

# Szakedolgozat

Diszjunkt legrövidebb utak keresése

Zentai Márton

Matematika BSc

Alkalmazott matematikus szakirány

**Témavezető:** Király Zoltán  
egyetemi docens  
ELTE TTK Matematika Intézet  
Számítógéptudományi Tanszék



**ELTE**  
**2022**

# NYILATKOZAT

Név: Zentai Márton

ELTE Természettudományi Kar, szak: alkalmazott matematikus

NEPTUN azonosító: ABATNC

Szakedolgozat címe:

Diszjunkt legrövidebb utak keresése

A **szakedolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.31



---

a hallgató aláírása

# Tartalomjegyzék

<b>Köszönetnyilvánítás</b>	<b>4</b>
<b>1. Bevezetés</b>	<b>5</b>
1.1. Diszjunkt útkeresési problémák története . . . . .	5
1.2. A szakdolgozat célja és felépítése . . . . .	6
1.3. Alap definíciók és jelölések . . . . .	6
<b>2. A bementként kapott gráf kezelése</b>	<b>8</b>
2.1. Felesleges élek és csúcsok elhagyása . . . . .	8
2.2. $G$ összefüggősége . . . . .	8
<b>3. A geometriai megközelítés és tulajdonságai</b>	<b>10</b>
3.1. A megközelítés bemutatása . . . . .	10
3.2. Geometriai jelölések . . . . .	10
3.3. Élek csoportosítása geometriai képük alapján . . . . .	12
3.4. Alapvető tulajdonságok . . . . .	12
3.5. Adott csúcson áthaladó legrövidebb út keresése . . . . .	13
<b>4. Geometriai állítások</b>	<b>15</b>
<b>5. Az algoritmus</b>	<b>22</b>
5.1. Az algoritmus leírása . . . . .	22
5.2. Az algoritmus futásideje . . . . .	26
<b>6. Implementáció</b>	<b>28</b>
6.1. Eltérések . . . . .	28
6.2. Input és output . . . . .	28
6.3. Példák . . . . .	29
<b>Irodalomjegyzék</b>	<b>31</b>

# Köszönetnyilvánítás

Diplomamunkámért köszönettel tartozom témavezetőmnek, Király Zoltánnak aki az elmúlt hónapokban szakértelmével és észrevételeivel segítette munkámat és a konzultációk során rengeteg tanácsot adott. Külön köszönöm, hogy figyelmembe ajánlotta ezt a témát és hogy segítségére mindig számíthattam.

Hálás vagyok a családomnak, akik türelmükkel és támogatásukkal segítették munkámat.

# 1. fejezet

## Bevezetés

A diszjunkt utak keresése alapvető és széles körben tanulmányozott kombinatorikai és gráfelméleti feladat, melynek sok variációja létezik. A  $k$  diszjunkt út probléma ( $k$ -DPP) azt kérdezi, hogy egy adott  $G$  gráfra és  $k$  darab végpont párra :  $(s_i, t_i)_{i \in [k]}$  léteznek-e  $P_1, P_2, \dots, P_k$  diszjunkt utak, úgy hogy  $P_i$  egy  $s_i$ -ben kezdődő,  $t_i$ -ben végződő út.

Felhasználása kiterjed szállítási hálózatok tervezésére, VLSI elrendezések optimalizálására és nagy méretű kommunikációs hálózatokban előforduló útkeresési problémákra.

### 1.1. Diszjunkt útkeresési problémák története

Történetileg fontos és sokat kutatott gráfelméleti probléma az ún. 2-DPP, (2 Disjoint Paths Problem) mely azt kérdezi, hogy létezik-e 2 diszjunkt út előre megadott  $s_1$  és  $t_1$  valamint  $s_2$  és  $t_2$  terminálok között. A feladatnak négy fő változata létezik attól függően, hogy a vizsgált  $G$  gráf irányított vagy irányítatlan, illetve, hogy a keresett utakra csúcs- vagy éldiszjunkságot követelünk meg, a továbbiakban mi csak a csúcsdiszjunkt esettel foglalkozunk, az éldiszjunkt eset erre könnyedén visszavezethető.

Egy általánosabb probléma, a fent említett  $k$ -DPP. Karp 1975-ben megmutatta, hogy  $k$ -DPP NP-nehéz, ha  $k$  az input része és NP-nehéz marad akkor is, ha inputként síkgráfot kapunk [8]. Régebbi kutatások arra fókuszáltak, hogy meghatározzák a probléma mely verziói oldhatók meg polinomiális időben, például Robertson és Seymour [9] megmutatták, hogy létezik  $k$ -DPP feladatot megoldó polinomiális algoritmus (ráadásul  $O(n^3)$  futásidejű) feltéve, hogy  $k$  konstans. Ezen algoritmus a minorokkal kapcsolatos munkásságok egyik mellékterméke. Ezt az eredményt javította nagyjából 16 évvel később Kawarabayashi et al. [10] egy  $O(n^2)$  idejű algoritmus megadásával ( $k$  persze itt is fix).

Sajnos irányított esetben a feladat már  $k = 2$  esetben is NP-nehéz [11], azonban körmentes irányított gráf esetén ismét polinomiális időben megoldható fix  $k$ -ra [11] és lineáris időben megoldható  $k = 2$ -re [2], ez utóbbi rendkívül fontos eredmény a szakdolgozat során ismertetett algoritmus szempontjából.

Térjünk rá, a 2-DPP feladat általunk vizsgált speciális verziójára, mikor az utaktól azt is elvárjuk, hogy legrövidebb utak legyenek, ez az ún. 2-DSPP (Disjoint Shortest Paths Problem). Ezt a verziót Eilam-Tzoref [3] vezette be 1998-ban és meg is mutatta, hogy  $k$ -DSPP is NP-nehéz, ha  $k$  az input része. Ezen túl megadott egy  $O(n^8)$  futásidejű, dinamikus programozást használó algoritmust 2-DSPP azon verziójára, melyben az élek pozitív súlyokkal rendelkeznek. Ezt az eredményt javították 2019-ben függetlenül Gottschau et al. [4], illetve

Kobayashi és Sako [5] egy nemnegatív élsúlyokra is működő, polinomiális algoritmus megadásával. Irányított gráfok esetén a kizárólag pozitív élsúlyokat használó verzióra Bérczi és Kobayashi [12] adott polinom idejű algoritmust. Vegyük észre, hogy nemnegatív élsúlyokat megengedve irányított gráf esetén 2-DSPP általánosítása lesz a 2-DPP feladatnak, az utóbbiról pedig tudjuk, hogy NP-nehéz [11], így ebben az esetben 2-DSPP is NP-nehéz.

A szakdolgozatom egy 2020-as eredményre épül [1], mely szerint 2-DSPP irányítatlan, súlyozatlan gráf esetében megoldható  $O(nm)$  időben, valamint  $k$ -DSPP megoldható  $O(kn^{16k \cdot k! + k + 1})$  időben. Meg kell jegyeznünk, hogy az [1] cikk egy elbírálás és lektorálás nélkül megjelent arXiv-cikk, ettől függetlenül, mint cikk hivatkozunk rá. Az ezt megelőző legfrissebb eredmény Akhmedov [6] munkájából származik, aki 2020-ban adott egy dinamikus programozásra épülő algoritmust mely 2-DSPP feladatot pozitív élsúlyok esetén  $O(n^7)$  időben oldja meg (a korábbi  $O(n^8)$  helyett), a súlyozatlan esetet (amivel itt foglalkozunk) pedig  $O(n^6)$  időben. Ehhez képest az [1] cikkben ismertetett  $O(nm)$  futásidő jelentős javítás, ezért nagyon fontos ez az eredmény. A korábbi megoldások dinamikus programozásra és bonyolult esetszétválasztásokra épülnek, ezt az általunk vizsgált módszer egy elegáns, geometriai szemlélet bevezetésével helyettesíti.

## 1.2. A szakdolgozat célja és felépítése

Bár az [1] cikkben ismertetett eredmény nagyon fontos, maga a leírás sok helyen hiányos és nehezen érthető, valamint több lényeges rész bizonyítása hiányzik. Dolgozatom célja a  $k = 2$  esetre a hiányzó bizonyítások pótlása, az algoritmust érthetőbb leírása és az algoritmus (kicsit módosított) implementációja.

A 2. fejezetben azzal foglalkozunk, hogy miért tehetjük fel, hogy az inputot a következő alfejezetben megadott formában kapjuk.

A 3. fejezetben a fent említett geometriai szemléletet mutatjuk be, bevezetjük az ehhez kapcsolódó jelöléseket és megemlítjük pár alapvető tulajdonságát. Itt esik még szó pár, az algoritmus során gyakran alkalmazott eljárás módjáról és futásidejéről.

A 4. fejezetben kerül sor az [1] cikkben is szereplő, lényeges állítások bizonyítására, ezek szükségesek az algoritmus helyességének igazolásához.

Az 5. fejezetben ismertetjük magát az algoritmust, bizonyítjuk jóságát. Itt bizonyítjuk azt is, hogy futásideje  $O(nm)$ .

Végül a 6. fejezetben esik pár szó az algoritmus implementációjáról.

## 1.3. Alap definíciók és jelölések

2 diszjunkt legrövidebb út probléma (2-DSPP):

**Input:**  $G = (V, E)$  irányítatlan, egyszerű, összefüggő gráf, valamint  $s_1, s_2, t_1, t_2 \in V$  csúcsok.

**Feladat:**  $P_1, P_2$  csúcsdiszjunkt utak megadása, ahol  $P_1$  legrövidebb  $s_1-t_1$  út,  $P_2$  pedig legrövidebb  $s_2-t_2$  út, ha létezik ilyen pár.

Az inputként kapott  $G$  gráfra azt is feltehetjük, hogy bármely él benne van legalább egy legrövidebb  $s_1-t_1$  vagy legrövidebb  $s_2-t_2$  útban, ugyanezt feltehetjük a csúcsokra is.

**1.3.1. Jelölés:**  $[n]$  az  $\{1, 2, \dots, n\}$  halmazt jelöli,  $G = (V, E)$  gráf esetén  $n$  a csúcsok,  $m$  pedig az élek száma, az egyszerűség kedvéért  $V := [n]$ .

Út alatt egy  $v_0v_1 \dots v_\ell$  csúcssorozatot értünk, melyre igaz, hogy semelyik csúcs nem ismétlődik és  $v_{i-1}, v_i$  csúcsok között létezik él  $G$ -ben. Ezen út hossza  $\ell$ . Tetszőleges  $u, v \in V$  csúcsokra  $d(u, v)$  jelölje az  $u, v$  csúcsokat összekötő legrövidebb út hosszát,  $\deg(u)$  jelölje  $u$  csúcs fokszámát. Ha  $P$  egy út és  $u, v \in P$ , akkor  $P[u, v]$  jelölje a  $P$  út  $u$ -tól  $v$ -ig tartó részútját, valamint  $P$  hosszát jelölje  $|P|$ .

Azt mondjuk, hogy  $P_1$  egy 1-út, ha  $P_1$  egy  $d(s_1, t_1)$  hosszú  $s_1$ - $t_1$  út. Hasonlóan  $P_2$  egy 2-út, ha  $P_2$  egy  $d(s_2, t_2)$  hosszú  $s_2$ - $t_2$  út.

Az  $u \in V$  csúcsra azt mondjuk, hogy 1-csúcs, ha létezik olyan 1-út mely tartalmazza  $u$ -t,  $u$  2-csúcs, ha létezik olyan 2-út mely tartalmazza  $u$ -t, végül  $u$  1-2-csúcs, ha  $u$  egyszerre 1-csúcs és 2-csúcs.

Ugyanezt a jelölést az élekre is bevezetjük,  $uv \in E$  él 1-él ha  $\exists P_1$  1-út, hogy  $uv \in P_1$ , 2-él ha  $\exists P_2$  2-út, hogy  $uv \in P_2$  és 1-2-él ha egyszerre 1-él és 2-él. Végül egy csúcs fontos ha 1-csúcs vagy 2-csúcs, egy él fontos ha 1-él vagy 2-él.

*1.3.2. Megjegyzés:* Tehát röviden úgy is mondhatnánk, hogy  $G$ -ről feltehető, hogy csak fontos élekből és csúcsokból áll.

## 2. fejezet

# A bementként kapott gráf kezelése

### 2.1. Felesleges élek és csúcsok elhagyása

A feladat formális kimondása során feltettük, hogy az inputként kapott  $G$  gráf bármely csúcsa és bármely éle fontos. Ebben az alfejezetben vizsgáljuk meg, hogy hogyan kellene eljárunk ha ezt nem tennénk fel.

Nevezzük most az inputként kapott gráfot  $G_0$ -nak, így is jelezve, hogy erre a gráfra csak azt tesszük fel, hogy összefüggő és egyszerű.  $G_0$  csúcs és élhalmaza  $V_0$  és  $E_0$ ,  $u, v \in V_0$  csúcsok  $G_0$ -beli távolságát jelölje  $d_0(u, v)$ . Célunk a  $G_0$  alapján előállítani egy  $G$  már csak fontos csúcsokból és élekből álló gráfot úgy, hogy a 2-DSPP pontosan ugyanakkor oldható meg a  $G_0$  és  $G$  gráfokon. Ezzel támasztjuk alá, hogy a  $G$ -re jogosak a feltételezéseink.

Négy szélességi keresés segítségével  $\forall u \in V_0$  csúcsra meghatározzuk  $d_0(s_1, u)$ ,  $d_0(s_2, u)$ ,  $d_0(u, t_1)$  és  $d_0(u, t_2)$  távolságokat. Ezen távolságok segítségével könnyedén jellemezhetjük az 1-éleket és 2-éleket  $G_0$ -ban.

**2.1.1. Lemma:**  $uv \in E_0$  1-él  $G_0$ -ban  $\Leftrightarrow d_0(s_1, u) + 1 + d_0(v, t_1) = d_0(s_1, t_1)$  vagy  $d_0(s_1, v) + 1 + d_0(u, t_1) = d_0(s_1, t_1)$ .

