Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Automatizálási és Alkalmazott Informatikai Tanszék

Erdős Szilvia

ZÁRÓVIZSGABEOSZTÁS KÉSZÍTÉSE LINEÁRIS PROGRAMOZÁSSAL

KONZULENS

Dr. Kővári Bence

BUDAPEST, 2019

Tartalomjegyzék

Összefoglaló

Az automatikus beosztástervezés évtizedek óta kutatott téma az irodalomban. Mivel a vizsgálandó állapottérre kisebb bementi változások is exponenciális hatással vannak, elsősorban heurisztikus és mesterséges intelligencia alapú módszerek hoztak sikereket.

A záróvizsga beosztások készítése a beosztástervezési feladat egy speciális részfeladata, ahol különleges követelmények (egy hallgatót pontosan egyszer kell beosztani, beosztott hallgatók és vizsgáztatók kapcsolatára vonatkozó kényszerek stb.) korlátozzák az állapotteret.

Előző évi TDK dolgozatomban a probléma automatizálását vizsgáltam egy genetikus algoritmus alapú és egy heurisztikus megközelítéssel. Algoritmusomat egy valós teszthalmazon, a Budapesti Műszaki és Gazdaságtudományi Egyetem (BME) 100 alapképzéses hallgatójának záróvizsga beosztásának elkészítésével teszteltem. Mivel egyik sem hozott teljesen kielégítő eredményt, kidolgoztam a probléma lineáris programozás alapú megoldását, mely minden korábbinál hatékonyabb és igazságosabb eredményt ad a BME adatbázison.

Eredményem jól mutatja, hogy erre a komplexitású feladatra lehetséges elfogadható megoldást adni, mely minden szigorú követelményt teljesít. A nemzetközi irodalmat áttekintve megoldásom egyedi módon átfogó választ ad a záróvizsgabeosztások tervezésének több kérdésére is.

Abstract

The automatic generation of schedules has been in focus of researchers for decades. Since small changes in the input have exponential impact on the tested state space, methods based on heuristics and artificial intelligence are the most successful.

The academic final exam scheduling is a special subtask of schedule generation, where special requirements (every student must be scheduled only once, constraints for relationship of students and examiners etc) reduce the state space.

In my dissertation from last year I analysed the opportunities of automatic generation of the final exam scheduling with genetic algorithm and heuristic approach. I evaluated the algorithms on a real test set of 100 students from one of the Bachelor courses of Budapest University of Technology and Economics (BME). As none of them gave completely acceptable solution, I elaborated a new approach based on linear programming, which gives better and fairer solution on BME database than the earlier algorithms.

My results show that for this complexity there is an acceptable solution, which fulfils every hard requirement. As I reviewed the international literature it seems like my solution gives uniquely a general answer for many problems of the final exam scheduling.

1 Bevezetés

Beosztástervezési feladatokkal először a BSc szakdolgozatomban [1] foglalkoztam, amikor is egy személyes indíttatású projekten kóstoltam bele a témába, ugyanis egy korábbi feladatom automatizálását valósítottam meg. Ennek kapcsán 200 hallgató beosztását oldottam meg egy egyhetes programsorozatra. Ezt a beosztást magyar módszer segítségével készítettem el, ami egy bővítésre szoruló, de a gyakorlatban már jól használható megoldást adott.

A mesterképzés során jobban beleástam magam a beosztástervezés témájába és a kutatásaim alatt megismertem a problémakör különböző területeit. A beosztástervezésnek egy speciális esetét, a záróvizsgabeosztás készítését választottam új kihívásnak, hiszen ez egy sokkal komplexebb probléma, mint a szakdolgozatom során tárgyalt beosztáskészítés. Arra a következtetésre jutottam, hogy ez a téma jelenleg az irodalomban tárgyalt beosztásoknál is bonyolultabb, ott ugyanis más komplexitású feladatokra próbálnak minél jobb algoritmusokat készíteni, míg a záróvizsga beosztás egy különösen összetett feladatnak minősült.

A záróvizsgák szóbeli beosztása legtöbbször egy manuálisan készült folyamat, melynek során cserélgetések, újra tervezések és többszöri újrakezdések során alakul ki a záróvizsgázók végleges beosztása. Ez a folyamat könnyen eredményezhet emberi hibákat, továbbá emberi szemmel nehéz átlátni azt, hogy minden szükséges feltétel teljesül-e, valamint minden résztvevő számára megfelelő-e az elkészült beosztás.

Ennek a folyamatnak az automatizálását vizsgáltam már korábbi TDK [2] dolgozatomban, amelynek során két különböző megközelítésű megoldásomat mutattam be. Az egyik megoldásom genetikus algoritmus alapú volt, melynek során egyre jobb és jobb beosztások elkészítésével, folyamatos javítások során éri el a végleges beosztást. A másik megközelítésem sorban, lépésről lépésre építi fel a záróvizsga beosztását, ami a heurisztikus alapú algoritmusommal készült.

Szükség volt először a probléma formális leírására, hiszen ez sem állt rendelkezésre. Kidolgoztam a beosztások követelményeit, amelyek között van olyan, amit mindenképpen teljesítenie kell minden esetben a beosztásnak, és vannak olyan előírások, amik nem annyira szigorúak, mindegyik betartása nem is lehetséges, mert több helyen ellentmondanak egymásnak. Ez adja a beosztás bonyolultságát, ugyanis ezek

alapján kell egy lehetőségekhez képesti minél jobb megoldást adni. A követelményekhez pontszámokat rendeltem, mellyel kialakult a pontrendszerem. Ezzel mérhetővé és összehasonlíthatóvá váltak az egyes beosztások. Ez a pontrendszer arányosan leírja, hogy melyik feltételt mennyire fontos teljesíteni.

A genetikus és heurisztikus algoritmusom ugyan teljesített minden szigorú követelményt a záróvizsgabeosztás egy szűkített halmazára, viszont a gyenge követelmények tekintetében nem volt teljesen megfelelő a beosztás. Ennek ellenére kiindulási alapnak természetesen nagyon hasznos megoldásokat nyújtottak.

Mivel egyik algoritmusom sem hozott teljesen kielégítő eredményt, kidolgoztam a probléma egy új megközelítését. Ez egy lineáris programozás alapú megoldás, mely minden korábbinál hatékonyabb és igazságosabb eredményt ad, mely jól mutatja, hogy erre a komplexitású feladatra lehetséges az adott kritériumrendszer mellett minimális büntetősúlyú megoldást adni.

A nemzetközi irodalmat is áttekintettem a hasonló problémakörökre alkotott megoldások terén. Ezek alapján arra jutottam, hogy megoldásom egyedi módon átfogó választ ad a záróvizsgabeosztások tervezésének több kérdésére is.

Dolgozatomban először részletezem a problémakört, bemutatom korábbi eredményeimet: az általam formalizált problémahalmazt, a követelményekből felépülő pontrendszert, a korábbi dolgozatomban elkészült algoritmusaimat.

Ezekután kitérek az új lineáris programozás alapú megoldásomra, azon belül is kifejtem, hogy miért volt szükséges egy új megközelítést alkalmazni, hogyan definiáltam a záróvizsgabeosztás problémáját lineáris programozással, valamint mennyivel hozott jobb eredményt, mint a korábbi algoritmusaim. Ezenkívül kitérek arra is, hogy az irodalomban foglalkozó megoldásokhoz képest miben nyújt többet az újonnan készült algoritmusom.

2 A problémakör

Ebben a fejezetben bemutatom a záróvizsga beosztásának problémáját, annak lényegét, valamint azt, hogy miért is bonyolult ennek megoldása.

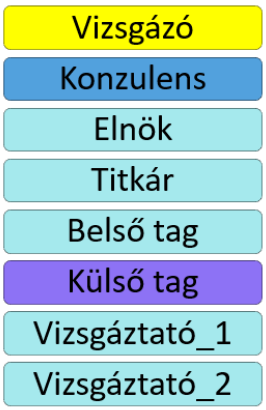
2.1 A záróvizsga beosztás felépítése

Először részletezem a záróvizsgák szóbeli rendszerét, annak működésébe adok bepillantást a következőkben, hogy jobban érthető legyen, miért is különleges ez az egyedi beosztástervezési probléma.

2.1.1 Általános esetű záróvizsgabeosztás

Először is bemutatom a szóbeli záróvizsga beosztás működését, azon belül is a Budapesti Műszaki és Gazdaságtudományi Egyetem (BME) Villamosmérnöki és Informatikai Karára (VIK) koncentrálva. Ennek pontos leírását az egyetemi Tanulmányi és Vizsgaszabályzat [3], valamint annak kari kiegészítése [4] tartalmazza. Az 1. ábra tartalmazza egy záróvizsga számunkra releváns szereplőit.

Természetesen a résztvevők között szerepel egy vizsgázó hallgató, akinek a vizsgája esedékes. Hozzá tartozik a konzulense (témavezetője), aki segítette a szakdolgozat vagy a diplomamunka elkészítésében. Minden vizsga levezetéséhez szükség van egy elnökre, egy titkárra és egy belső tagra, amely szerepköröket csak bizonyos feltételeket teljesítő oktatók tölthetnek be. Látható még a külső tag, aki nem áll jogviszonyban a szervező karral, nem függ a többi szereplőtől, csupán a saját elérhetősége a meghatározó. A vizsgázó képzési szintje, szakja és a választott vizsgatárgyai határozzák meg, hogy ki(k) lehet(nek) vizsgáztató(k) egy vizsgán.



1. ábra - Egy záróvizsga résztvevői

A szabályoknak megfelelő intervallumon belül meghatározzák a vizsgázók száma és az elérhető termek alapján, hogy legalább hány nap szükséges a vizsgák lebonyolításához, valamint, hogy pontosan mely napok lesznek ezek. Az oktatók megadják elérhetőségeiket, valamint az is befolyásoló tényező, hogy melyik oktató melyik képzési szinten, melyik szakon lehet a vizsgáztató bizottság tagja.

A záróvizsgák különlegessége, hogy nem szükséges minden szerepkör betöltéséhez külön-külön oktatónak megjelenni, lehetséges bizonyos feladatok összevonása. Erre egy példa, hogy az elnök lehet egyben a hallgató konzulense és egy vizsgatárgyból vizsgáztató is, a titkár szintén lehet konzulens és vizsgáztató. Viszont fontos, hogy az elnök és a titkár nem összevonható szerepkörök.

2.1.2 Az állapottér szűkítése

A 2.1.1 fejezetben taglalt általános beosztást egy szűkített térre, mégpedig a Villamosmérnöki és Informatikai kar legnagyobb homogén hallgatói csoportjára korlátoztam. Algoritmusaimat így ezen 100 fős BSc-s hallgatói csoporton teszteltem, hiszen a tanszékek többnyire homogén blokkokat hirdetnek, így nem okoz különösebb változtatást a problémakör ilyenfajta lecsökkentése. Ez azt jelenti, hogy nem kellene lényeges változásokat eszközölni annak érdekében, hogy például az MSc-s hallgatókat is figyelembe vegyük, így éltem ezzel az egyszerűsítéssel.

A továbbiakban a külső tagok beosztását is figyelmen kívül hagytam, mivel teljesen függetlenek a többi szereplőtől, nem vonatkozik rájuk semmilyen általunk figyelembe vehető követelmény, csupán rá kell érniük az adott vizsga időpontjában. Ennek megfelelően egy véletlenszerű beosztás is elfogadható a számukra.

A fentieken felül nem dolgoztam párhuzamos vizsgákkal, hogy jobban átlátható legyen a megoldásom. Ennek ellenére természetesen könnyen párhuzamosíthatóvá bővíthetők az algoritmusaim.

2.2 A probléma komplexitása

A beosztástervezés bizonyítottan NP-teljes probléma [5], ez adja a nehézségének egy részét. Ezen felül a záróvizsgabeosztás bonyolultságát két részre bontottam: először is meghatároztam az állapottér nagyságát egy 100 fős hallgatói csoport esetében; másrészt bizonyos feltételek, melyeket teljesíteni kell egy beosztásnak, sokszor ellentmondásosak.

2.2.1 Az állapottér nagysága

Kiszámítottam a problématér nagyságát, egészen pontosan meghatároztam azt, hogy egy 100 fős hallgatói csoport esetében nagyságrendileg mennyi lehetséges beosztás létezik.

Adott 100 vizsgaidőpont és 100 hallgató, őket $100!$ féleképpen lehet beosztani egy-egy vizsgaidőpontra, ha feltételezzük, hogy folyamatosan jönnek a hallgatók, az ebédszüneteken kívül nincs más szünet az adott napban. Hozzájuk egyértelműen meghatározható a konzulensük.

A teszt adatbázisomban 4 olyan oktató van, aki elnöki szerepet tölthet be. Teoretikusan mindegyik vizsgán bármelyik elnök részt vehet, ami 4^{100} lehetőséget jelent a 100 vizsga esetén. A titkárok beosztása hasonló módon felírva a 9^{100} lehetőséget foglal magában, míg a belső tagok esetében ez a szám 10^{100}.

Végül a vizsgáztatókat kell figyelembe venni, amely egy BSc-s záróvizsgázó esetében egy oktatót jelent, mert egy tárgyból kell vizsgáznia. Általgosan egy tárgyból 3 oktató vizsgáztathat, ami további 3^{100} lehetőséget ad.

Mindezt összegezve $100! * 4^{100} * 9^{100} * 10^{100} * 3^{100}$ lehetséges beosztás készíthető, ami megközelítőleg 10^{462} lehetőség, mely egy alsó becslés, hiszen feltételeztem, hogy előre adott a 100 időpont, ami a valóságban szintén egyeztetés tárgya lehet.

2.2.2 Egymásnak ellentmondó feltételek

A záróvizsgabeosztás komplexitását mutatja, hogy bizonyos feltételek, melyeket teljesítenie kell a beosztásnak, könnyen ellentmondanak egymásnak, melyre ebben a fejezetben hozok néhány példát.

A leginkább különleges szempont az oktatók terhelésének eloszlása, hiszen erre nagyon különböző feltételeket lehet megfogalmazni. Az egyenletes eloszlás fogalma már nevében is ellentmondásos, hiszen, ha jobban belegondolunk, egyes oktatóknak biztosan sokkal több konzultát hallgatója van, mint másoknak, ami már önmagában ellehetetleníti a feladat maximális teljesítését. Ezenkívül bizonyos szerepköröket csak bizonyos oktatók tölthetnek be (pl. elnök, titkár), ami már önmagában egyenlőtlenné teszi a beosztást.

