Robosztus megoldások kombinatorikus optimalizálási feladatokra

Szakdolgozat

Írta: Dibuz Dániel

Matematika BSc
Alkalmazott Matematikus szakirány

Témavezető:

Jüttner Alpár, tudományos főmunkatárs
Operációkutatási tanszék

Eötvös Loránd Tudományegyetem
Természettudományi Kar
2013

Tartalomjegyzék

1. fejezet

Bevezetés

A kombinatorikus optimalizálás során egy véges halmazon keresünk egy adott célfüggvény szerint optimálisat. A gyakorlati alkalmazás során esetenként felmerülő probléma, hogy a döntéshozás pillanatában nem rendelkezünk elegendő információval a célfüggvény pontos felírásához. Ennek hiányában a szokásos algoritmusok nem használhatóak. A robosztus optimalizálás célja olyan megoldást keresni, ami minden szóbajöhető célfüggvényre viszonylag jó eredményt ad, és ezáltal a bizonytalanságra a lehető legkevésbé érzékeny.

Panos Kouvelis és Gang Yu két fontos modellt adtak a probléma kezelésére. Feltételezve, hogy ismereteink alapján a lehetséges célfüggvényeket egy jól kezelhető halmazra tudjuk szorítani, kereshetjük a legnagyobb biztos eredményt adó megoldást, vagy az optimálistól való távolságra minimális megoldást. Az eredeti problémát ezzel egy másik optimalizálási feladattá transzformáljuk, ami figyelembe veszi az eredeti célfüggvény pontatlanságát is.

A módszereknek számos gyakorlati alkalmazása van, hiszen a bizonytalanság melletti döntéshozás egy gyakran előforduló dilemma, ami az optimalizálási módszerek használóit sem kerüli el.

2. fejezet

Robosztus modellek

2.1. Robosztusság

A dolgozatban lineáris célfüggvényű kombinatorikus optimalizásási feladatokkal fogunk dolgozni. Először tekintsük át ezeknek az általános leírását. Adott egy véges E halmaz, annak elemein pedig egy $c : E \to \mathbb{R}$ költségfüggvény, valamint E részhalmazainak egy R családja (megengedett halmazok vagy megoldások). Egy $e \in E$ elem költségét c_e-vel fogjuk jelölni. Egy $r \in R$ megoldás költségén az alkotóelemei összköltségét értjük: $c_r = \sum_{e \in r} c_e$.

A feladat megkeresni a legkisebb (legnagyobb) költségű megengedett halmazt: $\min_{r \in R} c_r$.
(A dolgozatban végig minimumfeladatokkal fogunk folgozni. A legtöbb kimondott állítás viszont egyszerű meggondolásokkal átvihető maximumfeladatokra is.)

A probléma, hogy a gyakorlatban a költségfüggvény nem mindig tekinthető ismertnek a döntéshozás pillanatában. Ekkor persze a fenti feladatot nem tudjuk ilyen formában megoldani. A robosztus modellben ezért fix költségfüggvény helyett költségfüggvények egy családjával dolgozunk, amiről feltételezzük, hogy köztük van az a példány, ami a végén megvalósul. Az optimalizáláshoz egy olyan kritériumot keresünk, ami ezt a többértékűséget figyelembe veszi, és biztosítja, hogy a választott megoldás minden esetben viszonylag hatékony marad.

Definíció. *A költségek egy realizálódását (vagyis egy konkrét célfüggvényt a sok közül) a szakirodalom szcenáriónak nevezi. Ezek halmazát S fogja jelölni.*

Világos, hogy az egyes szcenáriókhoz más és más optimális megoldás, illetve optimum tartozhat. Egy megoldás lehet optimális az egyikben, de nem optimális a másikban.

Jelölések:

· c_e^s és c_r^s fogja jelölni az s szcenárióban az e elem, ill. r megoldás költségét

· c_*^s jelöli az s szerinti optimális megoldásnak a költségét.

A szcenáriók leírásának két fő módját különböztetjük meg:

- A diszkrét vagy explicit megadásnál egyszerűen felsoroljuk S elemeit. Ez persze implikálja, hogy a szcenáriók száma véges.
- Intervallum, avagy implicit megadásnál az egyes elemek egymástól függetlenül egy $[l_e, u_e]$ intervallumból kapnak értéket. Ekkor az S halmaz az egyes intervallumok Descartes-szorzata: $S = [l_1, u_1] \times [l_2, u_2] \times \ldots \times [l_{|E|}, u_{|E|}]$

Természetesen ez a két változat nem meríti ki az összes lehetőséget, de a gyakorlatban ezek használatosak.

2.2. Minimax kritérium

Kézenfekvő ötlet a megoldás robosztusságát az összes szcenárión vett költségek közül a legrosszabbal jellemezni. Ez az a legkisebb határ, amire biztosan tudjuk garantálni, hogy a költség alatta fog maradni. A minimax kritériumban tehát vesszük minden szcenárióban a megoldásunk költségét, és a legnagyobb előforduló eredményt próbáljuk minimalizálni: $\min_{r \in R} \max_{s \in S} c_r^s$

Ez a megközelítés hasznos olyan alkalmazásokban, ahol a legnagyobb biztonság elérése a cél. Hozzá kell azonban tennünk, hogy mivel a modell nem foglalkozik a kedvezőbb esetekkel, ezért más profilú feladatokra esetleg használhatatlanul költséges megoldásokat adhat.

2.2.1. Általános eredmények a diszkrét szcenárió változatra

Egy nem túl erős, de könnyen bizonyítható állítás, hogy ha egy minimalizálási feladatunk van k szcenárióval, akkor k-közelítő megoldást kapunk az alapfeladat egy példányának megoldásával.

2.1. Állítás. *Tegyük fel, hogy k szcenárió van, és minden költség nemnegatív. Jelölje m azt a fiktív szcenáriót, ahol minden elem a szcenáriókon vett átlagköltségét kapja. Erre a (determinisztikus) rendszerre az alapfeladat optimuma a robosztus változatnak k-approximációja lesz, vagyis* $c_*^m \leq k \cdot (\min_{r \in R} \max_{s \in S} c_r^s)$.

Bizonyítás. Legyen X a robosztus változat optimális megoldása, Y pedig az átlag-szcenárió optimális megoldása. Indirekt tegyük fel, hogy $\max_{s \in S} c_Y^s >$ $> k \cdot \max_{s \in S} c_X^s$. Tudjuk továbbá, hogy egy r megoldás költsége az m szcenárióban definíció szerint

$$c_r^m = \sum_{e \in r} c_e^m = \sum_{e \in r} \frac{\sum_{s \in S} c_e^s}{k} = \frac{1}{k} \sum_{s \in S} \sum_{e \in r} c_e^s = \frac{1}{k} \sum_{s \in S} c_r^s \tag{2.1}$$

Ezek alapján

$$c_X^m = \frac{\sum_{s \in S} c_X^s}{k} \leq \max_{s \in S} c_X^s < \frac{1}{k} \max_{s \in S} c_Y^s \leq \frac{1}{k} \sum_{s \in S} c_Y^s = c_Y^m \tag{2.2}$$

lenne, ami ellentmond Y definíciójának. □

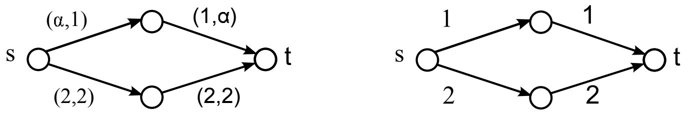
Megjegyzés: Maximumkeresére is megfogalmazhatnánk az állítás ellenpárját (ez esetben a k-approximáció azt jelentené, hogy az optimumnak legalább a k-adrészét kapjuk, azaz $k \cdot c_*^m \geq (\max_{r \in R} \min_{s \in S} c_r^s)$), de ez nem igaz. Legyen ugyanis az alapprobléma egyszerűen maximális elem keresése, két elemmel (e_1, e_2) és két szcenárióval (s_1, s_2) az alábbiak szerint:

	e_1	e_2
s_1 szerinti érték	1	2
s_2 szerinti érték	1	$\varepsilon < 1$
$\frac{s_1 + s_2}{2}$ szerinti érték	1	$\mathbf{1 + \frac{\varepsilon}{2}}$
legrosszabb érték	**1**	ε

Az átlagszcenárió e_2-t választja az optimális e_1 helyett, és $1/\varepsilon$ nem korlátos.

Hasonló eredményt ad, ha átlag helyett maximumot veszünk:

2.2. Állítás. *Legyen ismét minimalizálási feladat k szcenárióval és nemnegatív költségekkel. Most kapja minden elem az összes előforduló költsége közül a legnagyobbat. (Nevezzük max-szcenáriónak és jelöljük M-mel.) Erre az alapfeladat optimális megoldása szintén a robosztus feladat k-approximációja lesz.*



2.1. ábra. Maximumkeresésre a minimum-szcenárió nem approximál. Bal oldalt látható az eredeti gráf (c_e^1, c_e^2) formátumban, ahol α tetszőlegesen nagy lehet; jobb oldalt a minimum-szcenárió megvalósulása. A módszer a robosztus értelemben nem optimális alsó utat választja. A robosztus költségek hányadosa, $(\alpha+1)/4$ nem korlátos.

Bizonyítás. Legyen ismét X a robosztus feladat optimális megoldása, Y a max-szcenárió optimális megoldása, és tegyük fel indirekt, hogy $\max_{s\in S} c_Y^s >$ $> k \cdot \max_{s\in S} c_X^s$. Ekkor

$$c_X^M \leq \sum_{s\in S} c_X^s \leq k \cdot \max_{s\in S} c_X^s < \max_{s\in S} c_Y^s \leq c_Y^M \tag{2.3}$$

Ez ellentmond Y definíciójának. □

Ez az állítás ugyancsak nem terjed ki a maximalizálási változatra.