2-éleket hasonlóan lehet jellemezni, így egy élről konstans időben eldönthető, hogy fontos-e. Könnyű látni azt is, hogy ezután  $G_0$ -ban pontosan azok a csúcsok nem fontosak, amelyekre nem illeszkedik fontos él.

### 2.2. $G$ összefüggősége

$G$  legyen az a gráf melyet a  $G_0$ -ból úgy kapunk, hogy csak a fontos éleket és csúcsokat tartjuk meg, a többit elhagyjuk. Ekkor  $G$ -nek persze bármely csúcsa és éle fontos és teljesül rá, hogy a  $G$  gráfon pontosan akkor oldható meg 2-DSPP, ha a  $G_0$  gráfon megoldható.

*2.2.1. Megjegyzés:* Meg kell említenünk azonban, hogy előfordulhat, hogy  $G$  nem lesz összefüggő (élek és csúcsok elhagyása miatt).

Mivel  $s_1$  és  $t_1$  között létezik legrövidebb út (mivel  $G_0$  összefüggő) és ennek minden éle 1-él, ezért ezek a csúcsok  $G$ -n belül is ugyanazon komponensen belül lesznek, ugyanez igaz  $s_2$  és



$t_2$  csúcsokra. Olyan  $K \subset V_0$  komponense nem lehet  $G$ -nek melyben  $s_1$ ,  $s_2$ ,  $t_1$  és  $t_2$  csúcsok közül egyik sem szerepel, hiszen ekkor  $K$ -ban nem szerepelhet fontos csúcs.

Tehát  $G$  vagy összefüggő vagy 2 komponensből áll úgy, hogy  $s_1$  és  $t_1$  csúcsok az egyik,  $s_2$  és  $t_2$  csúcsok a másik komponensbe esnek. Ekkor 2 darab egymástól független szélességi kereséssel található  $P_1$  1-út és  $P_2$  2-út és ezek szükségképpen csúcsdiszjunktak is lesznek. Ez esetben  $P_1$  és  $P_2$  kiírása után leállhatunk.

$G$ -ről tehát jogosak a feltételezéseink, hiszen  $O(m)$  időben tetszőleges  $G_0$  egyszerű, összefüggő gráfot át tudunk alakítani csak fontos éleket és csúcsokat tartalmazó gráffá, a 2-DSPP megoldhatóságára invariáns módon (vagy a feladatot  $O(m)$  időben a 2.1.2 megjegyzésnek megfelelő módon megoldjuk).

## 3. fejezet

# A geometriai megközelítés és tulajdonságai

### 3.1. A megközelítés bemutatása

A továbbiakban alkalmazott módszerek, illetve az algoritmus átláthatósága érdekében vezessük be egy geometriai leképezést, melynek tulajdonságait és jelöléseit felhasználva nagy mértékben egyszerűsödik az egyébként csak a gráfon definiált feladat, valamint a későbbi állítások nagy része is erősen használja ezen geometriai leképezés ebben a fejezetben ismertetett tulajdonságait.

Helyezzük el a gráf csúcsait a síkon a következő módon: Egy  $u$  csúcs kerüljön a  $(d(u, s_1), d(u, s_2))$  koordinátájú pontba, egy él képe legyen a két végpontjának képét összekötő szakasz. Erre a leképezésre vezessünk be formális jelölést. Legyen  $\varphi : V \rightarrow \mathbb{R}^2$  függvény a csúcsok leképezését megadó függvény, azaz  $\varphi(u) = (d(u, s_1), d(u, s_2))$ .

Ezek az értékek minden csúcsra kiszámolhatók összesen  $O(m + n)$  időben két szélességi keresés segítségével. Még két szélességi kereséssel a  $t_1$  és  $t_2$  pontoktól való távolság is kiszámolható, ezekre azért van szükség, hogy meg tudjuk állapítani mely csúcsok 1-csúcsok vagy 2-csúcsok (élekre ugyanez).

*3.1.1. Megjegyzés:* Ekkor előfordulhat, hogy két különböző gráfcúcs ugyanolyan távolságra van az  $s_1$  és  $s_2$  csúcsoktól, így képük ugyanaz lesz, azaz  $u \neq v$  esetén  $\varphi(u) = \varphi(v)$  lehetséges. Különböző élek képei is eshetnek egybe, viszont  $\varphi(u) \neq \varphi(v)$  esetén persze  $u \neq v$  következik.

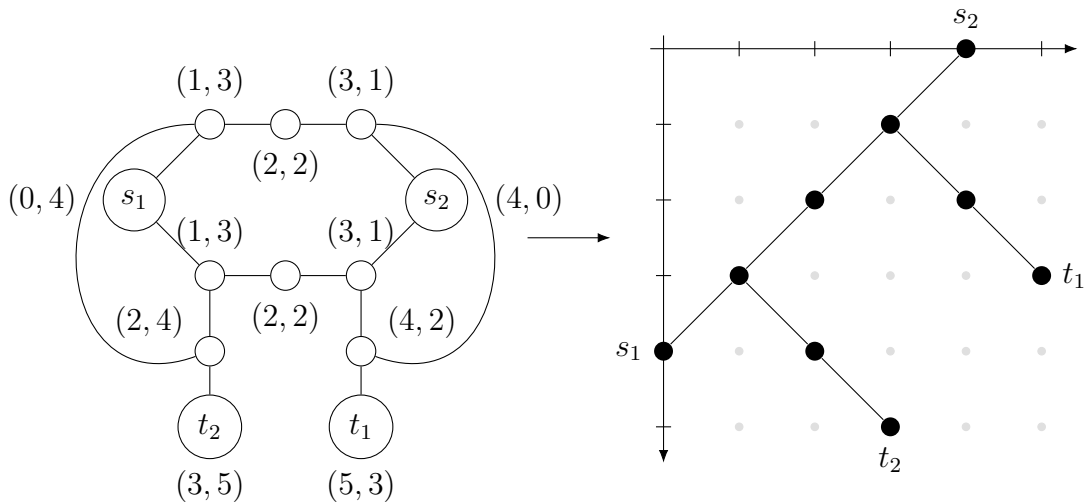
A geometriai leképezés bevezetésével újabb jelölésekre van szükségünk.

### 3.2. Geometriai jelölések

Az átláthatóság érdekében legyen  $\mathbf{u} = \varphi(u)$  és  $\varphi(\mathbf{u})^{-1}$  legyen azon csúcsok halmaza melyekre  $\varphi(u) = \mathbf{u}$ . A  $\mathbf{z} \in \mathbb{R}^2$  pont 1-pont, ha létezik  $u \in V$  1-csúcs, melyre  $\mathbf{u} = \mathbf{z}$ , azaz  $\varphi(\mathbf{z})^{-1}$  tartalmaz 1-csúcsot. Hasonlóan  $\mathbf{z}$  2-pont, ha  $\varphi(\mathbf{z})^{-1}$  tartalmaz 2-csúcsot és  $\mathbf{z}$  1-2-pont, ha 1-pont és 2-pont egyszerre.

$\mathbf{V}$  legyen azon  $\mathbf{z} \in \mathbb{R}^2$  pontok halmaza melyekre  $\varphi(\mathbf{z})^{-1} \neq \emptyset$ , azaz egész koordinátájú pontok (rácspontok) halmaza melyekhez tartozik gráfcúcs. Egy  $\mathbf{z} \in \mathbb{R}^2$  pont első koordinátáját jelölje  $x(\mathbf{z})$ , második koordinátát pedig  $y(\mathbf{u})$ .

$\mathbf{u}, \mathbf{v} \in \mathbf{V}$  esetén  $(\mathbf{u} \diamond \mathbf{v})$  jelölje a  $\{\mathbf{z} \in \mathbb{R}^2 : \|\mathbf{u} - \mathbf{z}\|_\infty + \|\mathbf{z} - \mathbf{v}\|_\infty \leq \|\mathbf{u} - \mathbf{v}\|_\infty\}$  halmazt,



3.1. ábra. Geometriai leképezés bemutatása egy példán

mely nem más, mint egy zárt téglalap, melynek szemközti csúcsai  $\mathbf{u}$  és  $\mathbf{v}$ , valamint oldalai 45-fokos szöget zárnak be a tengelyekkel.

Az  $\mathbf{u} \in \mathbf{V}$  ponthoz definiálunk négy területet, melyeket jelöléseik legyenek  $\mathbf{E}(\mathbf{u})$ ,  $\mathbf{D}(\mathbf{u})$ ,  $\mathbf{NY}(\mathbf{u})$  és  $\mathbf{K}(\mathbf{u})$  (az égtájaknak megfelelően). Ezek legyenek szemléletesen az  $\mathbf{u}$  szögcsúcsú, rendre függőlegesen felfelé és lefelé és vízszintesen balra és jobbra mutató 90-fokos zárt szögtartományok, melyek szögcsúcsai a tengelyekkel 45-fokos szöget zárnak be a 3.2 ábrán illusztrált módon. Formálisan adjunk meg ezeket a halmazokat az egyértelműség érdekében :

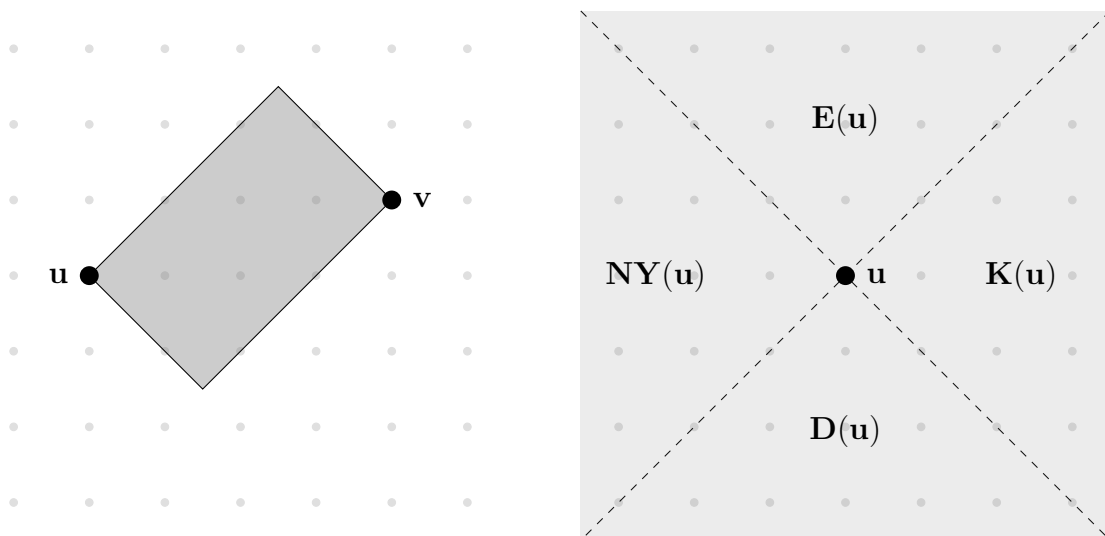
$$\mathbf{E}(\mathbf{u}) = \{\mathbf{z} \in \mathbb{R}^2 : y(\mathbf{u}) - y(\mathbf{z}) \geq |x(\mathbf{z}) - x(\mathbf{u})|\}$$

$$\mathbf{D}(\mathbf{u}) = \{\mathbf{z} \in \mathbb{R}^2 : y(\mathbf{z}) - y(\mathbf{u}) \geq |x(\mathbf{z}) - x(\mathbf{u})|\}$$

$$\mathbf{NY}(\mathbf{u}) = \{\mathbf{z} \in \mathbb{R}^2 : x(\mathbf{u}) - x(\mathbf{z}) \geq |y(\mathbf{z}) - y(\mathbf{u})|\}$$

$$\mathbf{K}(\mathbf{u}) = \{\mathbf{z} \in \mathbb{R}^2 : x(\mathbf{z}) - x(\mathbf{u}) \geq |y(\mathbf{z}) - y(\mathbf{u})|\}$$

Végül ha  $P$  egy út, legyen  $\mathbf{P} \subset \mathbb{R}^2$  az a töröttvonal mely a  $\{\mathbf{z} \in \mathbf{V} : \exists u \in P, \text{ melyre } \mathbf{u} = \mathbf{z}\}$  halmaz pontjait köti össze a  $P$  szerinti sorrend szerint.



3.2. ábra.  $(\mathbf{u} \diamond \mathbf{v})$  és  $\mathbf{E}(\mathbf{u})$ ,  $\mathbf{D}(\mathbf{u})$ ,  $\mathbf{NY}(\mathbf{u})$  és  $\mathbf{K}(\mathbf{u})$  területek szemléltetése

3.2.1. Megjegyzés: Egy  $\mathbf{z} \in \mathbb{R}^2$  pont tehát pontosan akkor 1-pont, ha létezik olyan  $P_1$  1-út,

melyre  $\mathbf{z} \in \mathbf{P}_1$ , azaz a  $P_1$ -hez tartozó töröttvonal áthalad a  $\mathbf{z}$  ponton.

3.2.2. *Megjegyzés:* Előfordulhat, hogy  $\mathbf{u}, \mathbf{v} \in \mathbf{V}$  esetén  $(\mathbf{u} \diamond \mathbf{v})$  egy szakasz lesz csak, ez akkor áll fenn, mikor  $\mathbf{u}$  és  $\mathbf{v}$  ugyanarra az 1 vagy  $-1$  meredekségű egyenesre esnek.

### 3.3. Élek csoportosítása geometriai képük alapján

Soroljuk  $G$  éleit geometriai képük alapján 4 osztályba:

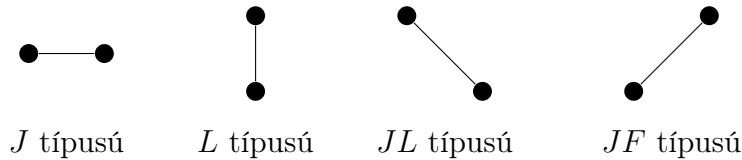
$J$  típusú : Az él végpontjai  $(x, y)$  és  $(x + 1, y)$  pontok, azaz vízszintes (jobbra) él

$L$  típusú : Az él végpontjai  $(x, y)$  és  $(x, y + 1)$  pontok, azaz függőleges (lefelé) él

$JL$  típusú - Az él végpontjai  $(x, y)$  és  $(x + 1, y + 1)$  pontok, azaz jobbra-le irányban álló él

$JF$  típusú - Az él végpontjai  $(x, y)$  és  $(x + 1, y - 1)$  pontok, azaz jobbra-fel irányú él

Olyan él, mely két végpontjának képe megegyezik, nem létezhet  $G$ -ben hiszen ez nem lehetne se 1-él se 2-él (mivel nem növeli a távolságot  $s_1$  vagy  $s_2$  csúcstól).



