Eötvös Loránd Tudományegyetem

Matematika Intézet

KISS MÁTYÁS JÓZSEF

Vasútimenetrend-tervezés

BSc szakdolgozat

Alkalmazott matematikus szakirány

Témavezető: Kis Tamás

ELTE Operációkutatási Tanszék

2014. Budapest

Köszönetnyilvánítás

Szeretném köszönetemet kifejezni konzulensemnek, Kis Tamás tanár úrnak, mert mindig előre tudta mozdítani a munkámat, amikor úgy éreztem, hogy elakadtam. Mindenképpen szeretnék hálás köszönetet mondani Horváth Markó doktorandusznak, aki időt és energiát nem sajnálva rengeteg technikai tanácsot, útmutatást adott, és aki nélkül ez a dolgozat egészen biztosan nem valósulhatott volna meg.

Szeretnék még köszönetet mondani Vörös Attila és Mózes Krisztián szakújságíróknak, amiért hasznos ötleteket adtak illetve információt a magyar vasút működésével kapcsolatban. Szeretném még megemlíteni Szabó Zsolt matematikus-informatikus hallgatót, aki a legkülönbözőbb napszakokban is szívesen nyújtott segítséget programozással kapcsolatban. És végül, de nem utolsó sorban köszönettel tartozom Hörcher Dániel közlekedésmérnöknek, aki először keltette fel a figyelmemet a téma iránt, több szakirodalmat és témalehetőséget a figyelmembe ajánlott.

Tartalomjegyzék

Bevezetés, téma ismertetése

Szakdolgozatom témája a vasútimenetrend-tervezés matematikai modellezési lehetőségei, illetve egy ilyen modell implementálása C++ programnyelven.

A gőzhajtású vasút XIX. század eleji feltalálása óta a vasúti közlekedés rengeteg fejlődött és világszerte elterjedt: az összes kiépített vonalhossz meghaladja az 1 000 000 km-t. Dolgozatomban csak személyszállítással fogok foglalkozni, ennek kapcsán érdemes bevezetni egy másik mérőszámot, az utaskilométert: ha egy személy megtesz 1 kilométert vonattal, akkor ezt a mérőszámot eggyel növeli. Ennek tükrében figyelemre méltó adat, hogy a világ összes utaskilométere több, mint 2765 milliárd évente.

A rengeteg utas és a kiszámíthatóság miatt a vasúttársaságok ütemes menetrendet használnak. Azonban ezeknek a menetrendeknek az összeállítása nagyon nehéz és kapacitásigényes feladat, hiszen rengeteg korlátozó tényezőt kell figyelembe venni. A teljesség igénye nélkül: nem mehet két vonat ugyanakkor ugyanazon a vágányon az ellenkező irányba; egy vonat sem mehet gyorsabban, mint az alatta lévő vágány által meghatározott maximális sebesség; egy személyvonat nem lehet hosszabb, mint az érintett összes állomás peronhosszainak minimuma.

Ebből kifolyólag az elmúlt évtizedek vívmánya, hogy matematikai modellt építenek fel a fenti, reménytelennek tűnő problémák megoldására. Dolgozatom 1. fejezetében röviden összefoglalom, hogy a menetrendtervezés modellezése hány különböző lépcsőből áll, és nagy vonalakban melyik során mi történik.

A szakdolgozat 2. fejezetében az egyik ilyen közbenső lépcsőre fogok koncentrálni: a hosszútávú menetrend-tervezési problémára. Hosszútávú, tehát több hónappal előre, esetleg a következő évre lehet ilyen módon a menetrendet meghatározni. És a későbbi, előre nem látott események miatti menetrend-módosítások már egy másik kérdéskörbe tartoznak. A menetrendtervezés pedig ebben az esetben a következőt jelenti: feltételezzük, hogy van egy ideális menetrendünk, ami csak az utasigényeknek megfelelően indítja a vonatokat, tehát figyelmen kívül hagyja azt, hogy a vonatok elférnek-e a vágányokon. Ezután ezt az ideális menetrendet javítjuk ki, tehát keressük azt a menetrendet, ami egy bizonyos célfüggvény szerint a lehető "legközelebb" van az ideális menetrendhez. A célfüggvény tehát valamilyen értelemben tárolja két menetrend "távolságát", és ezt próbáljuk minimalizálni (vagy a közelségüket maximalizálni). Ennek különböző lehetséges módjairól a 2.2.4 fejezet szól.

A 2.1. fejezetben egy lehetséges modellt ismertetek a hosszútávú menetrendtervezésre, amit a [2] forrás alatt megjelölt cikkben már közzé tettek. Ennek során egy gráfot fogunk felépíteni, majd ebben keresünk optimális többtermékes folyamot, ami aztán meg fog határozni egy menetrendet.

Ezt a modellt a 2.2 fejezetben tovább fogom bővíteni, és teljes részletességgel ki fogom fejteni. Egy helyett két irányba engedem majd közlekedni a vonatokat, illetve a szükséges szinten lemodellezem az állomásokat is. És a 2.1 fejezetben ismertetett modellhez hasonlóan itt is optimális többtermékes folyamot kell keresni.

Ez a modell tehát kétféle kérdésre adhat választ. Tegyük fel, hogy egy meglévő, működő menetrenden kisebb módosításokat vagyunk kénytelenek eszközölni. Ez a modell alkalmas arra, hogy a nem megengedett menetrendet (amiben tehát ütköznének a vonatok) optimálisan, minél kevesebb változtatással kijavítsa. Másrészt ha még nincsen semmilyen menetrendünk, csak tudjuk, hogy bizonyos vonatokat mikor akarunk üzemeltetni, akkor ez a modell alkalmas arra, hogy ebből egy megengedett menetrendet gyártson, vagyis tartalmazza a pontos indulási és érkezési időket az egyes állomásokon, és a vonatok összeltérése a kívánt indulási és érkezési időktől minimális.

Ezt a bővített modellt C++ nyelven implementáltam, aminek során a LEMON (Library for Efficient Modeling and Optimization in Networks) környezetet használtam, illetve a FICO Xpress 7.5 programmal számoltattam ki az optimális megoldást. A programot CD-mellékleten csatoltam a dolgozathoz.

A programot tesztelés után valós adatokon is lefuttattam. A Budapest–Szob viszonylaton közlekedő személyszállító vonatok menetrendjét vizsgáltam meg. Mivel a modell egy ideális (de nem megengedett) menetrendet feltételez, ezért a valós menetrendet valamilyen módon el kellett rontanom, hogy utána a leprogramozott modellel kijavíthassam. Emiatt tehát abból indultam ki, hogy valamilyen külső ok miatt kicsit megváltozni kényszerül a menetrend, de így már nem lesz megengedett. Például a nemzetközi vonatoknak máskor kellene mennie, vagy állomásfelújítás miatt sebességkorlátozást vezetnek be. Ezeket a lehetséges eseményeket, ezeknek a hatásait és a modell által javított menetrendeket a 3. fejezetben ismertetem.

1. fejezet

Menetrend tervezési problémák

A vasútimenetrend-tervezés egy nagyon komplex, sokrétű probléma. Elsőre talán nem is gondolnánk, de rengeteg optimalizálási probléma merül fel a témában, de persze attól függően, hogy mennyire tágan értelmezzük a menetrendtervezés fogalmát. A vasúti közlekedés alapvetően két részre bontható: személy-, és teherszállításra. Ebben a dolgozatban csak személyszállítással foglalkozom, bár a kettő teljes szétválasztása jelentős egyszerűsítés, hiszen ugyanazokat a vágányokat, kapacitásokat használják, tehát nagyon befolyásolják egymást.

A továbbiakban szeretném először összefoglalni, hogy milyen részfeladatokra bonthatjuk a problémát az optimalizálás tárgya szerint, itt alapvetően az [1] forrásra fogok támaszkodni. Második lépésben egy másik dimenzió szerint is rendezem a részfeladatokat: a szerint, hogy milyen időtávra tervezünk vele. Az itt leírtak forrásaként a [4] szám alatt megjelölt phd-dolgozat szolgált.

1.1. Részfeladatokra bontás az optimalizálás tárgya szerint

Az első részfeladat meghatározásánál abból induljunk ki, hogy a vasúthálózat és az utasok igényei már adva vannak. Ekkor az első feladat a vonaltervezés (Line Planning Problem). Ennek során jelöljük a személyszállítást teljesítő járatok útját, típusát és gyakoriságát. Ezután optimalizáljuk a menetrendet úgy, hogy megengedett menetrendet kapjunk, és ezt lerögzítjük (Train Timetabling Problem). Ha ez sikerült, a következő lépés, hogy minden állomáson belül meghatározzuk, melyik beérkező vonat melyik vágányra álljon be (Train Platforming Problem). Ezután minden vonat esetében meghatározzuk, pontosan hány darab, milyen típusú kocsiból álljon a szerelvény, és ezek milyen sorrendben legyenek (Rolling Stock Circulation Problem), majd ennek függvényében a mozdonyok és kocsik állomáson belüli tolatását és tárolóvágányra kihúzását (Train Unit Shunting Problem). Végül pedig a személyzetet (mozdonyvezetők, kalauzok) osztjuk be különféle szempontok alapján a vonatokhoz (Crew Planning Problem).

Ezeket a feladatokat természetesen jobb lenne egyben megoldani, a probléma csak az, hogy úgy túl összetettnek tűnik. A feladat ily módon dekomponálása tehát egyszerűsítés, hiszen mindig az előző alfeladat eredménye lesz a következő bemeneti adata. Valójában hatékonyabb lenne a tervezés folyamata, ha valamilyen módon oda-vissza csatolás lenne a részfeladatok között.

A továbbiakban ebbe a hat részfeladatba adok egy rövid betekintést. A feladatok nevét angolul is odaírtam, hogy könnyebben beazonosíthatóak legyenek a nemzetközi szakirodalomban.

1.1.1. Vonaltervezési probléma (Line Planning Problem)

Az európai személyszállító vasúti menetrendre túlnyomó többségében jellemző a ciklikusság. Vagyis az üzemeltetett vonatok halmazokba partícionálhatóak, ahol az egyes halmazokon belüli vonatok között egy (esetleg fél vagy kettő) ciklusnyi idő az eltérés. Az LLP probléma célja, hogy kielégítse az utazási keresletet úgy, hogy közben minél jobban

1. maximalizálja a célhoz szállított utasok számát, és
2. minimalizálja az üzemeltetés költségét.

Érdemes bevezetni a direkt utazás fogalmát, ami annyit jelent, hogy egy utas átszállás nélkül jut el az induló állomásából a célállomásába. Ezen utazások maximalizálása azt eredményezné, hogy nagyon hosszú járatok jelennének meg, aminek két hátránya van. Egyrészt könnyebben előfordulhatnak késések, másrészt a vasúti kocsik elosztása kevésbé lesz hatékony, mert így kevésbé tud reflektálni a csúcsidőben történő megnövekedett kocsiigényre. Világos tehát a dilemma, hogy gyakran rövidebb vonatokat érdemes-e üzemeltetni, vagy ritkábban hosszúakat. A kettő közötti egyensúlyozást a fenti két célfüggvény fejezi ki.

Ez tehát az az első probléma, ami az utasigényekből és a meglévő vasúti vonalakból létrehozza azt az ideális menetrendet, ami még nem foglalkozik vágánykapacitásokkal. Ezt a következő feladat megoldása adja, az, amivel dolgozatomban is részletesen foglalkozok.

1.1.2. Menetrend optimalizálása (Train Timetabling Problem)

A menetrend optimalizálását alapvetően kétféle típusra lehet bontani, a szerint, hogy ciklikus menetrendet fogunk készíteni, vagy aciklikusat. Mindkét esetben egy ideális menetrendet kapunk, és a cél az, hogy a vágánykapacitás miatti korlátokat betartsuk.

Első esetben minden vonatot tetszőlegesen módosíthatunk, ami azt eredményezi, hogy a ciklikus ideális menetrendtől 1-1 vonat el fog kicsit térni. Ez azért lehet rossz, mert ha minden órában ugyanakkor megy a vonat, de egyszer 5 perccel hamarabb, akkor az utasok ezt valószínűleg nehezebben tartják észben, sőt a személyzet dolgát is bonyolítja. Másrészt ha az eredetihez képest kicsit később megy, az nem feltétlenül baj (ha nem késik le másik csatlakozást).