Megjegyzés: Az előző két becslés éles.

Bizonyítás. Az átlag-becsléshez triviális példa lehet két elem közül a minimális kiválasztása. Az egyik elem költsége legyen minden esetben 1, a másiké pedig az egyik szcenárióban k, a többiben 0. Az első megoldás optimális, 1 maximális költséggel. A második elem maximális költsége k. A két költségátlag megegyezik, ezért az átlag-szcenárióban mindkettő optimális, így a második is.

A maximum-becslés élességére példa lehet az alábbi gráf a minimális út problémával: a gráf álljon két diszjunkt k élű útból. Egyikben minden él költsége mindig k. A másodikban az i-edik él költsége az i-edik szcenárióban k, a többiben 0. Könnyű belátni, hogy az első út M-re optimális, és k-approximáció. □

2.2.2. Megjegyzés az intervallum szcenárió változathoz

Az intervallum eset, ahol az egyes elemek költségei egymástól függetlenül az $[l_e, u_e]$ intervallumból kerülnek ki, triviális. A legrosszabb esetet az adja, ha minden elem a felső korlátját kapja. Így csak az alapfeladatot kell megoldani erre. Ez persze azt is jelenti, hogy a robosztus változat bonyolultsága megegyezik az eredeti feladatéval.

2.2.3. Bonyolultságelméleti eredmények

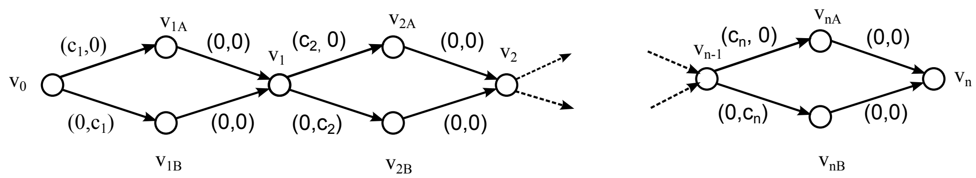
Az előző megjegyzés értelmében csak az diszkrét változatot vizsgáljuk. Sajnos ez még polinomidőben megoldható alapproblémák mellett is gyakran NP-nehézzé válik. Példaként belátjuk, hogy a legrövidebb út már két szcenárió mellett is NP-nehéz.

2.3. Állítás. *A legrövidebb út min-max robosztus változata $k \geq 2$ szcenárió mellett NP-nehéz.*

Bizonyítás. $k = 2$-re látjuk be az állítást, a többire ugyanúgy működik. A 2-partíció problémát vezetjük vissza a robosztus legrövidebb útra. A 2-partíció problémában adott egy n elemű, pozitív egész számokból álló H multihalmaz, amiről el szeretnénk dönteni, hogy felbontható-e két diszjunkt H_1 és H_2 részhalmazra úgy, hogy H_1-ben és H_2-ben a számok összege megegyezik. (A multihalmaz hasonló a halmazhoz, annyi különbséggel, hogy egy elem többször is előfordulhat.)

Legyenek S elemei $c_1, c_2 \ldots c_n$. Konstruáljuk ehhez a 2.2 ábrán látható gráfot. Látható, hogy minden $v_0 \to v_n$ út végigjárja a $v_1, v_2, \ldots v_n$ csúcsokat ilyen sorrendben, és minden $v_{i-1} \to v_i$ átmenetnél választhatunk, hogy a v_{iA} vagy a v_{iB} csúcs felé megyünk. Előbbinek a költsége az 1. szcenárió mellett c_i, a másodiknál 0. Utóbbinak a költsége az 1. szcenárióban 0, a másodikban c_i. Egy p út költsége összesen tehát $c_p^1 = \sum\limits_{\substack{v_{iA} \in p \\ 1 \leq i \leq n}} c_i$ az első szcenárióban, és $c_p^2 =$ $= \sum\limits_{\substack{v_{iB} \in p \\ 1 \leq i \leq n}} c_i$ a másodikban. Vagyis a v_{iA} ill. v_{iB} választásokat megfeleltethetjük annak, hogy az i-edik elemet a H_1, vagy a H_2 halmazba tesszük, és az így kapott út költsége az 1. szcenárió szerint $\sum\limits_{c_i \in H_1} c_i$, a második szcenárió szerint $\sum\limits_{c_i \in H_2} c_i$.

Világos, hogy mivel H_1 és H_2 együtt kiadják az egész H halmazt, valamelyikben legalább $\frac{\sum_{c_i \in H} c_i}{2}$ az elemek összértéke, és egyenlőség áll fent akkor és csak akkor, ha a két részhalmaz értékösszege megegyezik. Vagyis a robosztus legrövidebb út feladat megoldása $\min\limits_{(H_1,H_2)} \max(\sum\limits_{i \in H_1} c_i, \sum\limits_{i \in H_2} c_i) = \frac{\sum_{c_i \in H} c_i}{2}$ akkor és csak akkor, ha H-nak létezik a kívánt szétbontása.



2.2. ábra.

□

Ennek bizonyítása eredetileg Yu és Yang nevéhez fűződik. Kouvelis és Yu hasonló eredményt bizonyított a feszítőfa, párosítás és hátizsák-problémára, Aissi, Bazgan és Vanderpooten pedig a minimális s-t vágásra. Ha a szcenáriók száma az input része, akkor ezek mind erősen NP-nehézzé válnak. A minimális vágás viszont, ahol a gráfot csúcsait két nemüres halmazra szeretnénk vágni kijelölt (s,t) csúcsok nélkül, konstans sok szcenárió esetén polinomidőben megtalálható.

Min-max kritérium			
Probléma	Diszkrét, k szcenárióval		Intervallum
	k konstans	k az input része	
Legrövidebb út	Gyengén NP-nehéz	Erősen NP-nehéz	Polinomiális
Legolcsóbb feszítőfa	Gyengén NP-nehéz	Erősen NP-nehéz	Polinomiális
Párosítás	NP-nehéz	Erősen NP-nehéz	Polinomiális
Minimális s-t vágás	Erősen NP-nehéz	Erősen NP-nehéz	Polinomiális
Minimális vágás	Polinomiális	Erősen NP-nehéz	Polinomiális
Hátizsák-feladat	Gyengén NP-nehéz	Erősen NP-nehéz	Gyengén NP-nehéz

2.3. Robosztus eltérés kritérium

Karasan, Pinar és Yaman a min-max változatnál kevésbé óvatos megközelítést javasolt: a legrosszabb eset költsége helyett az optimumtól való legnagyobb eltérésre minimalizáljunk: $\min_{r \in R} \max_{s \in S} (c_r^s - c_*^s)$

Ez a robosztus eltérés, vagy más néven a min-max regret kritérium. A "regret", szó szerinti fordításban "megbánás" kifejezés egy döntéselméletből származó fogalom, a magyar megfelelője "elmulasztott haszon". A hozott döntéssel kapott haszon és a más döntéssel elérhető haszon közötti különbséget jelenti. A gazdasági szereplők viselkedése gyakran ezt a minimax regret elvet követi.

Definíció. *robosztus eltérésnek nevezzük egy adott szcenárióban az optimumtól való eltérést, vagyis a $c_r^s - c_*^s$ értéket.*

Jelölése: $RD^s(r)$ jelöli az s szcenárióban r robosztus eltérését. (Az RD az angol robust deviation rövidítése.) A legnagyobb robosztus eltérést jelölje $RD^{max}(r) = \max_{s \in S} RD^s(r)$

2.3.1. Elméleti eredmények a diszkrét változatra

2.4. Állítás. *A min max eltérés változatra is érvényes a (2.1) állítás, miszerint k szcenárió esetén az átlag-szcenárió optimális megoldása k-approximációt ad a robosztus feladatra.*

Bizonyítás. Hasonlóan bizonyítunk. Legyen X a robosztus feladat optimális megoldása, Y az átlag-szcenárió optimális megoldása, m jelölje az átlag-szcenáriót és tegyük fel indirekt, hogy $\max_{s \in S}(c_Y^s - c_*^s) > k \cdot \max_{s \in S}(c_X^s - c_*^s)$. Ekkor

$$c_X^m - \frac{\sum_{s \in S} c_*^s}{k} = \frac{\sum_{s \in S} c_X^s}{k} - \frac{\sum_{s \in S} c_*^s}{k} = \frac{\sum_{s \in S}(c_X^s - c_*^s)}{k} \leq \tag{2.4}$$

$$\leq \max_{s \in S}(c_X^s - c_*^s) < \frac{1}{k} \max_{s \in S}(c_Y^s - c_*^s) \leq \tag{2.5}$$

$$\leq \frac{1}{k} \left(\sum_{s \in S} c_Y^s - \sum_{s \in S} c_*^s \right) = c_Y^m - \frac{\sum_{s \in S} c_*^s}{k} \tag{2.6}$$

Aminek az elejét és a végét összevetve a $c_X^m \leq c_Y^m$ ellentmondás adódik. □

Megjegyzés: A min-max változattal ellentétben a módszer itt negatív költségek mellett és maximalizálási problémára is működik. Utóbbi esetben a bizonyítás ugyanaz, csak mindent szorzunk (-1)-gyel.

Megjegyzés: Itt is mutathatunk olyan példát, ahol egyenlőség teljesül. Vegyük a legegyszerűbb feladatot, a minimális elem kiválasztását. Legyen két szcenárió és három elem, melyek költségei (c_e^1, c_e^2) formában megadva (1,1) , (0,2) ill. (2,0). Az átlag-szcenárió szerint mindegyik optimális, a legnagyobb robosztus eltérések pedig rendre 1, 2 és 2. k szcenárióra ennek általánosítását adhatjuk k+1 elemmel, ahol az elsőnek a költségei $(1,1 \ldots 1)$, a többinek pedig $(k, 0, 0 \ldots 0)$, $(0, k, 0, \ldots 0)$, $\ldots$ $(0, 0 \ldots 0, k)$.