3.3. ábra. A 4 lehetséges éltípus

3.3.1. *Megjegyzés:* Bármely él valamelyik osztály tagja kell hogy legyen, hiszen szomszédos csúcsok távolsága maximum 1-gyel térhet el egy tetszőleges harmadik csúcstól, így  $s_1$  és  $s_2$  csúcsoktól is, azaz az  $x$  és  $y$  koordináták maximum 1-gyel nőhetnek/csökkenhetnek.

### 3.4. Alapvető tulajdonságok

Mondjuk ki pár könnyen látható tulajdonságát a geometriai leképezésnek. Ezek fő célja az, hogy a későbbi ábrákat alátámasszák (pl.  $s_1$  és  $s_2$  miért úgy helyezkednek el mindig), de párat a későbbi bizonyításokban is felhasználunk.

**3.4.1. Tulajdonság:** Az  $\mathbf{s}_1$  és  $\mathbf{s}_2$  pontok az origótól ugyanolyan távolságra helyezkednek el az  $x$ , illetve  $y$  tengelyeken, mivel  $\mathbf{s}_1 = (0, d(s_1, s_2))$  és  $\mathbf{s}_2 = (d(s_1, s_2), 0)$ . Ezt a két rácspontot a leképezés után mindig egyenes vonal köti össze, mivel  $G$  összefüggő, így létezik legrövidebb  $s_1$ - $s_2$  út, ennek éleinek képe pedig egy  $\mathbf{s}_1\mathbf{s}_2$  szakasz.

**3.4.2. Tulajdonság:** Legyen  $P_1$  1-út,  $P_2$  2-út. Ekkor  $P_1$  csúcsainak képeinek  $x$ -koordinátái szigorúan monoton növekvőek  $P_1$ -en végighaladva  $s_1$ -től  $t_1$ -ig. Ennek oka, hogy az 1-utak  $d(s_1, t_1) + 1$  csúcsból állnak,  $x(\mathbf{s}_1) - x(\mathbf{t}_1) = d(s_1, t_1)$  és 2 szomszédos csúcs  $x$ -koordinátája közötti eltérés maximum 1 lehet.  $P_2$  esetén az  $y$ -koordináták növekednek.

Ezek alapján  $\mathbf{P}_1$  csak  $J, JL$  és  $JF$  típusú,  $\mathbf{P}_2$  pedig csak  $L, JL$  és  $JF$  típusú élekből állhat. Ebből következik, hogy  $\mathbf{P}_1 \cap \mathbf{P}_2$  pedig csak  $JF$  és  $JL$  típusú élekből állhat.

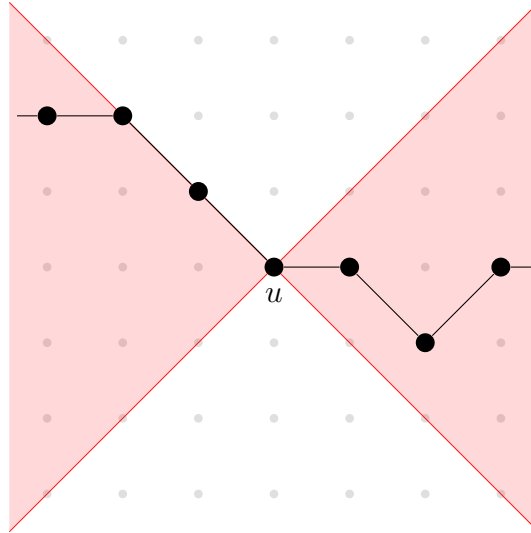
3.4.3. *Megjegyzés:* Ha a metszetben létezik  $JL$  típusú él, azt mindkét út ugyanabban az irányban használja, míg ha a metszetben  $JF$  típusú él létezik akkor azt  $P_1$  jobbra-fel,  $P_2$  pedig balra-le irányokban használják.

3.4.4. **Állítás:** Legyen  $P_1$  1-út,  $u \in P_1$  tetszőleges csúcs. Ekkor  $\mathbf{P}_1[\mathbf{s}_1, \mathbf{u}] \subseteq \mathbf{NY}(\mathbf{u})$  és  $\mathbf{P}[\mathbf{u}, \mathbf{t}_1] \subseteq \mathbf{K}(\mathbf{u})$ . Az állítás persze  $P_2$ -re 2-útra is érvényes,  $\mathbf{E}(\mathbf{u})$  és  $\mathbf{D}(\mathbf{u})$  szögtartományokkal.

**Bizonyítás:** Tudjuk, hogy  $P_1$ -en végighaladva a csúcsok  $x$ -koordinátája szigorúan monoton nő. Mivel a koordináták maximum 1-el változhatnak lépésenként, ezért bármely  $v \in P_1$  csúcsra az  $u$  és  $v$  csúcsok  $y$ -koordinátája közötti eltérés legfeljebb annyi lehet, mint az  $x$ -koordinátájuk közötti eltérés, tehát  $|x(\mathbf{u}) - x(\mathbf{v})| \geq |y(\mathbf{u}) - y(\mathbf{v})|$  egyenlőtlenség áll fenn, azaz bármely  $w \in P_1$  csúcsra amire  $x(\mathbf{u}) \geq x(\mathbf{w})$  az  $x(\mathbf{u}) - x(\mathbf{w}) \leq |y(\mathbf{u}) - y(\mathbf{w})|$  egyenlőtlenség, és bármely  $z \in P_1$  csúcsra amire  $x(\mathbf{z}) \geq x(\mathbf{u})$  az  $x(\mathbf{z}) - x(\mathbf{u}) \geq |y(\mathbf{u}) - y(\mathbf{z})|$  egyenlőtlenség teljesül.

Ezek az egyenlőtlenségek pontosan megfelelnek az  $\mathbf{K}(\mathbf{u})$  és  $\mathbf{NY}(\mathbf{u})$  halmazokat megadó egyenlőtlenségekkel.

A bizonyítás  $P_2$  esetén nagyon hasonló. □



3.4. ábra. Az út többi része csak a pirossal jelölt szögtartományokon belül lehet.

## 3.5. Adott csúcson áthaladó legrövidebb út keresése

Az algoritmus során szükségünk lesz arra, hogy egy adott  $v \in V$  1-csúcsához megadjunk egy olyan  $P_1$  1-utat, hogy  $v \in P_1$ .

3.5.1. **Állítás:** Legyen adott  $v \in V$  1-csúcs, ekkor  $O(m)$  időben megadható egy  $P_1$  1-út, mely tartalmazza a  $v$  csúcsot.

**Bizonyítás:**  $v$  szomszédjai közül könnyen találunk olyan  $u$  és  $w$  1-csúcsokat, melyekre  $x(\mathbf{u}) + 1 = x(\mathbf{v}) = x(\mathbf{w}) - 1$ . A  $v$  csúccsal szomszédos csúcsokon végighaladva ellenőrizzük, hogy az adott csúcs 1-típusú-e, illetve, hogy az  $x$ -koordinátája 1-el nagyobb vagy kisebb-e, mint  $x(\mathbf{v})$ .

Mivel  $u$  1-csúcs, így létezik rajta áthaladó 1-út, így  $u$  és  $w$  is biztosan léteznek, legfeljebb  $\deg(v)$  darab csúcs vizsgálata után meg is találjuk őket.

Ezt hasonló módon folytatjuk, csökkenő  $x$ -koordinátát keresve  $u$ -ból és növekvő  $x$ -koordinátát keresve  $w$ -ból, egészen addig míg elérjük  $s_1$ -et, illetve  $t_1$ -et, a kapott út hossza  $d(s_1, t_1)$  lesz és áthalad a  $v$ -csúcson, ezt volt a célunk.

Összesen legfeljebb  $\sum_{v \in V} \deg(v) = 2 \cdot m$  csúcsot vizsgáltunk meg, egyenként konstans időben, így  $O(m)$  időben találunk kívánt utat.  $\square$

**3.5.2. Következmény:** Ezek alapján  $uv \in E$  1-élhez is könnyen található  $O(m)$  időben 1-út mely tartalmazza  $uv$  élt. Ekkor  $u$  és  $v$  csúcsok 1-csúcsok és feltehetjük, hogy  $x(u) < x(v)$ . Az előző állítás értelmében  $O(m)$  időben találunk  $P'$  és  $P''$  1-utakat rajtuk keresztül, ekkor pedig  $P'[s_1, u]$  és  $P''[v, t_1]$  utak összefűzésével olyan 1-utakat kapunk, mely áthalad az  $uv$  élen.

*3.5.3. Megjegyzés:* Az állítások persze 2-csúcsokkal és 2-úttal is érvényes, a bizonyítások nagyon hasonlóak.

## 4. fejezet

# Geometriai állítások

Ebben a fejezetben kimondunk néhány állítást, melyek elengedhetetlenek az algoritmus helyességének bizonyítása során. Ahogyan az algoritmus is, ezen állítások és bizonyítások is főként a geometriai kép tulajdonságaival foglalkoznak, főként azt vizsgáljuk, hogy ha létezik  $P_1$  1-út és  $P_2$  2-út, melyek csúcdiszjunktak, akkor  $\mathbf{P}_1$  és  $\mathbf{P}_2$ , valamint a  $\mathbf{P}_1 \cap \mathbf{P}_2$  metszet hogyan helyezkedhetnek el síkon.

**4.1. Lemma:**  $\forall u, v \in V : \|\mathbf{u} - \mathbf{v}\|_\infty \leq d(u, v)$ , azaz a síkon, max norma szerinti távolsága két csúcsnak nem lehet nagyobb, mint távolságuk a gráfban.

**Bizonyítás:** Legyen  $P$  egy legrövidebb  $u$ - $v$  út, melynek élhalmaza  $E_P$ . Bármely  $wz$   $E_P$ -beli élre igaz, hogy  $\|\mathbf{w} - \mathbf{z}\|_\infty \leq 1$  (ez triviális), így a háromszög egyenlőtlenség miatt  $\|\mathbf{u} - \mathbf{v}\|_\infty \leq \sum_{e \in E_P} 1 = d(u, v)$ .  $\square$

**4.2. Állítás:** Legyen  $P_1$  egy 1-út, ekkor  $\forall u \in P_1$  csúcs esetén  $\mathbf{u} \in (\mathbf{s}_1 \diamond \mathbf{t}_1)$ , azaz a  $P_1$  út csúcsait  $\varphi$  az  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$  téglalapon belülre képezi.

**Bizonyítás:** A  $P_1$ -beli csúcsok mind 1-csúcsok és egy  $u \in V$  csúcs pontosan akkor 1-csúcs, ha  $d(s_1, u) + d(u, t_1) = d(s_1, t_1)$ , így ezt és az előző lemmát felhasználva  $\|\mathbf{s}_1 - \mathbf{u}\|_\infty + \|\mathbf{u} - \mathbf{t}_1\|_\infty \leq d(s_1, u) + d(u, t_1) = d(s_1, t_1) = \|\mathbf{s}_1 - \mathbf{t}_1\|_\infty$ , így tetszőleges  $u \in P_1$  csúcsra  $\mathbf{u} \in (\mathbf{s}_1 \diamond \mathbf{t}_1)$ .

Egy kevésbé formális, de talán szemléletesebb indoklás lehet a következő. Bármely  $u \in P_1$  csúcsra a 3.4.4 állítás miatt  $\mathbf{u} \in \mathbf{K}(\mathbf{s}_1)$  és  $\mathbf{u} \in \mathbf{NY}(\mathbf{t}_1)$ . Könnyen látható, hogy mivel  $\mathbf{t}_1 \in \mathbf{K}(\mathbf{s}_1)$  és  $\mathbf{s}_1 \in \mathbf{NY}(\mathbf{t}_1)$ , ezért  $\mathbf{K}(\mathbf{s}_1) \cap \mathbf{NY}(\mathbf{t}_1) = (\mathbf{s}_1 \diamond \mathbf{t}_1)$ , így  $\mathbf{u} \in (\mathbf{s}_1 \diamond \mathbf{t}_1)$  is teljesül.  $\square$

*4.3. Megjegyzés:* Ekkor persze  $\mathbf{P}_i \subseteq (\mathbf{s}_i \diamond \mathbf{t}_i)$  is teljesül, hiszen az élek képei csak egyenes szakaszok lehetnek és  $(\mathbf{s}_i \diamond \mathbf{t}_i)$  konvex.

**4.4. Lemma:** Az  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$  és  $(\mathbf{s}_2 \diamond \mathbf{t}_2)$  téglalapok nem lehetnek diszjunktak, azaz  $(\mathbf{s}_1 \diamond \mathbf{t}_1) \cap (\mathbf{s}_2 \diamond \mathbf{t}_2) \neq \emptyset$ .