Mindezeken felül olyan helyzet is előfordulhat, hogy egy hallgatóhoz szükségszerűen megjelenő oktatók elérhetőségének metszete üres halmazt ad, például a hallgató konzulense nem elérhető egyetlen olyan időpontban sem, mikor a hallgató vizsgatárgyából oktató vizsgáztatók ráérnek.

3 Irodalmi áttekintés

Ebben a fejezetben részletezem a beosztástervezés különböző megközelítéseit, valamint bemutatom ezek eredményeit és korlátait, koncentrálva az egyetemi vizsgabeosztások problémájára.

3.1 Beosztástervezés az irodalomban

A beosztástervezés egy széles körben kutatott téma az irodalomban, hiszen nagyon sok lehetőség rejlik benne. Az életünk sok területén van szükség beosztások elkészítésére: lehet szó akár egy órarend elkészítéséről, napirendi beosztásokról, de a dolgozók beosztása egy munkahelyen is ebbe a témakörbe tartozik.

Évtizedek óta próbálkoznak bizonyos különböző feladatokra minél hatékonyabb, minél gyorsabb és minél jobb eredményt adó megoldásokat keresni. Ezt bizonyítja, hogy évről évre megrendezésre kerül a nemzetközi beosztástervezési verseny [6], valamint már nagy hagyományra visszatekintő, két évente megrendezésre kerülő nemzetközi konferencia, ami az automatikus beosztáskészítés elméleti és gyakorlati szempontjait taglalja [7].

A beosztástervezésnek három különböző alap fajtáját különböztetik meg az irodalomban [8], amelyek mentén a kutatások fő vonala zajlik napjaink során is. Az első a *School timetabling,* mely egy általános vagy középiskolai órarend elkészítését mintázza, beleértve az osztályok, tanárok és termek beosztását. Második típus a *Course timetabling,* mely már komplexebb problémát, mégpedig az egyetemi vagy főiskolai kurzusok beosztását veszi górcső alá. Itt a nehézséget az okozza, hogy nincsenek fix osztályok, csak hallgatói csoportok, melyek között lehetnek átfedések. A harmadik pedig az *Examination timetabling*, vagyis a vizsgarendek beosztása egyetemi vagy főiskolai környezetben, mely probléma áll a legközelebb az általam vizsgált témakörhöz.

Ez a terület azt a problémát hivatott tárgyalni, hogy egy felsőoktatási intézmény vizsgaidőszakának beosztását hogyan készítsük el. Itt több célunk is lehet a beosztás elkészítése során. Lehet például az, hogy azoknak a vizsgáknak, amelyeket vélhetően ugyanazoknak a hallgatóknak kell teljesíteni, ne legyenek egy időpontban. Ezenkívül a hallgatók vizsgáit amennyire csak lehet, „szét kell szórni”, ami azt jelenti, hogy minél több idejük legyen egy-egy vizsga között, ezzel elősegítve a vizsgák sikerességét. A

termekre is lehet figyelni, ha a maximális befogadóképességet nem lépheti túl a vizsgázók száma, de érdekesség, hogy több kisebb létszámú, azonos időtartamú vizsgát meg is lehet tartani akár egy teremben.

3.2 Egyetemi vizsgabeosztás megközelítései

Az egyetemi vizsgák beosztásának megoldása során több különböző szempontot vehetünk figyelembe attól függően, hogy az adott algoritmussal a vizsgák mely tulajdonságaira szeretnénk jobban koncentrálni. Az irodalomban is eltérő megoldásokat találunk, attól függően, hogy ki milyen feltételeket tartotta szükségesnek a saját algoritmusa elkészítése során.

Wijgers, és Hoogeveen 2007-ben írt tanulmányában [9] például az egyetemi vizsgák beosztásának egy fajtáját kutatták, ahol előre meghatározott napokon belülre osztottak be vizsgákat. A hallgatóknak itt több vizsgájuk is lehetett, viszont megadták feltételnek, hogy egy hallgató egy napon csak egy vizsgán vegyen részt. Figyelembe vették az oktatók elérhetőségét, azonban azzal már nem számoltak, hogy mennyire terhelhető egy-egy oktató.

A 2010-ben Al-Yakoob, Sherali és Al-Jazzaf által írt cikkben [10] figyelembe vették a termek elérhetőségét, az egyes kampuszok elhelyezkedésével is számoltak, valamint a vizsgák levezetőinek preferenciát is figyelték az egyetemi vizsgabeosztás elkészítése során. Sőt, még bizonyos nemhez kötött feltételeket is bevezettek. Azonban nem kezelték az egyedi eseteket, ha bizonyos oktatókat csak meghatározott vizsgához lehet hozzárendelni.

Az egyetemi szóbeli vizsgák egy szűkített problémájára adott megoldást Kochaniková és Rudová egy 2013-ban írt cikkében [11]. Itt minden egyes vizsgához egy vizsgáztatót és egy hallgatót rendeltek, miközben figyelték a szereplők elérhetőségeit. Dolgoztak párhuzamos vizsgákkal, és kezelték az ütközéseket. Követelményeiket még meg is különböztették, voltak, melyek teljesülése kötelező volt, míg mások csak opcionálisak. Az oktatók viszont sokszor vizsgáztattak, a köztük lévő terheléseloszlást nem vették figyelembe.

Az egyetemi vizsgabeosztás témakörében írt Bergmann, Fischer és Zurheide 2014-es cikkében [12] például figyeltek a különböző erőforrások, termek, hallgatók egyenletes eloszlására, valamint lehetőséget adtak arra is, hogy az egyes vizsgák

egymással párhuzamosan kerültjenek megtartásra. Azonban Al-Yakoob (*2010*) [10] cikkéhez hasonlóan itt sem számoltak azzal, ha egyes vizsgákhoz például meg kell jelenniük bizonyos oktatóknak, helyette véletlenszerűen lehetett az elérhető oktatókat beosztani.

Wijgers (*2007*) [9] és Bergmann (*2014*) [12] tanulmányaihoz hasonlóan Ivancevic, Knezevic és Lukovic szintén 2014-es cikkében [13] sem vette figyelembe az oktatók túlterheltségét, és elérhetőséget sem kért be az oktatóktól, hanem beosztotta a hallgatókat a különböző vizsgaidőpontokra, és hozzájuk rendelt egy-egy oktatót. Azonban az általuk leírt algoritmus támogatta a párhuzamos vizsgákat, és figyelt a köztük lévő ütközésekre. Emellett a vizsgaidőszakot blokkokra osztották, és az oktatók ezekre az egybefüggő blokkokra lettek beosztva, nem váltogatták egymást folyton.

Aslan, Şîmşek és Karkacier 2017-es írásában [14] nagy hangsúlyt fektetett többek között a terhelések egyenletes eloszlására az oktatók esetében, valamint az esetleges idő- és térbeli ütközéseket is kezelték. Mindezek mellett a felállított követelményeik között nem tettek fontossági különbséget, azokat egyformán fontosnak kezelték. Ezenkívül az oktatók elérhetőségére sem tértek ki.

Látható az előző példáimból is, hogy hiába beszélünk az irodalomban ugyanarról a problémáról (*Examination timetabling*), mégis más-más kutatók más rendszer szerint építik fel azt, és így az elkészített modellük, illetve algoritmusuk csak bizonyos feltételek mentén ad megfelelő beosztást.

4 Korábbi eredményeim

A 2019. évi TDK dolgozatban [2] bemutatott eredményeim jelen dolgozat számára is lényeges pontjait foglalom össze a következő fejezetben, hiszen a jelenlegi kutatásaim alapját képezték az ott elértek. Ennek része volt a problémakör formalizálása, valamint a genetikus algoritmus és a magyar módszer alapú algoritmusaim elkészítése, valamint ezek értékelése.

4.1 A probléma formalizálása

A probléma megoldásához szükség volt először is a záróvizsgabeosztás problémájának formális megfogalmazására. Először meghatároztam egy egységes terminológiát az egyes szereplőkre, hogy egyértelmű legyen, mert egyes szabályzatok, leírások, megszokott elnevezések máshogy hivatkoznak ezekre (ennek pontos részleteit korábbi dolgozatom [2] tartalmazza).

Fontos megemlíteni a *blokk* fogalmát, mert jelen dolgozatomban egy blokknak nevezem egy homogén hallgatói csoport egy délelőttre vagy egy délutánra eső időszakát, vagyis egy fél napot, amikor ugyanazon szakon tanuló hallgatók vizsgáznak.

4.1.1 Követelmények meghatározása és a pontrendszer

A záróvizsgabeosztás szabályokban meghatározott kereteit és a gyakorlatban alkalmazott metodikákat közös nevezőre hozva meghatároztam bizonyos követelményeket, melyeket a beosztásnak teljesíteni kell.

Ezeknek a feltételeknek két típusa van. Az egyik csoportba azokat soroltam, melyek teljesülése feltétlenül szükségszerű a beosztás megvalósíthatósága szempontjából. Ezeket szigorú, azaz *hard* követelményeknek neveztem el.

A másik típusba azok a feltételek tartoznak, melyek teljesülése nem feltétlenül szükségszerű, de minél több teljesül belőlük, annál jobbnak tekinthető a beosztás. Ezeket gyenge, vagy *soft* követelményeknek nevezem.

A meghatározott követelményekhez egy-egy büntető pontszámot rendeltem, mely meghatározza azt, hogy az adott követelmény nem teljesülése valójában mennyire rossz hatással van a teljes beosztásra. Így a követelmények összehasonlíthatókká válnak. A büntető pontok összességéből kialakult pontozási rendszer alapján egyértelműen

összehasonlíthatók az egyes elkészült beosztások, és mérhetők, hogy mennyire jó megoldást adnak.

Minden követelményhez egy *1* és *1000* közé eső számot rendeltem attól függően, hogy mennyire fontos a feltétel teljesülése. Minél nagyobb egy pontszám, annál fontosabb, hogy teljesítve legyen az adott követelmény, amelyhez a pontszám tartozik.

A teljesség igénye nélkül a következőkben felsorolok néhány követelményt a hozzájuk tartozó pontszámokkal együtt. A feltételekről részletesebb leírást a korábbi dolgozatom [2] tartalmazza, továbbá a GitHub oldalamon is megtalálhatóak a követelményeim leírásai [15].

4.1.2 Gyenge követelmények pontszámainak alakulása

Követelmény	**Pontszám**
Elnökök saját konzultált hallgatója nem az elnök saját blokkjában vizsgázik	*2 pont máshol vizsgázó hallgatónként*
Titkárok saját konzultált hallgatója nem a titkár saját blokkjában vizsgázik	*1 pont máshol vizsgázó hallgatónként*
Vizsgáztató nem az elnök, pedig más napon elnök	*1 pont*
Az elnök képzési szintje nem egyezik meg a vizsgázó képzésével	*1 pont*
Az ebédszünet 11:30 előtt kezdődik, illetve 13:40 után végződik	*korábban kezdődik: 40 pont* *később végződik: 40 pont*
Az ebédszünet 40 percnél rövidebb, (az elvárt 60 perc)	*40 perc: 2 pont* *50 perc: 1 pont* *60 perc és több: 0 pont*
A belső tag nem elérhető a vizsga alatt	*5 pont*
A konzulens nem elérhető a vizsga alatt	*5 pont*
Az elnök terhelése a többi elnöktől eltérő	*30-0: az eltérés mértékétől függ*
A titkár terhelése a többi titkárétól eltérő	*30-0: az eltérés mértékétől függ*
A belső tag terhelése a többi tagtól eltérő	*30-0: az eltérés mértékétől függ*

4.1.3 Szigorú követelmények és pontszámaik

Követelmény	Pontszám
Az elnök nem elérhető a vizsga alatt	*1000 pont*
A titkár nem elérhető a vizsga alatt	*1000 pont*
A vizsgáztató nem elérhető a vizsga alatt	*1000 pont*
Az elnök megváltozik a blokkban (változásonként)	*1000 pont*
A titkár megváltozik a blokkban (változásonként)	*1000 pont*
A vizsga megkezdődik 8:00 előtt	*140-0: kezdés időpontjától függ*
A vizsga 18:00 után végződik	*140-0: zárás időpontjától függ*

4.2 Kidolgozott algoritmusok

A kutatásaim során kétféle megközelítést alkalmaztam a záróvizsga beosztásának problémájára. Az első a genetikus algoritmus alapú megoldás volt [16], mely folyamatos tanulás során éri el a kívánt eredményt. A második megoldásom a heurisztikus alapú algoritmus volt, amelyet azért választottam, mert ez áll legközelebb a kézzel készült beosztások menetéhez. Ezeket az elkészült algoritmusaimat mutatom be a következő fejezetben, hiszen ezek adták az alapját a lineáris programozás alapú algoritmusomhoz.

4.2.1 Genetikus algoritmus

A genetikus algoritmus gének kombinációival készíti el a beosztást. A génekből kromoszóma épül fel, melyekből különböző méretű populációt lehet generálni. Először is el kell végezni az inicializálást, ami kezdeti beosztás elkészítését jelenti. A kiértékelés fázisban egy fitness függvényt vizsgálok meg, amely a beosztás jóságát vizsgálja.

Ha még nincs kész a beosztás, akkor egy megadott méretű populációt hozok létre a kromoszómák különböző kombinálásával. Utódok létrehozásához kiválaszthatók tetszőleges egyedek, majd ezeket keresztezéséből és mutációjából eljut az algoritmus egy minél jobb beosztás felé.

4.2.1.1 Az algoritmusom működése

Először is létrehoztam a kezdeti egyedeket, illetve saját mutációkat implementáltam. A legnagyobb kihívást azonban a fitness függvény létrehozása adta, ami alapján értékelni lehetett a beosztásokat.

Egy génnek egy-egy hallgató záróvizsgáját választottam. A kromoszómámat ezekből a génekből építettem fel, így egy kromoszóma egy teljes záróvizsga beosztásnak felel meg. A kezdeni adathalmazt nem teljesen véletlenszerűen választottam meg, hanem az egyes résztvevők beosztásánál figyelembe vettem, hogy ki milyen szerepkörben lehet.

A fitnesz függvény határozza meg az egyes kromoszómákról, hogy mennyire megfelelőek és használhatók, így ennek a metódusnak a felépítése egy igazán nagy kihívást nyújt és meghatározó az algoritmus szempontjából. Az alapjául a 4.1.1. fejezetben bemutatott pontrendszert választottam, hiszen itt valójában azt kell meghatároznom, hogy az egyes beosztások mennyire jók.