Megjegyzés: A max-szcenárió viszont ebben az esetben nem ad semmilyen approximációt. Könnyen konstruálhatunk akár olyan példát is, ahol az egyik megoldás robosztus eltérése 0, a másiké pozitív, a maximumuk viszont megegyezik.

2.3.2. Elméleti eredmények az intervallum változatra

Először is jegyezzük meg, hogy a min-max változattal ellentétben itt nem triviális a megoldás megtalálása. (Sőt, mint látni fogjuk, sok esetben NP-nehéz lesz.)

Definíció. *nevezzük egy r megoldás legkedvezőbb szcenáriójának azt, ahol*

$$c_e = \begin{cases} l_e & e \in r \\ u_e & e \notin r \end{cases} \tag{2.7}$$

Jelölés: $r+$

Definíció. *nevezzük egy r megoldás legkedvezőtlenebb szcenáriójának azt, ahol*

$$c_e = \begin{cases} u_e & e \in r \\ l_e & e \notin r \end{cases} \tag{2.8}$$

Jelölés: $r-$

2.5. Állítás. *egy r megoldás robosztus eltérése a legkedvezőtlenebb szcenáriójában a legnagyobb.*

Bizonyítás. A megoldásban szereplő egyik elemet k-val növelve annak költsége k-val nő, az optimális megoldás költsége pedig legfeljebb k-val. Ekkor a robosztus eltérés, ami a kettő különbsége, nem csökkenhet. Hasonlót mondhatunk az r-ben nem szereplő elemek költségének csökkentéséről is. □

Következmény:

$$RD^{max}(r) = \sum_{e \in r} u_e - c_*^{r-}. \tag{2.9}$$

Definíció. *Nevezzük extrém szcenáriónak azokat a szcenáriókat, ahol minden elem a költségintervallumának vagy a felső, vagy az alsó korlátját kapja.*

Be fogjuk látni, hogy bár végtelen sok szcenárió van, elég az extrém szcenáriókat vizsgálni. Ehhez előbb lássunk be egy segédállítást:

2.6. Állítás. *Tetszőleges $A \in R$, $B \in R$ megoldásokra*

$$RD^{max}(B) \leq RD^{max}(A) + \sum_{e \in B \setminus A} u_e - \sum_{e \in A \setminus B} l_e \tag{2.10}$$

Bizonyítás. Könnyen belátható, hogy

$$c_B^{B-} = c_A^{A-} + \sum_{e \in B \setminus A} u_e - \sum_{e \in A \setminus B} u_e \tag{2.11}$$

Megmutatjuk, hogy

$$c_*^{B-} \geq c_*^{A-} - \sum_{e \in A \setminus B} (u_e - l_e) \tag{2.12}$$

is igaz. Tegyük fel indirekt, hogy nem így van. Vegyük az $(A \cup B)-$ szcenáriót, vagyis ahol

$$c_e = \begin{cases} u_e & e \in A \cup B \\ l_e & e \notin A \cup B \end{cases}$$

Jelölje B' az $B-$ szcenárió optimális megoldását. Ekkor állításunk szerint

$$c_*^{A-} > c_{B'}^{B-} + \sum_{e \in A \setminus B} (u_e - l_e) \geq c_{B'}^{(A \cup B)-} \geq c_{B'}^{A-} \tag{2.13}$$

,ami ellentmondás. (Egy optimális megoldás költsége nem lehet nagyobb semelyik másik megoldásénál.) Ekkor (2.11) - (2.12) a bizonyítandó állítást adja. □

Mostmár bizonyíthatjuk az alábbi állításunkat:

2.7. Állítás. *legyen X az implicit min-max regret feladat optimális megoldása. Ekkor X optimális megoldása a saját legkedvezőbb szcenáriójának.*

Bizonyítás. Indirekt tegyük fel, hogy nem így van; legyen az X+ szcenárió optimális megoldása $Y \neq X$. Ekkor a feltevésünk szerint $c_Y^{X+} < c_X^{X+}$. Innen

$$\sum_{e \in (Y \setminus X)} u_e + \sum_{e \in (Y \cap X)} l_e \leq \sum_{e \in (X \setminus Y)} l_e + \sum_{e \in (X \cap Y)} l_e \tag{2.14}$$

tehát

$$\sum_{e \in (Y \setminus X)} u_e - \sum_{e \in (X \setminus Y)} l_e < 0 \tag{2.15}$$

adódik. Ezt, és az előbbi (2.10) eredményt felhasználva

$$RD^{max}(Y) \leq RD^{max}(X) + \sum_{e \in Y \setminus X} u_e - \sum_{e \in X \setminus Y} l_e < RD^{max}(X) \tag{2.16}$$

,ami ellentmond X definíciójának. $\square$

Következmény: Végtelen sok helyett elég csak a $2^{|E|}$ darab extrém szcenáriót vizsgálni, a keresett megoldás ezek optimális megoldásai közül fog kikerülni. Persze ez még mindig exponenciálisan sok példány, a gyakorlatban nem megvalósítható, hogy az összeset végignézzük.

Az előzőekhez hasonlóan itt is adhatunk egy viszonylag egyszerű approximációt.

2.8. Állítás. *Legyen m az átlag-szcenárió, ahol minden elem költsége* $\frac{u_e + l_e}{2}$. *Ennek az optimális megoldása a robosztus feladat 2-approximációját adja, vagyis a robosztus feladat optimális megoldását X-szel, az átlagszcenárióét Y-nal jelölve* $RD^{max}(Y) \leq 2 \cdot RD^{max}(X)$.

Bizonyítás. Legyen $A \in R$ és $B \in R$ két tetszőleges megoldás. Lássuk be a következő egyenlőtlenséget:

$$RD^{max}(B) \geq \sum_{e \in B \setminus A} u_e - \sum_{e \in A \setminus B} l_e \tag{2.17}$$

A definíciókból azt kapjuk, hogy

$$RD^{max}(B) = c_B^{B-} - c_*^{B-} \geq c_B^{B-} - c_A^{B-} \tag{2.18}$$

Könnyen ellenőrizhető, hogy

$$c_B^{B-} - c_A^{B-} = \sum_{e \in B \setminus A} u_e - \sum_{e \in A \setminus B} l_e \tag{2.19}$$

,ami az előzővel együtt (2.17)-t adja.

Most be fogjuk látni, hogy tetszőleges $A \in R$ megoldásra $RD^{max}(Y) \leq$ $\leq 2 \cdot RD^{max}(A)$, és így persze az X robosztus optimális megoldásra is.

Y definíció szerint az átálagszcenárió mellett optimális, ezért bármely $A \in R$ megoldásra $c_Y^m \leq c_A^m$, vagyis

$$\frac{1}{2} \cdot \left(\sum_{e \in Y} u_e + \sum_{e \in Y} l_e \right) \leq \frac{1}{2} \cdot \left(\sum_{e \in A} u_e + \sum_{e \in A} l_e \right) \tag{2.20}$$

Amit átrendezve

$$\sum_{e\in Y\setminus A} u_e - \sum_{e\in A\setminus Y} l_e \leq \sum_{e\in A\setminus Y} u_e - \sum_{e\in Y\setminus A} l_e \tag{2.21}$$

adódik. Innen (2.10), (2.21) és (2.17) alkalmazásával azt kapjuk, hogy

$$RD^{max}(Y) \leq RD^{max}(A) + \sum_{e\in Y\setminus A} u_e - \sum_{e\in A\setminus Y} l_e \leq \tag{2.22}$$

$$\leq RD^{max}(A) + \sum_{e\in A\setminus Y} u_e - \sum_{e\in Y\setminus A} l_e \leq \tag{2.23}$$

$$\leq 2 \cdot RD^{max}(A) \tag{2.24}$$

Ezt akartuk bizonyítani. □

Megjegyzés: A teljesség kedvéért ellenőrizzük egy példával a becslés élességét. Legyen az alapprobléma csak egy minimális elem kiválasztása három elemből. A költség-intervallumok legyenek [1,1] , [0,2] és [2,0]. Az átlag-szcenárióra mindhárom elem optimális, a legnagyobb robosztus eltérések 1, 2 és 2.

2.3.3. Bonyolultságelméleti eredmények

Bonyolultságelméleti eredmények az explicit költségű változathoz: rögzített k szcenárió esetén Kouvelis és Yu kimutatta az NP-nehézséget a feszítőfa, párosítás és hátizsák problémára, Yu és Yang pedig a legrövidebb út problémára. Aissi, Bazgan és Vanderpooten megmutatták ugyanezt a minimális s-t vágásra. Ha k az input része, akkor ezek mind erősen NP-nehézzé válnak.

Megjegyzés: A diszkrét min-max eltérés legrövidebb út probléma NP-teljességét ugyanazzal a példával be lehet látni, amit a min-max esetben használtunk. Ehhez vegyük észre, hogy a gráfban mindkét szcenárió mellett a legolcsóbb út költsége 0.

Az implicit (intervallum alakú) költségek esetén erős NP-teljességet bizonyítottak a legrövidebb út és feszítőfa (Averbakh és Lebedev), továbbá a párosítás és s-t vágás (Aissi, Bazgan és Vanderpooten) problémákra. A minimális vágás (kijelölt s,t csúcsok nélkül) viszont polinomidőben megtalálható.