Ha mindenképpen ciklikus menetrendet kell kapni kimenetként, akkor típusokra bonthatjuk a vonatokat, és a típusokon belül lévő vonatokat egyszerre tudjuk módosítani. Ennek előnye, hogy ciklikus marad a menetrend, de esetleg a reggeli vonatokon is kell módosítani a csúcsforgalmi ütközések miatt, így lehet hogy a reggeli vonat korábban megy, mint optimális lenne.

Dolgozatomban az első típusú optimalizálást választottam, tehát minden egyes vonat menetrendjét egyesével lehet majd módosítani. A bemeneti adatok sem kell, hogy ciklikusak legyenek (míg ciklikus menetrend gyártása esetén ez szükséges).

1.1.3. Állomáshasználat optimalizálása (Train Platforming Problem)

Az előző problémában az állomásokkal csak annyira foglalkoztunk, hogy sosem lehet (állhat meg vagy haladhat keresztül) több vonat egy állomáson, mint ahány vágány van. Viszont nem világos egyelőre, hogy melyik vonat mikor melyik vágányra áll be. Egy kisebb állomásnál, ahol 4 vagy 6 vágány van egyáltalán nem komoly probléma. De van olyan pályaudvar, ahol 20-30 vágány is van. Az is elképzelhető, hogy mindkét irányba el lehet hagyni a pályaudvart. Ebben az esetben már nagyon összetett kérdésről van szó.

Figyelembe kell venni egyrészt a peronok hosszát és magasságát, ennek kompatibilisnek kell lennie a beálló vonattal. Az is előfordulhat, hogy nem lehet akármelyik vágányról akármelyikre beállni, tehát a váltók struktúráját mindenképpen bele kell számolni.

Ezek után pedig lehet optimalizálni a minél kevesebb váltóhasználatra, vagy a tolatások minimalizálására.

1.1.4. Kocsibeosztás optimalizálása (Rolling Stock Circulation Problem)

Ebben a feladatban a cél az, hogy az ismert időpontokban haladó vonatokra csatolt kocsik számát és típusát meghatározzuk. A feladat ismertetéséhez forrásként felhasználtam az [5] forrás alatt megjelölt előadáson elhangzottakat.

Az, hogy egy adott vonatra milyen kocsit kapcsolunk fel, sok mindentől függ. Egyrészt az előző részhez hasonlóan itt is figyelembe kell venni, hogy a kocsik például egymással (és a peronnal) egy magasak legyenek. Az is nagyon fontos, hogy ha egy vonatot szétkapcsolnak, akkor a lecsatolandó kocsik praktikusan a vonat végén legyenek, ne a közepén.

A cél pedig természetesen az, hogy minden kocsi minél nagyobb kihasználtsággal közlekedjen, de egy utasnak se kelljen állnia. A probléma bonyolultságát növeli az is, hogy figyelembe kell venni az utasok egyéni optimalizálását. Vagyis ugyan tudjuk egy utasról, hogy az A állomásról akar a B állomásra eljutni, de nem tudjuk, hogy ezt milyen módon fogja megtenni. Vagyis hogy egy esetleg kicsit gyorsabb vonalon megy ahol át kell szállni, vagy a lassabb is jó neki. Ez pedig alapvetően befolyásolja a felkapcsolandó kocsik számát.

1.1.5. Tolatási mozgások optimalizálása (Train Unit Shunting Problem)

Ez a probléma kevésbé szervesen függ össze a menetrendtervezés többi lépcsőjével. Egyrészt ide sorolhatjuk azt is, amikor a rendező-pályaudvarokon történő kocsikapcsolásokat végzik tolatómozdonyokkal. Másrészt a kevésbé forgalmas napszakokban és főleg éjszaka kihúzzák a vasúti kocsikat és mozdonyokat a tárolóvágányokra. A beállás után már nehéz módosítani a benn álló kocsik sorrendjén, ezért érdemes olyan rendszerben elhelyezni a kocsikat, hogy a következő forgalmasabb napszakban könnyebb legyen őket előhozni.

Korlátnak tehát leginkább a vasúti pálya adottságai adódnak, leginkább a vágányok és váltók száma és elhelyezkedése. Adott még a tárolásra elhelyezendő vonatok száma és ideje. Optimalizálni pedig például a tolatások és vontatások minimalizálására lehet.

1.1.6. Személyzetelosztás optimalizálása (Crew Planning Problem)

Ennek az alfeladatnak a célja, hogy az összes személyzetet (mozdonyvezetők és kalauzok, egyéb kiszolgáló személyzet) elossza a vonatok között. Figyelembe kell venni, hogy minden alkalmazottnak ott kell végeznie a napját ahol elkezdte, illetve az egyes szakszervezetektől függően olyan igények is felmerülhetnek, hogy legyen változatos az útvonal, amelyek szolgálatot teljesítenek. Ezen kívül a mozdonyvezetőknek vasúttársaságtól függően kellhet pályaismereti vizsgát tenni, tehát nem mehet minden mozdonyvezető minden pályaszakaszon. Az is még felmerülhet korlátként, hogy mennyi pihenőt kell biztosítani legalább a vezetőknek illetve az egyéb személyzetnek munka közben.

Ezt a feladatot két lépésben érdemes megoldani. Első lépésben névtelenül beosztják, hogy az egyes vonatokra hány alkalmazott jusson. Az is könnyen előfordulhat, hogy két mozdonyvezető megy a vonaton, hiszen lehet, hogy a következő állomásról két vonat megy tovább, és ahhoz két vezető kell. A további személyzet mennyisége pedig függ a várható utaslétszámtól, illetve praktikusan a felcsatolt kocsik számától.

Második lépésben pedig az egyes embereket név szerint osztják be a fent említett helyekre. Itt lehet figyelembe venni a kijelölt útvonalak változatosságát, és optimalizálni a holtidők minimalizálására.

1.2. Részfeladatokra bontás az optimalizálás hossza szerint

Ebben a fejezetben a menetrendtervezés problémáját másik szempont szerint fogom szétbontani a 1.1 fejezethez képest. Most az lesz a kérdés, hogy az optimalizálás során milyen időtávra tervezünk. Négy részre érdemes bontani az időhorizontot. A továbbiakban ezeket sorolom fel, illetve jellemzem.

1.2.1. Stratégiai tervezés

A stratégiai tervezés jelenti a több évre előre tekintést. Ekkor nagy vonalakban meg kell tervezni, hogy mit és milyen színvonalon akar megvalósítani a vasúttársaság, és hogy ehhez honnan lesz erőforrás.

Természetesen ez nagyban függ a jövőbeni várható kereslettől, erre különböző sztochasztikus modellek léteznek. Ez viszont azért nagyon bonyolult, mert a kereslet függ attól is, hogy milyen menetrend születik (hiszen mindig van helyettesítő termék, például távolsági busz vagy vonat)

A stratégiai menetrendtervezés során általában egy mátrixot becsülnek meg, melynek a_{ij} eleme adja meg, hogy az i. állomásból mennyi utas mehet később a j.-be. Tapasztalatok alapján a stratégiai menetrend nem gyakran változik, hiszen az utasigények sem változnak gyorsan, mert a városok méretének, ingázók számának változása is lassú folyamat.

Mindenképpen fontos még a stratégiai kocsibeosztás megtervezése. Valójában ez arról szól, hogy előre számoljanak a szükséges kocsik számával, a szükséges beszerzéseket időben elindítsák. Ehhez viszont legalább egy becslésre szükség van arról, hogy mennyi járat lesz, egy-egy vonaton hány kocsi lesz. Ehhez tehát megint a kereslet becslése szükséges.

Érdemes még megemlíteni a személyzet alkalmazásának stratégiai tervezését. Mivel új mozdonyvezetők, kalauzok kiképzése időbe kerül, ezért itt is fontos lenne néhány évre előre látni abban, hogy várhatóan mennyi alkalmazottra lesz szükség. Itt van egy másik fontos szempont is: a személyzetnek a nap végén abban a városban kell végeznie, ahol kezdett. Ebből az is következik, hogy a személyzet alkalmazásánál figyelembe érdemes venni a lakóhelyét is.

1.2.2. Taktikai tervezés

A taktikai tervezés során 1 évtől 2 hónapig terveznek előre. A MÁV esetében például ezt a szerepet a minden év november közepén közzé tett menetrend tölti be, amihez később még több módosítást mellékelnek.

A menetrendtervezésnél itt kell döntést hozni arról, hogy milyen hosszú mintázatok legyenek a menetrendben (jellemzően 60 vagy 120 perces), vagy milyen típusú napok legyenek (munkanap, hétvége, ünnepnap). Ennél a lépcsőnél már arra is készítenek tervet, hogy a vasúti kocsikat hogyan osztják el a vonatokon, vagy az állomásokon belül hogyan mozoghatnak a vonatok.

A személyzet elosztásánál a taktikai tervezés két lépcsőből áll. Először névtelenül elosztják az embereket a vonatokon, tehát beosztják, hogy például melyik vonatra hány kalauz kell, hogy mindig minden vonatra jusson. Ezután pedig valamilyen szempontok alapján pontosan beosztják az alkalmazottakat a pozíciókba.

1.2.3. Operatív tervezés

Ennél a lépcsőfoknál nagyjából három naptól két hónapig terveznek előre. Itt tehát a korábbi lépcsőfokok során kialakult előzetes menetrend már ismert, és ezen történnek módosítások különböző előre látható események miatt. Ilyen lehet valamilyen kulturális vagy sportrendezvény, például egy focimeccs vagy egy balatoni fesztivál. Az is ide tartozik, ha egy vágányszakaszt karbantartás miatt ideiglenesen lezárnak, de nem olyan hosszú ideig, mintha átépítenék.

Az operatív tervezési szakasznál lesz először fontos az, hogy a tervezés folyamata nem lehet akármilyen hosszú. Itt már nincs idő akármilyen hosszan tárgyalni a különböző szempontokat, alkudni a pályafenntartó vállalattal.

A módosítás mélységétől függően természetesen mind a menetrend, mind a kocsibeosztás és a személyzet elosztása változhat

1.2.4. Azonnali tervezés

Ez az utolsó lépcsőfok nem előre tervezett. Minden, 3 napon belüli menetrend-módosítás ide sorolható, és jellemzően a nem várt eseményekre történő azonnali reagálást jelenti. Ilyen lehet például egy fa kidőlése a vágányra, vagy a felsővezeték szakadása, de akár egy biztosítóberendezés meghibásodása.

A reagálási időnek nagyon rövidnek kell lennie (maximum néhány óra), és természetesen a cél az, hogy minél kevesebb változtatással folytatódhasson az eltervezett menetrend. Általában nem kifejezetten az optimumra törekednek: bármilyen megoldás, aminek során a rendszer tovább tud működni már jó.

A probléma leginkább a sűrűn használt vasúthálózatokban súlyos. Hollandiában például, ahol csúcsidőben akár 15 percenként járnak a vonatok, egy apró változtatás az egész rendszeren végiggyűrűzik. Arra kell tehát törekedni, hogy a vonatok összes késése minél kevesebb legyen. Ehhez aztán igazítani kell a személyzetbeosztást is.

2. fejezet

Hosszútávú ütemes menetrendtervezési modell

2.1. A Caprara-Fischetti-Toth modell

A dolgozat második felében a menetrendtervezés egyik közbenső lépcsőjére fogok koncentrálni, a hosszútávú menetrend-tervezési problémára. Ennek során a 1.1.1 fejezet szerint egy ideális menetrendet kapunk, ami a vágánykapacitásokat figyelmen kívül hagyva minden vonatot akkor futtat, amikor az önmagától menni szeretne (az utasigényekből következően).

A problémával részletesen a [2] cikk foglalkozik, ebben pontosan kifejtik azt, amiről én most egy rövidebb összefoglalót adok, és ami Train Timetabling Problemként (TTP) ismert.

2.1.1. TTP probléma

1. Definíció. Adott két nagyobb vasútállomás. Ha a kettő között van egy egyirányú, egyvágányos pálya, ami a két állomás között még kisebb állomásokat érint, akkor ezt hívjuk *korridornak*.

Egyszerűnek tűnhet, mégis van már gyakorlati haszna a korridoroknak. Vegyük észre, hogy két, ellenkező irányú korridor egymás mellé rakásával már kétirányú közlekedést modellezünk. A valóságban pedig sokszor jó modellt ad ez a fővonalak közlekedésére, például Budapest–Székesfehérvár esetében.