Ellenőrzőm a követelmények teljesülését minden beosztásra, és a sérült követelményhez hozzárendelem a pontszámát, melyeket különböző súlyfüggvényekkel vizsgálok. Végül ezeket összegzem, mely egyben kiadja a teljes beosztás fitnesz értékét. A genetikus algoritmus sajátossága, hogy minél nagyobb pontszámú fitnesz érték elérésére törekszik, így a pontszámaim büntetőpontokként foghatók fel, vagyis minél fontosabb követelmény sérül, annál több pontot vonok le.

A fitnesz érték javulása és annak mértéke határozza meg az algoritmus futási idejét, vagyis azt, hogy mennyi generáción át próbálkozik a minél jobb eredmény elérésére. Ha egy bizonyos generációszám eltelte után már nem javul tovább a fitness függvényem, leállítom az algoritmus futását, hiszen nála jobb eredményt nem tud elérni a megadott beállításokkal.

A záróvizsgára specifikus megoldások elérésére saját mutációkat fejlesztettem, melyek során a gének módosításával is érjek el javulást az algoritmusomban, ezáltal a saját követelményeim teljesülései felé mozdítsam el a beosztást.

Annak érdekében, hogy ne lokálisan optimális beosztás készüljön bevezettem egy véletlenszerű mutációt, mely kis valószínűséggel megváltoztat egy-egy gént. További mutációimat különböző, a 4.1.1. fejezetben meghatározott követelmények érdekében alakítottam ki, melyek közül a következőkben felsorolom és értékelem a legfontosabbakat. Az algoritmusom végül a legjobb eredményt akkor érte el, ha az összes saját mutációmat alkalmaztam, mindegyiket kis valószínűségekkel.

Készítettem különböző mutációkat a szigorú feltételek teljesülésére. Ezek közül kiemelkedik az elnökökre és a titkárokra vonatkozók, hiszen hozzájuk kapcsolódik a legtöbb ilyen feltétel. Egyik ilyen mutáció, amelyik azt valósítja meg, hogy ezekben a

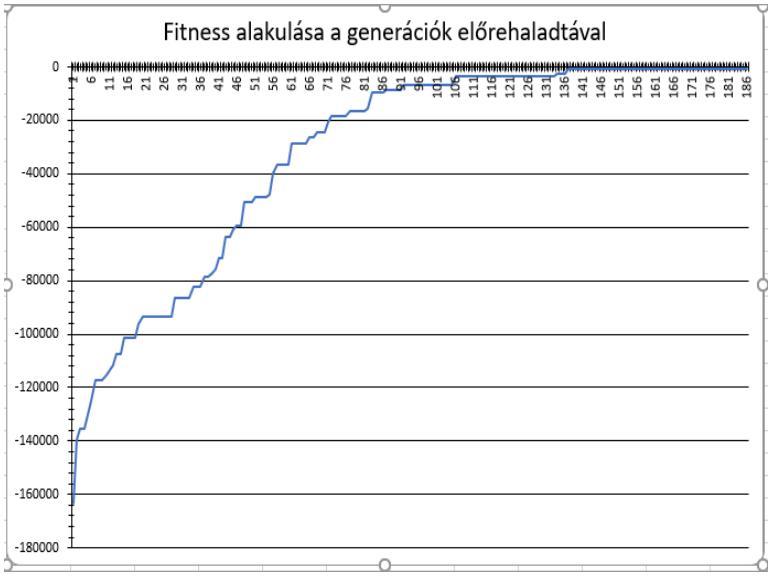
szerepkörökben lévő oktatók egy-egy blokk alatt ne cserélődjenek. Ezt úgy valósítom meg, hogy megvizsgálom, melyik elnök/titkár fordul elő a legtöbbször az adott blokkban, és a többi blokkon belüli időszeletbe is az adott elnököt/titkárt írom be bizonyos valószínűséggel. Az elnökökkel és titkárokkal kapcsolatban további fontos tényező, hogy elérhetőek legyenek, így ennek elérésére is készítettem saját mutációt.

Mindezeken kívül a gyenge követelményeimre is készítettem mutációkat, mert ezeket a beosztást jobbá tevő feltételeket sokszor nem tudná elérni ezek nélkül az algoritmus. Létrehoztam több olyan mutációt is, mely a különböző szerepkörök összevonására irányul: vagyis, ha egy szereplő betölthetné más feladatát is, akkor ne osszon be mást az algoritmus, hanem a már jelen lévőt vegye figyelembe. Erre egy példa, hogy ha a hallgató konzulense lehetne elnök, de nem ő az, akkor kicseréli az adott elnököt bizonyos valószínűséggel a hallgató konzulensére.

Észrevehető, hogy a korábban említett feltételek, és az ezek érdekében végrehajtott mutációk is egymásnak ellent mondhatnak. Az elnököt például több esetben is megváltoztatom: ha nem ugyanaz, mint az adott blokkon belül a többi, vagy nem a hallgató konzulense, pedig lehetne. Ennek érdekében alkalmaztam az egyes mutációkat csak egy kis valószínűséggel.

4.2.1.2 Elkészült beosztás

A beosztásom akkor hozta a legjobb eredményt, amikor az összes mutációm alkalmaztam. A 2. ábra mutatja, hogy a generációk előrehaladtával hogyan alakult ki a megoldás a legjobb elért beosztás esetén. Mindez összesen 643 büntetőpontot jelent, melynek többsége a nem kiegyenlített terhelésekből, valamint a nem összevont szerepekből származnak. Mindezek mellett fontos elmondani, hogy kizárólag enyhe kritériumok nem teljesültek.



2. ábra - Fitness alakulás a legjobb beosztás esetén

4.2.2 Heurisztikus algoritmus

A problémakört egy kézzel való elkészítéséhez közelebb álló módon is automatizáltam. Ez a megközelítés egy teljesen más irányból ad megoldást a problémára. A heurisztikus algoritmust egyébként gyakran használják NP-teljes problémák megoldására [17].

Megoldásomban a nagy problémateret úgy csökkentem, hogy hierarchizálom, majd az egyes szinteken lokálisan optimális döntéseket hozok. Ez azt jelenti, hogy sorrendet állítok fel az egyes szerepkörök között fontossági sorrendben, és ezek sorrendjében készítem el e beosztást. Vagyis nem teljes sorokat, vizsgákat veszek figyelembe, nem egy vizsgához tartozó összes szereplőt osztom le egyidőben, hanem szereplőnként sorban.

Először az elnököket osztom be, hiszen ők adják a legszűkebb keresztmetszetet, hozzájuk köthető a legtöbb fontos kritérium. Hozzájuk közel hasonlóan nehézkes a titkárok beosztása, így őket osztom be másodikként. Ezek után elhelyezem a hallgatókat, majd hozzájuk egyértelmű megfeleltetéssel a konzulenseiket. Ekkor hátravan még a vizsgáztatók és a belső tag beosztása, őket ebben a sorrendben osztom be, ugyanis a belső tag sok esetben egy „kiegyenlítő szerepet” tölt be, vagyis, ha valaki már be lett osztva, aki lehet belső tag, akkor nem kell további oktatót beosztani.

4.2.2.1 Magyar módszer

A heurisztikus algoritmusom megoldása során felhasználtam a magyar módszert, melynek segítségével páros gráfokban lehet maximális elemszámú párosítást keresni polinom időben [18].

Esetemben a szereplőket kell az adott záróvizsgával összepárosítani. Ha páros gráfként értelmezzük a problémát, akkor egyik oldalon vannak az emberek (oktatók vagy hallgatók), másikon a záróvizsgák (azok időpontjai) szerepelnek, ezeket között húzódnak élek, különböző súlyokkal. Ezeket a súlyokat kell nekem meghatározni, hogy a lehető legnagyobb összsúlyú párosítás jöjjön létre. Az összepárosítás után a megmaradt élek fogják jelezni, hogy melyik szereplőhöz melyik záróvizsgaidőpont jutott.

4.2.2.2 Az algoritmusom működése

Az elnökök és titkárok beosztását egyből egész blokkokra készítem el, és nem záróvizsgánként sorban. A páros gráfban a súlyozás az alapján történik, hogy az adott

oktató a megadott blokkban elérhető-e vagy sem. Az egyik dimenziómon találhatóak a záróvizsga blokkjai, a másikon pedig az elnökök/titkárok, mégpedig mindegyik elnököt/titkárt többszörösen, de ugyanannyiszor veszek fel, ezzel elősegítve azt a kritériumot, hogy hasonló legyen az elnökök terheléseloszlása. Amiatt kell mindegyikből többet felvennem, mert több blokk van, mint amennyi ilyen szerepkörben lévő oktató, tehát egy-egy oktató biztosan többször lesz beosztva.

A hallgatók beosztásánál több szempontot kellett figyelembe vennem. Először is fontos volt a már elkészült beosztás részletet szempontnak venni, mégpedig azért, mert több vizsgázóra vonatkozó követelményem kapcsolódik az elnökökhöz és a titkárokhoz. Ezek közül az egyik, hogy a hallgató konzulense amennyiben lehet elnök vagy titkár, lehetőleg ő legyen az.

A hallgató beosztásánál figyelembe kell venni a későbbi oktatói szerepkörök beosztását; például a hallgatóhoz tartozó konzulens legyen elérhető akkor, amikorra a hallgatót beosztom. Továbbá a vizsgáztatók elérhetősége is meghatározó, ugyanis fontos, hogy legyen elérhető vizsgáztató, valamint, ha lehet egyben elnök is, akkor ez az összevonás is teljesüljön.

A páros gráfom dimenzióiban a hallgatók, valamint a vizsgák szerepelnek, köztük a súlyok pedig azt határozzák meg, mennyire lenne jó egy adott vizsgára beosztani a hallgatót. Ezt a korábban említett feltételek figyelembevételével állapítom meg. Így ki is alakul a vizsgázók beosztása, akikhez egyből hozzá is rendelem a konzulenseiket.

A vizsgáztatók beosztását szintén a magyar módszer egy másfajta felhasználásával készítettem el. A beosztást itt tantárgyanként sorban egymás után készítettem el. Összegyűjtöttem, hogy az adott tárgyhoz melyik hallgatók tartoznak, majd a tárgyhoz tartozó vizsgáztatók egyenletes terhelését figyelembe véve alakítottam ki a párosítandó elemeket. A súlyozás alapja jelen esetben vizsgáztatók elérhetősége, valamint a lehetséges szerepösszevonások megvalósulása.

A belső tagok beosztását a valóságos beosztás elkészítésének reprezentálásával készítettem el, mégpedig úgy, hogy ha van már olyan oktató, aki betöltheti a belső tag szerepét, akkor nem rendelünk hozzá új embert. A többi vizsgánál szerepet kapott a belső tagok terheléskiegyenlítő szerepe. Mindez azt jelenti, hogy kiszámolom minden lehetséges tagra a korábbi beosztásainak számát. Ennek mennyiségével súlyozva,

valamint az adott vizsgaidőpontbeli ráéréseik alapján szintén magyar módszer segítségével összepárosítottam a megfelelő tagokat a hiányzó vizsgákra.

4.2.2.3 Elkészült beosztás

Ezen lépések végeztével minden szereplő helyét betöltötte ekkor valaki a beosztásban, és a szerepeknek megfelelő, pozitív súlyozású beosztás készült. A korábbi követelményeim alapján 450 büntetőpontot ért el, melyek a terhelésekből adódtak.

Az 1. táblázatban látható egyes szerepkörök terheléseinek eloszlása. Az elnököknél és titkároknál elfogadható beosztás készült, az eltéréseket nagyrészt a ráérések változatossága adja. A belső tagok terheléskülönbsége változatosabb képet ad. Nem teljesen egyenletes, mert a kiugróan magas számokkal rendelkező oktatóknak nagyon sok konzultált hallgatójuk van, illetve van olyan vizsgatárgy, melyből kevés oktató vizsgáztathat, melyből szintén adódik terheléskülönbség.

Presidents	Nr of exams	Secretaries	Nr of exams	Members	Nr of exams
Charaf Hassan	30	Braun Patrik János	15	Asztalos Márk	11
Forstner Bertalan	25	Budai Ádám	15	Blázovics László	3
Lengyel László	25	Fekete Tamás	15	Csorba Kristóf	5
Vajk István	20	Gazdi László	15	Dudás Ákos	14
		Hideg Attila	10	Ekler Péter	5
		Jánoky László Viktor	10	Iváncsy Szabolcs	6
		Pomázi Krisztián	10	Kovács Tibor	6
		Somogyi Ferenc Attila	5	Kővári Bence András	39
		Tömösközi Máté	5	Mezei Gergely	7
				Recski Gábor András	4

1. táblázat - Heurisztikus által készült beosztások néhány terheléseloszlása

4.2.3 Eredményeim

Algoritmusaimat a BME 100 fős homogén hallgatói csoportján teszteltem. A 4.2.1.2 fejezetben bemutattam a genetikus algoritmusom által generált beosztásom legjobb eredményeit, valamint a 4.2.2.3 fejezetben a heurisztikus megoldásom által generált beosztást. Mindegyik beosztásom kizárólag enyhe követelményeket sértett meg.

Az elkészült algoritmusaim teljesítették minden fontos általam elvárt követelményt. Mindkettő hasonló eredményeket ért el, legnagyobb gyengesége mindegyiknek a megfelelő terheléseloszlás elérése volt. A heurisztikus jelen követelményeimet jobban teljesítette, viszont nem számottevő módon.

5 Lineáris programozás alapú megközelítés

Az alábbi fejezetben fogom bemutatni az új megközelítés szükségességének okait, valamint kifejtem a lineáris programozás alapú megoldásomat. Továbbá értékelem az azáltal elért eredményeket.

5.1 Genetikus és heurisztikus algoritmus hibái, hiányosságai

A 4.2. fejezetben bemutatott algoritmusaim ugyan minden szigorú követelményt teljesítettek, és emiatt használhatónak mondhatók a szűkített problémakörre, mégsem adnak teljesen kielégítő megoldást. Továbbfejleszthetőség tekintetében nem teljesít mindegyik túl jól, valamint a gyenge követelményeket sem teljesítik maradéktalanul. Ezek a sérült feltételek elsősorban az egyes szereplők egyenletes terhelésére irányultak, valamint bizonyos szerepkörök összevonása sem teljesült (pl elnök hallgatója akkor vizsgázzon, amikor az oktató éppen elnök is). Ezek viszont a valós beosztás elkészülése során nem elhanyagolhatók, hiszen az oktatók egyéni beosztásánál nagy szerepet játszanak. Ezenkívül egyéb korlátjai is vannak az egyes algoritmusoknak, melyeket ebben a fejezetben fejtek ki.

