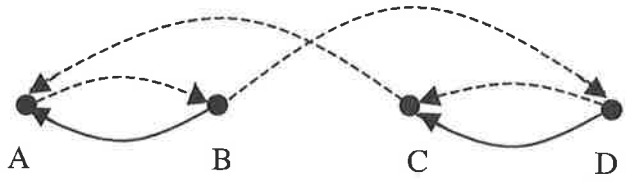
$A = 1 - B$

$B = A \cdot (1 - D)$

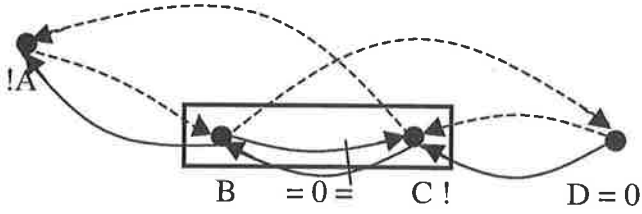
$C = 1 - A$

$D = C \cdot (1 - C)$

A feladat gráfja:



(Minden él releváns, hiszen csupa egyenlőségből álló feltételrendszerünk van, sőt az A és a C csúcsokból csak egy-egy él indul.) Használjuk a 3.9 lemma d) illetve e) élhozzáadását a $C! - - \rightarrow A! - - \rightarrow B$ részhez mindkét irányban, vagyis vegyük fel a B és C közti folytonos nem releváns éleket mindkét irányba. Ezek alapján egyenlősíthetjük e két csúcsot. Másrészt az eredeti gráf két helyen is tartalmazza a kételemű alapsémát, aminek alapján a D és a B értéke csak 0 lehet a megoldásban. A $C = B$ miatt pedig a $C = 0$ is fennáll. Lássuk, hogyan is néz ki a gráfunk:



Most vizsgáljuk meg, hogy mely élek váltak irrelevánssá. A B-ből a D-be mutató szaggatott irreleváns (3.11 lemma h) pont). Ekkor viszont a B-ből az A-ba mutató folytonos nyíl lesz az egyedüli releváns, ami a B csúcsból indul. Ekkor viszont az A és a B közt van egy, a 3.8 megjegyzés feltételeinek eleget tevő körünk, vagyis feltételrendszerünk ellentmondó, az adott megszorításokkal a feladatnak nincs megoldása. (Egyébként az ellentmondást megkapnánk úgy is, ha az A értékét a $B = 0! \longrightarrow A$ kiértékelő nyíl (3.10 lemma c) pont: $A = 0$), illetve a $A! - - B = 0$ kiértékelő nyíl (3.10 lemma f) pont: $A = 1$) felhasználásával határoznánk meg.)

4 Boole programozás a gyakorlatban

A gyakorlatban hasonló jellegű feladatok merülhetnek fel, amikor egyes kapcsolókra vannak feltételeink. Olyan feladatok lehetséges megoldásait kereshetjük így, ahol minden változó arról szól, hogy az adott dolog kell vagy nem kell. Egyes típusú logikai fejtörőknél a probléma igen hasonló, a (**) alakú feltételeknek megfelelő alakba írható ([6], [8]) és tárgyalható ezzel a módszerrel [7]. A Boole-programozási feladatot könnyen implementálhatjuk mátrixok segítségével. Például alapsémát jelent, ha a szaggatott élek mátrixában a főátlóban nem 0 áll stb.

5 Összefoglalás

A gráfelméleti megközelítést felhasználva speciális egészértékű programozási feladatokat, ún. Boole-programozási feladatokat oldottunk meg ezzel az új módszerrel. A gráf-lokális módszer, ami a gráfban a csúcsok közvetlen kapcsolataiból von le következtetéseket, a legtöbb esetben hatékonyan használható, a bemutatott backtrack algoritmus segítségével megvalósított esetszétválasztás módszerével pedig minden esetben meghatározhatjuk a lehetséges megoldásokat. A későbbiekben tervezzük a módszer kiterjesztését más alakú, illetve nem csak $\{0,1\}$ értékhalmazon adott feladatokra is.

6 Köszönetnyilvánítás

A szerző ezúton is szeretné megköszönni Vízvári Bélának a hasznos észrevételeit és tanácsait.

Irodalom

1. Forgó Ferenc, *Nemkonvex és diszkrét programozás*, Budapest, Közgazdasági és Jogi Könyvkiadó, 1978
2. Glevitzky Béla, *Az operációkutatás elemei*, egyetemi jegyzet, Debrecen, 1992.
3. Glevitzky Béla, *Operációkutatás*, egyetemi jegyzet, Debrecen, 1980.
4. Hajnal Péter, *Gráfelmélet*, Polygon, Szeged, 1997.
5. Kovács László Béla, *A diszkrét programozás kombinatorikus módszerei*, Budapest, Bolyai János Matematikai Társulat, 1969
6. Nagy Benedek, Kósa Márk: Logical puzzles (Truth-tellers and liars), *ICAI'01*, Eger, 2001. 105–112.
7. Nagy Benedek, Igazmondó-hazug feladatok gráfelméleti megközelítésben, XXV. MOK, Debrecen, 2001.
8. Raymond Smullyan, *Forever Undecided*, Alfred A. Knopf, New York, 1987.
9. Vízvári Béla, *Egészértékű programozás*, Budapest, Tankönyvkiadó, 1989, 1992.

BOOLE-PROGRAMMING AND RELATED GRAPHS

In this paper, we investigate special integer-valued programming problems in which each variable is either 0 or 1. We study the basic and the modified Boole-programming problems. In the basic Boole-programming problems we can write to linear form to our conditions. We represent these conditions in a graph and we can solve the problem using this graph. In the modified case our conditions are non-linear, but we can use nearly similar graph representation and solving method. In graph representation we use so-called 'relevant edges' to represent the non-linear conditions. We present a general algorithm to get the solutions of these problems.