Min-max eltérés kritérium			
Probléma	Diszkrét, k szcenárióval		Intervallum
	k konstans	k az input része	
Legrövidebb út	Gyengén NP-nehéz	Erősen NP-nehéz	Erősen NP-nehéz
Legolcsóbb feszítőfa	Gyengén NP-nehéz	Erősen NP-nehéz	Erősen NP-nehéz
Párosítás	NP-nehéz	Erősen NP-nehéz	Erősen NP-nehéz
Minimális s-t vágás	Erősen NP-nehéz	Erősen NP-nehéz	Erősen NP-nehéz
Minimális vágás	Polinomiális	Erősen NP-nehéz	Polinomiális
Hátizsák-feladat	Gyengén NP-nehéz	Erősen NP-nehéz	Gyengén NP-nehéz

2.4. Néhány szó a sztochasztikus megközelítésről

Eddig végig olyan modellekről volt szó, ahol nem feltételeztük valószínűségi eloszlás ismeretét a lehetőségeken. Ez a gyakorlatban sokszor tükrözi is a valóságot. Mi van akkor, ha mégis ismerünk egy eloszlást az S halmazon, és szeretnénk a várható érték minimumát megkeresni?

2.9. Állítás. *Ez a feladat megoldható az alapfeladat egy példányának megoldásával, amikor minden elem a várható értékét kapja költségül.*

Bizonyítás. a várható érték linearitását felhasználva $E_s(c_r^s) = E_s(\sum_{e \in r} c_e^s) = $
$= \sum_{e \in r} E_s(c_e^s)$, vagyis egy r halmaz költségének várható értéke egyenlő az elemei várható értékeinek az összegével. Ebből már következik az állítás. $\square$

Megjegyzés: érdemes megjegyezni, hogy a sztochasztikus modell nem követi ugyanazt a filozófiát, mint a robosztus. A robosztus optimalizálásnál arra törekszünk, hogy a megoldásunk minden kimenetel esetén megállja a helyét. A várható érték maximalizálása csak akkor kifizetődő modell, ha a költséget nem csak egyszer, hanem rendszeresen ki kell fizetni, és kizárólag a hosszútávú eredmény számít. Vagyis nem probléma, ha alkalomadtán nagyon költséges eredményt kapunk.

3. fejezet

A feladat egészértékű lineáris programként

Az intervallum változatú minimális robosztus eltérés feladatot fel tudjuk írni $\{0,\ 1\}$-értékű lineáris programként.

Ehhez először az eredeti feladatot írjuk fel ilyen alakban. A megoldást reprezentáló $x \in \{0{,}1\}^{|E|}$ vektor i-edik eleme 1, ha az i-edik elemet bevesszük a megoldásba, és 0, ha nem. A költségfüggvény ekkor egy minden s szcenárió mellett egy $c^s \in \{0{,}1\}^{|E|}$ vektorral írható le, és cx az x megoldás költsége.

$$\min cx \tag{3.1}$$

$$x_i \in \{0, 1\} \quad \forall i \in \{1, 2 \ldots |E|\} \tag{3.2}$$

$$Ax \leq b \tag{3.3}$$

A robosztus változatot definíció szerint az alábbi programmal írhatjuk le:

$$\min_{x_1} \qquad \max_{c}(cx_1 - \min_{x_2} cx_2) \tag{3.4}$$

$$Ax_2 \leq b \tag{3.5}$$

$$x_2 \in \{0, 1\} \tag{3.6}$$

$$l \leq c \leq u \tag{3.7}$$

$$Ax_1 \leq b \tag{3.8}$$

$$x_1 \in \{0, 1\} \tag{3.9}$$

Ez (2.9) alapján átírható

$$\min_{x_1}(ux_1 - \min_{x_2} c^{x_1-} x_2) \tag{3.10}$$

$$Ax_2 \leq b \tag{3.11}$$

$$x_2 \in \{0,1\} \tag{3.12}$$

$$Ax_1 \leq b \tag{3.13}$$

$$x_1 \in \{0,1\} \tag{3.14}$$

alakba, ahol c^{x_1-} az x_1 legkedvezőtlenebb szcenáriójának a költségvektora. (Ez $l+(u-l)x$ -szel egyenlő.) Keressünk az alapproblémára olyan felírást, ahol az optimum megegyezik a relaxált feladat optimumával, vagyis az optimális megoldás külön kikötés nélkül egy $\{0,1\}$-értékű vektor. Ekkor a belső min-probléma egy közönséges LP-feladat, amire a dualitástételt alkalmazva a

$$\min_{x_1}(ux_1 - \max_{y}\{y \ : \ yA = c\}) \Leftrightarrow \tag{3.15}$$

$$\Leftrightarrow \min_{x_1,y}(ux_1 - y) \tag{3.16}$$

optimalizálandó kifejezést nyerjük. Vegyük észre, hogy a külső min-hez tartozó változókra továbbra is explicite ki kell kötni az egészértékűséget, ugyanis az eredeti feladat automatikus egészértékűsége nem biztosítja ugyanezt a robosztus megoldásra is. Összességében tehát a

$$\min_{x,y}(ux - y) \tag{3.17}$$

$$Ax \leq b \tag{3.18}$$

$$yA = l + (u-l)x \tag{3.19}$$

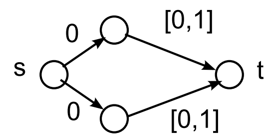
$$y \leq 0 \tag{3.20}$$

$$x \in \{0,1\} \tag{3.21}$$

$$\tag{3.22}$$

programot kell megoldanunk.

Az egészértékű lineáris program nem oldaható meg polinomidőben, hacsak nem $P = NP$. Viszont léteznek olyan számítógépes programok, amik az ilyen feladatokat próbálják meg minél hatékonyabban megoldani.



3.1. ábra. A legrövidebb s-t út folyamproblémaként automatikusan egészértékű lesz, a robosztus optimális megoldás azonban nem feltétlenül. A fenti két út legnagyobb robosztus eltérése 1, de az optimumot a mindkét úton $\frac{1}{2}$ nagyságú folyam adja.

3.1. Legrövidebb út

Legyen G = (V,E) irányított gráf. A legrövidebb s-t út probléma LP-változata:

$$\min c_{ij}x_{ij} \tag{3.23}$$

$$\forall i \sum_{j\in V} x_{ij} - \sum_{j\in V} x_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & egyébként \end{cases} \tag{3.24}$$

$$x \geq 0 \tag{3.25}$$

feltételezve, hogy a gráf nem tartalmaz negatív kört. (A megoldások között nem csak utak szerepelnek, hanem az (út) $\cup$ (néhány kör) alakú objektumok, és egy élt többször is bevehettünk. Ha nincs negatív kör, akkor a körök elhagyásával $\min cx$ értéke nem nő, ezért az optimális megoldások között ott lesz az egyszerű s-t út is. Ha van negatív kör, akkor a poliéder nem korlátos. Ezzel az esettel most nem foglalkozunk. Mivel egy 1 értékű s-t folyamként írtuk le, és a kapacitások egészek, ezért a megoldás is egész.

A duálisa:

$$\max(y_t - y_s) \tag{3.26}$$

$$\forall i,j \quad y_j - y_i \leq c_{ij} \tag{3.27}$$

Ezek alapján a robosztus LP-feladat

$$\min_{x,y}(ux - y_t) \tag{3.28}$$

$$y_j \leq x_j + l_{ij} + (u_{ij} - l_{ij})x_{ij} \;\; \forall (i,j) \in E \tag{3.29}$$

$$\forall i \sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & egyébként \end{cases} \tag{3.30}$$

$$y_s = 0 \tag{3.31}$$

$$y_i \geq 0 \;\; \forall i \tag{3.32}$$

$$x_{ij} \in \{0,1\} \;\; \forall i,j \tag{3.33}$$

3.2. Legolcsóbb feszítőfa

A legolcsóbb feszítőfa egyik LP-modelljében a feszítőfát úgy fogjuk fel, mint a gráf egy tetszőleges csúcsából (ez kapja az 1-es sorszámot) indított $(n-1)$ nagyságú folyamot (n a gráf csúcsszáma), amiből minden csúcs elnyel 1 egységet. Továbbra is az x vektor tárolja, hogy mely élek kerülnek be a gráfba, f_{ij} pedig az i csúcsból a j csúcsba menő folyam nagysága. Most jelöljük E-vel az irányítatlan értelemben vett élhalmazt, és A-val, ha az éleket oda-vissza megirányítjuk.