5.1.1 A genetikus algoritmus korlátjai

A genetikus algoritmusom, melyet a 4.2.1. fejezetben részleteztem, természetéből adódóan a kisebb változtatások követésére nem túl rugalmas [19]. Ez azt jelenti, hogy egy-egy paraméter vagy mutációs valószínűség megváltoztatása ugyan okoz változást az eredményben, viszont a generációk előrehaladásának természetes kiszámíthatatlansága nem feltétlenül determinisztikus, ezért nem lehet egyértelműen megállapítani, hogy az adott kismértékű változás az eredményben a saját paramétereink változtatásának az eredménye, vagy csupán az algoritmus most ebben az irányban ment el a véletlen valószínűségek következtében. Ezáltal az apróbb javítások, a paraméterek finomhangolása az optimalizálás érdekében megnehezedik, és ha ezen változtatások mentén szeretnénk még használhatóbb algoritmust létrehozni, akkor egy rendkívül időigényes folyamattal találjuk szemben magunkat.

Ezeken túl a futási idővel is gondjaink akadhatnak, ugyanis jelenlegi minőségében is több, mint 20 percig fut az algoritmus (egy 2,4 GHz-es i7-es processzorral rendelkező gépen, 8 szálon futtatva, 8 GB RAM mellett), ami nem tűnik soknak, viszont messze

elmarad már a heurisztikus algoritmus alig 1,5 perces futási idejétől. Ez az idő viszont a létrehozott populációk számának növekedésével közel exponenciálisan növekszik [20], valamint az algoritmus bővítésével, további mutációk behozatalával még sokkal tovább növekedhet.

5.1.2 A heurisztikus algoritmus korlátjai

A 4.2.2. fejezetben bemutatott magyar módszer alapú heurisztikus algoritmusom is rendelkezik különböző korlátozó tényezőkkel. A legnagyobb probléma az algoritmus lineáris előrehaladása, mely során élni kell azzal a feltételezéssel, hogy az egyes szerepkörök adott sorrendben való beosztása megfelelő megoldást ad. Ezt csupán a kézi eljárás bizonyítja, hiszen ez a megoldás annak a mintájára készült.

A lineáris előrehaladás eredménye, hogy bizonyos szerepkörök beosztásánál előre kell gondolkodni, vagyis a még be nem osztott szerepkörökben lévő résztvevők egyes tulajdonságait is figyelembe kell venni, mert az egyes szerepelők beosztása hatással van más emberek beosztására. Erre egy példa, hogy mikor a hallgatók beosztása készül, már figyelembe kell venni, hogy olyan időpontra legyen beosztva, amikor van elérhető vizsgáztató a hallgató vizsgatárgyából. Fordított sorrendben pedig szintén nem egyszerűbb a beosztás, hiszen a vizsgatárgyakat és így a hozzájuk kapcsolódó vizsgáztatókat pedig a hallgatók határozzák meg. Ennek értelmében a követelmények különböző összefonódásai, oda-vissza hatásai az egyes szerepkörökre nagyban megnehezítik a beosztás bővíthetőségét.

Ebből az egymásra határból ered a terhelések kiegyenlítésének problémája is, hiszen, ha egyes szereplőket már előre beosztottunk, annak nagy hatása lesz arra, hogy az adott oktató későbbi (még nem beosztott) szerepeiben mennyire legyen terhelve, viszont a lineáris haladás következtében erre nem tud az algoritmus figyelni.

5.2 Lineáris programozás alapú algoritmus

A korábbi megoldások hiányosságaiból adódóan egy új megközelítésű algoritmust dolgoztam ki, melyet lineáris programozással oldottam meg. Ilyen megközelítéssel próbálkoztak már korábban is különböző beosztástervezési feladatok megoldására [12], melyek a saját követelményrendszerük szerint jó eredményeket produkáltak, ezért választottam ezt a megközelítést. Azonban az elkészült beosztásom az irodalomban taglalt hasonló témájú beosztásoknál (melyeket a 3.2. fejezetben taglalok)

átfogóbb megoldást ad a vizsgabeosztások problémájára, melynek részletes összehasonlítását a 6. fejezetben végzem el.

5.2.1 Lineáris programozás

A lineáris programozás (LP) alapjait George B. Dantzig alkotta meg [21] még a XX. század közepén. A módszert azóta is használják különböző optimalizációs feladatokban, döntési folyamatok, gazdasági problémák optimalizálására. Az algoritmus lényege, hogy egy adott döntési feladathoz matematikai modellt állítunk fel, melyben általában lineáris formulák, egyenlőségek és egyenlőtlenségek szerepelnek. Majd az egyenletrendszerek megoldási algoritmusára alapuló eljárásokkal kiszámoljuk a szóban forgó döntéshez tartozó legjobb változatokat.

A lineáris programozás alapfeladatát a következő standard formában fogalmazhatjuk meg: adottak $x_1, x_2, \cdots, x_n$ döntési változók, melyek értékeit keressük úgy, hogy $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ célfüggvény felvegye szélsőértékét (minimumát vagy maximumát), miközben teljesülnek a következő különböző korlátozó feltételek:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &\leq b_n \\ x_1, x_2, \cdots, x_n &\geq 0 \end{aligned}$$

Az adott paraméterekre igaz, hogy $a_{ij}, b_i, c_i \in \mathbb{R} \quad \forall i, j \in \{1, 2, \dots, n\}$.

A lineáris programozás alapjainak megalkotásakor Dantzig kidolgozott egy módszert [22] is az ilyen feladatok megoldására, mely a szimplex módszer nevet kapta. Az algoritmus műveleti ideje akár exponenciális is lehet, így azóta is foglalkozik az irodalom a szimplex módszer lehető legmegfelelőbb implementálásán. Napjaink egyik leghatékonyabb lineáris optimalizációs feladatok megoldója [23] a Gurobi [24], így én is ezt alkalmaztam.

5.2.2 Modellem felépítése a záróvizsgabeosztásra

A záróvizsgabeosztás problémájának lineáris programozás alapú modelljének felépítését mutatom be a következő fejezetekben. Először is bemutatom, hogy milyen jelöléseket használtam a záróvizsgabeosztás pontos leképezésére, melyet az 5.2.3. fejezetben mutatok be.

Ahogy az 5.2.1. fejezetben az általános problémát bemutattam, a saját modellemben is bevezetek döntési változókat, melyek értékét keresni fogja az algoritmusom. Ezek esetemben az egyes szereplők adott időpontban való beosztását, vagy nem beosztását jelentik, részletesen az 5.2.4. fejezetben írok róluk.

A döntési változókra építkezve létre kellett hoznom egy célfüggvényt, melynek alapjául bizonyos korábban megfogalmazott követelményeket vettem (4.1.2 és 4.1.3), hiszen ez határozza meg, hogy pontosan mit is szeretnénk elérni, mit szeretnénk optimalizálni az algoritmus során. Jellemzően ezek bizonyos gyenge követelmények teljesülésére irányuló törekvések, hiszen, ha elfogadhatónak mondható a beosztás (vagyis minden szigorú követelmény teljesült), akkor már csak a gyenge feltételek határozzák meg, hogy az elfogadható megoldások közül mennyire is jó az adott elkészült beosztás. A célfüggvényem felépítését az 5.2.5. fejezetben részletesen is bemutatom.

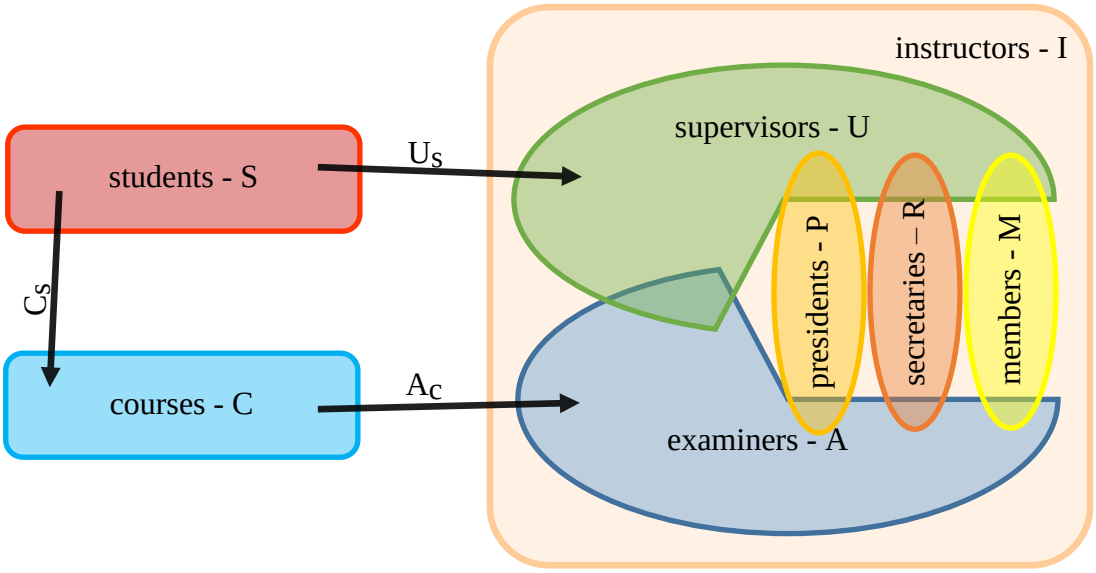
A szigorú feltételeimet, valamint néhány gyenge követelményt is a különböző modellemet korlátozó feltételekkel határozok meg, valamint itt található néhány olyan feltétel is, melyre ugyan nem készült külön követelmény, viszont elengedhetetlen a beosztás elfogadhatósága szempontjából. Ilyen például, hogy vegyen részt hallgató minden vizsgám, ami intuitívan is következik, hiszen hallgató nélkül nincs értelme vizsgát tartani, viszont a beosztás megvalósulása szempontjából ezeket a triviális feltételeket is bele kellett építenem a modellembe, melyek sokszor nem is voltak olyan egyértelműek vagy könnyen meghatározhatók. Minden korlátozó feltételemet az 5.2.6. fejezetben határoztam meg.

5.2.3 Modellben használt halmazok, konstansok és jelöléseik

A modell felépítésében különböző jelöléseket használok a záróvizsgabeosztás egyes erőforrásaira, szereplőire, melyeket ebben a fejezetben részletesen ki is fejtek a jobb érthetőség érdekében.

A vizsgázási időszak időszeletekre (*timeslots*) van felosztva, ahol minden időszeletben pontosan egy hallgató vizsgázik. Az összes időszelet halmazát T jelöli, ahol egy időszelet $t \in T$. A blokkokat (*blocks*) B jelöli, ahol egy blokk $b \in B$. A hallgatók (*students*) halmazát S jelöli.

A további szereplők bemutatásához a könnyebb megértés érdekében készítettem egy halmazábrát, melyet a 3. ábra mutat be.



3. ábra - Modellemben használt jelölések és kapcsolataik

Az oktatók (*instructors*) egy külön halmazt alkotnak, akiket összefoglalóan az I halmazba sorolok. Az oktatók különböző szerepköreit is jelölnöm kell, melyet az alábbi módon tettem meg:

- A konzulensek (*supervisors*) halmazát $U \subseteq I$ jelöli. Minden $s \in S$ -hez létezik egy $u \in U$, tehát $\{s \in S\} \rightarrow \{u \in U\}$. Egy konzulens viszont több hallgatóhoz is tartozhat, mégpedig: $\{u \in U\} \rightarrow \{s_1, s_2, \dots, s_n \mid \forall s_i \in S\}$. Ezenkívül u_s jelöli s hallgató konzulensét.
- Az elnökök (*presidents*) halmazát $P \subseteq I$ -vel jelöltem, akik azok az oktatók, akik a szabályoknak megfelelően betölthetik az elnök szerepét.
- A titkárok (*secretaries*) halmaza $R \subseteq I$, akik azok az oktatók, akik betölthetik a titkár szerepét.
- Végül a belső tagok (*members*) halmazát $M \subseteq I$ jelöli.

Egy további erőforrás a záróvizsgabeosztás során maga a vizsgatárgy, melyből minden hallgató esetemben egyet választ. A tantárgyak (*courses*) halmazát C-vel jelölöm. Minden hallgató választ egy tárgyat, melyet c_s jelöl. Minden tárgyhoz előre meghatározható, hogy mely oktatók vizsgáztathatnak belőle. Egy adott c vizsgatárgyhoz tartozó vizsgáztatók (*examiners*) halmaza $A_c \subseteq I$, vagyis ennek megfelelően a hozzárendelés az alábbi módon alakul: $\{c \in C\} \rightarrow \{a_1, a_2, \dots, a_n \mid \forall a_i \in A_c\}$.

Egy vizsgát (*exam*) e-vel jelöltem, a teljes beosztás pedig E. Így egy vizsga az alábbi módon épül fel, mely tartalmazza a szereplőket és a vizsga idejét:

$$e \in \{\mathrm{t}, s, u, p, r, m, a \mid \mathrm{t} \in \mathrm{T}, s \in S, u \in U, p \in P, r \in R, m \in M, a \in \mathrm{A}\}$$

A záróvizsgabeosztás feltételei során fontos ismeret az, hogy egy adott szerepkörben lévő oktató ideális esetben hány időpontban kerül beosztásra. Ez egy konstans érték, melyek egyszerűen úgy határozok meg, hogy figyelembe veszem az adott szerepkört betölteni képes oktatók számát és a vizsgaidőpontok számát, és egyenlő eloszlást feltételezek ideálisként. Ezeket a konstans ideális értékeket elnökök esetén D_p, titkároknál D_r, míg a belső tagoknál D_m jelöli.

Ezenkívül meghatároztam, hogy az egyes szerepkörökben mekkora értékek között mozogjon a beosztások száma, ha nem sikerül elérni az ideális értéket. Ezeket az értékeket elnökök esetén D_p^- (minimum) és D_p^+ (maximum) jelöli. Titkárok beosztásainak a száma D_r^- és D_r^+ közé kell hogy essen, míg a belső tag szerepét betölthető oktatók beosztásainak minimumát D_m^-, maximumát pedig D_m^+ jelöli a modellemben.

5.2.4 Döntési változók

A lineáris programozás során a legelső kihívás a megfelelő döntési változók kiválasztása. Ezek lehetnek valamilyen skálán mozgó értékek, bináris változók, vagy akár egész számok is, ahol a változó határértékeit is megadhatjuk. Fontos a döntési változók meghatározása során, hogy azok az adott problémakörre jól illeszkedjenek és a lehetőleg minél kevesebb változó leírja a teljes modellt.