Szeretnék bevezetni néhány jelölést:
A vizsgált időintervallumot osszuk fel egyenletesen q időpillanatra, tehát például egy nap percenkénti felosztása 1-től $q = 1440$ -ig lesz.
Legyen $S = \{1, ..., \Psi\}$ az állomások halmaza, olyan sorrendben megszámozva, ahogy a vonatok végigmennek rajtuk.
$T = \{1, ..., \Gamma\}$ legyenek a vonatok, amik tehát minden periódusban ugyanakkor mennek az ideális menetrend szerint.
Minden $j \in T$ vonathoz tartozik egy f_j kezdő és egy l_j végállomás ($l_j > f_j$). Jelölje $S^j := \{f_j, ..., l_j\} \subseteq S$

a $jinT$ vonat által rendre érintett állomások halmazát.

2. Definíció. A *menetrend* egy olyan függvény, amely minden $j \in T$ vonat esetében a kezdő f_j állomáshoz hozzárendel egy indulási időt, az utolsó l_j állomáshoz egy érkezési időt, illetve minden $s \in [f_j + 1, l_j - 1]$ állomáshoz egy érkezési és egy indulási időt.

3. Definíció. Egy $j \in T$ vonathoz tartozó *futási idő* legyen azon két időpillanat különbségének abszolútértéke, amikor a vonat elhagyja f_j-t és amikor megérkezik l_j-be.

A modell számára tehát bemeneti adatként szerepel az állomások sorrendje és száma. Ezen kívül valahogyan meg kell adni az utasigényeket is: ezt a szerepet az *ideális menetrend* tölti be. Ez egy olyan menetrend, ami minden vonat esetében tartalmazza, hogy az a vonat mikor szeretne menni attól függetlenül, hogy ilyen módon ütközik-e más vonatokkal. Más szóval figyelmen kívül hagyjuk a vágányszám miatti megkötéseket. Ezen kívül minden $i \in S$ állomáshoz bevezethetünk egy a_i és d_i korlátokat, amik azt mondják meg, hogy az i. állomáson legkevesebb mennyi időnek kell eltelnie két beérkező illetve kimenő vonat között.

A célfüggvényt pedig természetesen úgy fogjuk bevezetni, hogy valamilyen értelemben minél közelebb vagyunk az ideális menetrendhez, annál jobb legyen. Legyen π_j az a profit, amit a j. vonat indításával szerzünk. Jelölje $\nu_j \geq 0$ az abszolút különbségét az ideális és az eredményként kapott menetrendben lévő indulási idejét a j. vonatnak az f_j állomásról. Végül pedig jelölje $\mu_j \geq 0$ a j. vonathoz tartozó, ideális menetrendben lévő futási idő és az eredményként kapott menetrendben lévő futási idő különbségének abszolútértékét. Legyen továbbá Φ_j tetszőleges monoton növő függvény, amire $\Phi_j(0) = 0$, ez fogja büntetni az indulási időben való eltérést az ideális menetrendtől. Végül pedig γ_j legyen tetszőleges nem negatív paraméter, ami pedig az utazási idő hosszabbodását bünteti (tehát az utazási idő növekedésével lineárisan csökken a profit). Ekkor az egyes vonatokhoz tartozó profitfüggvényt a következő módon formalizálhatjuk:

$$\pi_j - \Phi_j(\nu_j) - \gamma_j \mu_j$$

Ez azt is jelenti, hogy egy vonat negatív profitot is tud eredményezni, ilyenkor természetesen az adott vonat már nem indul el.

4. Tétel. *A fent leírt TTP probléma NP-nehéz.*

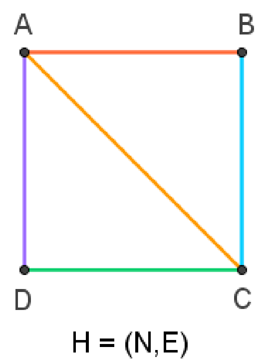
A tétel kimondása és bizonyítása eredetileg a [2] cikkben jelent meg.

Bizonyítás. A gráfbéli maximális független ponthalmaz keresését fogjuk visszavezetni a TTP-problémára. Adott a $H = (N, E)$ gráf, amiben keresünk egy legalább K elemű független csúcshalmazt. Legyen továbbá $E = e_1, e_2, ..., e_m$ az élek halmaza, ahol $e_i = (j_i, k_i)$, $\forall i = 1, ..., m$

A megfelelő TTP problémát a következőképpen hozzuk létre. Minden csúcs megfelel egy vonatnak, vagyis $\Gamma = |N|$. Olyan módon adjuk meg a profitfüggvény paramétereit, hogy egy vonat vagy az ideális menetrendje szerint közlekedik, vagy sehogy. Ezt a következő módon tehetjük: $\pi_j := 1$, $\Phi_j(0) := 0$,és

$\Phi_j(x) := 1$, ha $x \neq 0$, illetve $\gamma_j := 1$. Kihasználva, hogy az idő egész pontokra lett felosztva, ha a vonat legalább 1 perccel több ideig utazik vagy legalább 1 perccel később indul, a profit már nempozitív, tehát a vonatot törölhetjük. Az állomások száma legyen $m+1$, minden vonat az elsőtől az utolsó állomásig közlekedik ($\forall j \in T, f_j := 1, l_j := m+1$)

Az ideális menetrendet úgy fogjuk megalkotni, hogy a j és k vonatok pontosan akkor lesznek egymást kizáróak, ha a H gráfban $(j,k) \in E$. Ezt úgy fogjuk elérni, hogy az e_i él két végpontját jelentő j és k vonatok az i. állomásról pont ugyanakkor fognak indulni, és az $i+1$. állomásra pont ugyanakkor érkeznek meg. Ilyen menetrendet pedig már könnyű konstruálni. A könnyebb megértés érdekében egy példán fogom illusztrálni. Tekintsünk egy példagráfot:



2.1. ábra. Példagráf az NP-nehézséghez

Ehhez készítsük el az incidenciamátrix transzponáltját:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Majd ebből leírhatjuk a menetrendet:

Jól látható, hogy a készített új menetrend valamilyen értelemben örökli az incidenciamátrix transzponáltjának struktúráját. A menetrend a következő módon készül el: tekintsük egy párnak az i. állomás indulási és az $i+1$. állomás érkezési idejét adott j vonathoz. Iteráljunk végig ezeken a párokon először i szerint, azon belül pedig j szerint, és sorban töltsük ki a pozitív egész számokkal. Csak arra kell még figyelni, hogy az egymást kizáró vonatoknál (az ábrán színes cellák) ugyanazt az időt adjuk meg

Ha a menetrend-tervezési feladatra tudnánk találni legalább K profitú megoldást polinomiális időben, akkor a gráfban is találnánk egy legalább K elemű független ponthalmazt polinom időben, és fordítva. Tehát a TTP feladat NP-nehéz.

□

		Vonatok			
Áll.		A	B	C	D
1	ind.	1	1	3	5
2	érk.	2	2	4	6
	ind.	7	9	7	11
3	érk.	8	10	8	12
	ind.	13	15	17	13
4	érk.	14	16	18	14
	ind.	19	21	21	23
5	érk.	20	22	22	24
	ind.	25	27	29	29
6	érk.	26	28	30	30

2.2. ábra. A létrehozott menetrend

2.1.2. Gráf-reprezentáció

Ebben a fejezetben bemutatom, hogyan lehet a TTP problémát egy többtermékes folyam problémára visszavezetni. Először definiálok egy irányított gráfot, majd azon maximális egészértékű folyamot keresek. Minden megengedett folyam kölcsönösen egyértelműen meg fog határozni egy megengedett menetrendet.

A gráf csúcshalmaza legyen $V = \{s,t\} \cup (U^2 \cup ... \cup U^s) \cup (W^1 \cup ... \cup W^{s-1})$, ahol s jelöli majd a forrást, t a nyelőt, illetve U^i és W^i jelölik az i. állomásra való érkezést, és onnan indulást. A 2.3 ábrán látható vízszintes vonalak jelölik az U^i és W^i halmazokat, tehát azok valójában sok csúcs egymás mellett. Fontos megjegyezni, hogy bár az ábrán folytonos vonalakkal jelöltem, de ezek a halmazok végesek.

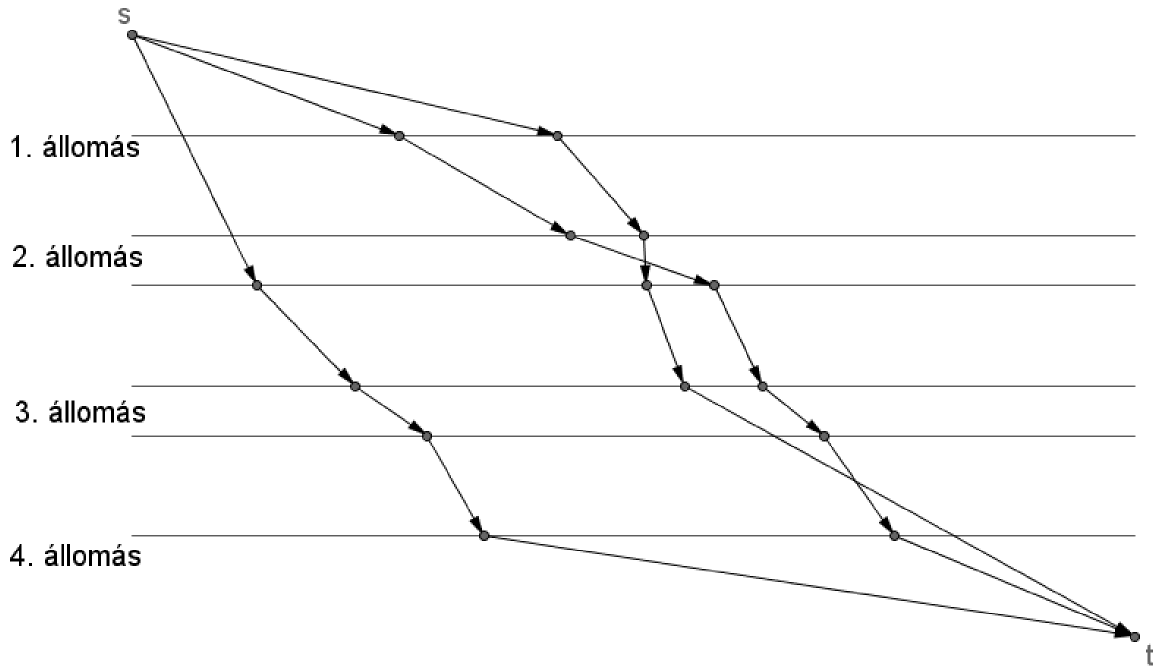
A gráf élhalmaza felbomlik $A^1, ..., A^\Gamma$ halmazokra, ahol minden élhalmaz egy-egy megfelelő vonatot reprezentál. Minden ilyen élhalmazban lévő él az alábbi típusok valamelyike:

- **Kezdőél**, azaz (s, v) alakú él, ahol $v \in W^{f_j}$, tehát a forrásból a kezdőállomás lehetséges indulási időpontjaiba kötött élek.

- **Állomásél**, azaz (u, v) alakú él, ahol $u \in U^i$ és $v \in W^i$ úgy, hogy $i \in S^j \backslash \{f_j, l_j\}$. Ez tehát minden érintett állomáson a lehetséges beérkezési időket és a lehetséges indulási időket összekötő él.

- **Állomásközi él**, azaz (v, u) alakú él, ahol $v \in W^i$ és $u \in U^{i+1}$ úgy, hogy $i \in S^j \backslash \{l_j\}$. Vagyis az adott állomást lehetséges elhagyási időpontokat és a következő állomásra való lehetséges beérkezési időpontokat összekötő él.

- **Befejező él**, azaz (u, t) alakú él, ahol $u \in U^{l_j}$, vagyis az utolsó állomásra való lehetséges beérkezési időket a nyelővel összekötő él.



2.3. ábra. Példa egy TTP problémából generált gráfra

Ebben a gráfban szeretnénk maximális megengedett egészértékű s-t folyamot keresni.

A fenti problémát át lehet írni egészértékű lineáris programmá (ILP - integer linear program). Minden egyes a élhez tartozzon egy x_a változó, ami pontosan akkor 1, ha az a él az optimális megoldásban szerepel. Jelölje továbbá $\delta_j^+(v)$ és $\delta_j^-(v)$ a $j \in T$ vonathoz tartozó, $v \in V$ csúcsból kilépő illetve csúcsba belépő éleket.

Korlátok:

1. $\sum\limits_{a\in\delta_{j+(s)}} x_a \leq 1$, vagyis a folyam értéke A^j élhalmazonként legfeljebb egy lehet, más szóval egy vonat nem indulhat többször.