$$\min cx \tag{3.34}$$

$$\forall i \in V \sum_{(i,j) \in A} f_{ij} - \sum_{(j,i) \in A} f_{ji} = \begin{cases} n-1 & ha \;\; i = 1 \\ -1 & \forall i \in V \setminus \{1\} \end{cases} \tag{3.35}$$

$$f_{ij} \leq (n-1)x_{ij} \;\; \forall \{i,j\} \in E \tag{3.36}$$

$$f_{ji} \leq (n-1)x_{ij} \;\; \forall \{i,j\} \in E \tag{3.37}$$

$$\sum_{e \in E} x_{ij} = n - 1 \tag{3.38}$$

$$f \geq 0 \tag{3.39}$$

$$x_e \in \{0,1\} \;\; \forall e \tag{3.40}$$

Ezt más alakban is felírhatjuk. Gondoljunk most úgy a feszítőfára, hogy az egyes csúcsokba indított folyamokat külön-külön tartjuk számon. f_{ij}^k jelöli a k-ba indított folyam nagyságát az (i,j) élen. Yaman, Karasan és Pinar adták a következő felírást:

$$\min c_{ij}(y_{ij} + y_{ji}) \tag{3.41}$$

$$\sum_{(j,1)\in A} f^k_{j,1} - \sum_{(1,j)\in A} f^k_{1,j} = -1 \quad \forall k \in V\backslash\{1\} \tag{3.42}$$

$$\sum_{(i,j)\in A} f^k_{j,i} - \sum_{(j,i)\in E} f^k_{i,j} = 0 \quad \forall k, i \in V\backslash\{1\}, k \neq i \tag{3.43}$$

$$\sum_{(j,k)\in A} f^k_{j,k} - \sum_{(k,j)\in A} f^k_{k,j} = 1 \quad \forall k \in V\backslash\{1\} \tag{3.44}$$

$$f^k_{ij} \leq y_{ij} \quad \forall (i,j) \in A \; és \; \forall k \in V\backslash\{1\} \tag{3.45}$$

$$\sum_{(i,j)\in A} y_{ij} = n - 1 \tag{3.46}$$

$$f \geq 0 \; és \; y \geq 0 \tag{3.47}$$

Szintén Yamantól és társszerzőitől származik az állítás, miszerint a lineáris programnak {0,1}-értékű megoldásai vannak. Ennek a feladatnak a duálisa:

$$max \sum_{k\in V\backslash\{1\}} (\alpha^k_k + \alpha^k_1) - (n-1)\mu \tag{3.48}$$

$$\sigma^k_{ij} \geq \alpha^k_j - \alpha^k_i \quad \forall (i,j) \in A \quad \forall k \in V\backslash\{1\} \tag{3.49}$$

$$\sum_{k\neq 1} \sigma^k_{ij} + \mu \leq c_{ij} \quad \forall \{i,j\} \in E \tag{3.50}$$

$$\sum_{k\neq 1} \sigma^k_{ji} + \mu \leq c_{ij} \quad \forall \{i,j\} \in E \tag{3.51}$$

$$\sigma, \alpha \geq 0 \tag{3.52}$$

Ezt és a legelső változatot felhasználva kapjuk a robosztus változat lineáris progjamját. Csak a második változat felhasználásával is dolgozhattunk volna, de így egyszerűbb alakot kapunk.

$$\min \left(ux - \sum_{k=V\setminus\{1\}} (\alpha_k^k - \alpha_1^k) - (n-1)\mu \right) \tag{3.53}$$

$$\sigma_{ij}^k \geq \alpha_j^k - \alpha_i^k \quad \forall (i,j) \in A \quad \forall k \in V\setminus\{1\} \tag{3.54}$$

$$\sum_{k\neq 1} \sigma_{ij}^k + \mu \leq l_{ij} + (u_{ij} - l_{ij})x_{ij} \quad \forall \{i,j\} \in E \tag{3.55}$$

$$\sum_{k\neq 1} \sigma_{ji}^k + \mu \leq l_{ij} + (u_{ij} - l_{ij})x_{ij} \quad \forall \{i,j\} \in E \tag{3.56}$$

$$\sum_{(i,j)\in A} f_{ij} - \sum_{(j,i)\in A} f_{ji} = \begin{cases} n-1 & ha\ i = 1, \\ -1 & \forall i = V\setminus\{1\} \end{cases} \tag{3.57}$$

$$f_{ij} \leq (n-1)x_{ij} \quad \forall \{i,j\} \in E \tag{3.58}$$

$$f_{ji} \leq (n-1)x_{ij} \quad \forall \{i,j\} \in E \tag{3.59}$$

$$\sum_{e\in E} x_e = n-1 \tag{3.60}$$

$$f, \sigma, \alpha \geq 0 \tag{3.61}$$

$$x_e \in \{0,1\} \quad \forall e \in E \tag{3.62}$$

Hasonlóan próbálhatjuk meg felírni a többi feladatra is. Egy TU-mátrix például garantálhatja az egészértékűséget, vagy ha a feladatot visszavezetjük minimális költségű, egész nagyságú s-t folyamkeresésre egész kapacitássokal.

3.3. Előfeldolgozás

A $\{0,1\}$ értékű lineáris programozás NP-nehéz feladat. Bár a legújabb megoldóprogramok ehhez képest jó futásidőket adnak, fontos, hogy a változók és egyenlőtlenségek számát a minimálisra csökkentsük. Ezért a lineáris program végrehajtása előtt érdemes megkeresni azokat az elemeket, amik biztosan benne lesznek, illetve biztosan nem lesznek benne a robosztus optimális megoldásban. Ezzel elhagyhatjuk a hozzájuk tartozó változókat és egyenleteket.

Definíció. *Gyenge megoldásnak nevezzük azokat a megoldásokat, amik legalább egy szcenárióra nézve optimálisak.*

Definíció. *Gyenge elemeknek (gráfelméleti feladatban gyenge éleknek) nevezzük az olyan elemeket, amik része egy gyenge megoldásnak.*

3.1. Állítás. *A robosztus optimális megoldás gyenge megoldás.*

Bizonyítás. Ez a (2.7) állítás egyenes következménye, miszerint a robosztus optimális megoldás mindig optimális a saját legkedvezőbb szcenárióján. □

3.2. Állítás. *Egy r megoldás gyenge akkor és csak akkor, ha optimális a saját legkedvezőbb szcenáriójára ($r+$).*

Bizonyítás. Ha r optimális $r+$-ra, akkor definíció szerint gyenge. Másrészt, ha r gyenge, akkor létezik egy s szcenárió, amire $\forall r' \in R : c_r^s \leq c_{r'}^s$.

$$\Rightarrow \sum_{e \in r \setminus r'} c_e^s + \sum_{e \in r \cap r'} c_e^s \leq \sum_{e \in r' \setminus r} c_e^s + \sum_{e \in r' \cap r} c_e^s \tag{3.63}$$

$$\Rightarrow \sum_{e \in r \setminus r'} c_e^s \leq \sum_{e \in r' \setminus r} c_e^s \tag{3.64}$$

$$\Rightarrow \sum_{e \in r \setminus r'} l_e \leq \sum_{e \in r' \setminus r} u_e \tag{3.65}$$

$$\Rightarrow \sum_{e \in r \setminus r'} l_e + \sum_{e \in r \cap r'} l_e \leq \sum_{e \in r' \setminus r} u_e + \sum_{e \in r \cap r'} l_e \tag{3.66}$$

$$\Rightarrow c_r^{r+} \leq c_{r'}^{r+} \tag{3.67}$$

Ezt akartuk bizonyítani. □

Következmény: Ha egy megoldás nem gyenge, akkor nem lehet optimális a robosztus feladatra. Ha egy elem nem gyenge, akkor nem része a robosztus optimális megoldásnak.

Definíció. *Egy elem erős, ha minden szcenárió mellett része egy optimális megoldásnak. Egy elem szigorúan erős, ha minden szenárió mellett minden optimális megoldásnak része.*

Yaman, Karasan és Pinar közös cikkükben bebizonyították, hogy robosztus optimális megoldások között mindig van olyan, ami az összes erős élet tartalamzza. A robosztus feladat megoldásánál ezért elég ezekre az esetekre szorítkozni. A bizonyítás eredetileg feszítőfákra szól, de minden egyéb struktúrára is általánosítható. A továbbiakban feltételezzük, hogy egyik költség-intervallum sem elfajuló, azaz $\forall e \in E : l_e \neq u_e$. (Ha az intervallum mégis egy konstans a számmá fajulna, módosítsuk úgy a feladatot, hogy veszünk helyette egy $[a - \varepsilon, a + \varepsilon]$ intervallumot, ahol $\epsilon > 0$ tetszőlegesen kicsi pozitív szám.)

3.3. Állítás. *Mindig létezik olyan robosztus megoldás, ami az összes erős elemet tartalmazza.*

A bizonyításhoz szükség van néhány lemmára.

A továbbiakban az *egyedüli optimális megoldás* azt fogja jelenteni, hogy a megoldás (az adott szcenárió mellett) optimális, és nincs rajta kívül másik ilyen tulajdonságú.

3.4. Állítás. *Ha egy r megoldás nem az egyedüli optimális megoldás az ő legkedvezőbb szcenáriójában ($r+$), akkor létezik egy olyan $\hat{r} \neq r$, amire $c_r^s \geq$ $\geq c_{\hat{r}}^s \ \forall s \in S$. Ha r nem gyenge, akkor szigorú egyenlőtlenség teljesül minden eseteben.*

Bizonyítás. Legyen $\hat{r} \neq r$ az $r+$ szcenárió egy optimális megoldása. Ekkor $\forall s \in S$:

$$c_r^s - c_{\hat{r}}^s = \sum_{e \in r \setminus \hat{r}} c_e^s - \sum_{e \in \hat{r} \setminus r} c_e^s \geq \sum_{e \in r \setminus \hat{r}} l_e - \sum_{e \in \hat{r} \setminus r} u_e = c_r^{r+} - c_{\hat{r}}^{r+} \geq 0 \qquad (3.68)$$

Az állítás második feléhez a (3.2) állítás alapján ha r nem gyenge, akkor nem optimális a $r+$ szcenárión, ezért $c_r^{r+} > c_{\hat{r}}^{r+}$. Az előbbi egyenlőtlenségekből már következik az állítás. □

3.5. Állítás. *Ha $r \neq \hat{r}$, akkor létezik olyan s szcenárió, ahol $c_r^s \neq c_{\hat{r}}^s$.*

Bizonyítás. Vegyük az L szcenáriót, vagyis ahol minden elem a lehetséges legkisebb költségét kapja. Ha $c_r^L \neq c_{\hat{r}}^L$, akkor készen vagyunk. Ha $c_r^L =$ $= c_{\hat{r}}^L$, akkor r-nek van olyan eleme, ami nincs $\hat{r}$-ban. (Vagy fordítva.) Ennek az elemnek a költségét megnövelve r költsége nő, $\hat{r}$-é viszont nem, ezért a költségek már nem lesznek egyenlőek. □

3.6. Állítás. *Ha r egyik szcenárióban sem az egyedüli optimális megoldás, akkor létezik egy olyan $\hat{r} \neq r$, amire $c_r^s \geq c_{\hat{r}}^s \ \forall s \in S$, úgy, hogy $\hat{r}$ az egyedüli optimális megoldás valamelyik szcenárióban.*

Bizonyítás. Ha r nem az egyedüli optimális megoldás minden szcenárióban, akkor (3.4) szerint van olyan $r_1 \neq r$, amire $c_r^s \geq c_{r_1}^s \ \forall s \in S$. Ha r_1 az egyedüli optimális megoldás valamelyik szcenárióban, akkor készen vagyunk. Ha nem, akkor r_1-re alkalmazva a lemmát, egy hasonló tulajdonságú r_2 létezését kapjuk, vagyis amire $c_{r_1}^s \geq c_{r_2}^s \ \forall s \in S$. A (3.5) lemma szerint $r_2 \neq r$, mert különben $\forall s \in S : c_{r_2}^s = c_{r_1}^s = c_r^s \Rightarrow r_1 = r$ következne.