A modellemben két alapvető döntési változó halmazt hoztam létre, amely illeszkedik a záróvizsgabeosztás kétféle emberekből álló csoportjához. Az egyik ilyen az oktatókhoz kapcsolódik, mégpedig úgy, hogy egy kétdimenziós mátrixként foghatjuk fel, ahol az egyik dimenzióban található minden oktató ($i \in I$), a másikban pedig minden időszelet ($t \in T$), a mátrix egy példájának részletét mutatja be a 4. ábra. A benne található értékek binárisak, vagyis 0 vagy 1 szerepel a mátrix minden helyén. Ez azt jelöli, hogy minden egyes oktatónál minden időszeletben meg van határozva, hogy be van-e osztva, vagy nincs. Értelemszerűen, ha 0 szerepel egy $i_k \in I$ oktatónál a $t_j \in T$ időszeletben, akkor nem került beosztásra ez az oktató ebben az időpontban, míg, ha 1-es az érték, akkor be lett osztva.

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
Ács Judit	1	1	1	0	0	0	0	1	1	1
Albert István	0	0	0	0	1	1	1	1	1	1
Asztalos Márk	0	0	0	0	0	1	1	1	1	1
Benedek Zoltán	0	1	1	1	1	1	0	0	0	0
Blázovics László	1	1	1	1	1	0	0	0	0	0
Charaf Hassan	0	0	0	0	0	1	1	1	1	1
Cserkúti Péter	0	0	0	1	1	0	0	0	0	0
Csorba Kristóf	1	0	0	1	0	1	1	0	0	0
Dudás Ákos	0	0	0	0	0	0	1	1	1	1
Dunaev Dmitriy	1	1	1	0	0	0	0	0	0	0
Ekler Péter	1	1	1	1	1	0	0	0	0	0

4. ábra - Oktatók bináris változóinak reprezentálása

Az oktatók bináris változóját $x_{i,t}$-vel jelölöm, ahol i az adott oktatót, t az adott időszeletet jelenti. Formálisan ezt az alábbi módon fogalmaztam meg a modellemben:

$$x_{i,t} \in \{0,1\} \qquad \forall i \in I, \forall t \in T$$

Fontos megjegyezni, hogy az összes oktatót egyben kezelem, nem különítem el ennél a döntési változónál a különböző szerepkörökbe léphető oktatókat. Erre azért volt szükség, mert egy-egy ember sokféle szerepkört betölthet, és nagymértékű redundanciát okozna, ha minden szerepnél külön kellene adminisztrálni az oktatókat, ami nemcsak az átláthatóságot, hanem az algoritmus hatékonyságát és kezelhetőségét is megnehezítené.

Mindemellett az egyes korlátozó feltételek meghatározásakor szükség van egyes szerepkörökhöz rendelt változókat figyelembe venni, ekkor viszont fontos látni, hogy nem új változókat hozok létre, hanem kiválasztom az adott éppen szóban forgó halmazba eső oktatók $x_{i,t}$ változóit. Erre egy példa, ha az elnök szerepkörbe léphető oktatókat veszem figyelembe, ezt $x_{p,t}$-vel jelölöm, ami valójában azokat az $x_{i,t}$ döntési változókat jelenti, ahol $i \in P$, vagyis az oktató az elnökök csoportjába tartozik.

Az oktatókhoz hasonlóan egy kétdimenziós mátrixként lehet elképzelni a hallgatók beosztására készített döntési változóimat. Itt az egyik dimenzióban a hallgatók szerepelnek ($s \in S$), a másikban pedig szintén a vizsgaidőszak időszeletei ($t \in T$). Szintén bináris változókat használtam, ha 0 szerepel az adott s hallgatónál t időpontban, akkor nincs beosztva, vagyis nem akkor vizsgázik, ha pedig 1-es szám van, az azt jelenti, hogy akkor vizsgázik. Ezt formálisan az alábbi módon írtam le:

$$x_{s,t} \in \{0,1\} \qquad \forall s \in S, \forall t \in T$$

Ez eddig megfogalmazott döntési változók azok, melyeket optimális módon meg kell határoznia az algoritmusomnak, ezek határozzák meg a tényleges beosztást. Ezeken kívül viszont be kellett vezetnem néhány számított döntési változót is a modellembe, melyek valójában csak segítik kialakítani a végleges beosztást. Ezek jellemzően valamilyen követelmény megfelelő teljesülése érdekében jöttek létre, melyeket a következőkben fogok részletezni.

Fontos követelményem, melyet már korábban említettem, hogy az elnökök és a titkárok egy-egy teljes blokkra kerüljenek beosztásra, ne legyenek cserék ezekben a szerepkörökben. Ennek érdekében létrehoztam egy olyan számított döntési változót, mely ezen szerepkörök esetén figyeli a teljes blokkra való beosztást. Szintén kétdimenziós változókról van szó, ahol az egyik dimenziókban az elnöki/titkári szerepet betölteni képes oktatók szerepelnek ($p \in P$ illetve $r \in R$), a másikban pedig a záróvizsgaidőszak blokkjai ($b \in B$). Ezek is bináris változók, melyek azt jelölik, hogy az adott szereplő az adott teljes blokkra beosztásra került-e vagy sem. Látható, hogy ez természetesen összefügg az $x_{i,t}$ döntési változóval, ezért lesz ez az változó számított döntési változó. Pontos meghatározása korlátozó feltételként lehetséges, melyet az 5.2.6.1. fejezetben részletezek. Az elnökökre, illetve a titkárokra vonatkozó számított döntési változókat formálisan az alábbi módon határoztam meg:

$$x_{p,b} \in \{0,1\} \qquad \forall p \in P, \forall b \in B$$
$$x_{r,b} \in \{0,1\} \qquad \forall r \in R, \forall b \in B$$

Mindezeken felül további segédváltozókat is bevezettem, melyeket a terhelések egyenlőbb elosztása érdekében alkalmaztam a modellemen (az erre vonatkozó feltételek a 4.1.2. fejezetben találhatóak). Ezeken belül is az elnökök, a titkárok és a belső tagok terhelésére készítettem külön változókat, melyek amiatt voltak szükségesek, mert az optimalizáció során meg kellett tudnom határozni az adott szereplő terhelésének és az ideális beosztások számának a különbségét. Különböző döntési változók műveleteinek az abszolút értékét egy lineáris programozási feladatban nem lehet meghatározni, így az egyenletet átalakítva segédváltozók segítségével tudtam megoldást találni a problémára.

Mindegyik szerepkörhöz két-két segédváltozóra volt szükségem a modellem felépítéséhez, melyek mindegyiket pozitív egész szám. Ennek miértjeit és pontos működését korlátozó feltételek formájában lehet meghatározni, melyeket az 5.2.6.3 fejezetben fejtek ki a terhelések kiegyenlítése érdekében létrehozott feltételeknél,

valamint alkalmazása megjelenik az 5.2.5. fejezetben kifejtett célfüggvényben is. Formálisan az alábbi módon határoztam meg az említett változókat:

Az elnökök terhelésének kiegyenlítésére létrehozott segédváltozók:

$$x_p^{\alpha} \in \mathbb{Z}^+ \qquad \forall p \in P$$
$$x_p^{\beta} \in \mathbb{Z}^+ \qquad \forall p \in P$$

A titkárok terhelésének egyenlőbbé tételére ezeket a változókat készítettem:

$$x_r^{\alpha} \in \mathbb{Z}^+ \qquad \forall r \in R$$
$$x_r^{\beta} \in \mathbb{Z}^+ \qquad \forall r \in R$$

A belső tag szerepét betölthető oktatók terhelésére az alábbiakat alkalmaztam:

$$x_m^{\alpha} \in \mathbb{Z}^+ \qquad \forall m \in M$$
$$x_m^{\beta} \in \mathbb{Z}^+ \qquad \forall m \in M$$

5.2.5 Célfüggvény

A célfüggvény általános esetben bizonyos döntési változók felhasználásával készült lineáris kifejezés, amelyet az algoritmus futása során optimalizál (minimalizál vagy maximalizál) úgy, hogy közben minden korlátozó feltétel teljesüljön.

A záróvizsgabeosztás problémakörére ezt a kifejezést megfeleltettem a követelményeim teljesüléseként, mégpedig az alábbi módon: a szigorú követelményeim teljesülése nélkül nem tekinthető jónak a beosztás, így ezeket a feltételeket korlátozó feltételként értelmeztem, míg a gyenge követelményeimet nem tekinthetem ilyennek, hiszen azok könnyen ellent is mondanak egymásnak (erre példákat a 2.2.2. fejezetben adok), így 100 %-os teljesülésük nem lehetséges, viszont az optimális felé kell törekedni. Ennek megfelelően bizonyos gyenge követelményeim adják a célfüggvényem alapját.

Ezen belül is a legkritikusabb és legnehezebben elérhető feltételek az elérhetőségek és a terhelések lehető legjobb eloszlása, melyek az előző algoritmusaim legnagyobb gyengeségei voltak, ahogy ezt a 4.2.3. fejezetben is említettem. Így ezen feltételek alapján készítettem el a célfüggvényemet, melyet formálisan az alábbi módon építettem fel a modellemben:

$$min \sum_{i}^{i \in I} \sum_{t}^{t \in T} \left(x_{i,t} * Cost_{i,t}\right) + \sum_{p}^{p \in P} \left(x_p^{\alpha} + x_p^{\beta}\right) + \sum_{r}^{r \in R} \left(x_r^{\alpha} + x_r^{\beta}\right) + \sum_{m}^{m \in M} \left(x_m^{\alpha} + x_m^{\beta}\right)$$

Látható, hogy a kifejezést minimalizálni szeretném, hiszen az egyes tagok egy-egy követelmény megszegését jelentik, amiből szeretném, hogy minél kevesebb legyen. A célfüggvény egyes részeit sorban bemutatom az alábbiakban.

5.2.5.1 Elérhetőségekre vonatkozó célfüggvény részlet

$$\sum_{i}^{i\in I}\sum_{t}^{t\in T}\left(x_{i,t} * Cost_{i,t}\right)$$

A fenti összefüggésben $Cost_{i,t}$ egy pozitív egész konstans, mely megfelel annak a büntetőpontnak, mikor az $i \in I$ oktató a $t \in T$ idősávban ráérése szempontjából nem elérhető (a büntetőpontszám részletei a 4.1. fejezetben kerültek részletezésre). Ezt szorzom össze az adott i oktató t időszeletbeli beosztásához kapcsolódó döntési változóval, és ezt megteszem minden oktatóval minden időszeletben, melyeket mind összegzek. Ez a tag különböző értékeket vehet fel az adott oktató ráérésétől és beosztásától függően:

- Ha i oktató a t időszeletben beosztásre került és el is érhető, vagyis beosztási döntési változója $x_{i,t} = 1$ lett és mivel elérhető $Cost_{i,t} = 0$, akkor a szorzat 0 lesz, ami megfelelő, hiszen ezesetben megengedhető, hogy az oktató be legyen osztva.
- Ha i oktató a t időszeletben nem került beosztásra, vagyis döntési változója $x_{i,t} = 0$ lett, a szorzat szintén 0 lesz függetlenül attól, hogy ráért volna-e vagy sem, ami szintén megfelelő, hiszen nem sérül a ráéréssel kapcsolatos követelmény, ha nem is lett beosztva.
- Ha i oktató a t időszeletben beosztásra került, de nem volt elérhető, vagyis $x_{i,t} = 1$ és $Cost_{i,t} > 0$ a nem elérhetősége miatt, a szorzat pontosan megfelel annak a büntetőpontnak, hogy az adott oktató nem elérhető. Tehát ez az egyetlen eset, mikor nem teljesülnek a követelmények, így ez meg is jeleik a célfüggvényben pozitív egész számként.

5.2.5.2 Terhelések eloszlására vonatkozó célfüggvény részlet

A célfüggvény további részét egyben kezelek, mert hasonló követelmények teljesülése érdekében készült, melyet ebben a fejezetben fejtek ki.

$$\sum_{p}^{p\in P}\left(x_p^{\alpha}+x_p^{\beta}\right)+\sum_{r}^{r\in R}\left(x_r^{\alpha}+x_r^{\beta}\right)+\sum_{m}^{m\in M}\left(x_m^{\alpha}+x_m^{\beta}\right)$$

A fenti három tagból álló forma sorban az elnökök, titkárok, valamint a belső tagok terhelésének eloszlására vonatkoznak, mégpedig mindegyik esetében az összes adott szerepet betölthető oktatóra összeadom az 5.2.4. fejezetben bemutatott segéd döntési változókat. Ezek a változó értékek valójában azt jelentik az egyes résztvevők esetében, hogy mekkora az eltérés az ideális beosztási mennyiséghez képest.

Például az elnökök esetén x_p^{α} jelenti azt, hogy az ideális beosztáshoz képest mekkora a pozitív irányú eltérés, tehát mennyivel több időszeletben lett beosztva az adott elnök beosztva az ideálishoz képest; míg x_p^{β} jelöli azt, ha kevesebb időszeletre lett az adott oktató beosztva, mint ideális lenne, és az értéke megadja az eltérés mértéket. Ebből egyértelműen látszik, hogy x_p^{α} és x_p^{β} közül legalább az egyik mindig 0 értéket vesz fel, és ha ezeket összeadjuk, megkapjuk az eltérések mértékét függetlenül attól, hogy pozitív vagy negatív irányú volt-e az eltérés. Ha mindkét változó értéke 0, az azt jelenti, hogy pontosan annyi időszeletre lett beosztva az adott oktató, mint amennyi a pont megfelelő, így ilyenkor nem emeli semmi a célfüggvény értékét.

A titkárok és a belső tagok esetében ugyanígy számolom ki a terhelések ideálistól való eltérését. Titkároknál a pozitív, illetve negatív irányú eltéréseket x_r^{α} és x_r^{β} jelöli, míg a belső tag szerepét betöltő oktatóknál ez a két változó x_m^{α} és x_m^{β}. Mindegyik változó értékének pontos meghatározásához korlátozó feltételeket kellett bevezetnem, melyeket az 5.2.6.1. fejezetben részletesen is kifejtek.

5.2.6 Korlátozó feltételek

Különböző követelményeknek kell megfelelnie a beosztásnak, melyek között található egészen alapvető elvárás (például a hallgatónak a tényleges konzulense szerepeljen a vizsgán), valamint a többi szigorú követelmény is a korlátozó feltételek által tud maradéktalanul teljesülni. Ezeken kívül a modellemben feltételként kell felvennem a számított döntési változók valódi értékeit is. Ezeknek a követelményeknek a modellemben való megvalósulását mutatom be ebben a fejezetben.