2. $\sum\limits_{a\in\delta_{j-(v)}} x_a = \sum\limits_{a\in\delta_{j+(v)}} x_a$ ahol $j \in T, v \in V \backslash \{s,t\}$ vagyis a folyammegmaradásnak igaznak kell lennie pontonként és vonatonként.

3. $x_a + x_b \leq 1$, ahol $(a,b) \subseteq E \times E$ jelölik az egymást kizáró élpárokat. Lényegében ez garantálja, hogy megengedett menetrendet kapjunk, a feladat lényege ebben rejlik. Ennél a modellnél ezt

nem fejtem ki részletesen, a 2.2.2 és a 2.2.3 fejezet nagy része erről fog szólni.

4. $x_a \geq 0$

5. x_a egész

A célfüggvény pedig

$$\max \sum_{j \in T} \sum_{a \in A^j} p_a x_a,$$

ahol A^j a $j \in T$ vonathoz tartozó éleket jelöli, p_a pedig az éleken értelmezett profitfüggvény, ami az ideális menetrend élein a legmagasabb, attól távolodva pedig egyre alacsonyabb.

2.2. A modell bővítése

Ahhoz, hogy implementálni lehessen a 2.1.2 részben leírt modellt, olyan szinten, hogy azt valós adatokon futtatni lehessen, néhány dolog még feltétlenül hiányzik:

- Ki kell fejteni és pontosan leírni 2.1.2 fejezetben leírt 3. sorszámú korlátokat
- Le kell modellezni az állomások kapacitását is. Az előző modellben ugyanis arra volt csak megkötés, hogy két vonat nem érkezhet be vagy indulhat el egymáshoz időben túl közel. Nem volt arra korlát, hogy hány vonat tartózkodhat egyszerre egy állomáson
- Az előző megkötés miatt újra kell értelmezni azt, hogy mit kezdünk azokkal a vonatokkal, amik nem az egész vonalon, hanem annak csak egy részén közlekednek (tehát letér, vagy becsatlakozik)

A továbbiakban azt a modellt fogom részletesen és pontosan leírni, amit implementáltam.

Bemeneti adatként adva van:

1. Az állomások száma és neve
2. Az egyes állomásokon lévő használatban lévő vágányok száma
3. Az ideális menetrendben lévő vonatok száma
4. Minden egyes vonathoz az ideális menetrend szerint érintett állomások sorban
5. Minden vonathoz és minden érintett állomáshoz az ideális menetrend szerinti érkezési és indulási idő.

   A modell egyéb paraméterei:

6. **Denzitás**, azaz a diszkrét időpillanatok száma. Ezt a továbbiakban jelölje Δ.

7. **Flexibilitás**, azaz hogy az egyes járatok mennyire térhetnek el legfeljebb az ideálistól. Ez egy pozitív egész szám, és minden vonat minden érintett állomásánál a beérkezési idők és az indulási idők különbsége nem lehet ennél nagyobb az aktuális és az ideális menetrendben.
8. **Minimális idő két vonat között**, vagyis ha egy adott állomásra azonos irányból érkezik két vonat, vagy ha adott állomásról azonos irányba indul vonat, akkor legalább ennyi időnek el kell telnie közöttük.

2.2.1. Gráf-reprezentáció

Pontok létrehozása

A $G = (V, E)$ gráf létrehozásához először is szükségünk van a pontok meghatározására. Mindenekelőtt lesz a gráfnak egy s és t pontja, amik a keresett egészértékű folyamnak a forrása és nyelője lesznek. A további pontokhoz bevezetek egy fogalmat:

5. Definíció. *Idővonalnak* fogom hívni a gráfnak Δ darab különböző pontját, amik 0-tól Δ-ig az összes időpillanatot képviselik.

Az állomások lemodellezése miatt úgy döntöttem, hogy az első és az utolsó állomást is két idővonal fogja reprezentálni. Így tehát minden állomáson az egyik, (ábrán felső) pontsor fogja az adott időpillanatban, megfelelő állomásra való beérkezést jelenteni, a másik (ábrán alsó) pontsor pedig az állomás adott pillanatban történő elhagyását.

6. Definíció. Egy adott állomáshoz tartozó *érkező* illetve *induló idővonalnak* fogom hívni a gráf pontjainak azon részhalmazát, melyek az állomásra való beérkezést és állomásról való elindulást jelölik minden időpillanatban 1-től Δ-ig.

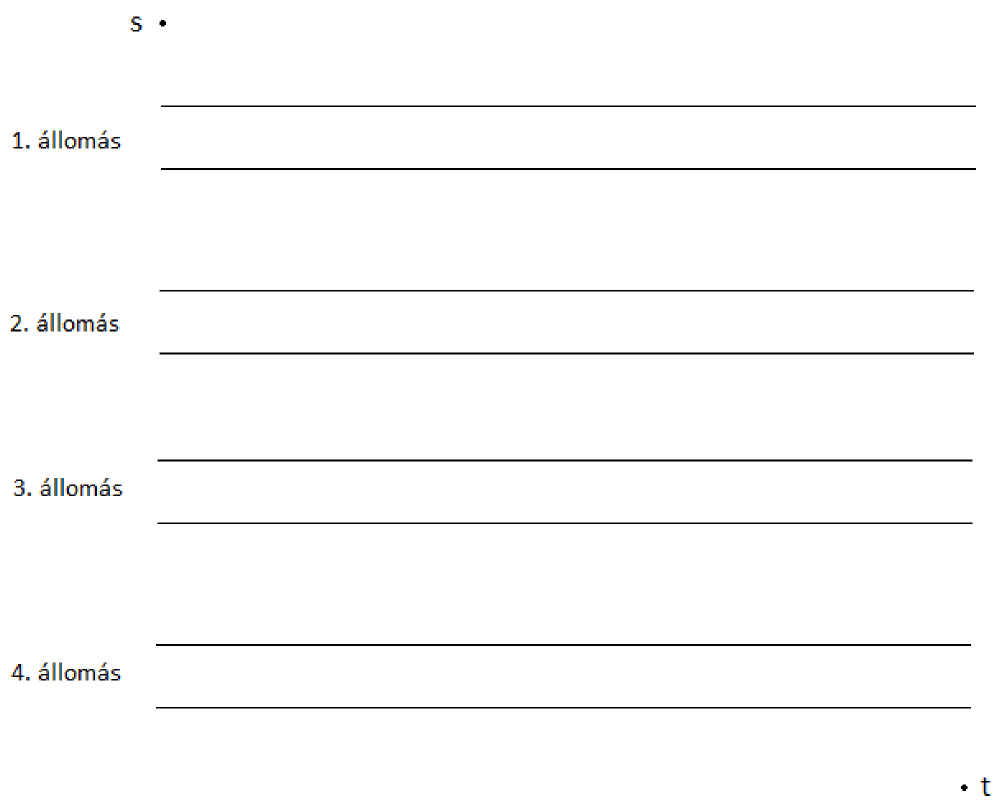
Ezen kívül pedig még szeretnék bevezetni egy jelölést: jelölje $A_x[y], x \in \{erk, ind\},\ y \in [0, \Delta]$ azt a pontot, ami az 'A' állomás (x-től függően) érkező vagy induló idővonalában van, és ott az y. időpillanatnak felel meg (vagyis az $y - 1$. pont, mert az időpillanatok 0-tól vannak indexelve). Például $Aprajafalva_{ind}[46]$ jelöli azt, hogy egy vonat Aprajafalváról a 46. percben elindul.

Ehhez tehát ismernünk kell a bemeneti adatok közül az 1.-t azaz az állomások számát és nevét, és a 6.-at azaz a denzitást (Δ).

Élek behúzása

A gráf élei fogják kijelölni azt, hogy egy járat mikor közlekedhet. Minden él tehát egy-egy vonathoz is tartozik, másképpen szólva egy él azt jelenti, hogy egy bizonyos adott vonat mikor megy.

Az élek behúzásához ismernünk kell először is az ideális menetrendet. Ez a bemeneti adatok közül a 3-as, 4-es és 5-ös pont. Az alábbi példában egy vonat szerepel a menetrendben, az halad végig négy állomáson, és minden állomásnál ismerjük az ideális érkezési és indulási időt:



2.4. ábra. 4 állomáshoz felvett pontok

Állomás neve	Beérkezési idő	Indulási idő
1. állomás	5	8
2. állomás	14	16
3. állomás	21	21
4. állomás	26	28

2.1. táblázat. Példa menetrend

Szeretném megjegyezni, hogy könnyen előfordulhat az is, hogy egy vonat egy adott állomáson nem áll meg, ezt úgy kezeltem le, hogy a beérkezési és az indulási idő megegyezik. Erre példa a 3. állomás.

Ezen kívül most fogom felhasználni a bemeneti adatok közül a flexibilitást, jelölje ezt F.

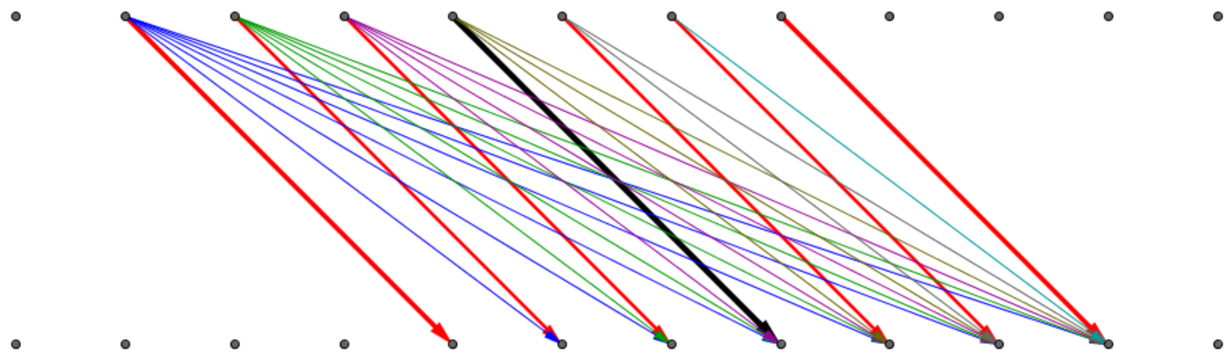
Az éleket pontosan ugyanúgy mint a 2.1.2 fejezetben négy részre lehet bontani: kezdőélre, befejező élre, állomásélre és állomásközi élre.

A kezdő és befejező élek behúzása egyszerű: jelölje q_{ind} és q_{bef} az első érintett állomásra való beérkezési időt, illetve az utolsó érintett állomásnál lévő indulási időt, továbbá jelölje E az első, U pedig az utolsó érintett állomást. Az adott vonathoz tartozó induló élek az s-et kötik össze rendre a $E_{erk}[i]$ pontokkal, ahol $i \in [q_{ind} - F, q_{ind} + F] \cap [0, \Delta] \cap \mathbb{N}$. A befejező élek pedig értelemszerűen a

$U_{ind}[j]$ pontokat kötik össze t-vel, ahol $j \in [q_{bef} - F, q_{bef} + F] \cap [0, \Delta] \cap \mathbb{N}$.

Az állomásokon belüli éleket és az állomások közötti éleket pedig a következő módon húzom be: jelölje I_j a j. vonathoz tartozó ideális menetrend által meghatározott st út éleit, kivéve az s-ből induló és a t-be érkező éleket. Továbbá legyen $\Upsilon : V \to \mathbb{N}$ az a függvény, ami a gráf egy adott pontjához hozzárendeli azt az időpillanatot, amit az reprezentál, és $\alpha, \omega : E \to V$ azon függvények, amik hozzárendelik egy adott élhez a kezdő, illetve végpontját.

Ekkor ha adott $e \in I_j$ akkor behúzzuk azokat az éleket, amik nincsenek időben ettől messzebb mint F, és nem is meredekebbek. Precízebben kifejezve a élt akkor húzzuk be, ha $|\Upsilon(\alpha(a)) - \Upsilon(\alpha(e))| \leq F$ és $|\Upsilon(\omega(a)) - \Upsilon(\omega(e))| \leq F$ és $\Upsilon(\omega(e)) - \Upsilon(\alpha(e)) \leq \Upsilon(\omega(a)) - \Upsilon(\alpha(a))$.