Ha r_2 jó, akkor készen vagyunk, ha nem, akkor hasonlóan létezik egy, az eddigiektől különböző r_3 is, és így tovább. A megoldások száma véges (E elemű alaphalmaz esetén legfeljebb $2^{|E|}$), ezért az utóbbi lehetőség csak véges sokszor fordulhat elő. Ennek alapján fogunk találni egy megfelelő r_i-t. □

Definíció. *Megoldások egy* Γ *halmazát nevezzük állandónak, ha tetszőleges szcenárióban* Γ *tartalmaz optimális megoldást.*

Definíció. *Megoldások egy* Γ *halmazát nevezzük minimális állandó halmaznak, ha állandó, viszont bármelyik elemét elhagyva már nem az.*

3.7. Állítás. *Legyen* Γ *azon megoldások halmaza, amik egyedüli optimális megoldásai a saját legkedvezőbb szcenáriójuknak. Ekkor* Γ *minimális állandó halmaz.*

Bizonyítás. Először belátjuk, hogy Γ állandó. Indirekt tegyük fel, hogy nem így van. Akkor létezik egy s szcenárió, aminek egyik optimális megoldása sincs Γ-ban. Legyen r egy optimális megoldás s szerint. Mivel $r \notin \Gamma$, ezért r nem egyedüli minimális megoldás $r+$ szerint. Az (3.4) lemma következményeként r egyik szcenárión sem az egyedüli optimális megoldás. (3.6) alapján ekkor létezik egy $\hat{r}$ megoldás, ami legalább egy szcenárióban az egyedüli optimális, és ezen kívül minden szcenárióban legfeljebb annyi a költsége, mint r-nek. A (3.4) lemma következményeként $\hat{r}$ egyedüli optimális $\hat{r}+$ -ban. Tehát $\hat{r} \in \Gamma$.

Azt mondtuk, hogy $\hat{r}$ költsége minden szcenárióban, így s-ben is legfeljebb annyi, mint r-nek. Másrészt r a definíciója szerint optimális s-ben, Γ viszont nem tartalmaz s-re optimális megoldást, ezért $c_r^s < c_{\hat{r}}^s$, ami ellentmondás. Másrészt Γ minimális, mert minden eleme egyedüli minimális megoldás valamely szcenárióra nézve. □

3.8. Állítás. *A minimális állandó halmaz egyértelmű.*

Bizonyítás. Egy minimális állandó halmaz szükségképpen tartalmazza Γ minden elemét, mert azok egyedüli minimális megoldások valamelyik szcenárióra. Más elemet viszont nem tartalmazhat, mert Γ állandó, ezért a többi elem elhagyásával kapott halmaz állandó maradna. □

3.9. Állítás. *Egy elem erős akkor és csak akkor, ha a minimális állandó halmaz összes elemének része.*

Bizonyítás. Ha egy elem erős, akkor definíció szerint minden szcenárió esetén része egy minimális megoldásnak. Másrészt Γ minden eleme egyedüli optimális megoldás valamelyik szcenárión. Ezért egy erős elemnek mindegyikben ott kell lennie. Megfordítva, minden szcenárió esetén Γ elemei között van optimális, ezért ha egy elem mindegyiknek része, akkor erős. □

3.10. Állítás. Γ *elemei között van robosztus optimális.*

Bizonyítás. Tegyük fel, hogy nincsen. Vegyünk akkor egy r robosztus optimális megoldást. Mivel $r \notin \Gamma$, ezért r nem az egyedüli optimális megoldása $r+$ szerint. Ezért az (3.4) lemma alapján egyik szcenáriónak sem lehet az egyedüli optimális megoldása. A (3.6) lemma szerint létezik olyan $\hat{r}$ megoldás, amire $\forall s \in S : c^s_{\hat{r}} \leq c^s_r$ és $\hat{r}$ egyedüli optimális megoldás valamelyik szcenárióra. Az (3.4) lemma szerint ekkor $\hat{r}$ egyedüli optimális megoldás $\hat{r}+$-ra, amiből $\hat{r} \in \Gamma$ következik. Tehát

$$RD^{max}(\hat{r}) = c^{\hat{r}-}_{\hat{r}} - c^{\hat{r}-}_{*} \leq c^{\hat{r}-}_{r} - c^{\hat{r}-}_{*} = RD^{\hat{r}-}(r) \leq RD^{max}(r) \tag{3.69}$$

Mivel r robosztus optimális volt, ez csak úgy lehet, hogy $\hat{r}$ is robosztus optimális. Viszont láttuk, hogy $\hat{r} \in \Gamma$. □

A gyenge és erős elemek kiszűrése feladatonként eltérő nehézségű. A két leginkább vizsgált feladat a legrövidebb út és a legolcsóbb feszítőfa.

3.3.1. Legrövidebb út

A legrövidebb útnál annak eldöntése, hogy egy él gyenge-e, NP-teljes. A szigorúan erős elemeket jellemzi az alábbi állítás:

3.11. Állítás. *Legyen $G = (V, E)$ egy irányított gráf, az éleinek lehetséges költségei $0 \leq l_e \leq c_e \leq u_e$. Jelölje a a kezdőcsútot és b a célcsúcsot. Tegyük fel, hogy a (k, l) él elhagyásával b továbbra is elérhető a-ból. Ekkor (k, l) szigorúan erős él akkor és csak akkor, ha a (k, l) él elhagyásával kapott új gráfban minden q útra és minden $s \in S$ szenárióra $c^s_q > c^s_*$.*

Bizonyítás. Ha (k, l) szigorúan erős, akkor q nem lehet optimális, mert $(k, l) \notin q$. Ezért $c^s_q > c^s_*$. Visszafelé, tetszőleges q út, ami (k, l)-t nem tartalmazza és $\forall s \in S : c^s_q > c^s_*$, az egyik szcenárióra sem lehet optimális. Tehát bármely szcenárióra csak olyan optimális út van, ami (k, l)-t tartalmazza, ezért (k, l) szigorúan erős. □

Megjegyzés: Ha a (k,l) él elhagyása után b nem érhető el a-ból, akkor (k,l) szigorúan erős, mert az összes a-b út ezen az élen halad keresztül.

Használjuk az alábbi jelöléseket:

- U Jelölje ismét a maximum-szcenáriót, ahol minden él a lehető legnagyobb költséget kapja.
- p legyen az U-szcenárió optimális megoldása.
- $(k,l) \in p$ tetszőleges él.
- P^* azon G-beli a-b utak halmaza, amik nem tartalmazzák (k,l)-t.
- q' a P^*-beli utak közül a $p-$ szcenárióra nézve optimális.

3.12. Állítás. *Tegyük fel, hogy (k,l) elhagyása után b még elérhető a-ból. Ha $c_{q'}^{p-} > c_p^{p-}$, akkor (k,l) egy szigorúan erős él.*

Bizonyítás. A fenti jelöléseket és feltételeleket használva

$$\forall q \in P^* : c_q^{p-} \geq c_{q'}^{p-} > c_p^{p-} \tag{3.70}$$

adódik, aminek következményeként

$$\sum_{e \in q \setminus p} l_e = \sum_{e \in q \setminus p} c_e^{p-} > \sum_{e \in p \setminus q} c_e^{p-} = \sum_{e \in p \setminus q} u_e \tag{3.71}$$

Ezért $\forall s \in S$:

$$\sum_{e \in q \setminus p} c_e^s \geq \sum_{e \in q \setminus p} l_e > \sum_{e \in p \setminus q} u_e \geq \sum_{e \in p \setminus q} c_e^s \tag{3.72}$$

Ehhez $\sum\limits_{e \in p \cap q} c_e^s$ -t hozzáadva kapjuk, hogy $\forall s \in S : c_q^s > c_p^s \geq c_*^s$, így a (3.11) állítás szerint (k,l) egy szigorúan erős él. □

Az állítás szerint tehát megkeressük a p utat, kiszámítjuk ebből a $p-$ szcenáriót, majd p éleire egyesével elvégezzük a vizsgálatot. Azaz a vizsgált élt elhagyva a gráfból, összehasonlítjuk a $p-$ szcenárió mellett az optimális út költségét a $c_p^{p-} = \sum\limits_{e \in p} u_e$ értékkel. Ha a q' optimális költsége nagyobb, akkor az elhagyott él szigorúan erős, ezért biztosan benne lesz a robosztus optimális megoldásban. Szintén szigorúan erős az elhagyott él, ha az elhagyása után nem létezik a-b út a gráfban.

3.3.2. Legolcsóbb feszítőfa

A legolcsóbb feszítőfa problémában a gyenge és az erős élek is polinomidőben felismerhetők.