5.2.6.1 Számított döntési változókat meghatározó feltételek

Az 5.2.4. fejezetben bemutattam olyan döntési változókat is, melyek más döntési változók alapján bizonyos feltételek mentén alakulnak ki. Ilyen feltételek lehetnek például bináris változókon végzett lineáris műveletek (*és* művelet, *vagy* művelet).

Az első ilyen számított bináris döntési változóim az elnök szerepét betölteni képes oktatók teljes blokkokra való beosztásait mutatja meg. Az $x_{p,b}$ értéke csak akkor 1, ha minden adott $b \in B$ blokkon belüli $t \in T$ időszeletben beosztásra került az adott oktató, vagyis $\forall t \in b$-re $x_{p,t} = 1$. Viszont, ha bármelyik b-ben lévő t időszeletben $x_{p,t} = 0$, vagyis nincs beosztva az adott elnök, akkor már nincs beosztva a teljes blokkra sem, vagyis $x_{p,b} = 0$ kell legyen. Ezt legegyszerűbben az adott blokkon belüli az időszeletekre vonatkozó döntési változókon végzett *és* művelettel lehet meghatározni, melyet az alábbi feltétel felvételével oldottam meg minden egyes blokkban minden elnöki szerepet betölthető oktatóra:

$$x_{p,b} = \bigwedge_{t \in b} x_{p,t} \qquad \forall p \in P\,, \forall b \in B$$

Az elnökökhöz teljesen hasonló módon építettem fel a titkárok egy blokkra való beosztásához szükséges számított döntési változó meghatározását, vagyis itt is a blokkon belüli időszeletekhez tartozó döntési változókon végzett *és* művelettel számítottam ki a titkárhoz tartozó blokk számított döntési változóját, mely felétel formálisan az alábbi módon néz ki:

$$x_{r,b} = \bigwedge_{t \in b} x_{r,t} \qquad \forall r \in R\,, \forall b \in B$$

A célfüggvény meghatározásakor kifejtettem (az 5.2.5.2. fejezetben), hogy a terhelések eloszlására hogyan használtam az ezt meghatározó számított döntési változókat, melyek pontos értékeinek meghatározásához alkalmazott feltételeket az alábbiakban részletesen is kifejtek.

Az alapkoncepcióm az volt, hogy kiszámítom, mekkora az eltérés az adott szerepkörű oktatók beosztásainak száma és az ideális beosztások száma között. Ehhez elnökök esetében elegendő volt a teljes blokkokra való beosztásokat ($x_{p,b}$) figyelembe venni, hiszen minden esetben teljes blokkban kerülnek beosztásra az elnökök. D_p jelenti azt a konstanst, ahány blokkra kellene ideális esetben beosztani az elnököket. Ha

összeszámolom az elnökök összes beosztását a blokkokra és kivonom belőle D_p-t, akkor ennek abszolút értékéből megkapom, hogy mennyi az eltérés, vagyis mekkora a hiba. Ezek alapján, amit szeretnék meghatározni, formálisan így írnám le:

$$\left| \sum_{b}^{b \in B} x_{p,b} - D_p \right|$$

Ezt az optimalizálandó kifejezést szerettem volna megadni a célfüggvényben is, de sajnos döntési változókból álló lineáris kifejezés abszolút értékét nem lehet megadni a lineáris programozási feladat célfüggvényének. A problémát körüljárva találtam rá egy megoldásra [25], melyet átalakítva alkalmazni is tudtam az abszolút érték problémájának megoldására, melyhez két segédváltozót kellett bevezetnem (x_p^{α} és x_p^{β}). A segédváltozókra megkötés, hogy mindegyik egy nemnegatív egész szám (ezt már definiáláskor korlátoztam az 5.2.4. fejezetben), melyekre az alábbi feltétel teljesülése esetén helyettesíthető az abszolút értékben megadott feltétel csupán a két változó használatával.

$$x_p^{\alpha} - x_p^{\beta} = \sum_{b}^{b \in B} x_{p,b} - D_p \qquad \forall p \in P$$

Ezt a fenti feltételt kellett alkalmaznom minden elnöki szerepet betölteni képes oktatóra, és így a célfüggvényben ki tudtam váltani az abszolút értékes kifejezést a két segédváltozó összegére:

$$\left| \sum_{b}^{b \in B} x_{p,b} - D_p \right| \quad \Longrightarrow \quad x_p^{\alpha} + x_p^{\beta}$$

A titkárok egyenletes eloszlására szintén az elnökökhöz hasonló módon adtam megoldást. Az ő esetükben D_r jelzi az ideális beosztások számát, szintén blokkokban megadva. Így a célfüggvényben hasonlóképpen az alábbi abszolútértékes kifejezés helyett szintén a titkárok terhelési segédváltozóinak összeadását jelenítettem meg.

$$\left| \sum_{b}^{b \in B} x_{r,b} - D_r \right| \quad \Longrightarrow \quad x_r^{\alpha} + x_r^{\beta}$$

Ezesetben szintén fontos, hogy a segédváltozók pozitív egész számok, valamint a az alábbi feltételnek is teljesülni kell, melyet korlátozó feltételként vezettem be:

$$x_r^{\alpha} - x_r^{\beta} = \sum_{b}^{b \in B} x_{r,b} - D_r \qquad \forall r \in R$$

A belső tagok esetében az előző két szereplőhöz képest annyi a különbség, hogy a belső tagok nem teljes blokkokra kerülnek beosztásra, hanem csupán bizonyos időszeletekre, így náluk az időszeletekre vetített összes beosztásuk számát kellett figyelembe venni, illetve D_m az ideális beosztási mennyiséget határozza meg a titkárok esetén időszeletenként számolva. Itt az ideálistól való eltérés abszolút értékét szintén a segédváltozók összegével helyettesítettem az alábbi módon:

$$\left| \sum_{t}^{t \in t} x_{m,t} - D_m \right| \quad \Longrightarrow \quad x_m^{\alpha} + x_m^{\beta}$$

Szintén fontos, hogy pozitív számok legyenek az x_m^{α} és x_m^{β} segédváltozók, valamint, hogy teljesüljön az alábbi korlátozó feltétel.

$$x_m^{\alpha} - x_m^{\beta} = \sum_{t}^{t \in T} x_{m,t} - D_m \qquad \forall m \in M$$

5.2.6.2 Beosztás alapjait meghatározó feltételek

Vannak olyan alapvető elvárások egy záróvizsgabeosztással szemben, melyek valójában azt határozzák meg, hogy ténylegesen egy vizsgáról beszélhetünk-e. Ezeket a feltételeket viszont szintén meg kellett határoznom a modellemben feltételek formájában, melyeket ebben a fejezetben taglalok.

Először is fontos, hogy egy időpontban maximum 5 oktató legyen beosztva, hiszen több szereplőre nincs szükség. Ezt az alábbi egyenlőtlenség teljesíteni tudja, hiszen itt megvizsgálom minden időszeletben, hogy hány vizsgáztató van beosztva, melyet úgy kapok meg, hogy összeadom az összes oktató beosztási változóját az adott időszeletben. Mivel, ha be van osztva valaki az adott időpontra, akkor a változó értéke 1, ha pedig nincs, akkor a változó értéke 0, így ezek összeadásából valóban kiderül, hány oktató lett beosztva.

$$\sum_{i} x_{i,t} \leq 5 \quad i \in I, \forall t \in T$$

A következő korlátozó tényezőm az elnökökre vonatkozik. Fontos ugyanis, hogy egy időszeletben legfeljebb egy elnöki szerepet betölteni képes oktató legyen beosztva, hiszen erőforrás pazarlás lenne, ha egynél több elnök lenne beosztva, viszont az is szükséges, hogy jelen legyen legalább egy elnök a vizsgán, aki levezeti az eseményt. Így alakul ki az a feltétel, hogy pontosan egy elnöknek kell beosztva lennie egy időszeletben. Ehhez a szükséges feltételt szintén minden egyes időszeletre ($t \in T$) értélmezem, és összeadom az elnök szerepkörbe léphető oktatók beosztási változóit ($x_{p,t} \in x_{i,t} \mid i \in P$), és ezen összegnek pontosan 1-et kell kiadnia.

$$\sum_{p} x_{p,t} = 1 \quad p \in P, \forall t \in T$$

Az elnökökhöz hasonló módon a titkárokra is vonatkozik az a feltétel, hogy pontosan egy szerepelhet ebből a szerepkörű oktatóból egy időpillanatban az elnökhöz hasonló okok miatt. Így az ő beosztásukra is vonatkozik minden időszeletben ($t \in T$) az alábbi feltétel. Itt a titkár szerepkörbe beosztható oktatókhoz tartozó döntési változókat ($x_{r,t} \in x_{i,t} \mid i \in R$) adom össze egy-egy időpillanatra vetítve.

$$\sum_{r} x_{r,t} = 1 \quad r \in R, \forall t \in T$$

Belső tagokra szintén szükség van egy záróvizsga során a feltételek szerint, így elmondható, hogy minden időszeletben legyen legalább egy olyan oktató beosztva (vagyis legyen 1 a döntési változója), aki betöltheti a belső tag szerepkörét. Viszont egy beosztás során könnyen előfordulhat, hogy több olyan szereplőnek is részt kell vennie egy vizsgán, akik lehetnének belső tagok. Ez abból adódik, hogy a többi különleges szerepkörhöz képest (elnök és titkár) sokkal többen lehetnek belső tagok. Például ha egy hallgató vizsgatárgyából vizsgáztató oktatók mindegyike lehetne belső tag, illetve a hallgató konzulense is lehet belső tag, de a konzulense egyben nem töltheti be a vizsgázó szerepét, akkor már fixen két olyan oktató kerül beosztásra, akik szerepük szerint lehetnek belső tagok. Így összegezve a belső tagokra azt a megállapítást tettem, hogy minden időszeletben ($t \in T$) a belső tag szerepben megjelenhető oktatók beosztásához tartozó döntési változók ($x_{m,t} \in x_{i,t} \mid i \in M$) összegének legalább 1-nek kell lennie.

$$\sum_{m} x_{m,t} \geq 1 \quad m \in M, \forall t \in T$$

A belső tagokra mindemellett azt a megkötést is alkalmazhatjuk, hogy az előzők alapján viszont 2-nél több belső tag szerepben lévő oktató nem vehet részt a vizsgán, így az előbbi összeg legfeljebb 2 minden időszeletben.

$$\sum_{m} x_{m,t} \leq 2 \quad m \in M, \forall t \in T$$

További fontos alapvető követelmény, hogy minden idősávba legyen beosztva hallgató, viszont ne is legyen egynél több, így a hallgatók beosztására vonatkozó döntési változók ($x_{s,t}$) összege minden időszeletre vonatkozóan ($t \in T$) legyen pontosan 1, melyet az alábbi feltétel ír le.

$$\sum_{s} x_{s,t} = 1 \quad s \in S, \forall t \in T$$

Emellett fontos, hogy minden hallgató vizsgázzon le, de ne vizsgázzon egy hallgató többször. Így arra is kellett egy feltételt megfogalmaznom, hogy minden hallgató pontosan egy időszeletre legyen beosztva. Így a hallgatók döntési változóit ($x_{s,t}$) hallgatókként külön adom össze, tehát minden hallgatóra megvizsgálom külön-külön, hogy az összes időszeletre a döntési változóik összege 1 legyen.

$$\sum_{t} x_{s,t} = 1 \quad t \in T, \forall s \in S$$

A hallgatók mellé fontos, hogy a konzulense is beosztásra kerüljön, tehát amikor a hallgató be van osztva egy $t \in T$ időszeletre, akkor ebben a t időpontban a konzulense is be legyen osztva. Viszont az nem mondható el, hogy a konzulens csak akkor legyen beosztva, mikor a hallgató, hiszen más szerepeket is betölthet az adott oktató más hallgatók vizsgáin, valamint más hallgatónak is lehet konzulense. Így erre a feltételre egy bonyolultabb egyenlőtlenséget kellett konstruálnom.

Megállapítottam, hogy ha a hallgató be van osztva (tehát az adott időpillanatban ezen hallgatóra $x_{s,t} = 1$), akkor a konzulensének (u_s) a döntési változójának is 1 kell, hogy legyen az értéke, vagyis $x_{u_s,t} = 1$. (Itt $x_{u_s,t}$ azt az $x_{i,t}$ döntési változót jelenti, amelyik oktató az adott s hallgató konzulense.) Ekkor a két szereplő döntési változója ($x_{s,t}$ és $x_{u_s,t}$) közötti különbség 0.

Azonban, ha a hallgató nincs beosztva egy adott $t \in T$ időszeletre, tehát $x_{s,t} = 0$, akkor a konzulense ebben a t időszeletben akár be lehet osztva, akár nem, erre nincs

semmilyen megkötés, tehát ha más adott t pillanatban vizsgázó hallgatónál betölt valamilyen más szerepet, akkor $x_{u_s,t} = 1$, ha pedig nem, akkor $x_{u_s,t} = 0$. Ebben az esetben a két döntési változó ($x_{s,t}$ és $x_{u_s,t}$) különbsége 0 vagy -1 lesz.

A két eset összevonásából észrevettem azt, hogy a konzulensek beosztására tudok megfelelő feltételt felírni, mégpedig az alábbi módon. A hallgató döntési változójának értékéből kivonva a konzulense döntési változójának értékét az eredmény biztosan 0 vagy -1 lesz, amit röviden úgy is megfogalmazhatunk, hogy legyen legfeljebb 0. Ezzel kizárhatjuk azt az egyetlen esetet, ami nem megfelelő beosztást eredményezne. Ez a rossz beosztás nem más, mint hogyha a hallgató be van osztva adott t időszeletben, tehát $x_{s,t} = 1$, viszont a konzulense nincs, vagyis $x_{u_s,t} = 0$, akkor a hallgató változójából kivonva a konzulens változóját 1 eredményt kapnánk.

$$x_{s,t} - x_{u_s,t} \leq 0 \quad \forall t \in T\,, \forall s \in S$$

A vizsgáztatók beosztása szintén egy hasonlóan bonyolult feladat abból adódóan, hogy itt is csak annyit tudunk, hogy amikor a hallgató be van osztva ($x_{s,t} = 1$) egy $t \in T$ időszeletben, akkor legyen a vizsgatárgyából vizsgáztatható oktatók közül legalább egy beosztva. Ezt úgy fogalmaztam meg, hogy az összes lehetséges vizsgáztató döntési változóit, ha összeadom az adott t időszeletben, akkor az összegnek legalább 1-et kell kiadnia. (Az összeg $\sum x_{a_{c_s},t}$ ahol c_s jelenti az adott s hallgató vizsgatárgyát, az ebből a tárgyból vizsgáztatható oktatókat a_{c_s} jelöli, azokat az oktatói döntési változókat pedig $x_{a_{c_s},t} \subseteq x_{i,t}$ jelöli, amely ezekhez az oktatókhoz tartoznak.) Így, ha kivonom a hallgató döntési változójából a hallgató vizsgatárgyából vizsgáztatható oktatók döntési változóinak az összegét, akkor biztosan legfeljebb 0-t kapok eredményül.