2.5. ábra. A sáv összes éle két idővonal között

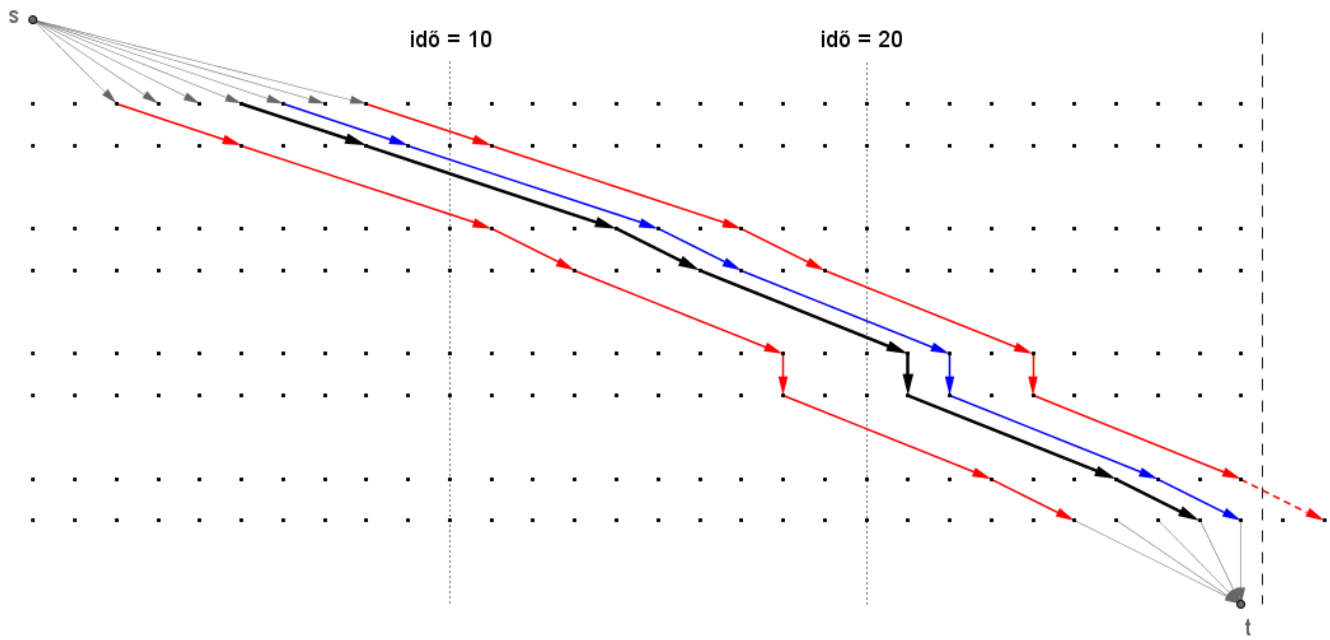
Az ábrán fekete éllel jelöltem e-t, tehát az ideális élt, pirossal azokat, amiknél nincs késleltetés, tehát két állomás között a maximális sebességgel mennek, állomáson belül pedig nem várakozik többet a vonat, mint szeretne. Az egyes pontokból induló késleltetett éleket pedig különböző színnel jelöltem aszerint, honnan indulnak.

Jól láthatóan tehát egy $2F$ széles "sávon" belül húzom be az éleket. Egy ilyen vonathoz tartozó teljes sávot láthatunk a 2.6 ábrán, de itt már a sávon belüli éleket az átláthatóság érdekében nem húztam be.

Fontos még megjegyezni, hogy amennyiben ez a sáv a szélessége miatt kilóg az idővonalról, akkor a kilógó éleket nem húzom be.

Illetve szeretném még arra is felhívni a figyelmet, hogy elképzelhető, hogy egy vonat az ideális menetrend szerint nem állna meg egy adott állomáson, azonban mégis meg fog állni a megengedett menetrend szerint.

7. Állítás. *Egy járat eltérése az ideális menetrendtől monoton nő.*



2.6. ábra. A 2.1 táblázatban leírt menetrendhez tartozó sáv

Bizonyítás. Az állítás majdnem triviális, hiszen az ideálisnál gyorsabban sosem mehet vonat, illetve az állomásokon sem várakozhat kevesebb ideig. □

A 7. állítás miatt az is könnyen látható, hogy az előző ábrán amiatt, hogy a járat "kilóg" az idővonalról, emiatt a kék éleken kívülre már nem is mehet folyam, hiszen nincsenek már összekötve a t-vel. Ilyenkor tehát az összes, kéktől jobbra lévő él tulajdonképpen fölösleges.

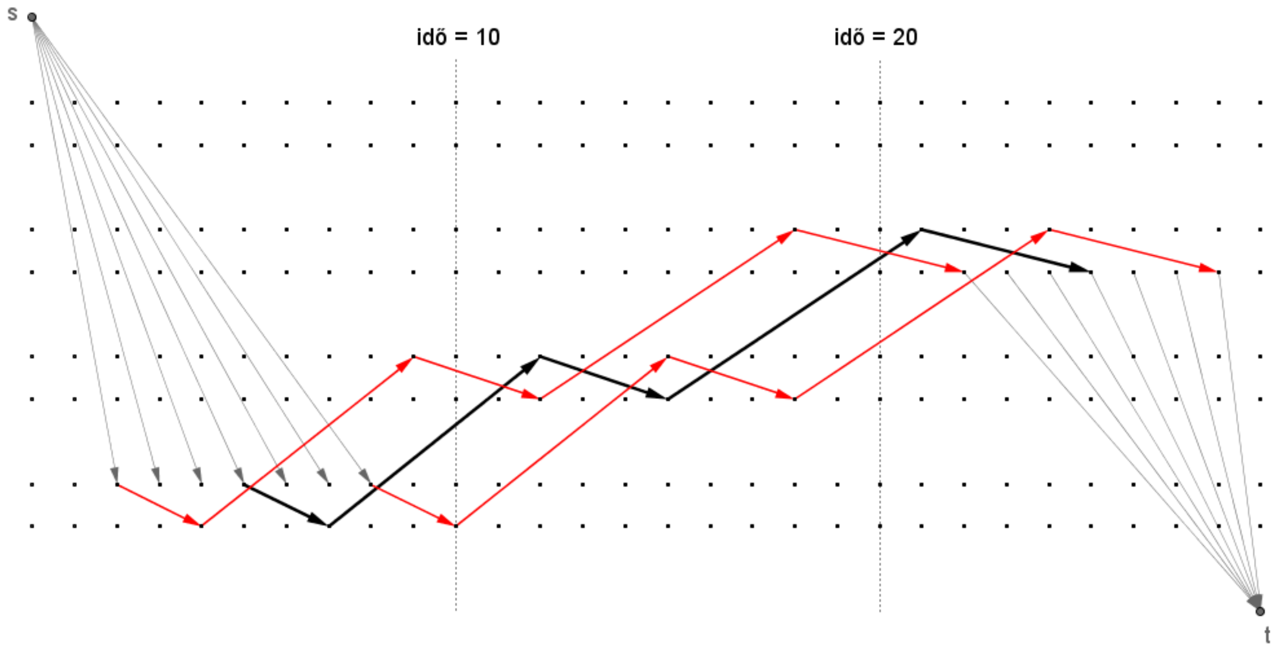
Végezetül pedig még egy ábra (2.7), amin egy visszafelé haladó vonaton ábrázolok, az alábbi menetrend szerint:

Állomás neve	**Beérkezési idő**	**Indulási idő**
4. állomás	5	7
3. állomás	12	15
2. állomás	21	25

2.2. táblázat. Példa menetrend visszafelé jövő vonattal

8. Állítás. *Legyen egy adott járat által érintett állomások száma A. Ekkor ehhez a vonathoz pontosan $(4A-2)F^2+(6A+1)F+1$ él tartozik a gráfban.*

Bizonyítás. Először is van $2F+1$ kezdőél, illetve $2F+1$ befejező él.



2.7. ábra. A 2.2 menetrendhez tartozó gráf

Ezen kívül az állomásokon belül és két állomás között is két idővonal között lévő élek száma $(2F+1)+2F+(2F-1+...+1)=\frac{(2F+1)(2F+2)}{2}$. Összesen A helyen vannak állomáson belüli élek, és $A-1$ helyen állomások közötti.

Összesen tehát az élek száma $4F+2+\frac{(2F+1)(2F+2)}{2}(2A-1)=(4A-2)F^2+(6A+1)F+1$ $\square$

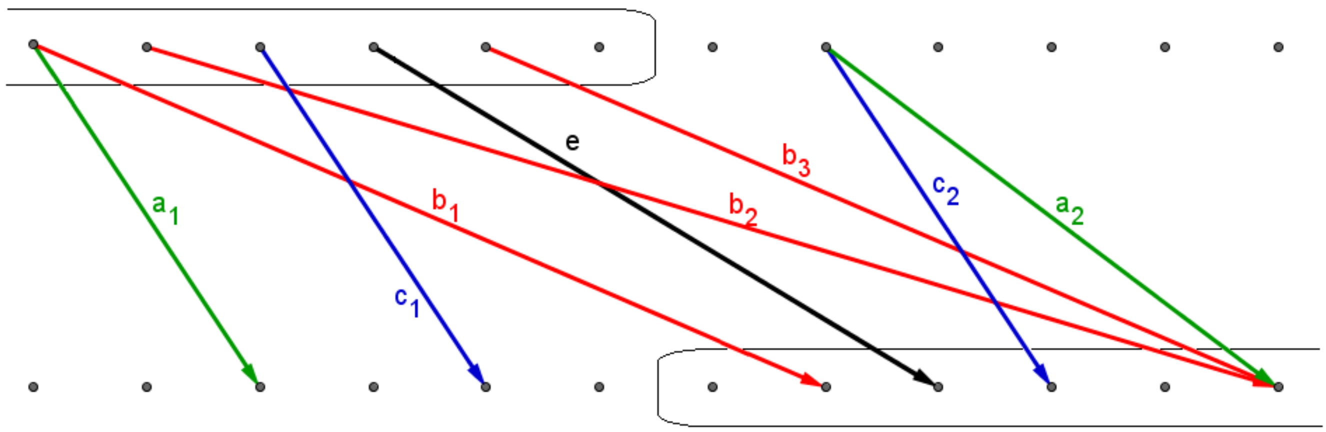
2.2.2. Állomások közötti korlátok

Az állomások közötti megszorításokkal két dolgot szeretnék korlátozni: az egyik, hogy semmikor nem előzhet meg két állomás között egy vonatot egy másik, amelyik ugyanarra megy. Azért csak egy irányba menő vonatokra van most ez a megszorítás, mert a pálya, amire le fogom futtatni a program kétvágányú. Azzal a feltételezéssel élek, hogy két állomás között az egyik vágányon csak az egyik irányba mehet vonat, a másikon a másikba. A másik, hogy két vonat nem indulhat el vagy érkezhet meg ugyanarra az állomásra időben túl közel. Ezt a két dolgot egyben fogom lekezelni, méghozzá úgy, hogy felsorolok nagyon sok élpárt, amik egymást kizáróak, és páronként megkötöm, hogy legfeljebb az egyik szerepelhet a folyamban. Szükségünk lesz a bemeneti adatok közül a 8-asra, vagyis a minimális időre két vonat között, ezt jelölje a továbbiakban τ. Ez a gráfra vonatkoztatva annyit jelent, hogy két induló folyamél között legalább $\tau-1$ pontnak kell lennie.

Először végigiterálok az összes állomások közötti élen (2.8 ábrán feketével jelöltem). Az aktuális élt ahol az iterálás tart jelölje a továbbiakban e.

Ezután minden olyan a él kizáró lesz az e-vel, amire

- $\Upsilon(\alpha(a))-\Upsilon(\alpha(e))<\tau$, és



2.8. ábra. Két állomás közötti korlátok. Az ábrán $\tau = 3$

- $\Upsilon(\omega(e)) - \Upsilon(\omega(a)) < \tau$, és
- a és e másik vonathoz tartozó élek,

ahol a jelölések ugyanazt jelentik, mint a 2.2.1 fejezetben.

Ez szemléletesen azt jelenti, hogy minden él, ami a fenti kijelölt halmazból indul és a lenti kijelölt halmazba érkezik, az kizáró él e-vel. Az, hogy másik vonathoz tartozik, nem szükséges megkötés, de ha lenne, fölösleges lenne, és rengeteg új sort produkálna az IP-ben. Fölösleges, hiszen egy vonat csak egy időben mehet (s-ből kilépő adott folyamhoz tartozó folyamérték legfeljebb 1, lásd 2.2.5 fejezeten belül a (3.2) korlátozást), nem kell azt külön kizárni, hogy önmagát ne előzze meg.