**Bizonyítás:** Mivel minden él fontos, így minden csúcs vagy él képe az  $(\mathbf{s}_1 \diamond \mathbf{t}_1) \cup (\mathbf{s}_2 \diamond \mathbf{t}_2)$  területen belülre esik, ekkor viszont tetszőleges  $Q$  legrövidebb  $s_1$ - $s_2$  út képét vizsgáljuk (ez létezik, mivel  $G$  összefüggő).  $\mathbf{Q}$  nem más, mint az  $\mathbf{s}_1\mathbf{s}_2$  szakasz és  $\mathbf{Q} \subseteq ((\mathbf{s}_1 \diamond \mathbf{t}_1) \cup (\mathbf{s}_2 \diamond \mathbf{t}_2))$ , így  $(\mathbf{s}_1 \diamond \mathbf{t}_1) \cap (\mathbf{s}_2 \diamond \mathbf{t}_2)$  nem lehet üres.  $\square$

**4.5. Lemma:** Legyen  $P_1$  egy 1-út,  $P_2$  pedig egy 2-út. Ekkor  $\mathbf{P}_1 \cap \mathbf{P}_2$  lehet üres, egyetlen pont vagy egyetlen olyan szakasz mely a tengelyekkel 45-fokos szöveget zár be. Úgy is mondhatnánk,

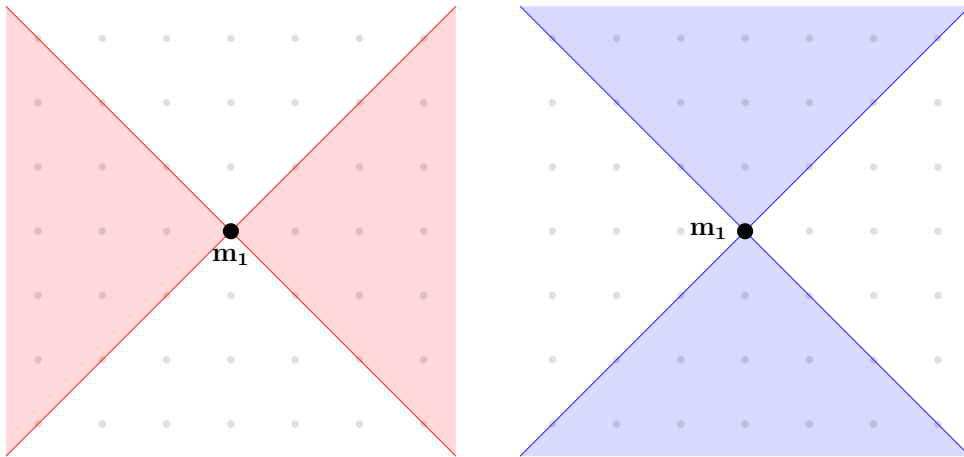
hogy ha  $|\mathbf{P}_1 \cap \mathbf{P}_2| > 1$  akkor nem tartalmazhat egyszerre  $JL$  és  $JF$  típusú élt, továbbá a metszet összefüggő a síkon (A 3.4.2 tulajdonság szerint a metszetben csak  $JL$  és  $JF$  típusú élek lehetnek).

**Bizonyítás:** Tegyük fel, hogy a metszet nem üres és nem is egyetlen pont. Nem nehéz látni, hogy ekkor  $\mathbf{P}_1 \cap \mathbf{P}_2$  tartalmaz legalább 2 darab 1-2-pontot is, legyen egy ilyen pont  $\mathbf{m}_1$  (mivel egy 1-út és egy 2-út metszetéről beszélünk, ezért a metszetben lévő  $\mathbf{V}$ -beli csúcsok mindegyike 1-2-pont. A 3.4.4 állításból következik, hogy  $\mathbf{P}_1 \subseteq \mathbf{NY}(\mathbf{m}_1) \cup \mathbf{K}(\mathbf{m}_1)$  és  $\mathbf{P}_2 \subseteq \mathbf{E}(\mathbf{m}_1) \cup \mathbf{D}(\mathbf{m}_1)$ . Ebből adódik, hogy a metszet tetszőleges  $\mathbf{V}$ -beli pontjára igaz, hogy a metszet többi része csak az adott ponton keresztül haladó 1 és  $-1$  meredekségű egyeneseken helyezkedhet el, hiszen  $\mathbf{P}_1 \cap \mathbf{P}_2 \subseteq (\mathbf{NY}(\mathbf{m}_1) \cup \mathbf{K}(\mathbf{m}_1)) \cap (\mathbf{E}(\mathbf{m}_1) \cup \mathbf{D}(\mathbf{m}_1))$ , ahol a jobb oldal pontosan az előbb említett 2 egyenes, a továbbiakban ezt jelölje  $\mathbf{X}(\mathbf{m}_1)$ .

Legyen az  $\mathbf{m}_2$  a  $\mathbf{P}_1 \cap \mathbf{P}_2$  a metszet egy  $\mathbf{m}_1$ -től különböző 1-2-pontja (ilyen biztosan létezik). Tudjuk, hogy  $\mathbf{P}_1 \cap \mathbf{P}_2 \subseteq \mathbf{X}(\mathbf{m}_2)$  is igaz, így a metszet kénytelen  $\mathbf{X}(\mathbf{m}_1) \cap \mathbf{X}(\mathbf{m}_2)$  halmazon belül lenni, mely már csak egy egyenes lehet.

Innen már az is könnyen adódik, hogy a metszet egyetlen szakasz lehet, mivel ha két 1-2-pont ugyanazon a  $\pm 1$  meredekségű egyenesen helyezkednek el, akkor bármely mindkét pontot tartalmazó  $P$  1-út (vagy 2-út) képe tartalmazza a két pontot összekötő szakaszt is (a monoton  $x$ -, illetve  $y$ -koordináták miatt).

□



4.1. ábra.  $\mathbf{P}_1$  csak a piros,  $\mathbf{P}_2$  pedig csak a kék területen belül helyezkedhet el.

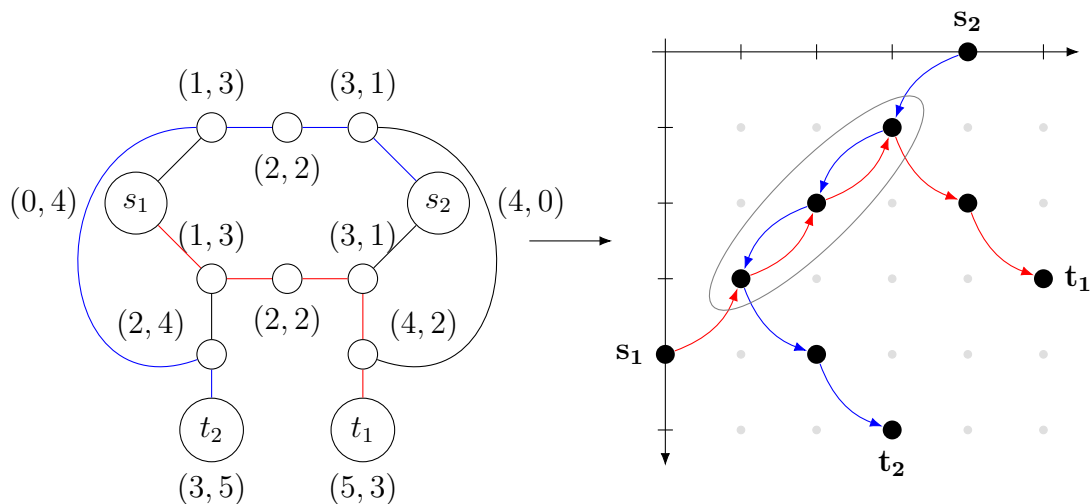
**4.6. Állítás:** Tegyük fel, hogy létezik  $P_1$  1-út és  $P_2$  2-út, melyekre  $\mathbf{P}_1 \cap \mathbf{P}_2$  tartalmaz  $JF$  típusú élt. Ekkor  $s_1^* = s_1$ ,  $s_2^* = t_2$ ,  $t_1^* = t_1$ ,  $t_2^* = s_2$  végpontokat bevezetve, a gráfot most az  $s_1^*$  és  $s_2^*$  csúcsoktól való távolságok alapján ábrázolva, a  $P_1$  út  $\mathbf{P}_1^*$  képének és  $P_2$  út  $\mathbf{P}_2^*$  képének  $\mathbf{P}_1^* \cap \mathbf{P}_2^*$  metszetében már nem lesz  $JF$  típusú él.

**Bizonyítás:** Ha  $u, v \in V$  2-csúcsok akkor  $d(u, s_2) + d(u, s_2^*) = d(s_2, t_2)$  egyenlet teljesül, azaz a csere hatására bármely 2-pont tükröződik az  $x = \frac{d(s_2, t_2)}{2}$  egyenletű egyenesre, tehát bármely 2-élnek megfelelő szakasz is erre az egyenesre tükröződik. Így bármely  $JF$  típusú 2-él a csere után  $JL$  típusú lesz. Mivel feltettük, hogy a  $\mathbf{P}_1 \cap \mathbf{P}_2$  metszet tartalmaz  $JF$  típusú élt és a metszeten belül csak 2-él képe lehet, így ennek az élnek képe tükrözés után  $JL$  típusú lesz. Ezek alapján az új  $\mathbf{P}_1^* \cap \mathbf{P}_2^*$  metszet tartalmaz  $JL$  típusú élt, ekkor pedig a 4.5 lemma alapján  $JF$  típusút már nem tartalmazhat.

□

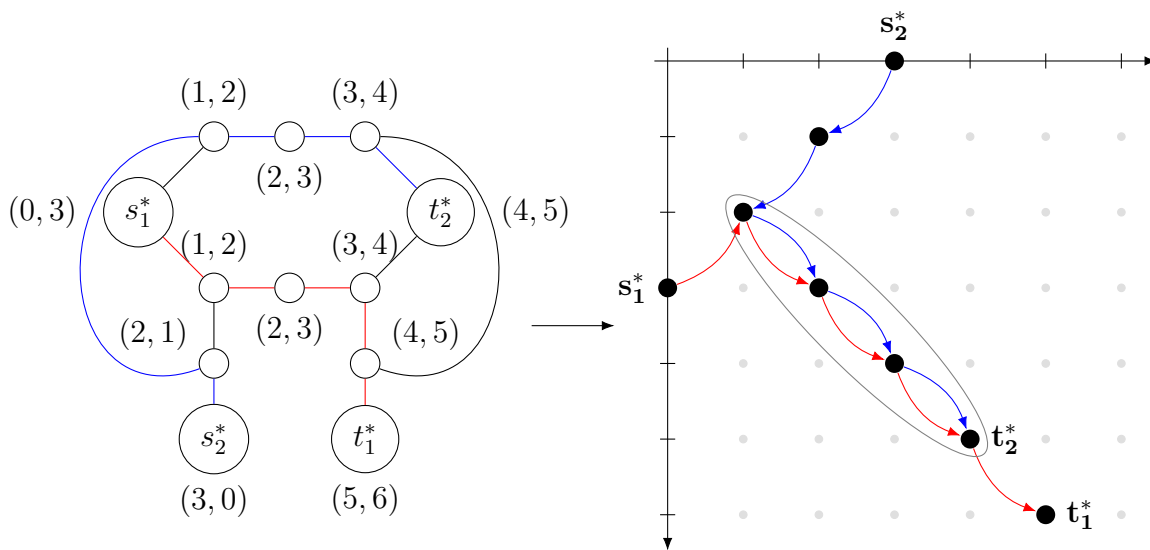


**4.7. Példa:** Nézzünk meg egy példát  $s_2$  és  $t_2$  szerepeinek cseréjére a már korábban is ábrázolt példagráfon.



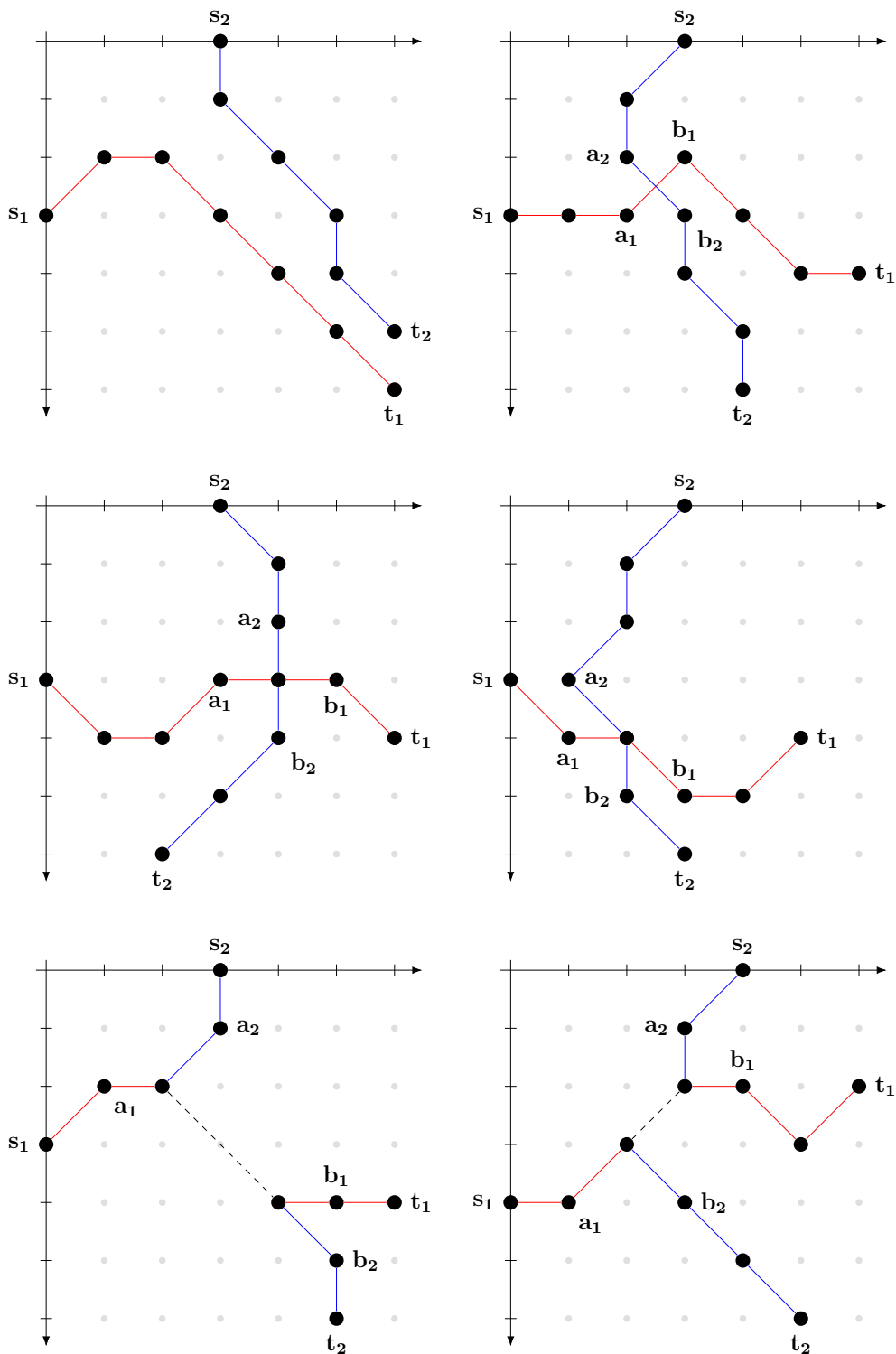
4.2. ábra. Eredeti geometriai kép, pirossal jelöljük  $P_1$ -et, kékkel  $P_2$ -t. A jobb oldalon látható, hogy a metszet csak  $JF$ -típusú élekből áll.