Amikor nincs az adott hallgató beosztva ($x_{s,t} = 0$), akkor a hozzá tartozó vizsgáztatók szabadon akár be lehetnek osztva az adott t időpillanatban egy másik éppen akkor vizsgázó hallgatónál, akár nem. Ilyenkor a vizsgáztatók döntési változóinak összege lehet 0 vagy akár egy pozitív egész szám, attól függően, hányan vannak beosztva. Viszont a hallgató változójából ha kivonjuk az előbbihez hasonló módon a vizsgáztatók változóinak összegét, biztosan legfeljebb 0-t kapunk eredményül.

Az egyetlen nem elfogadható eset, hogyha a hallgató be van osztva ($x_{s,t} = 1$), de egyetlen vizsgáztató sincs beosztva azok közül, akik vizsgáztathatnának a hallgató vizsgatárgyából. Ilyenkor a vizsgáztatók döntési változóinak összege 0, vagyis

formálisan $\sum x_{a_{cs},t} = 0$. Ebben az esetben, ha elvégezzük a korábbi kivonást, vagyis a hallgató döntési változójából ($x_{s,t}$) kivonjuk a vizsgáztatók döntési változóinak összegét ($\sum x_{a_{cs},t}$), akkor 1-et kapunk eredményként.

Így a lehetséges megoldásokat úgy tudom összefoglalni, hogy az említett különbség legfeljebb 0 lehet, ami egyben kizárja az utolsó nem elfogadható esetet. Ennek megfelelően a vizsgáztatók beosztásához kapcsolódó feltételem az alábbi módon alakult:

$$x_{s,t} - \sum x_{a_{cs},t} \leq 0 \qquad \forall t \in T\,, \forall s \in S$$

5.2.6.3 További követelmények teljesülésére létrehozott feltételek

A modellemnek további feltételeket is teljesítenie kell, melyeket a 4.1. fejezetben határoztam meg. Ezek közül az első, amit részletezek, az elnökök egy blokkra való beosztása. Az 5.2.6.1. fejezetben részleteztem, hogy az ezen feltétel teljesülésére létrehozott számított döntési változóm hogyan alakul ki, viszont szükség van arra, hogy ennek a segítségével meg is fogalmazzam azt a feltételt, hogy az elnökök teljes blokkra legyenek beosztva, mégpedig mindegyik blokkban pontosan egy elnök legyen.

Mindezt úgy értem el, hogy az elnökökre vonatkozó számított döntési változókat ($x_{p,b}$) összeadom egy-egy blokkban, és ennek az összegnek meghatározom, hogy 1 értéket vegyen fel, így fixen egy elnök szerepét betölthető oktató lesz beosztva az adott teljes blokkban. Ezt a feltételt meghatározom minden blokkban, így végül formálisan az alábbi feltételemet építettem fel:

$$\sum_{p} x_{p,b} = 1 \qquad p \in P, \forall b \in B$$

Az elnökökhöz hasonlóan a titkárokra is meghatároztam, hogy minden blokkban legyen pontosan egy titkár szerepét betölthető oktató beosztva. Erre a feltételt az elnökökhöz teljesen hasonló módon az alábbiak szerint határoztam meg a titkárok teljes blokkra való beosztását meghatározó számított döntési változó $x_{r,b}$ segítségével:

$$\sum_{r} x_{r,b} = 1 \qquad r \in R, \forall b \in B$$

A záróvizsgabeosztásom egy további fontos követelménye, hogy az elnökök csak olyan időpontban kerüljenek beosztásra, amikor ráérnek. Mivel ez egy szigorú követelményem (mely részletesen a 4.1.3. fejezetben található), korlátozó feltételt

készítettem rá. Minden egyes elnök szerepet betölthető oktatónál megvizsgáltam minden egyes időszeletben, hogy ráér-e, ezt pedig a következőképpen tettem meg: Az adott oktató beosztásához kapcsolódó döntési változó értékét ($x_{p,t}$ ami $x_{i,t}$-k közül az, ahol $i \in P$) megszorzom annak a súlyával, hogy az adott oktató az adott időszeletben elérhető-e vagy sem ($Cost_{p,t}$). Utóbbi értéke 0, ha az oktató elérhető az adott t időszeletben, pozitív konstans, ha nem elérhető. Ennek a szorzatnak az értéke két esetben lehet 0, mégpedig vagy akkor, ha nem kerül beosztásra az adott oktató (ekkor $x_{p,t} = 0$), vagy ha elérhető az adott időpontban, így nem kap büntetőpontot. Így az összes $p \in P$ oktatónál minden $t \in T$ időpontban az említett szorzatnak 0-t kell adnia, amely feltételt úgy lehet a leegyszerűbben felírni, hogy az összeset összeadom az alábbi módon:

$$\sum_{p}^{p \in P} \sum_{t}^{t \in T} \left(x_{p,t} * Cost_{p,t}\right) = 0$$

Az elnökökhöz hasonló módon szintén fontos feltétel, hogy a titkárok is elérhetőek legyenek minden olyan időszeletben, amikor beosztásra kerülnek. Ehhez is összeszorzom a titkár szerepét betölthető oktatók döntési változóját ($x_{r,t}$) azzal a súllyal, ami jelöli, hogy ráér-e az adott időpontban az adott oktató, vagy sem ($Cost_{r,t}$). Ezeket összegezve az alábbi formális feltételt határoztam meg a modellemben:

$$\sum_{r}^{r \in R} \sum_{t}^{t \in T} \left(x_{r,t} * Cost_{r,t}\right) = 0$$

Végül további feltételként bevezettem egy alsó és egy felső korlátot az elnökök, titkárok és belső tagok beosztásainak számában. Ugyan ennek a gyenge feltételnek a kezelésére már a célfüggvényben adtam egy optimalizációs lépést (az 5.2.5.2 fejezetben), viszont szerettem volna nagyobb hangsúlyt adni a kiemelkedően rossz terhelések számának elkerülésére. Ezért ezt a három szerepet betölthető oktatók beosztásainak számát korlátok közé helyeztem egy-egy korlátozó feltétel meghatározásával.

Az elnöki szerepet betölthető oktatók esetében a korlátok meghatározása beosztott blokkokra vonatkozik, hiszen ezen szerepkör beosztása teljes blokkokra történik, ami már egy másik feltételem által biztosítva van. Itt az elnök blokkokra vonatkozó számított döntési változóit adom össze minden blokkra, ezzel megkapva azt, hogy az adott p elnöki szerepet betölthető oktató hány blokkra lett beosztva. Ennek a számnak kell egy minimum és egy maximum érték között mozognia, mely konstansokat az 5.2.3. fejezetben

definiáltam. Így az alábbi két feltételt kellett meghatároznom az elnökök terheléseire, minden egyes elnökre külön vetítve:

$$\sum_{b}^{b \in B} x_{p,b} \geq D_p^- \quad \forall p \in P$$

$$\sum_{b}^{b \in B} x_{p,b} \leq D_p^+ \quad \forall p \in P$$

Az elnökökhöz teljesen hasonló módon határozom meg a titkárok terhelésének korlátait. Szintén teljes blokkokra kell figyelembe venni őket, így itt is a számított döntési változók összegzésének módszerét tudom használni a beosztások számának meghatározásához. Így az alábbi két feltételt határoztam meg a titkár szerepét betölthető oktatók beosztásainak számára, minden egyes titkárra:

$$\sum_{b}^{b \in B} x_{r,b} \geq D_r^- \quad \forall r \in R$$

$$\sum_{b}^{b \in B} x_{r,b} \leq D_r^+ \quad \forall r \in R$$

A belső tag szerepkörének beosztása nem teljes blokkokra történik, ezért az ő esetükben a terhelések megfelelő elosztásának vizsgálatához a belső tag szerepét betölthető oktatók döntési változóit összegzem minden $t \in T$ időszeletre. Az ehhez kapcsolódó két feltételt formálisan pedig az alábbi módon határoztam meg minden egyes belső tag szerepkörét betölthető oktatóra:

$$\sum_{t}^{t \in T} x_{m,t} \geq D_m^- \quad \forall m \in M$$

$$\sum_{t}^{t \in T} x_{m,t} \leq D_m^+ \quad \forall m \in M$$

Ezzel felépítettem a modellemet a záróvizsgabeosztás problémájára, melyben meghatároztam minden általam elvárt követelményt, és így az algoritmus futtatásával elkészült a beosztás, melynek eredményeit a 6.1. fejezetben taglalom.

6 Eredmények értékelése

Az elkészült lineáris programozás alapú beosztásom különösen jónak mondható, igazságosabb és használhatóbb megoldást adott a korábbiaknál, melynek részleteit ebben a fejezetben fejtem ki részletesen. Először kitérek az algoritmus által elért eredményekre, majd a saját korábbi algoritmusaimmal való összehasonlításra, utána pedig az irodalomban taglalt megoldásokkal vetem össze algoritmusom képességeit.

6.1 Lineáris programozás alapú beosztás

Az algoritmusomat a BME 100 fős homogén hallgatói csoportján teszteltem, mint ahogy korábban a genetikus és heurisztikus algoritmusaimat is. Az elkészült beosztásom vizsgálatára egyrészt felhasználtam a Gurobi [24] adta lehetőségeket, ugyanis a modell futtatása során automatikusan megjelenít bizonyos információkat. Ezeket az 5. ábra tartalmazza. Az információk között található, hogy 260 korlátozó feltételt adtam meg összesen, 15106 változót definiáltam a modellemben, melyek közül 15060 bináris. Látható továbbá, hogy összesen 2.66 másodpercig tartott a teljes algoritmus lefutása 8 szálon futva, valamint az is, hogy ezalatt talált megoldást a megadott feltételek mellett.

```
Optimize a model with 20801 rows, 15106 columns and 94831 nonzeros
Model has 260 general constraints
Variable types: 0 continuous, 15106 integer (15060 binary)
Coefficient statistics:
  Matrix range     [1e+00, 5e+00]
  Objective range  [1e+00, 5e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+01]
Presolve removed 12029 rows and 3239 columns
Presolve time: 0.71s
Presolved: 8772 rows, 11867 columns, 63706 nonzeros
Variable types: 0 continuous, 11867 integer (11837 binary)

Root relaxation: objective 5.000000e+00, 7008 iterations, 1.40 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

H    0     0                       5.0000000 -106.00000  2220%     -    2s
     0     0     cutoff    0         5.00000    5.00000  0.00%     -    2s

Explored 0 nodes (9287 simplex iterations) in 2.66 seconds
Thread count was 8 (of 8 available processors)

Solution count 1: 5

Optimal solution found (tolerance 1.00e-04)
Best objective 5.000000000000e+00, best bound 5.000000000000e+00, gap 0.0000%
Obj: 5
```

5. ábra - Gurobi által készített összefoglaló

Emellett az elkészült beosztást én is elemzés alá vetettem, mégpedig kilistáztam, hogy milyen követelményeim nem teljesültek. Ezek alapján kiderült, hogy mindösszesen 40 büntetőpontot szerzett az algoritmusom a 4.1. fejezetben meghatározott pontrendszer alapján, melyek mindegyike terhelésekből adódott, így ezeket jobban megvizsgáltam, hogy pontosan mik voltak azok. Kettő titkár szerepbe beosztott oktatónál sérült a követelmény, mely $2 * 10$ büntetőpontot okozott, továbbá két belső tag esetében mutatott egy-egy enyhébb eltérést a többi tag terhelésétől, melynek köszönhető a további $2 * 10$ büntetőpont.

A terhelések eloszlását a 2. táblázat tartalmazza, ahol a számok azt jelentik, hogy a 100 vizsgából hányra lett beosztva az adott oktató a megadott szerepkörbe.

Elnökök	Nr	Titkárok	Nr	Belső tagok	Nr
Charaf Hassan	25	Braun Patrik János	10	Asztalos Márk	10
Forstner Bertalan	25	Budai Ádám	10	Blázovics László	10
Lengyel László	25	Fekete Tamás	10	Csorba Kristóf	10
Vajk István	25	Gazdi László	10	Dudás Ákos	10
		Hideg Attila	10	Ekler Péter	11
		Jánoky László Viktor	15	Iváncsy Szabolcs	8
		Pomázi Krisztián	10	Kovács Tibor	9
		Somogyi Ferenc Attila	15	Kővári Bence András	12
		Tömösközi Máté Ferenc	10	Mezei Gergely	10
				Recski Gábor András	10

2. táblázat - Terhelések eloszlása egyes szereplőknél

Látható, hogy a két oktató, aki 10 helyett 15 időpontba lett beosztva, ők okozzák a kapott büntetőpontokat, hiszen eltérnek a többi titkár beosztásainak számától. Egy blokkom 5 idősávból áll a teszthalmazomban, és mivel teljes blokkokra kell beosztanom a titkárokat, így egyértelmű, hogy a beosztásaik számának öttel oszthatónak kell lennie. Azonban a tesztesetemben 9 olyan oktató van, aki betöltheti a titkár szerepét, ami a 100 vizsgára leosztva $100 \div 9 = 11.11$-et ad, vagyis természetes, hogy nem lehet mindenkinek pontosan 10 beosztása, mert akkor maradna még 10 vizsga, vagyis 2 teljes blokk, melyekre a legmegfelelőbb eljárás, hogy 1-1 oktatót még be kell osztani. Ebből látható, hogy a lehető legmegfelelőbb megoldást alkalmazta az algoritmusom ebben a helyzetben az adott követelmények között.