Nézzük, hogy ez mit jelent. Az ábrán zölddel jelölt a_1 és a_2 éleket nem zárja ki a megkötés, és nem is kell, mert mehet mind a kettőn egyszerre folyam. A b_1 él kizáró, mert egymáshoz túl közel érkeznek időben a következő állomásra. Hasonlóan b_3 azért kizáró, mert egymáshoz túl közel indulnak az előző állomásról. b_2 pedig azért kizáró, mert az e élen haladó vonat megelőzné a b_2-n haladót, pedig egy irányba csak egy vágány van. Ezen kívül a c-vel jelölt éleknek is kizárónak kellene lenniük, hiszen e-hez túl közel indulnak illetve túl közel érkeznek. Ez itt mégsem kerül be a megkötések közé, azonban amikor az állomások közötti éleken való iterálás a c_1-nél illetve a c_2-nél fog tartani, akkor a kizáró élek között szerepelni fog az e (c_1-nél az e úgy lesz kizáró, mint a példában e-nél a b_3, illetve c_2-nél úgy lesz kizáró, mint e-nél a b_1).

Szeretném megjegyezni, hogy ez az eljárás a visszafelé haladó vonatok állomás közötti élein is végigiterál, illetve az eljárás során az $\alpha(e)$ és $\omega(e)$ csúcsokhoz tartozó idővonalakkal dolgoztam, ezért minden ugyanúgy működik ott is, csak az ábra máshogy nézne ki.

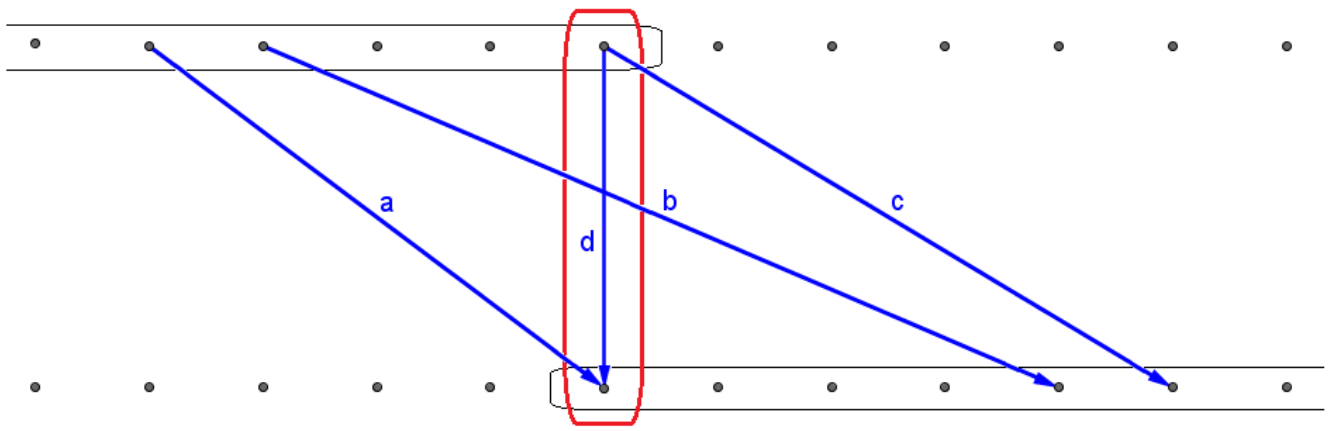
Végül pedig jelölje a további fejezetekben az itt egymást kizárónak nevezett élpárok halmazát K.

2.2.3. Állomásokon belüli korlátok

Ezeknél a megszorításoknál azt szeretném elérni, hogy sose álljon több vonat az állomáson belül, mint ahány vágány van. Másképpen megfogalmazva minden időpillanatban igaz az, hogy az állomáson lévő vonatok száma kisebb vagy egyenlő, mint a vágányok száma. Pontosan így is programoztam le: végigiterálok az időpillanatokon, jelölje az éppen aktuálist γ_0 (2.9 ábrán pirossal jelölve). Ezután minden e élt felsorolok, amire igaz, hogy

- $\Upsilon(\alpha(e)) \leq \gamma_0$ és
- $\gamma_0 \leq \Upsilon(\omega(e))$

Ez tehát azt jelenti, hogy vagy áthalad a piros időmetszeten, vagy abból indul, vagy oda érkezik. Valójában ez nagyon hasonló az előző, 2.2.2 fejezetben megismert korlátokhoz, úgy is lehet érteni, hogy a felső feketével keretezett halmazból induló, és a lenti feketével keretezett halmazba érkező éleket felsorolom.



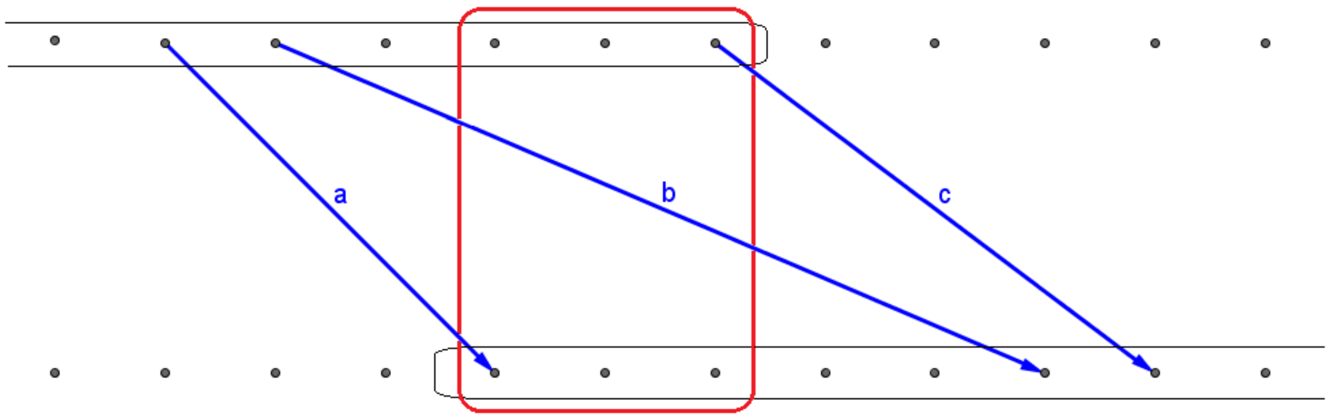
2.9. ábra. Pirossal jelölve az adott időpillanat ahol az iterálás tart, kékkel a számba vett élek

Ez az a négy típusú él lehet, ami az ábrán is a, b, c, d nevű élekkel jelöltem. Fontos megjegyezni azt is, hogy mindegy, melyik irányba áll be az állomásra a vonat, mert ugyanúgy használja a vágányt, és mivel bármerről jön, a felső pontsorba van bekötve, ezért ide már az is beszámít.

Most felhasználjuk az utolsó bemeneti adatot a 2.2 fejezetben megnevezettek közül a 2-es számút, nevezetesen, hogy hány vágány van egy állomáson belül. Ugyanis a megszorítás az lesz, hogy az adott időpillanathoz felsorolt éleken folyó folyam összege kisebb vagy egyenlő kell, hogy legyen, mint az adott állomáson lévő vágányok száma.

A további fejezetekben jelölje $J_{i,j}$ az i. állomás j. időpontjához tartozó olyan éleket, amik ezt az időpontot érintik a fentebb leírtaknak megfelelően.

Felmerült bennem egy lehetséges probléma: ha egy vonat elhagy egy vágányt, akkor a fenti leírás szerint a következő pillanatban már rögtön bejöhet ugyanarra a vágányra egy másik vonat. Ebben a modellben nincsen pontosan leírva, hogy melyik vonat mikor melyik vágányon áll, illetve hogy a váltókon tud-e úgy menni, hogy arra a vágányra kerüljön, amelyikre kellene neki. Én azt feltételeztem, hogy bárhonnan érkezve egy állomásra az összes vágányra el lehet jutni, és mindegyik vágány megfelelő minden vonatnak (lásd 1.1.3). Viszont ha n vágány van, és mindegyiken áll vonat, majd az egyik elmegy, akkor biztosan ugyanarra érkezne a következő akkor, ha rögtön utána érkezne. A modell egyik erőssége az, hogy a fenti megkötéseken nem kell sokat változtatni, és alkalmas lesz arra, hogy a két (ugyanazon állomásra egy irányból beérkező) vonat közötti minimális időt figyelembe vegye (2.10 ábra):



2.10. ábra. El kell telnie $\epsilon = 2$ egység időnek a vágány felszabadulása után a következő vonat előtt

Legyen az az idő, ameddig egy vágánynak üresnek kell lennie két vonat között ε. Annyi a különbség az előzőhöz képest, hogy most nem az időpillanatokon, hanem a $\varepsilon+1$ széles időintervallumokon iterálok végig, és minden él, ami a fenti fekete halmazból indul és a lentibe érkezik (tehát érinti a pirossal jelölt időintervallumot) az felsorolásra kerül, és végül az ezeken lévő folyamösszeg kell, hogy kisebbe vagy egyenlő legyen, mint a vágányszám.

Szeretném megjegyezni, hogy ez a minimális idő nem egészen ugyanaz, mint a bemeneti adatok közül a 8-as, vagyis a minimális idő két vonat között. Előbbi arra vonatkozik, hogy egy bizonyos vágányon például el kell telnie 1 percnek, mire a következő vonat beállhat, az utóbbi (ami szerepel a modellben) pedig arra vonatkozik, hogy például az állomás elején lévő váltót esetleg át kell állítani, ezért legalább τ időnek el kell telnie két, egy irányból beérkező vonat között.

2.2.4. Célfüggvény

Az eredeti cikkben ([2]) leírt profitfüggvényen is változtattam. Többet között a következő volt vele a problémám: a ν_j az ideálistól eltérő indulást, a μ_j pedig a ideálisnál hosszabb futási időt büntetni. Ebből az következik, hogy egy olyan vonat, ami mondjuk 15 perccel hamarabb indul, de rögtön az első állomáson 15 percet vár, onnantól kezdve további 15 állomáson keresztül végig az ideális éleken megy, az viszonylag rossznak számít. Úgy gondoltam, hogy ennek jónak kéne lennie, ezért minden élhez hozzárendeltem egy profitot. Ahogy egyre távolabb kerülünk az ideális éltől úgy fog monoton csökkenni az élen szerzett profit. Precízebben megfogalmazva ha e az ideális él, akkor az a élen keletkező profit

$$\pi(a) = 1 - \frac{1}{1000}\frac{|\Upsilon(\alpha(a)) - \Upsilon(\alpha(e))| + |\Upsilon(\omega(a)) - \Upsilon(\omega(e))|}{2}$$

Látható, hogy az egyes éleken szerzett profit az ideálistól távolodva lineárisan csökken. Tehát mindegy, hogy valahol két vonat késik egy percet, vagy egy késik kettőt.

Megfontolandó lenne más célfüggvények használata, illetve a különbözőség következménye. Előnyös lenne, ha az ideálistól való eltérést például négyzetesen büntetnénk. De szintén erősebbé tenné a modellt, ha minden vonathoz tartozna egy szám, ami azt jellemzi, hogy mennyire rossz, ha az a vonat késik.

2.2.5. Formalizálás, implementálás

Ebben a fejezetben szeretném összefoglalni és formálisan felírni, hogy a modellemben milyen megkötéseket tettem. Jelölje

- x_a azt az indikátorváltozót, ami pontosan akkor 1, ha az x-en megy folyam,
- π_a az a élhez tartozó profitot (2.2.4),
- továbbra is $\delta_j^+(v)$ és $\delta_j^-(v)$ a $j \in T$ vonathoz tartozó, $v \in V$ csúcsból kilépő illetve csúcsba belépő élek halmazát, ugyanúgy mint a 2.1.2 fejezetben,
- K az 2.2.2 fejezet alapján azon élpárok halmazát, amik az állomások közötti megszorítás miatt egymást kizáróak,
- $J_{i,j}$ a 2.2.3) fejezet alapján azon éleket, amik az i. állomás j. időpontjában az állomáson tartózkodást reprezentálják,
- $z : \mathbb{N} \to \mathbb{N}$ azt a függvényt, ami hozzárendeli az i. állomáshoz az ott lévő vágányok számát,
- $\{1, ...\Psi\}$ az állomások halmazát.