3.13. Állítás. *A legolcsóbb feszítőfa problémában egy e él gyenge akkor és csak akkor, ha a saját legkedvezőbb szcenáriójában (az e él költsége a lehető legkisebb, a többié a lehető legnagyobb) része egy optimális feszítőfának.*

Bizonyítás. Ha e része egy optimális feszítőfának, akkor definíció szerint gyenge. A másik irányhoz használjuk Kruskal algoritmuát. Az algoritmus egy üres halmazból kiindulva élenként építi fel a fát. Élköltség szerint növekvő sorrendben végignézi az éleket, és az adott élt beveszi a halmazba, ha nem alkot kört a már bennlévőkkel. Ismeretes, hogy az így készült feszítőfa minimális lesz. Módosítsuk az algoritmust, hogy döntetlen esetén a szóbanforgó élet részesítse előnyben. (Az eredeti algoritmus ilyenkor szabadon választ, tehát az általánosság csorbítása nélkül ezt megköthetjük.)

Legyen az algoritmus által kiválasztott élek halmaza L. Vegyük észre, hogy ha a Kruskal-algoritmus bevesz egy $\{a, b\}$ élet az L halmazba, akkor L-ben az egy $a \to b$ utat fog adni. Ha nem választja be, akkor az azért volt, mert már volt egy $a \to b$ út L-ben. Következtetésképpen ha az $\{a, b\}$ él olcsóbb e-nél, akkor már lesz egy $a \to b$ út L-ben, mire az algoritmus az e élhez jut. Következmény: e nem része az optimális feszítőfának akkor és csak akkor, ha létezik egy út e két végpontja között, csupa e-nél kisebb költségekkel.

Az $e+$ definíciójából adódóan, ha egy e' él az $e+$ szcenárióban olcsóbb e-nél, akkor a többiben is. Ha $e+$-ban e-t nem választottuk be, akkor volt egy út e két végpontja között, csupa e-nél kisebb költségekkel. De ha ezek az élek $e+$-ban olcsóbbak e-nél, akkor az összes többi szcenárióban is. Ezért e nem lesz része optimális feszítőfának. □

3.14. Állítás. *Egy él erős akkor és csak akkor, ha a legkedvezőtlenebb szcenáriójában része egy optimális feszítőfának.*

A bizonyítás hasonlóan történik, mint a gyenge élekre.

4. fejezet

Egzakt algoritmus a min-max robosztus eltérés legrövidebb útra

4.1. Elméleti háttér

Montemanni és Gambardella nevéhez fűződik ez az algoritmus, ami a legrövidebb út probléma min-max eltérés változatát oldja meg. Az algorimus futásideje legrosszabb esetben exponenciális, de a gyakorlatban sok esetben megtalálja az optimális megoldást viszonylag jó futásidővel.

Jelölés: Jelölje P a vizsgált gráfban az s-t utak halmazát.

Jelölés: Jelölje U azt a szcenáriót, ahol minden él a költségének felső korlátját kapja.

Jelölés: Jelölje p_i az U szcenárióban az i-edik legolcsóbb utat.

Jelölés: Jelölje p' a p út legkedvezőtlenebb szcenáriójának optimális útját.

Az algoritmus alapötlete, hogy az U szcenárió optimális vagy közel optimális megoldásai általában jó eredményeket adnak a robosztus feladatra is. (Ez csak egy heurisztika, semmilyen approximációt nem lehet bizonyítani. Lásd a (2.4) állítás után tett 3. megjegyzést) A módszer az U szerinti legolcsóbb utakat költség szerint nem csökkenő sorrendben tekinti át. Ehhez felhasználjuk, hogy tudunk hatékony algoritmust az első k legolcsóbb út felsorolására.

4.1. Állítás. $c^U_{p_1} \geq c^{p-}_{p'} \quad \forall p \in P.$

Bizonyítás. Mivel p' definíció szerint a $p-$ szcenárió legolcsóbb s-t útja, ezért

$c_{p'}^{p-} \leq c_{p_1}^{p-}$. Továbbá az U definíciója szerint bármelyik út legalább akkora költségű U-ban, mint bármely más szcenárióban. Ezért $c_{p_1}^{p-} \leq c_{p_1}^{U}$. A kettő együtt kiadja az állítást. □

4.2. Állítás. $RD^{max}(p_k) \geq c_{p_i}^U - c_{p_1}^U \quad \forall k \geq i.$

Bizonyítás. A (2.9) képlet és az előző állítás és a definíciók alapján

$$RD^{max}(p_k) = \sum_{e \in p_k} u_e - c_*^{(p_k-)} = c_{p_k}^U - c_{p'_k}^{(p_k-)} \geq c_{p_k}^U - c_{p_1}^U \tag{4.1}$$

□

4.3. Állítás. *Ha* $c_{p_i}^U - c_{p_1}^U \geq \min_{j \in \{1 \dots i\}} RD^{max}(p_j)$, *akkor* $\arg\min_{p \in \{p_1, \dots p_i\}} RDmax(p_j)$ *a robosztus optimális megoldás.*

Bizonyítás. Egyrészt triviális, hogy

$$\min_{j \in \{1 \dots i\}} RD^{max}(p_j) \leq RD^{max}(p_k) \quad \forall k \leq i \tag{4.2}$$

Másrészt az állítások alapján

$$\min_{j \in \{1 \dots i\}} RD^{max}(p_j) \leq c_{p_i}^U - c_{p_1}^U \leq RD^{max}(p_k) \quad \forall k \geq i \tag{4.3}$$

□

Ezzel már megadhatnánk az algoritmust, de előbb még nézzünk meg két szabályt, amivel bizonyos esetekben számítási időt spórolunk:

4.4. Állítás. *Legyen L az a szcenárió, ahol minden él a lehető legolcsóbb, tehát a költség-intervallum alsó határát kapja. Legyen l ennek az optimális megoldása, és tegyük fel, hogy egy* $p \in P$ *útra* $p \cap l = \emptyset$. *Ekkor* $p' = l$, *vagyis l az optimális megoldása a* $p-$ *szcenáriónak is.*

Bizonyítás. Tetszőleges x útra $c_x^L \leq c_x^{p-}$, hiszen utóbbi szcenárióban minden élköltség nagyobb-egyenlő, mint L-ben. Ezt és a definíciókat használva

$$c_l^{p-} = c_l^L \leq c_x^L \leq c_x^{p-} \quad \forall x \in P \tag{4.4}$$

□

Ennek következményeként ilyenkor $RD^{max}(p) = \sum_{e \in p} u_e - \sum_{e \in l} l_e$, így a p robosztus eltérését közvetlenül megkapjuk anélkül, hogy egy legrövidebb út feladatot meg kéne oldanunk. Ez adja az egyik szabályt, amivel javíthatunk az algoritmuson. A másikhoz először be kell látnunk az alábbi lemmát:

4.5. Állítás. *Továbbra is p' jelöli a $p-$ szcenárió optimális útját. Tegyük fel, hogy $i \geq j$ és $p'_j \cap p_i \subseteq p'_j \cap p_j$. Ekkor $c^{p_i-}_{p'_i} \leq c^{p_j-}_{p'_j}$.*

Bizonyítás. A feltétel szerint

$$c^{p_i-}_{p'_j} = \sum_{e \in p'_j \cap p_i} u_e + \sum_{p'_j \setminus p_i} l_e = \tag{4.5}$$

$$= \sum_{e \in (p'_j \cap p_j) \setminus ((p'_j \cap p_j) \setminus (p'_j \cap p_i))} u_e + \sum_{(e \in p'_j \setminus p_j) \cup (p'_j \cap p_j) \setminus (p'_j \cap p_i)} l_e = \tag{4.6}$$

$$= \sum_{e \in p'_j \cap p_j} u_e + \sum_{e \in p'_j \setminus p_j} l_e + \sum_{e \in (p'_j \cap p_j) \setminus (p'_j \cap p_i)} (l_e - u_e) \leq \tag{4.7}$$

$$\leq c^{p_j-}_{p'_j} + 0 \tag{4.8}$$

Vagyis innen

$$c^{p_i-}_{p'_j} \leq c^{p_j-}_{p'_j} \tag{4.9}$$

A definíció szerint

$$c^{p_i-}_{p'_i} \leq c^{p_i-}_{p'_j} \tag{4.10}$$

A kettő együtt adja, hogy

$$c^{p_i-}_{p'_i} \leq c^{p_j-}_{p'_j} \tag{4.11}$$

□

Innen megfogalmazhatjuk az alábbi észrevételt:

4.6. Állítás. *Ha $i > j$ és $p_i \cap p'_j \subseteq p_j \cap p'_j$, akkor $RD^{max}(p_i) \geq RD^{max}(p_j)$.*

Bizonyítás. Az előző lemmából

$$c^{p_i-}_{p'_i} \leq c^{p_j-}_{p'_j} \tag{4.12}$$

Definíció szerint

$$c^{U}_{p_i} \geq c^{U}_{p_j} \tag{4.13}$$

A kettőt összeadva és rendezve

$$RD^{max}(p_i) = c^U_{p_i} - c^{p_i-}_{p'_i} \geq c^U_{p_j} - c^{p_j-}_{p'_j} = RD^{max}(p_j) \tag{4.14}$$

□

4.2. Az algoritmus

Mivel az algoritmus futásideje exponenciális, érdemes a lépések számára egy K határt beiktatni, ami után mindenképp megállunk. Ha eddig nem találtuk meg az optimális megoldást, akkor az addig kapott legjobb eredményt vehetjük közelítő megoldásként.

SP(i) fogja a p'_i utat tárolni, RD(i) a p_i út robosztus eltérése, UB a (4.2) állításban kapott felső korlát $c^U_{p_i} - c^U_{p1}$ értékére, a stop változó pedig jelzi, hogy találtunk-e bizonyítottan optimális megoldást. Az eddig talált legkisebb robosztus eltérésű utat a path változó tártolja.