Cseréljük meg most  $s_2$  és  $t_2$  csúcsokat, számoljuk újra a távolságokat. Az alábbi ábrán látszik, hogy az így kapott metszet már nem tartalmaz  $JF$ -típusú élt.



4.3. ábra. Csere után a metszetben nincs  $JF$ -típusú él.

**4.8. Definíció:** Legyen  $P_1$  1-út és  $P_2$  2-út úgy, hogy ez a két út csúcdiszjunkt,  $P_1$ -et az  $s_1 \rightarrow t_1$ ,  $P_2$ -t pedig az  $s_2 \rightarrow t_2$  irányban képzeljük el. Legyen  $\mathbf{M} = \mathbf{P}_1 \cap \mathbf{P}_2 \neq \emptyset$  és legyen  $a_i \in V$  a  $P_i$  út utolsó olyan csúcsa, melynek képe még a metszet előtt helyezkedik el,  $b_i \in V$  pedig az első, aminek képe a metszet után. Ha  $\mathbf{M} \neq \emptyset$ , azaz  $P_1$  és  $P_2$  képeinek metszete nem üres, akkor ezek a csúcsok léteznek.



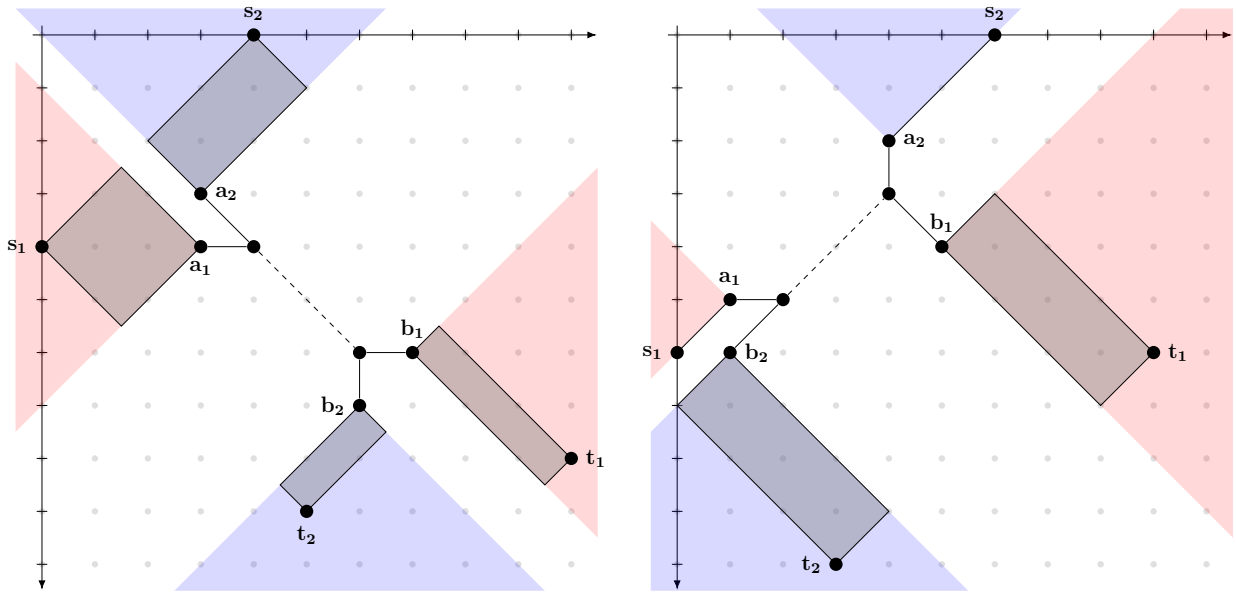
4.4. ábra.  $a_1, a_2, b_1$  és  $b_2$  pontok elhelyezkedése  $P_1 \cap P_2$ -től függően. *Bal-fent* : A metszet üres, így az említett csúcsok nem léteznek. *Jobb-fent* : A metszet nem tartalmaz egész koordinátájú pontot, így egyetlen, négy darab rácspont közötti pontból áll. *Bal-középen és jobb-középen* : A metszet egyetlen rácspontból áll. *Bal-lent és jobb-lent* : A metszet tartalmaz több egész koordinátájú pontot is (azaz  $V$ -beli pontot).

**4.9. Állítás:** Tegyük fel, hogy létezik  $P_1$  1-út és  $P_2$  2-út, melyekre  $P_1 \cap P_2 \neq \emptyset$ . Ekkor a 4.8 definícióban használt jelöléseket használva :  $(s_1 \diamond a_1)$ ,  $(b_1 \diamond t_1)$ ,  $(s_2 \diamond a_2)$  és  $(b_2 \diamond t_2)$  téglalapok

páronként diszjunktak.

**Bizonyítás:** Összesen 6 darab téglalap-párra kellene megmutatnunk, hogy diszjunktak, ezek közül  $(s_1 \diamond a_1) \cap (b_1 \diamond t_1) = \emptyset$  és  $(s_2 \diamond a_2) \cap (b_2 \diamond t_2) = \emptyset$  esetek az  $x$ , illetve  $y$ -koordináták monotonitása miatt könnyen adódnak. A maradék 4-ből szimmetria miatt elegendő csak a  $(s_1 \diamond a_1) \cap (s_2 \diamond a_2) = \emptyset$  esetet belátni, a maradék 3 ehhez hasonló módon bizonyítható.

Könnnyen látható, hogy bármilyen is legyen  $\mathbf{P}_1 \cap \mathbf{P}_2$ ,  $a_2$  az  $a_1$ -en átmenő 1 meredekségű egyenes felett kell, hogy legyen, így  $\mathbf{NY}(a_1)$  és  $\mathbf{E}(a_2)$  diszjunktak. Ekkor  $(s_1 \diamond a_1)$  és  $(s_2 \diamond a_2)$  téglalapok a megfelelő szögtartományokon belülré esnek, így a szögtartományok diszjunktóságából következik a téglalapok diszjunktága is.  $\square$



4.5. ábra. Illusztráció a 4.9 állításhoz. Látható, hogy a téglalapok diszjunkt szögtartományokon belül helyezkednek el, így a téglalapok maguk is diszjunktak lesznek. Csak azon esetet ábráztunk amikor a metszet nem csak egy pontból áll, de persze ezen ábrák alapján az az eset is könnyen elképzelhető. A jobb oldalon  $(s_1 \diamond a_1)$  és  $(s_2 \diamond a_2)$  téglalapok elfajultak szakaszokká.

**4.10. Állítás:** Ha létezik  $P_1$  1-út, illetve  $P_2$  2-út, melyek csúcdiszjunktak és melyekre  $\mathbf{P}_1 \cap \mathbf{P}_2 = \emptyset$  akkor létezik  $u \in V$  csúcs, melyre az  $u$  pont a  $t_1$ -től vízszintesen jobbra vagy a  $t_2$ -től függőlegesen lefelé helyezkedik el.

**Bizonyítás:** Az átláthatóság kedvéért definiáljuk az alábbi halmazokat :

$$\mathbf{T}_1 = \{z \in \mathbb{R}^2 : x(z) > x(t_1) \wedge y(z) = y(t_1)\}$$

$$\mathbf{T}_2 = \{z \in \mathbb{R}^2 : x(z) = x(t_2) \wedge y(z) > y(t_2)\}$$

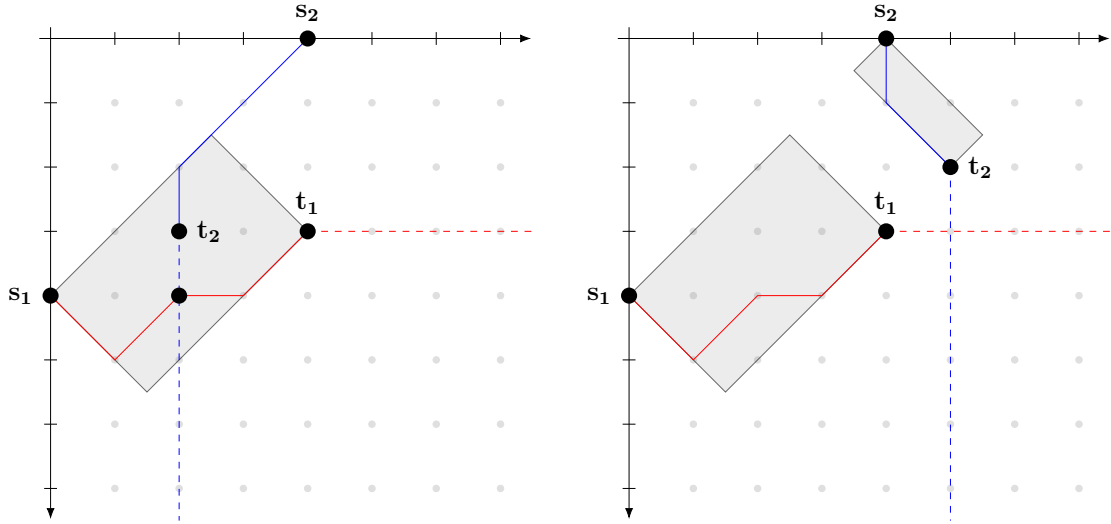
Ekkor  $\mathbf{T}_1 \cap \mathbf{V} \neq \emptyset \Leftrightarrow \exists P_2$  2-út, hogy  $\mathbf{P}_2 \cap \mathbf{T}_1 \neq \emptyset$ , mivel minden 1-pont  $(s_1 \diamond t_1)$  téglalapon belül van, így  $(\mathbf{T}_1 \cap \mathbf{V})$  halmazban csak 2-pontok lehetnek.

A  $\mathbf{T}_2 \cap \mathbf{V} \neq \emptyset \Leftrightarrow \exists P_1$  1-út, hogy  $\mathbf{P}_1 \cap \mathbf{T}_2 \neq \emptyset$  állítás ugyanígy érvényes.

Feltehetjük, hogy  $\mathbf{P}_2 \cap (\mathbf{P}_1 \cup \mathbf{T}_1) = \emptyset$ , különben az előző állítás értelmében  $\mathbf{T}_1 \cap \mathbf{V} \neq \emptyset$  vagy  $\mathbf{P}_1 \cap \mathbf{P}_2 \neq \emptyset$ . Ekkor  $t_2$  nem lehet  $\mathbf{P}_1 \cup \mathbf{T}_1$  töröttvonalon vagy alatta, tehát csak felette lehet. Ekkor  $\mathbf{T}_2 \cap (\mathbf{P}_1 \cap \mathbf{T}_1)$  metszet biztosan nem üres, így vagy  $\mathbf{T}_2 \cap \mathbf{P}_1 \neq \emptyset$  és a bizonyítás elején

szereplő állítás miatt  $\mathbf{T}_2 \cap \mathbf{V} \neq \emptyset$  vagy pedig  $\mathbf{T}_1 \cap \mathbf{T}_2 \neq \emptyset$ , ekkor viszont  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$  és  $(\mathbf{s}_2 \diamond \mathbf{t}_2)$  téglalapok diszjunktak, ami ellentmond a 4.4 lemmának.  $\square$

4.11. *Megjegyzés:* A  $\mathbf{t}_1$  ponttól vízszintesen jobbra csak 2-pontok helyezkedhetnek el, hiszen bármely 1-pont  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$  téglalapon belül van, ugyanígy  $\mathbf{t}_2$  ponttól függőlegesen lefelé csak 1-pontok lehetnek. Ezért volt elég nekünk a 4.10 állításban csak létezését bizonyítani (pl.  $\mathbf{T}_1 \cap \mathbf{V}$  halmazban csak 2-pontok lehetnek).

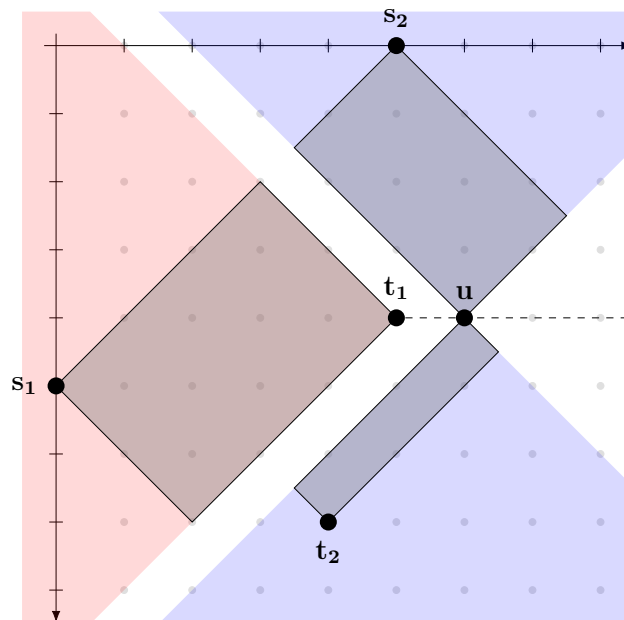


4.6. ábra. Illusztráció a 4.10 állítás bizonyításához.  $\mathbf{T}_1$  és  $\mathbf{T}_2$  halmazokat szaggatott piros és kék vonalak jelölik. Mindkét ábra esetén a  $\mathbf{t}_2$  pont a  $\mathbf{P}_1 \cap \mathbf{T}_1$  töröttvonal felett helyezkedik el. *Bal oldal* : a  $\mathbf{t}_2$  pont a  $\mathbf{t}_1$  ponttól jobbra van, ekkor  $\mathbf{T}_2 \cap \mathbf{P}_1 \neq \emptyset$ . *Jobb oldal* : a  $\mathbf{t}_2$  pont a  $\mathbf{t}_1$  ponttól jobbra van, ekkor pedig  $(\mathbf{s}_1 \diamond \mathbf{t}_1) \cap (\mathbf{s}_2 \diamond \mathbf{t}_2) = \emptyset$ .