Célfüggvény:

$$\max \sum_{x_a \in E} \pi_a x_a \tag{2.1}$$

Korlátok:

$$\sum_{a \in \delta_j^+(s)} x_a \leq 1 \qquad \forall j \ vonatra \tag{2.2}$$

$$\sum_{a \in \delta_j^+(v)} x_a = \sum_{a \in \delta_j^-(v))} x_a \qquad \forall v \in V \backslash \{s, t\} \tag{2.3}$$

$$x_a + x_b \leq 1 \qquad \forall (a, b) \in K \tag{2.4}$$

$$\sum_{a \in J_{i,j}} x_a \leq z(i) \qquad \forall i \in \{1, ...\Psi\}, j \in \{0, ..., \Delta\} \tag{2.5}$$

$$x_a \geq 0 \tag{2.6}$$

$$x_a \in \mathbb{Z} \tag{2.7}$$

A 2.2 megkötés garantálja, hogy egy vonat csak egyszer megy valahol, a 2.3a folyammegmaradást. A 2.4 jelenti azt a megkötést, hogy két állomás között nem előzhet meg egy vonat egy vele azonos irányba haladó másikat. A 2.5 korlát szerint pedig egy állomáson sem tartózkódhat több vonat egyidejűleg, mint a párhuzamosan használható vágányok száma. A 2.6 és 2.7 korlátok pedig technikai jellegűek (minden változó nemnegatív, és egész).

Ezt a modellt a 2.2 fejezet eddigi része során leírt részletekkel programoztam le C++ programnyelven. A gráf létrehozásához használtam a LEMON csomagot, majd miután elkészült a gráf, a FICO Xpress 7.5 programot használtam az egészértékű program megoldására.

2.2.6. Modell bővítésének lehetőségei

A forrásként felhasznált cikkben leírt modellen azért változtattam, hogy használhatóbb modellt kapjak a gyakorlati alkalmazás céljából. Azonban természetesen nem egyértelmű, hogy mit kell, mit nem célszerű bevenni a modellbe. Az alábbiakban felsorolok néhány lehetőséget ami a modell kialakítása során felmerült, de mégsem valósítottam meg. Ennek a modellnek a további alakításai során ezek mindenképpen megfontolandók.

Kézenfekvő lehetőség, hogy minden vonathoz tartozik néhány olyan időpont, hogy egy bizonyos állomásra addig az időpontig meg kell érkeznie, vagy ennél az időpontnál hamarabb nem hagyhatja el. Természetesen ennek a gyakorlati jelentősége abban áll, hogy az átszállásokat megkönnyítse. Megvalósítani sem lenne nehéz, egyszerűen a nem kívánt éleket nem húzzuk be. Amikor a program létrehozza a gráfot és húzza be az éleit, ott egy ellenőrző függvény, ami minden élre megvizsgálja, hogy az nem kívánt-e az átszállások miatt. Leginkább azért nem valósítottam meg, mert a modellem alapvetően korridort vizsgál, tehát nem igazán értelmes az átszállás fogalma.

Ezzel kapcsolatos a következő felvetés is. Sokat bonyolítana, de megvalósítható lenne az, hogy ne csak egy korridort modellezzek le, hanem egy bonyolultabb hálózatot, amiben lehetnek elágazások és körök is. Valójában már a saját modellemben sem használtam ki, hogy az állomások szigorúan rendezettek. Az inputból beolvasható, hogy mi a következő állomás, és annak az állomásnak a beérkező pontjaiba kell húzni az élt, nem szükséges, hogy ez szomszédos állomás legyen. Az állomáson belüli megkötések ugyanúgy működnének, az állomások közöttiek pedig minden elágazás közötti korridorra felírható.

Ezeken kívül a meglévő modell bővítése helyett "okosabbá" is lehet tenni. Az már most is állomásfüggő, hogy hány vágány van ott, de a minimális idő az nem. Pedig elképzelhető, hogy egyik állomáson modernebb, gyorsabb a váltó, jobb a biztosítóberendezés, úgyhogy kevesebb idő is elegendő két vonat között.

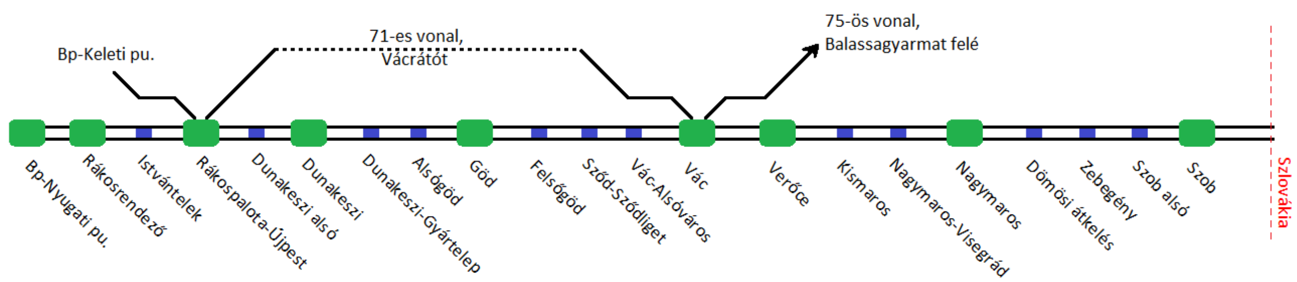
Másik irány annak a lehetőségnek a beépítése, hogy egy kétvágányú vonalon esetenként mindkét vágányon közlekedhet vonat ugyanabba az irányba. Például egy személyvonat megáll egy megállóban, majd ha már úgyis megállt átmegy a másik vágányra és ott megy a következő állomásig (természetesen közben más nem jöhet szembe). Addig pedig egy gyors, amelyik meg sem áll egyik állomáson se, a saját vágányán megelőzheti a személyvonatot, és így anélkül valósult meg az előzés, hogy a személynek hosszabban kellett volna várnia egyik állomásán.

3. fejezet

A saját modell futtatása valós adatokon

Ebben a fejezetben a le fogom írni és elemezni fogom a program valós adatokon való futtatásának eredményét. A Budapest–Szob viszonylatra esett a választásom két okból is. Egy olyan pályára volt szükségem, amire minél inkább igaz, hogy korridor, tehát ugyanolyan fontos vasúti pálya nem csatlakozik be vagy ágazik le róla. Budapest és Szob között ez nagyjából igaz, kevés kivétellel, mint azt később látni fogjuk. Ezen kívül fontos volt az is, hogy nagy legyen a pálya kihasználtsága. Ugyanis ha egy olyan vonalat néznék, ahol két óránként megy vonat, akkor nem lenne nagyon hasznos egy javító algoritmus, majdnem minden ideális menetrend megengedett is. Célszerűnek tűnt tehát olyan vonalat választani, ami Budapestet és agglomerációját köti össze, és emiatt csúcsidőben legalább félóránként megy vonat. A továbbiakban először szeretném leírni a Budapest–Szob vonal jellemzőit.

3.1. A Budapest–Szob viszonylatról



3.1. ábra. A 70-es vonal sematikus ábrázolása

A pálya száma 70-es, a Nyugati pályaudvartól indul és a Duna bal partján azzal többnyire párhuzamosan megy fel a szlovák határig. A pálya 64 km hosszú, végig villamosított és kétvágányú. Ezen a vonalon többnyire az elővárosi közlekedést kiszolgáló vonatok közlekednek. Egészen pontosan 5 féle járat közlekedik a viszonylaton:

- **Személyvonat**. Minden állomáson és megállóhelyen megáll.
- **Gyorsított**. Hasonló a személyvonathoz, kisebb utasforgalmú állomásokon nem áll meg.
- **Zónázó**. Nyugati pályaudvar és Vác között nem áll meg sehol, utána mindenhol.
- **Vácig közlekedő személy**. Nyugati pályaudvar és Vác között közlekedik, és megáll mindenhol.
- **EC - nemzetközi gyors**. Ezek a vonatok a Keleti pályaudvarról indulnak, és a 70-es vonalra Rákospalota-Újpesten csatlakoznak be. Szob után a határon átkelve folytatják útjukat.

A pályára vonatkozó érvényes menetrendet a [3] forrás alatt megjelölt weboldalról töltöttem le. Ezt utána legépeltem és átalakítottam olyan formátumúra, amilyen inputra a programnak szüksége volt.

A továbbiakban a modellt felhasználva két lehetséges esemény hatását fogom megvizsgálni. A vonalon egy nap alatt mindkét irányba összesen 132 vonat megy. Emiatt egyik példa esetében sem fogom az egész javított menetrendet bemutatni, hiszen nem férne el. A dolgozathoz tartozik CD-melléklet, azon megtalálható az egész javított menetrend.

3.2. ábra. Az összes állomás amit a vonal érint, zölddel a modell szempontjából fontos állomások

Km	Állomás neve	Vágányok száma	Menetrend szerint megáll-e: 1=igen, 0=nem					Megjegyzés
			Személy	Gyorsított	Zónázó	Vácig közlekedő	EC	
0	Bp-Nyugati	20	1	1	1	1		
3	Rákosrendező	20	1	1	0	1		
6	Istvántelek	2	1	0	0	1		Keleti pu. felől becsatlakozik egy párhuzamos vágány, de váltó itt még nincs
8	Rákospalota-Újpest	6	1	1	0	1	0	Leágazó pálya Vácrátót felé és becsatlakozó vágány a Keleti pu. felől
13	Dunakeszi alsó	2	1	0	0	1	0	Utasforgalmi okokból fontos állomás, egyébként váltó nincs
15	Dunakeszi	8	1	0	0	1	0	
17	Dunakeszi-Gyártelep	2	1	1	0	1	0	
21	Alsógöd	2	1	0	0	1	0	
22	Göd	6	1	0	0	1	0	
24	Felsőgöd	2	1	1	0	1	0	
27	Sződ-Sződliget	2	1	0	0	1	0	
32	Vác-Alsóváros	2	1	1	0	1	0	Valójában 3 vágány van a Vácrátót felől becsatlakozóval, de váltó csak Vácon van
34	Vác	8	1	1	1	1	1	Becsatlakozó vonal Vácrátót felől, leágazás Balassagyarmat felé
43	Verőce	4	1	1	1		0	Az állomás előtt, váltó nélkül letér a 75-ös vonal (Balassagyarmat felé)
46	Kismaros	2	1	1	1		0	
51	Nagymaros-Visegrád	2	1	1	1		1	
52	Nagymaros	6	1	1	1		0	
55	Dömösi átkelés	2	0	0	0		0	Márc. 15. és okt. 2. között a zónázó vonat itt is megáll
59	Zebegény	2	1	1	1		0	
63	Szob alsó	2	1	1	1		0	
64	Szob	8	1	1	1		1	Tovább megy a pálya Párkány, Szlovákia felé

3.2. Demonstráció: a nemzetközi vonatok módosulása

3.3. ábra. A példában az utolsó 3 oszlopnak megfelelő vonatokat fogom megvizsgálni

Km	MÁV-START Zrt.	❄ *E* 2134 2.◆ 🚲	*E* 2354 2.◆ 🚲 91		*E* 2144 2.◆ 🚲	❄ *O* EC 274 R◆ 🚲R	*E* 2214 2224 2.◆ 🚲 92
	Kiindulási állomás						
0	**Budapest-Nyugati** 2 71	13 07	13 11	...	Ⓐ13 28	...	13 41
3	**Rákosrendező**		13 16	...	13 33	...	13 46
6	Istvántelek		13 20	...		...	13 50
8	**Rákospalota-Újpest**		13 22	...	13 38	...	13 52
	0 **Budapest-Keleti**			...		13 25	
	Rákospalota-Újpest		13 23	...	13 39	JAROSLAV HAŠEK	13 53
13	Dunakeszi alsó		13 28	...			13 58
15	Dunakeszi		13 31	...			14 01
17	Dunakeszi-Gyártelep		13 34	...	13 47		14 04
21	Alsógöd		13 38	...			14 08
22	Göd		13 40	...			14 10
24	Felsőgöd		13 43	...	13 53		14 13
27	Sződ-Sződliget		13 46	...			14 16
32	**Vác-Alsóváros** 71		13 51	...	13 59		14 21
34	**Vác 75**	13 32	13 54	...	14 01	14 09	14 24
	Vác	13 34	...	...	14 04	◘ 14 10	Ⓐ14 28
43	Verőce	13 42	...	...	14 12		14 36
46	Kismaros *317*	13 45	...	...	14 15		14 39
51	Nagymaros-Visegrád	13 50	...	...	14 20	14 24	14 44
52	Nagymaros	13 52	...	...	14 22		14 46
55	Dömösi átkelés		...	...			
59	Zebegény	13 59	...	...	14 29		14 53
63	Szob alsó	14 03	...	...	14 33		14 57
64	**Szob *318***	14 05	...	...	Ⓐ14 35	14 35	Ⓐ14 59
	Szob	...	...	...	...	◘ 14 36	...
	Štúrovo	...	...	...	...	14 46	...
	Végállomás					Praha	

● Dömösi átkelésen csak III.15 - XI.2-ig áll meg.