1. A kezdeti értékek legyenek i:= 1, stop:= hamis, UB:= ∞.
2. Kiszámítjuk az l utat.
3. Amíg $i \leq K$ és stop = hamis:
   a) p_i:= az i-edik legrövidebb út U szerint (megkeressük a megfelelő algoritmussal, közben eltároljuk a $c^U_{p_i}$ értéket)

   b) Ha $\exists p_j, j < i : p_i \cap p'_j \subseteq p_j \cap p'_j$ VAGY $p_i \cap l = \emptyset$:
   $p'_i := l$
   Egyébként: kiszámoljuk a p'_i utat.

   c) RD(i):= $c^u_{p_i} - c^{p_i-}_{p'_i}$

   d) Ha RD(i) < UB, akkor UB:= RD(i), path = p_i.

   e) Ha $c^U_{p_1} - c^U_{p_1} > UB$, akkor stop:= igaz.

   f) i:= i+1

   RETURN path

Megjegyzés: Bár a módszer legrövidebb út keresésére lett kifejlesztve, ezt a tulajdonságot egyedül ott használtuk ki, hogy van gyors algoritmus az i-edik legrövidebb út kiszámítására. Bármilyen feladatra, ahol hasonló eredményt tudunk mutatni, az algoritmus egy az egyben átültethető, a két hatékonyságnövelő szabályt is beleértve.

5. fejezet

Megoldások a diszkrét feladatokhoz

5.1. Approximációk

A robosztus feladatok közeli rokonságban állnak a többszempontú optimalizálási feladatokkal. Utóbbiak a következőt jelentik: van egy optimalizálási feladatunk, de nem egy, hanem k költségfüggvénnyel. Szeretnénk felsorolni a Pareto-optimális megoldásokat. Pareto-optimális egy r megoldás, ha $\nexists r' : \forall k c^k_{r'} \leq c^k_r$ úgy, hogy legalább az egyik helyen szigorú egyenlőtlenség teljsül. Az ilyen megoldásokat más néven hatékonynak nevezik.

5.1. Állítás. *Egy kombinatorikus optimalizálási feladat min-max és min-max eltérés változatainak legalább egy optimális megoldása Pareto-optimális.*

Bizonyítás. Tegyük fel, hogy egy x megoldás dominálja az y megoldást. Ekkor a definíció szerint $\forall s \in S : c^s_x \leq c^s_y \Rightarrow \max_{s\in S} c^s_x \leq \max_{s\in S} c^s_y$, vagyis a hatékony megoldások közül az, amelyikre $\max_{s\in S} c^s_x$ a legkisebb, robosztus optimális. Hasonlót mondhatunk a min-max eltérés feladatra is, ha az első egyetlőtlenség mindkét oldalából levonunk c^s_*-ot. □

Ennek alapján ha a többszempontú optimalizálási feladatra tudunk valamilyen f(|E|)-approximációt, akkor a min-max robosztus feladatra szintén tudunk ilyet. A többszempontú feladat valamilyen α-approximációján azoknak a megoldásoknak a felsorolását érjük, melyeknek a költségét α-val leosztva a megoldás hatékony.

5.2. Állítás. *Ha egy probléma többszempontú optimalizálás változatára létezik* f(|E|)*-approximáció, akkor a min-max robosztus változatra is.*

Bizonyítás. Legyen F a hatékony megoldások $f(|E|)$-approximációjának halmaza. Az előzőekből tudjuk, hogy a hatékony megoldások között van min-max optimális, ezért $\exists y \in F: \forall s \in S: c_y^s \leq f(|E|)c_*^s$. Erre az y-ra ezek szerint $\max_{s\in S} c_y^s \leq \max_{s\in S} f(|E|)c_*^s = f(|E|) \max_{s\in S} c_*^s$, és ezt akartuk bizonyítani. □

Megjegyezzük, hogy a min-max eltérésre ez nem approximálja az optimumot. Az F halmaznak megengedtük az f(|E|)-szeres eltérést, de a robosztus eltérés ettől még lehet tetszőlegesen kicsi, ezért az arányuk nem korlátos.

A min-max eltérésre nézzünk ehelyett egy olyan eredményt, ami bizonyos esetekben lehetővé teszi a min-max változat átültetését erre a feladatra.

5.3. Állítás. *Vegyünk egy P problémát, ahol minden költség nemnegatív, és minden megoldás azonos, t elemszámmal rendelkezik. (Ilyen pl. a feszítőfa, amelyikből mindegyik példány azonos, n-1 darab élt tartalmaz.) Legyen P' a P-nek a kiterjesztése negatív megoldásokat nem, de negatív elemköltségeket megengedő esetre. Ha min max P'-nek van valamilyen* f(|E|)*-approximációja, akkor min max eltérés P -nek is.*

Bizonyítás. Vegyük a P problémának egy I példányát. Számítsuk ki minden szcenárióra a c_*^s értéket. Készítsük el I-nek egy $\overline{I}$ párját, ahol $\overline{c_e^s} := c_e^s - \frac{c_*^s}{t}$. Könnyen belátható, hogy az elemköltségek lehetnek negatívak, de a megoldások nem. Ekkor $\overline{c_r^s} = c_r^s - c_*^s$, tehát I-re a min-max eltérés feladat megoldása megegyzik $\overline{I}$-re a min-max feladat megoldásával. Ezt viszont a feltételünk szerint tudjuk approximálni. □

Approximációs eredmények				
Probléma	k konstans		k nem konstans	
	Min-max	Min-max eltérés	Min-max	Min-max eltérés
Legrövidebb út	fptas	fptas	nem (2-ε)-approx.	nem (2-ε)-approx.
Feszítőfa	fptas	fptas	nem $(\frac{3}{2} - \varepsilon)$-approx.	nem $(\frac{3}{2} - \varepsilon)$-approx.
Hátizsák	fptas	nem approx.	nem approx.	nem approx.

A fptas jelentése: teljesen polinomiális approximációs séma, ahol tetszőlegesen pontos közelítés elérhető polinomiális idő alatt. A táblázatban approx. = approximálható.

5.2. Egzakt megoldás

A diszkrét robosztus eltérés feladat általánosan felírható a következő alakban:

$$\min y \tag{5.1}$$
$$y \geq c_r^s - c_*^s \quad \forall s \in S \tag{5.2}$$
$$r \in R \tag{5.3}$$
$$y \geq 0 \tag{5.4}$$

A min-max feladat hasonlóan implementálható a c_*^s elhagyásával.

Tekintsük a következő relaxált problémát. Legyen $\mu = \mu_1, \mu_2 \dots \mu_k$ egy olyan nemnagatív vektor, ahol $\sum_{i=1}^{k} \mu_i = 1$. Legyen az új probléma

$$\min_{\mu} L(\mu) \tag{5.5}$$
$$L(\mu) = \sum_{s \in S} \mu_i (c_r^s - c_*^s) \tag{5.6}$$
$$r \in R \tag{5.7}$$

Ennek az optimális megoldása nyilván megoldása az eredeti változatnak is, viszont egy könnyebben kezelhető lineáris feladatot kaptunk. Ennek hatékony megoldási módszerei megtalálhatók Kouvelis és Yu *Robust discrete optimization and its applications* című könyvében.

6. fejezet

Összegzés

A dolgozat során áttekintettük a kombinatorikus optimalizálási feladatok robosztus változatait, az erre vonatkozó modelleket, elméleti eredményeket és megoldási módszereket. Mint kiderült, a legtöbb robosztus feladat még polinomidőben megoldható alapprobléma esetén is NP-nehéz. Ennek ellenére számos módszer született a probléma kezelésére, mind approximáció, mind egzakt algoritmus formájában.

Ezúton szeretném megköszönni témavezetőmnek, Jüttner Alpárnak a segítséget a szakdolgozat elkészítéséhez.

Irodalomjegyzék

[1] R. Montemanni, L.M. Gambardella: *An exact algorithm for the robust shortest path problem with interval data*, Computers & Operations Research, Volume 31, Issue 10, 2003.

[2] Hassene Aissi, Cristina Bazgan, Daniel Vanderpooten: *Min-max and min-max regret versions of combinatorial optimization problems: a survey*, European Journal of Operational Research DOI:10.1016/j.ejor.2008.09.012 pp.427-438, 2009.

[3] Hassene Aissi, Cristina Bazgan, Daniel Vanderpooten: *Approximation complexity of min-max (regret) versions of shortest path, spanning tree, and knapsack*, ESA'05 Proceedings of the 13th annual European conference on Algorithms Pages 862-873, 2005.

[4] Hande Yaman, Oya Ekin Karasan, Mustafa C. Pinar: *The robust spanning tree problem with interval data*, Operations Research Letters, 29 (2001) 31-403, 2001.

[5] Thomas L. Magnanti , Laurence A. Wolsey: *Optimal trees*, Center for Operations Research and Econometrics, Univ., 1994.

[6] Gang Yu, Jian Yang: *On the Robust Shortest Path Problem*, Computers and Operations Research, 1997.

[7] Oya E. Karasan, Mustafa C. Pinar, Hande Yaman: *The Robust Shortest Path Problem with Interval Data*, Bilkent University, Department of Industrial Engineering, Technical Report, 2001.

[8] Hassene Aissi, Cristina Bazgan, Daniel Vanderpooten: *Complexity of the min-max (regret) versions of min cut problems*, Lecture Notes in Computer Science Volume 3827, 2005, pp 789-798, 2005.

[9] Adam Kasperski, Pawel Zielinski: *An approximation algorithm for interval data minmax regret combinatorial optimization problems*, Information Processing Letters, Volume 97, Issue 5, 2006.