4.12. **Állítás:** Ha létezik csúcs, melynek képe  $\mathbf{t}_1$ -től vízszintesen jobbra van, legyen ez  $u$ , akkor  $(\mathbf{s}_2 \diamond u)$  és  $(u \diamond \mathbf{t}_2)$  diszjunktak  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$ -től. Ugyanígy, ha létezik csúcs, melynek képe  $\mathbf{t}_2$ -től függőlegesen lefelé van, legyen ez  $v$ , akkor  $(\mathbf{s}_1 \diamond v)$  és  $(v \diamond \mathbf{t}_2)$  diszjunktak  $(\mathbf{s}_2 \diamond \mathbf{t}_2)$ -től.

**Bizonyítás:** Tekintsük az  $u$  csúcshoz az  $\mathbf{E}(u)$  és  $\mathbf{D}(u)$  szögtartományokat. Az  $(\mathbf{s}_2 \diamond u)$  és  $(u \diamond \mathbf{t}_2)$  téglalapok ezen szögtartományokon belülre esnek,  $\mathbf{E}(u) \cup \mathbf{D}(u)$  pedig diszjunkt  $\mathbf{NY}(\mathbf{t}_1)$  szögtartománytól (ez  $x(u) > x(\mathbf{t}_1)$  és  $y(u) = y(\mathbf{t}_1)$  feltételekből könnyen adódik), mely tartalmazza  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$ -et. Mivel a tartalmazó szögtartományok diszjunktak, a téglalapok is diszjunktak.  $\square$

4.13. **Következmény:** Ezek szerint a 4.10 állítás megfordítása is igaz, azaz ha  $\mathbf{T}_1 \cap \mathbf{V}$  vagy  $\mathbf{T}_2 \cap \mathbf{V}$  nem üres akkor létezik  $P_1$  1-út és  $P_2$  2-út, melyek csúcdiszjunktak és  $\mathbf{P}_1 \cap \mathbf{P}_2 = \emptyset$ .



4.7. ábra. A 4.12 állítás szemléltetése, a  $\mathbf{t}_1$ -től jobbra lévő  $\mathbf{u}$  ponthoz illesztett  $(\mathbf{s}_2 \diamond \mathbf{u}$  és  $(\mathbf{u} \diamond \mathbf{t}_2)$  téglalapok diszjunktak  $(\mathbf{s}_1 \diamond \mathbf{t}_1)$  téglalappal.

## 5. fejezet

# Az algoritmus

### 5.1. Az algoritmus leírása

Ebben a fejezetben ismertetem a módosított algoritmust, mely a cikkben leírt algoritmus egy leegyszerűsített változata. A leírás során azt is belátjuk az előző fejezetben szereplő állítások segítségével, hogy ha létezik megoldása a feladatnak, akkor az algoritmus is talál megoldást, azaz csak akkor nem találunk megoldást, ha valóban nincs is.

**5.1.1. Algoritmus:** Az algoritmus 3 fő lépésből áll, a helyességet lépésenként bizonyítjuk.

1. lépés: Ebben a lépésben keressünk csúcdiszjunkt  $P_1$  1-utat és  $P_2$  2-utat úgy, hogy  $\mathbf{P}_1 \cap \mathbf{P}_2 = \emptyset$ .

Tudjuk, hogy a 4.10 állítás és megfordítása is igaz, tehát pontosan akkor létezik  $P_1$  1-út és  $P_2$  2-út, melyek csúcdiszjunktak és  $\mathbf{P}_1 \cap \mathbf{P}_2 = \emptyset$  ha létezik  $u \in V$  csúcs, melyre  $\mathbf{u} \in \mathbf{T}_1 \cup \mathbf{T}_2$ . Ilyen  $u$ -t keressünk úgy, hogy minden csúcson végighaladva ellenőrizzük, hogy az adott csúcs képe  $\mathbf{t}_1$ -től jobbra vagy  $\mathbf{t}_2$ -től lefelé van-e.

Ha nem találunk ilyen csúcsot, akkor tehát nincs is olyan megoldás melyet ebben a lépésben keressünk, az algoritmus a 2. lépéssel folytatódik.

Ha találunk ilyen csúcsot akkor a következő módon járunk el. Ha olyan  $u \in V$  csúcsot találunk, hogy az  $\mathbf{u}$  pont a  $\mathbf{t}_1$  ponttól vízszintesen jobbra van akkor a 4.12 állítás alapján egy tetszőleges, az  $u$  csúcsot tartalmazó 2-út és egy tetszőleges 1-út szükségképpen csúcdiszjunktak, mivel képeik diszjunktak. Tetszőleges 1-utat könnyen találunk egy szélességi kereséssel és az  $u$  csúcson átmenő 2-utat is találunk, a 3.5.1 állítás alapján. Hasonló módon járhatunk el, ha az  $u \in V$  csúcs képe a  $\mathbf{t}_2$  ponttól függőlegesen lefelé van.

Az így talált megoldás (ami a 4.10 állítás megfordítása szerint létezik) kiírása után algoritmusunk le is állhat.

2. lépés: Ebben a lépésben keressünk csúcdiszjunkt  $P_1$  1-utat és  $P_2$  2-utat úgy, hogy  $\mathbf{P}_1 \cap \mathbf{P}_2 \neq \emptyset$ , de a metszetben nincsen egész koordinátájú pont.

Könnyű látni, hogy ekkor a metszet egyetlen  $\mathbf{z} \in \mathbb{R}^2$  pontot tartalmaz, mely 4 rácspont között helyezkedik el (azaz  $(i + \frac{1}{2}, j + \frac{1}{2})$  alakú, ahol  $i, j \in \mathbb{Z}^+$ ). Ekkor a 4.8 definíciójában szereplő  $a_1, a_2, b_1, b_2$  csúcsok léteznek és képeik  $\mathbf{z}$ -hez képest kétféleképpen helyezkedhetnek el, ezek az 5.1 ábrán vannak szemléltetve. Ha léteznek ilyen utak és ilyen  $\mathbf{z}$  akkor  $\mathbf{s}_1 \in \mathbf{NY}(\mathbf{z})$ ,  $\mathbf{s}_2 \in \mathbf{E}(\mathbf{z})$ ,  $\mathbf{t}_1 \in \mathbf{K}(\mathbf{z})$  és  $\mathbf{t}_2 \in \mathbf{D}(\mathbf{z})$  ( $\mathbf{z} \notin \mathbf{V}$ , de a szögtartományok  $\mathbb{R}^2$ -beli pontokra is értelmezhetőek).



5.1. ábra.  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{d}_1$  és  $\mathbf{d}_2$  pontok kétféle lehetséges elhelyezkedése  $\mathbf{z}$ -hez képest.

Haladjunk sorban végig az összes gráfcsúcson, az éppen vizsgált csúcsot jelölje  $u_1$ . Jelölje ekkor  $\mathbf{z}^*$  az  $(x(\mathbf{u}_1) + \frac{1}{2}, y(\mathbf{u}_1) + \frac{1}{2})$  pontot, azaz  $\mathbf{u}_1$  a  $\mathbf{z}^*$  ponthoz képest balra-lent lévő rácspon.

Egy  $u_1$  csúcs vizsgálata 2 részből áll :

1. rész : Ha  $u_1$  1-csúcs akkor keressünk szomszédai között olyan  $v_1$  1-csúcsot melyre

$$\begin{aligned} x(\mathbf{v}_1) &= x(\mathbf{u}_1) + 1 \\ y(\mathbf{v}_1) &= x(\mathbf{u}_1) - 1 \end{aligned}$$

tehát az  $u_1v_1$  él képe jobbra-fel irányban keresztezi  $\mathbf{z}^*$  pontot. Ezután keressünk olyan  $u_2$  és  $v_2$  2-csúcsokat, hogy  $u_2v_2 \in E$  és  $u_2v_2$  képe jobbra-le irányban halad át a  $\mathbf{z}^*$  ponton. Ezt például az összes élt egyenként kipróbálva is kereshetjük. Ha  $v_1$  csúcs vagy  $u_2v_2$  él nem létezik a kívánt tulajdonságokkal akkor  $\mathbf{z}^*$  nem lehet egy 1-út és egy 2-út képeinek egyetlen nem egész pontból álló metszete, így a 2. részre lépünk tovább, ha pedig  $u_1, v_1, u_2$  és  $v_2$  csúcsok léteznek a kívánt módon akkor a 3.5.2 következmény alapján kereshetünk 1-utat  $u_1v_1$  élen át és 2-utat  $u_2v_2$  élen át. A 3.4.4 állítást felhasználva ekkor az 1-út a  $\mathbf{NY}(\mathbf{u}_1) \cup \mathbf{K}(\mathbf{v}_1)$  területen belül halad az  $\mathbf{u}_1\mathbf{v}_1$  szakasz kivételével, a 2-út pedig az  $\mathbf{E}(\mathbf{u}_2) \cup \mathbf{D}(\mathbf{v}_2)$  területen belül helyezkedik el, az  $\mathbf{u}_2\mathbf{v}_2$  szakasz kivételével, ezek a szögtartományok pedig 4.9 állítás alapján diszjunktak. A két út képe tehát nem metszi egymást egész pontban, így szükségképpen csúcdiszjunktak is.

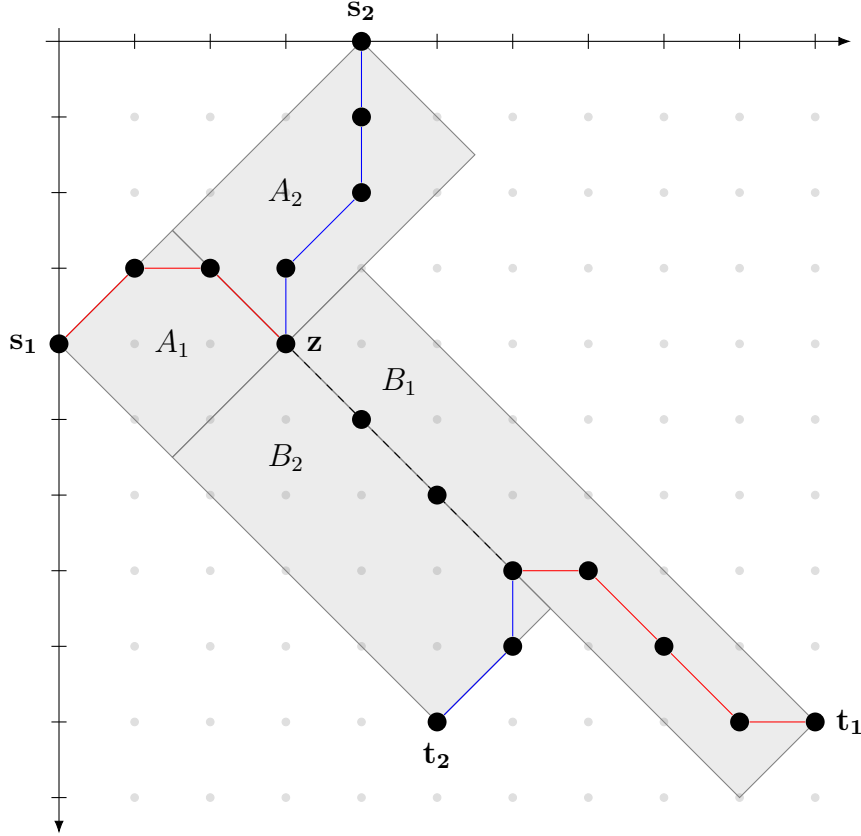
2. rész : Akkor hajtjuk végre az algoritmus ezen részét, ha  $u_1$  2-csúcs. Az előző részhez nagyon hasonlóan járunk el, csak most  $v_1$  2-csúcs,  $u_2$  és  $v_2$  pedig 1-csúcsok kell, hogy legyenek. Ha ezek közül bármelyik csúcs keresése sikertelen akkor az algoritmus a 3. lépéssel folytatódik, ha találtunk megfelelő csúcsokat akkor pedig az 1. részhez hasonlóan találunk megoldást is. Ha létezik olyan megoldás, melyet a 2. lépés során keresünk, akkor lesz olyan  $\mathbf{z}^*$  az összes csúcson végighaladva, hogy a  $\mathbf{z}^*$  pont éppen a megoldást jelentő útpár metszete, ekkor pedig meg is találjuk ezt az útpárt, az algoritmus a megoldás kiírása után leáll. Tehát ha a 2. lépésben nem találunk megfelelő útpárt akkor tudjuk, hogy a megoldás csak olyan lehet, hogy a két útpár egész pontban is metszi egymást.

3. lépés: Ebben a lépésben keressünk csúcdiszjunkt  $P_1$  1-utat és  $P_2$  2-utat úgy, hogy  $\mathbf{P}_1 \cap \mathbf{P}_2$  tartalmaz egész koordinátájú pontot.

Ezt a lépést 2-szer futtatjuk le, egyszer az eredeti  $s_1, s_2, t_1,$  és  $t_2$  csúcsokkal, egyszer pedig  $s_2$  és  $t_2$  csúcsok szerepeit megcserélve, így a 4.6 állítás alapján ha létezik olyan megoldás, hogy az utak képei egész pontban is metszik egymást, akkor a 2 futtatás közül legalább az egyikben ezen metszet nem tartalmaz  $JF$  típusú élt, a továbbiakban tehát feltehetjük, hogy  $\mathbf{P}_1 \cap \mathbf{P}_2$  nem tartalmaz  $JF$  típusú élt.