91 Közlekedik: XII.23-ig és I.2-től.

Első példaként azt szeretném illusztrálni, hogy egy egészen apró változás is mennyi változást tud okozni. Tegyük fel azt, hogy a nemzetközi vonatok menetrendje úgy módosul, hogy mindenhol 5 perccel később lesz. Ez egyébként azért sem irreális feltevés, mert ezek a járatok nagyon gyakran késnek. És mivel elsőbbséget élveznek a belföldi járatokkal szemben, ezért sokszor tovagyűrűző hatásuk van, és az összes többi vonat menetrendje is felborul. Persze a menetrend azonnali javítása egy egészen másik kérdés (lásd erről a 1.2 fejezetet). Mégis vizsgáljuk most meg azt, hogy ha a nemzetközi vonatok menetrendje is módosulna, akkor mi történne. A példában a Jaroslav Hašek nevű nemzetközi vonat jön 5 perccel később, és megvizsgáljuk mi történik a környékén lévő vonatokkal.

Várakozásainknak megfelelően a gyorsított vonat nem módosult, hiszen a mögötte lévő vonat még később megy, tehát nem konfrontálódnak. Az eredeti menetrend szerint 13:53-kor Rákospalota-Újpest állomáson megáll a személy, megvárja amíg a Keleti pályaudvar felől becsatlakozó EC elmegy, majd utána indul el. Azzal, hogy az EC később jön, ez csak úgy valósulhatna meg, ha a személyvonat is 5 perccel többet vár Rákospalotán. Viszont ha a személyvonat be tud érni a következő állomásra, akkor jobb, ha az EC már csak ott előzi meg. Pontosan ezt láthatjuk megvalósulni is, a személyvonat Gödre ér be éppen, és ott az eredeti 1 perc helyett 2 percet áll, és ezalatt az EC meg tudja előzni. A személyvonat pedig a többi vonat miatt csak Verőcén áll egy kicsit többet, hogy utána az ideális menetrend szerint menjen tovább.

3.4. ábra. A három vonat ideális és javított menetrendje. Naranccsal jelöltem azt, ahol az ideális menetrend szerint ütközés van, zölddel azt, ahol módosulás történt az ideálishoz képest

		Ideális menetrend		Javított menetrend	
	Állomás	Érkezés	Indulás	Érkezés	Indulás
Személy	Nyugati pályaudvar	13:40	13:41	13:38	13:39
	Rákosrendező	13:45	13:46	13:43	13:44
	Rákospalota-Újpest	13:52	13:53	13:50	13:51
	Dunakeszi	14:00	14:01	13:58	13:59
	Göd	14:09	14:10	14:07	14:09
	Vác	14:23	14:28	14:22	14:27
	Verőce	14:35	14:36	14:34	14:36
	Nagymaros	14:45	14:46	14:45	14:46
	Szob	14:59	15:00	14:59	15:00
Gyorsított	Nyugati pályaudvar	13:27	13:28	13:27	13:28
	Rákosrendező	13:32	13:33	13:32	13:33
	Rákospalota-Újpest	13:38	13:39	13:38	13:39
	Dunakeszi	13:44	13:44	13:44	13:44
	Göd	13:48	13:48	13:48	13:48
	Vác	14:01	14:04	14:01	14:04
	Verőce	14:11	14:12	14:11	14:12
	Nagymaros	14:21	14:22	14:21	14:22
	Szob	14:35	14:36	14:35	14:36
EC	Rákospalota-Újpest	13:58	13:58	13:58	13:58
	Dunakeszi	14:04	14:04	14:04	14:04
	Göd	14:08	14:08	14:08	14:08
	Vác	14:14	14:15	14:14	14:15
	Verőce	14:21	14:21	14:21	14:21
	Nagymaros	14:30	14:30	14:30	14:30
	Szob	14:40	14:41	14:40	14:41

3.3. Demonstráció: a váci állomás felújítása

Ebben a részben abból indultam ki, hogy a váci állomást felújítják (ez valóban így van, 2013. december 15-én kezdték, májusig még nem fejeződött be). Ez idő alatt változó korlátozások vannak érvényben. Én ezek közül egy elképzelhetőt vettem alapul: minden Vácra beérkező vonat és onnan kimenő menetideje megnő 3 perccel. Tehát a Vácra közlekedő személyvonatnak csak 3 perccel, de minden áthaladónak 6 perccel. Ez azért reális feltevés, mert ha a pálya előtt és után is felújítanak, akkor mindkét oldalon van jelentős sebességkorlátozás, ezért az állomás előtt és után is perceket veszíthet egy vonat. Illetve ezen kívül még azt is feltettem, hogy a nemzetközi vonat menetrend szerint 20 perccel hamarabb jön.

Az ideális menetrenddel több gond adódik. Egyrészt Nagymaros és Verőce között ütközne a személy és az EC, másrészt Vácra egyszerre érkezik be a zónázó és az EC. Ez azért baj, mert ha egy irányból jönnek, és egy percen belül akkor biztosan nincs elég idő váltót átkapcsolni (és muszáj is átkapcsolni, mert az EC van hátrébb, és meg fogja előzni a zónázót). A program pedig úgy javítja ki ezeket, hogy a személy kicsit korábban megy, így Verőcén el tudja engedni az EC-t, illetve a zónázó is 1 perccel hamarabb megy, hogy még éppen beérjen Vácon a nemzetközi vonat elé.

3.5. ábra. Négy vonat ideális és javított menetrendje. Pirossal jelöltem azt, ahol az ideális menetrend szerint ütközés van, zölddel azt, ahol módosulás történt az ideálishoz képest

		Ideális menetrend		Javított menetrend	
	Állomás	Érkezés	Indulás	Érkezés	Indulás
Személy	Szob	18:59	19:00	18:57	18:58
	Nagymaros	19:12	19:13	19:10	19:11
	Verőce	19:22	19:23	19:20	19:23
	Vác	19:34	19:37	19:34	19:37
	Göd	19:52	19:53	19:52	19:53
	Dunakeszi	20:01	20:02	20:01	20:02
	Rákospalota-Újpest	20:09	20:10	20:09	20:10
	Rákosrendező	20:16	20:17	20:16	20:17
	Nyugati pályaudvar	20:23	20:24	20:23	20:24
Zónázó	Szob	18:54	18:55	18:53	18:54
	Nagymaros	19:07	19:08	19:06	19:07
	Verőce	19:17	19:18	19:16	19:17
	Vác	19:29	19:31	19:28	19:31
	Göd	19:41	19:41	19:41	19:41
	Dunakeszi	19:46	19:46	19:46	19:46
	Rákospalota-Újpest	19:51	19:51	19:51	19:51
	Rákosrendező	19:56	19:56	19:56	19:56
	Nyugati pályaudvar	20:00	20:01	20:00	20:01
Vácig közle	Vác	19:03	19:04	19:03	19:04
	Göd	19:16	19:17	19:16	19:17
	Dunakeszi	19:25	19:26	19:25	19:26
	Rákospalota-Újpest	19:33	19:34	19:33	19:34
	Rákosrendező	19:40	19:41	19:40	19:41
	Nyugati pályaudvar	19:47	19:48	19:47	19:48
EC	Szob	19:04	19:05	19:04	19:05
	Nagymaros	19:14	19:14	19:14	19:14
	Verőce	19:21	19:21	19:21	19:21
	Vác	19:29	19:30	19:29	19:30
	Göd	19:38	19:38	19:38	19:38
	Dunakeszi	19:43	19:43	19:43	19:43
	Rákospalota-Újpest	19:47	19:47	19:47	19:47

3.4. A modell gyengeségei

Mindenképpen hátránynak minősül az, hogy a vonatokat nem lehet megkülönböztetni a célfüggvény szempontjából. Vagyis nincs beépítve, hogy az egyik vonat késése sokkal rosszabb, mint a másiké. A példákban is ha csak egy ütközés van, akkor mindegy, hogy a személyvonat jön hamarabb, vagy a nemzetközi később.

A magyar vasúti közlekedésben biztosítóberendezéseket használnak, ami szakaszokra bontja a pályát. Emiatt tehát megtörténhet az is, hogy egy berendezés a pálya közepén állít meg egy vonatot, és nem engedi tovább, amíg a következő szakasz ki nem ürül. A magyar vasútra alkalmazott programba ezt mindenképpen érdemes lenne beletenni. Ezt viszont forrás hiányában nem tudtam megtenni (tehát hogy hogyan működik ez pontosan, és hol vannak a biztosítóberendezések)

Nem szerepelnek a modellben tehervonatok. Ha egy vállalat egy tehervonatot akar üzemeltetni, olyankor egy időintervallumra kibérelnek egy pályaszakaszt. Ha késés miatt változik a menetrend, olyankor az utasforgalom előnyt élvez, de csak az azonnali menetrend-újraoptimalizálás esetén. Természetesen hosszútávú menetrendet nem lehet úgy tervezni, hogy ütközzön a teherszállítással. Valójában ezt be lehet tenni a modellbe egy-egy véletlenszerűen elhelyezett vonat formájában. Fontos még, hogy a tehervonat nem olyan nagy baj, ha perceket késik, ehhez tehát mindenképpen szükség lenne az első bekezdésben említett differenciáló profitfüggvényre.

A modell aciklikusságából (lásd 1.1.2) fakadóan minden javításnál a program elrontja a menetrend ciklikusságát. Ezen úgy lehetne javítani, hogy alapvetően máshogy oldjuk meg a feladatot: az egyes vonattípusokat együtt lehet módosítani, a ciklikusságukat megtartva.

Összefoglalás

Dolgozatom célja kettős. Egyrészt szerettem volna nagy vonalakban egy képet adni a vasútimenetrend-tervezés során felmerülő matematikai problémákról. Felvázoltam, hogy milyen részproblémákra lehet bontani a menetrendtervezést, és hogy melyik során mit tekinthetünk bemenetnek, mire optimalizálunk és mi lesz a kimenet. A dolgozat keretei között ezt nem volt lehetőség mélyebben kifejteni, csak egy átfogó képet szerettem volna adni.

Másrészt pontosan ismertettem a hosszútávú menetrendtervezés egy lehetséges modelljét. Egy szakirodalomban már kifejtett cikkből indultam ki, és azt tovább bővítettem, majd részletekbe menően kifejtettem. Így a modell már megfelelően részletgazdag volt ahhoz, hogy leprogramozhassam. Ezt C++ programnyelven tettem meg, amihez a LEMON csomagot használtam, és FICO Xpress 7.5 optimalizáló programmal oldottam meg a leírt problémát.

A felépített modell tehát egy nem megengedett menetrendhez keresi meg a hozzá legközelebb lévő megengedettet. Működő menetrenden ezt úgy lehet alkalmazni, hogyha a menetrendet "elrontjuk". Így tehát a Budapest–Szob, 70-es számú vonalon lévő menetrendben végeztem apró módosításokat úgy, hogy azok reális, elképzelhető eseményeket tükrözzenek. Az így kapott ideális menetrendeket aztán a programmal kijavítottam, hogy megengedett megoldást kapjunk.

Irodalomjegyzék

[1] Barnhart, C. & Laporte, G. (2007): Handbooks in operations research and management science. Chapter 3, pp. 129-187 3

[2] Caprara, A., Fischetti, M., & Toth, P. (2002): Modeling and solving the train timetabling problem. *Operations Research,*. Vol. 50, No. 5 (Sep. - Oct., 2002), pp. 851-861 1, 9, 10, 22

[3] Magyar Állami Vasutak: 2. sz. módosítás a 2013-2014. évi menetrendhez `http://www.mav-start.hu/res/2013-2014.evi_menetrend_-_2.potlek_mod.pdf` 27

[4] Maróti, G. (2006): Operations research models for railway rolling stock planning. *NUR 919, Phd thesis.* pp. 11-32. 3

[5] Maróti, G. (2014): Vasúti üzemzavarok kezelése: ütemezzük a kocsikat utasbarát módon!. *Optimalizálás szeminárium előadás, Budapesti Műszaki Egyetem, Differenciálegyenletek Tanszék* `//www.math.bme.hu/diffe/szeminarium/opt_2014_01/maroti20140320_ea.pdf` 5

Melléklet

A szakdolgozathoz CD-melléklet tartozik. Ezen található meg a program forráskódja, amivel a különböző ideális menetrendekből megengedett menetrendet csináltam.