Legyen  $\mathbf{z}$  a  $\mathbf{P}_1 \cap \mathbf{P}_2$  metszet azon pontja, melynek  $x$ -koordinátája legkisebb, ez létezik és egyértelmű. Itt is megjegyezzük, hogy ekkor  $\mathbf{s}_1 \in \mathbf{NY}(\mathbf{z}), \mathbf{s}_2 \in \mathbf{E}(\mathbf{z}), \mathbf{t}_1 \in \mathbf{K}(\mathbf{z})$  és  $\mathbf{t}_2 \in \mathbf{D}(\mathbf{z})$ . Tegyük fel, hogy  $\mathbf{z}$ -t ismerjük, ezt az előző esethez hasonlóan "ki tudjuk találni", ami

azt jelenti, hogy az összes csúcsot kipróbáljuk. Vizsgáljuk meg, hogy az  $(s_1 \diamond z)$ ,  $(s_2 \diamond z)$ ,  $(z \diamond t_1)$ ,  $(z \diamond t_2)$  téglalapokon belüli éleket az utak milyen irányban használják, jelölje ezeket a téglalapokat az egyszerűség kedvéért rendre  $A_1$ ,  $A_2$ ,  $B_1$  és  $B_2$ . Ekkor például azon  $uv$  éleket, melyekre  $u \in A_1 \setminus B_2$  és  $v \in A_1$  csak  $P_1$  használhatja, a növekvő  $x$ -koordináta irányába.



5.2. ábra. A 3. lépésben előforduló területek szemléltetése, csak az  $A_1 \cap B_2$  és  $A_2 \cap B_1$ -ben elhelyezkedő élek irányítása nem egyértelmű, ezeket használhatná  $P_1$  vagy  $P_2$  is.

Hozzunk létre négy darab irányított gráfot, (jelölje őket  $D_i, i \in [4]$ ) a következő módon :  $D_i$  csúcsai legyenek  $G$  csúcsai és  $D_i$ -ben akkor menjen él  $u$  csúcsból  $v$  csúcsba, ha  $uv \in E$ , valamint:

- (1)  $x(\mathbf{u}) + 1 = x(\mathbf{v})$
- (2)  $\mathbf{u} \in A_1 \setminus B_2 \wedge \mathbf{v} \in A_1$  vagy pedig  $\mathbf{u} \in B_1 \wedge \mathbf{v} \in B_1 \setminus A_2$

vagy

- (1)  $y(\mathbf{u}) + 1 = y(\mathbf{v})$
- (2)  $\mathbf{u} \in A_2 \setminus B_1 \wedge \mathbf{v} \in A_2$  vagy pedig  $\mathbf{u} \in B_2 \wedge \mathbf{v} \in B_2 \setminus A_1$

Fontosak még az  $A_1 \cap B_2$  és  $A_2 \cap B_1$ -be képződő élek (melyeknek mindkét végpontja ezekben a metszetekben van). Mivel a  $\mathbf{P}_1 \cap \mathbf{P}_2$  metszet kizárólag  $JL$  típusú éleket tartalmazhat és  $\mathbf{z}$  a legkisebb  $x$ -koordinátájú pont,  $(\mathbf{P}_1 \cap \mathbf{P}_2) \subseteq B_1 \cap B_2$  adódik, így a fent említett élek képei nem lehetnek egyszerre  $\mathbf{P}_1$  és  $\mathbf{P}_2$  töröttvonalak részei. Ez azt jelenti, hogy két olyan élre melyek végpontjai például  $A_1 \cap B_2$  halmazba képződnek, nem lehet igaz, hogy egyiket  $P_1$  másikat  $P_2$  használja (ugyanaz igaz az  $A_2 \cap B_1$  halmazra). Így ezen halmazok éleit ugyanolyan irányítással vehetjük be  $D_i$ -be, így négy különböző irányítást kell kipróbálnunk. Itt fog eltérni a négy létrehozott gráfunk,  $D_1$ -ben mutasson él  $u$  csúcsból  $v$  csúcsba, ha  $uv \in E$ , valamint :



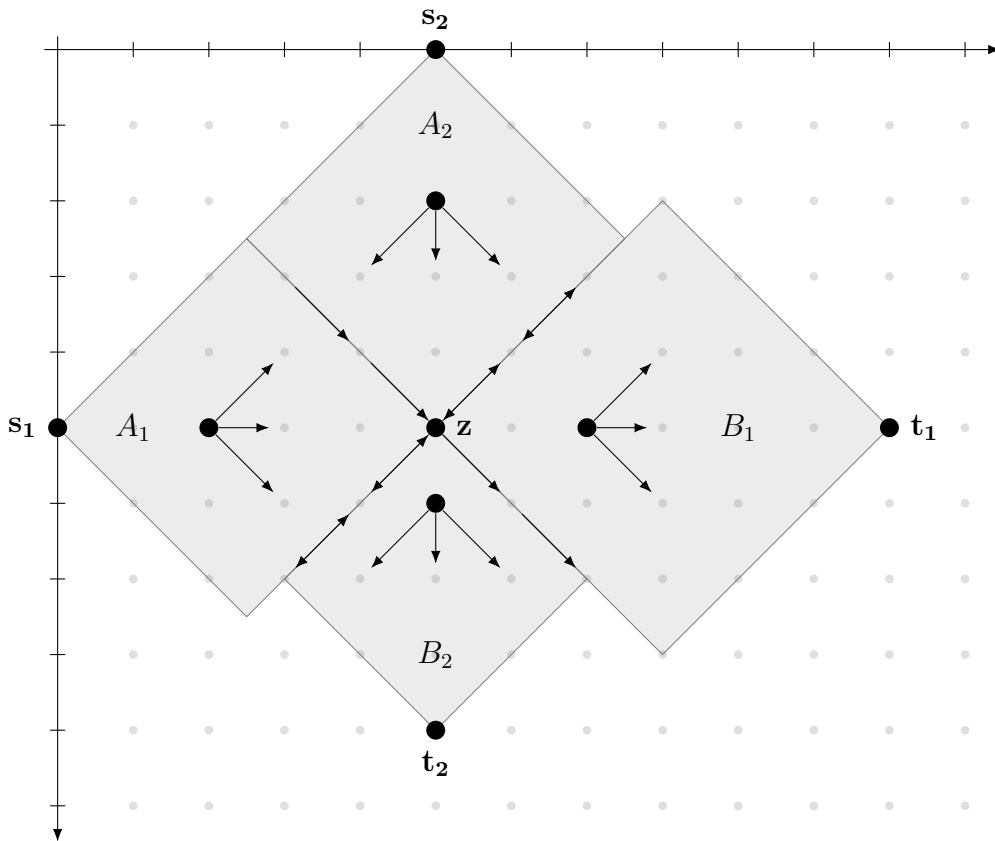
(1)  $x(\mathbf{u}) + 1 = x(\mathbf{v})$

(2)  $\mathbf{u}, \mathbf{v} \in A_1 \cap B_2$  vagy pedig  $\mathbf{u}, \mathbf{v} \in A_2 \cap B_1$

$D_2, D_3$  és  $D_4$  gráfokat az  $A_1 \cap B_2$  és  $A_2 \cap B_1$ -beli élek további 3 lehetséges irányítása szerint egészítjük ki.

Ekkor  $D_i$  irányított gráfok körmentesek és bármely  $s_1-t_1$  út  $D_i$ -ben egy 1-út  $G$ -ben, bármely  $s_2-t_2$  út  $D_i$ -ben egy 2-út  $G$ -ben, ezeket a tulajdonságokat az 5.1.2 és az 5.1.3 állításokban bizonyítjuk.

Így Tholey algoritmusát [2] (mely irányított körmentes gráfban oldja meg a 2-DPP feladatot) felhasználva, ha létezik megoldás akkor valamely  $i \in [4]$ -re  $D_i$ -ben diszjunkt  $s_1-t_1$  és  $s_2-t_2$  utakat keresve, találunk megoldást az eredeti problémára. Olyan megoldást sosem találunk, melyben az utak képeinek metszete tartalmaz  $JF$  típusú élt.



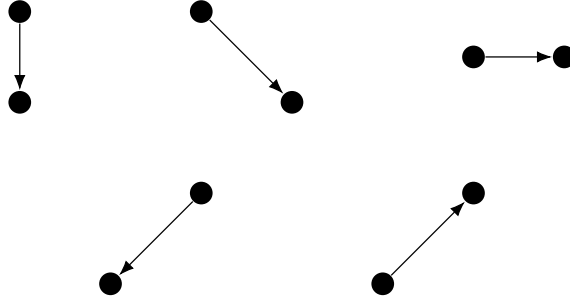
5.3. ábra. A 3. lépés során bevezetett irányítás. Csak a  $z$ -n áthaladó jobbra-fel irányú egyenesre eső élekre nem egyértelmű, hogy milyen irányban adjuk hozzá  $D_i$ -hez

Ha létezik tehát  $P_1$  1-út és  $P_2$  2-út melyek csúcdiszjunktak, akkor beleesnek a fenti 3 lépés során vizsgált 3 csoport valamelyikébe, ekkor pedig az algoritmus adott lépésen belül meg is találja ezen utakat, így az algoritmus valóban csak akkor nem talál megoldást, ha nincs is.

**5.1.2. Állítás:**  $D_i$  gráfban nincsen irányított kör.

**Bizonyítás:** A  $D_i$  irányított gráf bármely  $uv$  élére igaz, hogy  $x(\mathbf{u})+1 = x(\mathbf{v})$  vagy  $y(\mathbf{u})+1 = y(\mathbf{v})$  (mivel így vettük be az éleket). Ez alapján  $x(\mathbf{u}) + y(\mathbf{u}) \leq x(\mathbf{v}) + y(\mathbf{v})$  is teljesül, azaz

bármely  $uv \in E_i$  élre igaz, hogy nem csökkenti a koordináták összegét. (ahol  $E_i$  a  $D_i$  gráf élhalmazát jelöli). Ez a tulajdonság az alábbi ábra alapján is könnyen látható.



5.4. ábra. A  $D_i$ -ben előforduló élek lehetséges képei

Tegyük fel, hogy létezik irányított kör  $D_i$ -ben, jelölje ezt  $K$ . A koordináták összegének monotonitása miatt  $K$  csak olyan élekből állhat, melyek nem változtatnak a koordináták összegén (az 5.4 ábrán két alsó lehetőség). Ugyanakkor bármely  $u \in K$  csúcs esetén  $\mathbf{u}$  pontból indulva,  $K$  élei mentén haladva vissza kell hogy jussunk  $\mathbf{u}$ -ba, ez pedig azt jelenti, hogy  $E_i$ -ben van olyan él melynek képe a  $\mathbf{v}$  pontból mutat  $\mathbf{w}$  pontba és olyan is melynek képe  $\mathbf{w}$  pontból  $\mathbf{v}$  pontba mutat (valamely  $\mathbf{v}, \mathbf{w}$  pontokra). Könnyen látszik, hogy ez lehetetlen.  $\square$

**5.1.3. Állítás:** *Tetszőleges  $s_1-t_1$  út  $D_i$ -ben egy 1-út  $G$ -ben (ugyanígy tetszőleges  $s_2-t_2$  út 2-út lesz).*

**Bizonyítás:** Legyen  $u \in D_i$  olyan csúcs, hogy  $\mathbf{u} \in B_2^* = (B_2 \setminus (A_1 \cup B_1))$ . Mivel  $B_2^*$ -on belül csak  $y$ -koordinátát növelő éleket vettünk be és  $\mathbf{t}_1 \notin \mathbf{D}(\mathbf{u})$ , ezért nem létezhet  $u$ -ból  $t_1$ -be vezető út  $D_i$ -ben.

Legyen  $v \in D_i$  olyan csúcs, hogy  $\mathbf{v} \in A_2^* = (A_2 \setminus (A_1 \cup B_1))$ . Mivel  $A_2^*$ -on belülré képződő csúcsokba csak más  $A_2^*$ -on belülré képződő csúcsokból mutat él, így  $s_1$ -ből nem létezhet út  $v$ -be  $D_i$  gráfon belül.

Ezek alapján bármely  $P$   $D_i$ -beli  $s_1-t_1$  út esetén  $\mathbf{P} \subseteq (A_1 \cup B_1)$ , ebből pedig következik, hogy  $\mathbf{P}$  áthalad a  $\mathbf{z}$  ponton. Lehetnek olyan  $E_i$ -beli élek, melyeknek képe  $A_1 \cup B_1$  halmazon belül van és balra-lefelé mutat (a  $\mathbf{z}$ -n balra-lefelé áthaladó egyenesen).

Legyen  $vu$  olyan él, melynek képe balra-lefelé mutat. Ha  $\mathbf{v} \in A_2 \cap B_1$ , akkor a fentiek alapján  $D_i$ -ben nem vezet út  $s_1$ -ből  $v$ -be, ha pedig  $\mathbf{u} \in A_1 \cap B_2$ , akkor a fentiek alapján  $D_i$ -ben nem vezet út  $u$ -ból  $t_1$ -be. Tehát bármely más, az  $A_1 \cup B_1$  halmazba képződő  $uv$  élre  $x(\mathbf{u}) + 1 = x(\mathbf{v})$ , így bármely  $D_i$ -beli  $s_1-t_1$  út 1-út  $G$ -ben.

Az állítás második fele hasonló módon bizonyítható.  $\square$

## 5.2. Az algoritmus futásideje

**5.2.1. Állítás:** *2-DSPP megoldható  $O(nm)$  időben.*

**Bizonyítás:** Az előkészítés során négy szélességi keresés segítségével számoljuk a termináloktól való távolságokat, ez  $O(m)$  időt vesz igénybe.

Az 1. lépés során  $O(n)$  rácspontot kell megvizsgálnunk, hogy  $\mathbf{t}_1$  ponttól jobbra vagy  $\mathbf{t}_2$  ponttól lefelé helyezkedik-e el a képe, ez  $O(n)$  időben megy, majd ha találtunk ilyen csúcsot akkor

a 3.5.1 állítás miatt  $O(m)$  időben találunk megoldást, így ez a lépés csak  $O(n + m)$  időt igényel. A 2. lépés során  $O(n)$  darab csúcsot kell megvizsgálni, mint a  $\mathbf{z}$  pont körüli négy rácspont közül a bal-alsó. Keresztező élt keresve  $O(m)$  időben vizsgáljuk meg az összes élt, majd ha van ilyen élpár, akkor a 3.5.2 következmény miatt összesen  $O(m)$  időben találunk megoldást. Ha nem létezik ilyen élpár, akkor is összesen  $O(n)$  koordinátát vizsgáltunk meg, mint lehetséges metszéspont  $O(m)$  időben, ez összesen  $O(nm)$  futásidőt eredményez (a 2 részre bontás és hogy 2 élhez keresünk rajtuk átmenő 1-utat és 2-utat, ezen nem változtat, mivel ezek a futásidőre nézve csak konstans szorzókat hoznak be).

A 3. lépés során a  $\mathbf{z}$  pontot "kitalálása" során ismét  $O(n)$  pontot kell megvizsgálunk, ezen belül pedig a  $D_i$  gráfokat  $O(m)$  időben hozzuk létre, majd bennük  $O(m)$  időben keresünk diszjunkt utakat. Ezt a részt  $s_2$  és  $t_2$  szerepének cseréjével is le kell futtatni, de ez csak egy 2-es szorzót jelent, így ez az eset is összességében  $O(nm)$  időt vesz igénybe.

Az algoritmus összesített futásidője tehát  $O(nm)$ . □

## 6. fejezet

# Implementáció

Az 5. fejezetben ismertetett algoritmust Java nyelven implementáltam, a program forráskódja elérhető *itt*.

### 6.1. Eltérések

Az implementált algoritmus pár helyen eltér Az 5. fejezetben ismertetett algoritmustól :

(1) A bemenetként kapott gráfról az implementáció során nem tettem fel, hogy csak fontos élekből és csúcsokból áll. Ennek oka, hogy teszteléshez az Erdős-Rényi-modell alapján generáltam véletlen gráfokat, így persze nem várhatjuk el, hogy a kapott gráfok minden éle és csúcsa fontos legyen. A programom tehát azzal kezdődik, hogy az inputként kapott gráfot a 2. fejezetben leírtak alapján kezelem. Azt viszont elvárjuk a bemenetként kapott gráftól, hogy összefüggő legyen, ha ez nem igaz akkor a program "G\_0 nem összefüggő" üzenettel leáll.

(2) A 3. lépés során előállított körmentes irányított gráfokban 2-DPP megoldására nem Tholey [2] algoritmusát, hanem Perl és Shiloach [13] algoritmusát használom. Ez ugyan nem  $O(n + m)$ , hanem  $O(nm)$  futásidejű (így az implementált algoritmus futásideje  $O(n^2m)$ ), viszont sokkal egyszerűbb és átláthatóbb, ezért alkalmasabb volt az implementációra.

### 6.2. Input és output

Az inputot az "input.txt" nevű fájlból olvassuk be, ez alapértelmezetten a 3.1 ábrán látható gráfnak megfelelő bemenetet tartalmazza.

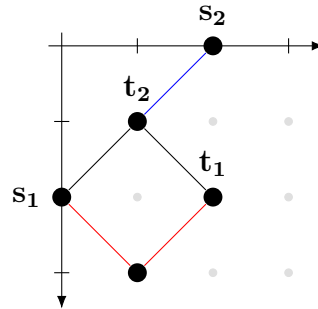
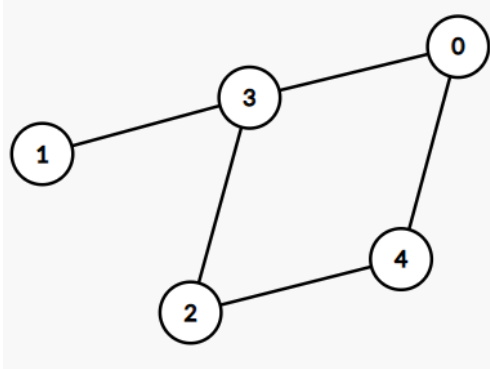
**Input** : A fájl első sorában szerepeljen  $n$  és  $m$ , második sorában pedig négy  $n$ -nél kisebb egész szám, ezek rendre  $s_1, s_2, t_1, t_2$  csúcsok címkéi. Az ezt követő  $m$  darab sorban szerepeljenek a gráf élei.

**Output** : Első sorában információ arról, hogy létezik-e megoldás és ha igen azt az algoritmus melyik lépésében találtuk meg. A második és harmadik sorokban írjuk  $P_1$  és  $P_2$  utakat, ezekről könnyen megállapítható, hogy valóban megoldásai 2-DSPP-nek (ellenőrzöm, hogy  $P_1$  1-út,  $P_2$  2-út és  $P_1$  és  $P_2$  csúcsdiszjunkt). Az output utolsó sora ennek az ellenőrzésnek az eredményét tartalmazza ( $P_1$  és  $P_2$  megfelelő vagy a program által adott útpár nem megoldása a feladatnak).

### 6.3. Példák

A továbbiakban mutatunk pár példát bemenetként kapott gráfokra és a hozzájuk tartozó, a programom által adott outputra. Az összes példában az egyszerűség kedvéért az  $s_1, s_2, t_1, t_2$  csúcsok legyenek rendre a 0, 1, 2, 3 címkéjű csúcsok (tehát 0-2 és 1-3 utakat keresünk). Az ábrák jobb oldalán pirossal jelöljük  $P_1$  és kékkel jelöljük  $P_2$  töröttvonalakat, ha létezik megoldás.

**6.3.1. Példa:** Példa az 1. lépés során megoldható gráfra.



A program outputja ebben az esetben:

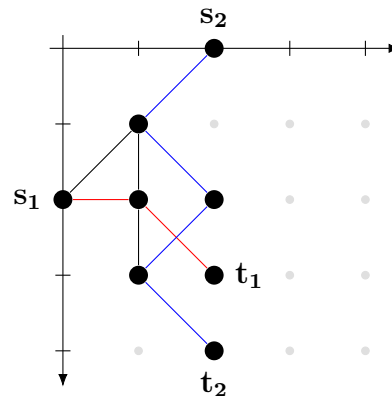
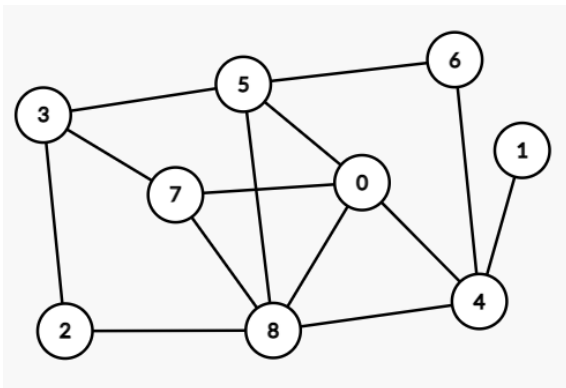
1. lépés során találtunk megoldást ( $t_2$ -től lefelé)

$P_1$  : 0 - 4 - 2

$P_2$  : 1 - 3

Ellenőrzés :  $P_1$  és  $P_2$  valóban megoldásai 2-DSPP-nek a  $G$  gráfon.

**6.3.2. Példa:** Példa a 2. lépés során megoldható gráfra.



A program outputja ebben az esetben:

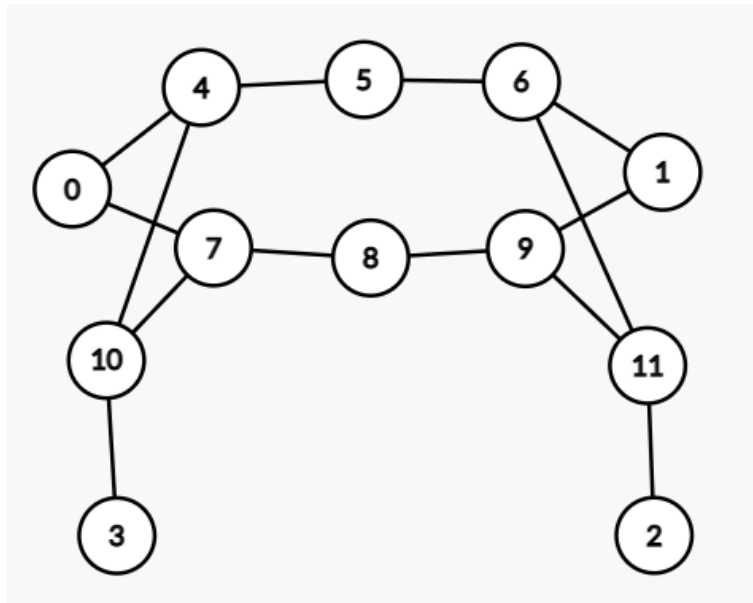
2. lépés során találtunk megoldást

$P_1$  : 0 - 8 - 2

$P_2$  : 1 - 4 - 6 - 5 - 3

Ellenőrzés :  $P_1$  és  $P_2$  valóban megoldásai 2-DSPP-nek a  $G$  gráfon.

**6.3.3. Példa:** Jó példa olyan gráfra melyben az algoritmus a 3. lépés során talál megoldást a 4.7 példában szereplő gráf. Látszik, hogy az 1. és 2 lépés során nem találhatunk megoldást,



6.1. ábra. Ezen gráf geometriai tulajdonságairól a 4.7 példa során bővebben beszéltünk.

valamint  $s_2$  és  $t_2$  csúcsok cseréje nélkül sem, azaz a megoldást a 3. lépés  $s_2$  és  $t_2$  cseréje utáni részben találhatjuk csak meg.

A program outputja ebben az esetben:

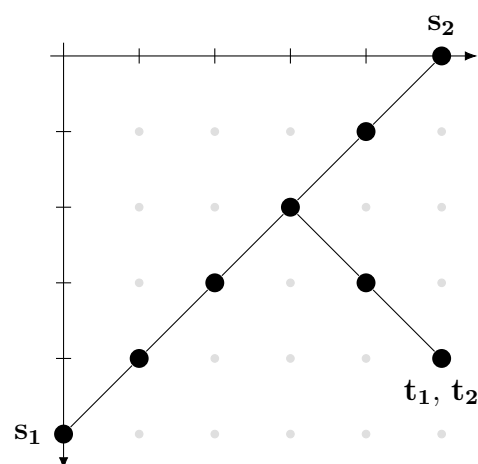
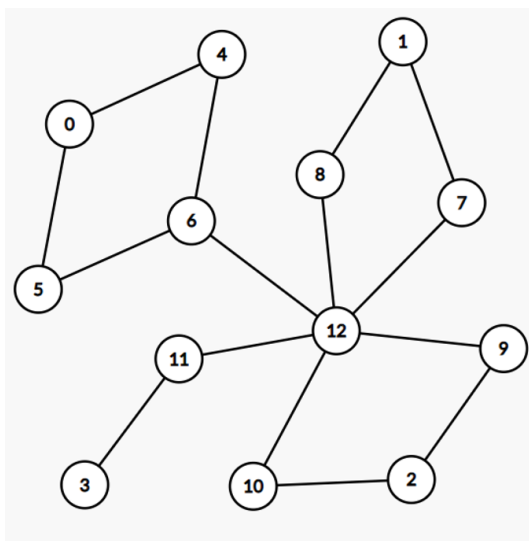
3. lépés során találtunk megoldást ( $s_2$  és  $t_2$  cseréje után)

$P_1 : 0 - 7 - 8 - 9 - 11 - 2$

$P_2 : 1 - 6 - 5 - 4 - 10 - 3$

Ellenőrzés :  $P_1$  és  $P_2$  valóban megoldásai 2-DSPP-nek a  $G$  gráfon

**6.3.4. Példa:** Példa olyan gráfra melyben nincs megoldás, ez esetben ez könnyen látszik, hiszen bármely 1-útnak és bármely 2-útnak át kell mennie a 12 címkeű csúcson.



A program outputja ebben az esetben:  
Nincs megoldás

# Irodalomjegyzék

- [1] Bentert, M.; Nichterlein, A.; Ranken, M.; Zschoche, P.: Using a Geometric Lens to Find  $k$  Disjoint Shortest Paths, *arXiv:2007.12502* (2020), <https://arxiv.org/abs/2007.12502>
- [2] Tholey, T.: Linear time algorithms for two disjoint paths problems on directed acyclic graphs, *Theoretical Computer Science* **465**, 35-48 (2012)
- [3] Eilam-Tzoref, T.: The disjoint shortest paths problem, *Discrete Applied Mathematics* **85** (2), 113-138 (2012)
- [4] Gottschau, M.; Kaiser, M.; Waldmann, C.: The undirected two disjoint shortest paths problem, *Operations Research Letters* **47** (1), 70-75 (2019)
- [5] Kobayashi, Y.; Sako, R.: Two disjoint shortest paths problem with non-negative edge length, *Operations Research Letters* **47** (1), 66-69 (2019)
- [6] Akhmedov, M.: Faster 2-disjoint-shortest-paths algorithm, *Computer Science - Theory and Applications* **12159**, 103-116 (2020)
- [7] Karp, R. M.: On the Computational Complexity of Combinatorial Problems, *Networks* **5** (1), 45-68 (1975)
- [8] Kramer, M. R.: The complexity of wirerouting and finding minimum area layouts for arbitrary VLSI circuits, *Advances in computing research* **2**, 129-146 (1984)
- [9] Robertson, N.; Seymour, P. D.: Graph minors. XIII: the disjoint paths problem, *Journal of Combinatorial Theory, Series B* **63** (1), 65-110 (1995)
- [10] Kawarabayashi, K.; Kobayashi, Y.; Reed, B. A.: The disjoint paths problem in quadratic time, *Journal of Combinatorial Theory, Series B* **102** (2), 424-435 (2012)
- [11] Fortune, S.; Hopcroft, J. E.; Wyllie, J.: The directed subgraph homeomorphism problem, *Theoretical Computer Science* **10** (2), 111-121 (1980)
- [12] Bérczi, K.; Kobayashi, Y.: The Directed Disjoint Shortest Paths Problem, *25th Annual European Symposium on Algorithms (ESA 2017)* **87**, 13:1-13:13 (2017)
- [13] Perl, Y.; Shiloach, Y.: Finding Two Disjoint Paths Between Two Pairs of Vertices in a Graph, *Journal of the ACM* **25** (1), 1-9 (1978)