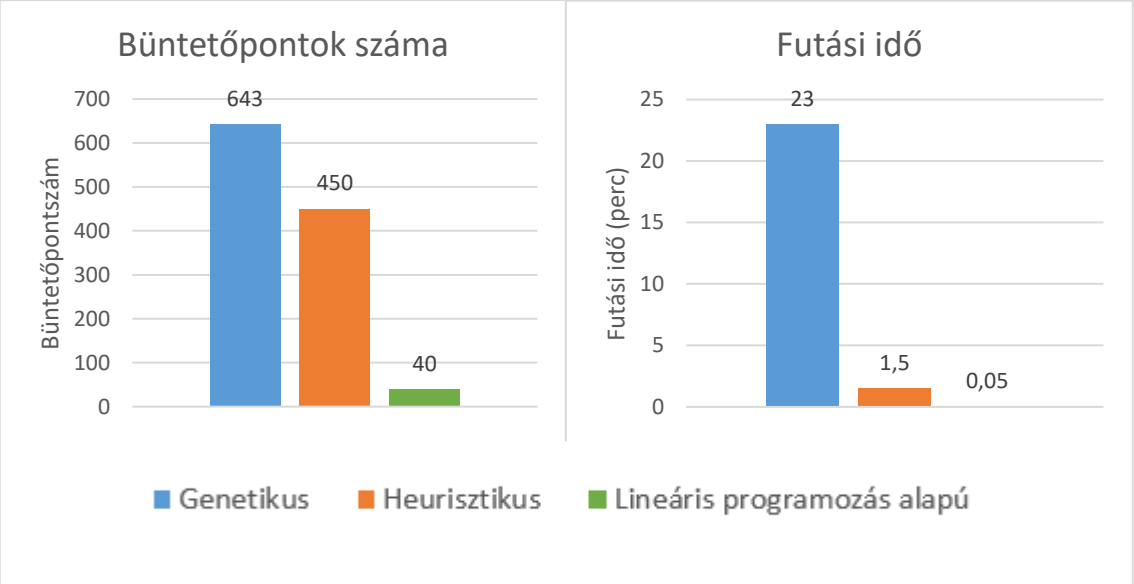
A belső tagok esetében a büntetőpontot a 8, illetve a 12 beosztással rendelkező oktatók okozták, hiszen itt az ideális 10 beosztás lenne (a kisebb eltéréseket még hibahatáron belülinek tekinti az algoritmusom). Utánajártam a problémának, és az okozta

az eltérést, hogy azoknak az oktatóknak, akiknek 10-nél több beosztása lett, több saját konzultált hallgatójuk van. Ezért azokon a vizsgákon, mikor a hallgató vizsgázik, mindenképp jelen kell lenniük, és a szerepkörök összevonása miatt az algoritmus a belső tag szerepébe is őket osztotta be, ami megfelelő eljárás, hiszen így nem kell plusz oktatót behívni az adott vizsgára.

Így látható, hogy a követelményeim mellett a lineáris programozás alapú algoritmusom megtalálta a lehető legmegfelelőbb beosztást.

6.2 Összehasonlítás a saját korábbi eredményeimmel

A genetikus és heurisztikus algoritmusaim eredményeit a 4.2.3. fejezetben foglaltam össze. A 6. ábra mutatja, hogy összehasonlítottam a három algoritmusom által elért legjobb beosztások által kapott büntetőpontokat, valamint az algoritmusok futási időit.



6. ábra - Beosztások összehasonlítása

Látható, hogy messze a legjobb eredményt érte el mind büntetőpontszám tekintetében, mind pedig futási időben az új lineáris programozás alapú megoldásom. A büntetőpontszám esetében ez várható is volt, hiszen mint ahogy a 6.1. fejezetben azt kifejtettem, a meghatározott követelményeim alapján megtalálta a lehető legideálisabb megoldást. A másik két algoritmusomban ugyan szintén csak gyenge követelmények nem teljesültek, azonban ezek mértéke láthatóan sokkal nagyobb volt, és azok a beosztások az egyének számára a terhelések tekintetében nem voltak megfelelőek.

6.3 Összehasonlítás az irodalomban taglalt algoritmusokkal

Az irodalomban taglalt egyetemi vizsgabeosztási algoritmusok mind-mind különböző szempontokat vettek figyelembe, és különböző követelményeket teljesítettek, melyeket a 3.2. fejezetben már taglaltam.

Az algoritmusok átfogóbb tulajdonságai mentén összehasonlítást végeztem a saját lineáris programozás alapú algoritmusom, valamint az irodalmi áttekintésben is taglalt cikkek között. Ennek eredményét a 3. táblázat mutatja be, ahol ✓ szimbólum jelzi azt, ha az algoritmus az adott képességet teljesíti, × jel pedig azt jelzi, hogy nem veszi figyelembe az adott tulajdonságot, nem teljesíti azt.

	hard és soft követelmények megkülönböztetve	terheléseloszlás	elérhetőség	blokkok kezelése	párhuzamos vizsgák támogatása	specifikus oktatók a vizsgákhoz
saját lineáris programozás alapú algoritmus	✓	✓	✓	✓	✓/×	✓
Wijgers, és Hoogeveen [9] (2007)	×	×	✓	✓	×	×
Al-Yakoob, Sherali és Al-Jazzaf [10] (2010)	✓	✓	✓	×	✓	×
Kochaniková és Rudová [11] (2013)	✓	×	✓	×	✓	✓
Bergmann, Fischer és Zurheide [12] (2014)	✓	✓	✓	×	✓	×
Ivancevic, Knezevic és Lukovic [13] (2014)	×	×	×	✓	✓	✓
Aslan, Şîmşek és Karkacier [14] (2017)	×	✓	×	×	✓	×

3. táblázat - Algoritmusok összehasonlítása képességek szerint

A következőkben sorban végig veszem a képességeket, melyek mentén összehasonlítom az algoritmusokat.

Az első képesség a *hard és soft követelmények megkülönböztetve*. Itt azt veszem figyelembe, hogy az adott megoldás elkülöníti-e, diverzifikálja-e a követelményeit, ezzel is egy árnyaltabb megközelítést alkalmazva a feltételeknek, mellyel egyben fontossági sorrendet is ad a követelmények teljesülésének. Ahogy a 4.1 fejezetben részletesen is bemutattam, a saját algoritmusom elkészítésekor különböző követelményeket határoztam meg, melyeket nem csak gyenge és szigorú kategóriák közé soroltam, hanem pontrendszert is készítettem rá, hogy pontosan meg lehessen határozni, melyik feltételem teljesülése mennyi lényeges.

Látható, hogy az általam említett irodalmi példák elég változatos képet adnak ebben a témában. Jellemzően ott, ahol kevesebb követelményt határoznak meg, nem választják külön a feltételeket, míg ahol bonyolultabb követelményrendszert használnak, ott megkülönböztetik azok fontosságát egymástól.

A következő vizsgálandó pont a *terheléseloszlás*: Az egyetemi beosztástervezésnek egy különleges pontja az egyes oktatók, termek, hallgatók terhelésének megfelelő eloszlása, hogy senki vagy semmi ne legyen a többihez képest alul- vagy túlterhelve. Ahogy korábban a 6.1. fejezetben is részleteztem az eredményeknél, erre a pontra különös hangsúlyt fektettem algoritmusomban, hiszen a célfüggvényemnek egy része is erre a problémára koncentrál (ahogy az 5.2.5.2. fejezetben erről írtam is).

Az irodalomban ezzel a témával kapcsolatban is változatos a kép, sok esetben nem foglalkoznak ezzel a problémával, és ahol mégis, ott is változó, hogy kiknek a terhelésére figyelnek. Például Bergmann, Fischer és Zurheide [12] cikkében a hallgatók és a termek terheltségére fordítanak figyelmet, viszont az oktatók ezen tulajdonságára már nem, Aslan, Şîmşek és Karkacier [14] tanulmányában az oktatók terhelését vizsgálják.

A harmadik képesség, amit vizsgálok az *elérhetőség*. Ez annyit jelent, hogy az adott algoritmus figyelembe veszi-e az egyes erőforrások elérhetőségeit, hogy megadhatja-e az oktató, hogy mikor elérhető, és annak megfelelően kerül-e beosztásra. Az én lineáris programozás alapú modellem figyelembe veszi ezt a követelményt, több feltételem (4.1.1) is vonatkozik erre.

Az irodalomban bemutatott algoritmusok többsége teljesíti ezt a képességet, hiszen a valós beosztások elkészítésénél is a leggyakrabban elvárható ez a feltétel.

A következőkben a *blokkok kezelése* alatt azt a képességet értem, hogy az algoritmus tud-e a vizsgaidőszakon belül bizonyos részeket egyben kezelni, figyel-e arra, hogy egyes erőforrások, oktatók beosztásakor hasznos, ha nem folyamatosan cserélődnek, hanem egy-egy blokkban végig beosztásra kerülnek. Az algoritmusom teljesít ilyen feltételt, a teljes vizsgaidőszakot blokkokra osztom, ami esetemben egy-egy délelőttöt, illetve délutánt jelent. Az elnökök és a titkárok beosztásánál pedig figyelek arra, hogy teljes blokkra kerüljenek beosztásra.

Ezt a képességet csak néhány másik, az irodalomban bemutatott algoritmus teljesít, Ivancevic, Knezevic és Lukovic [13] cikkében például csak egy-egy oktató van jelen a vizsgákon a hallgatók mellett, és ezek az oktatók nem cserélődnek a blokkokban.

A következő szempont a *párhuzamos vizsgák támogatása*, mely arról szól, hogy lehetséges-e egyidőben több vizsgát lebonyolítani. Az általam elkészített algoritmus jelenlegi formájában a szűkített állapottérre lett tesztelve (lásd 2.1.2. fejezet), mely nem támogatja a párhuzamos vizsgákat. Viszont azért jelöltem a táblázatban ezt köztes megoldásként, mert valójában a modellem fel van készítve arra, hogy több teremben is zajlani tudjanak a vizsgák, ehhez több időszeletet kellene felvennem, valamint plusz egy feltételt megfogalmazni, ami biztosítja, hogy egy ember ne legyen egyszerre két helyre beosztva. Így ez egy továbbfejlesztési lehetőség az algoritmusomban, de elméletben bővíthető a modellem ebbe az irányba is.

Az irodalomban taglalt algoritmusok többségében van lehetőség párhuzamosításra, ahol is vagy több vizsgát tartanak egy időben, vagy akár az is előfordul, hogy egy teremben engedik meg egyszerre több vizsga megtartását.

Az utolsó összehasonlítási szempontom a *specifikus oktatók a vizsgákhoz* elnevezést kapta. Ez nem jelent mást, mint hogy egyes vizsgákhoz szükséges lehet, hogy egy bizonyos meghatározott oktató legyen jelen, vagy az oktatók egy meghatározott csoportjából legyen beosztva valaki vagy valakik. Az én algoritmusom nagyban épít erre a tulajdonságra, hiszen meghatározott szerepkörű oktatóknak kell jelen lenni. Vannak olyan szereplők, akik egy-egy csoportból kerülnek ki (elnök, titkár, belső tag), van, aki egyértelműen adott egy hallgató vizsgájához (konzulens), és előfordul olyan is, akit a

hallgató által választott csoportból kell kiválasztani (vizsgáztató, amihez a hallgató választja a vizsgatárgyat).

Tapasztalataim alapján az irodalomban vizsgált esetek többségében nincs erre lehetőség, nincs meghatározva, hogy kiket kell beosztani, csak az oktatói halmazból tetszőleges valakit. Kochaniková és Rudová [11] tanulmányában például egyetlen szereplő kerül megnevezésre, az elnök, így oda csak az adott szerepkört betölthető oktató kerül beosztásra.

Mindezek alapján látható, hogy algoritmusom a nemzetközi irodalmat áttekintve az eddigieknél egyedibb módon átfogó megoldást ad a záróvizsgabeosztások tervezésének több kérdésére is, több különböző képességet ötvöz, mint amennyit az irodalomban taglalt hasonló témakörű tanulmányok.

7 Összefoglalás

A záróvizsgák szóbeli vizsgája félévente minden tanszéken megrendezésre kerül, az egységes szabályrendszer szerint, amely nagyon sok vizsgát jelent a nagy hallgatói létszám miatt. Az állapottér nagysága exponenciális hatással van a lehetséges bemenetek számára, így minél több hallgatóval állunk szemben, annál komplexebb a probléma.

Korábbi TDK dolgozatomban ezen szívemhez közeli problémára adtam két különböző megközelítésű megoldást. A kutatásaim során először a záróvizsga rendszerének formalizálását végeztem el, amikor is meghatároztam a szükséges követelményeket, hozzájuk pontszámokat rendeltem annak megfelelően, hogy mennyire fontos a teljesülésük. Ezek alapján egy genetikus és egy heurisztikus algoritmus alapú megközelítést dolgoztam ki a problémára. Mindkét beosztásom egy használható, de közel sem tökéletes beosztás adott, így szükségesnek éreztem a továbbfejlesztésüket.

A korábbi algoritmusaim legnagyobb hibája az oktatók nem egyenletes terhelése volt, valamint az ott alkalmazott megközelítések nem voltak könnyen továbbfejleszthetők. Így egy új megközelítést építettem fel a korábban formalizált problémakörre, mely ezúttal lineáris programozás alapú volt.

Felépítettem a lineáris programozás alapú modellemet, melyben a legnagyobb kihívást a megfelelő döntési változók kiválasztása, valamint az azok alapján készült célfüggvény és a korlátozó feltételek megalkotása jelentette. Egyes feltétételek meghatározásához szükség volt további változók és segédfeltételek meghatározására is a lineáris programozás korlátjai miatt.

Az új megközelítésű algoritmusommal viszont sikerült egy minden eddiginél igazságosabb megoldást adnom a záróvizsgabeosztás problémájára, hiszen a követelményeim keretein belül ideális megoldást talált.

Összehasonlítottam eredményeimet a korábbi algoritmusaimmal is, melyből szintén látható, hogy a jelenlegi megoldás nagymértékben felülmúlja a korábbiakat. Emellett a nemzetköri irodalomban kutatott egyetemi vizsgabeosztásokkal is végeztem összehasonlítást, melynek eredményeképp elmondhatom, hogy megoldásom egyedi módon átfogó választ ad a záróvizsgabeosztások tervezésének több kérdésére is.

8 Irodalomjegyzék

[1] S. Erdős, Beosztástervezési algoritmusok fejlesztése, 2017.

[2] S. Erdős, „Automatizált záróvizsga beosztás készítése,” *Tudományos Diákköri Konferencia,* 2019.

[3] Rektori Kabinet Oktatási Igazgatóság, „A Szenátus X./10./2015-2016. (2016. VII. 11.) számú határozata A BME TANULMÁNYI ÉS VIZSGASZABÁLYZATÁRÓL,” 1. szeptember 2016.. [Online]. Available: http://www.kth.bme.hu/document/2061/original/BME_TVSZ_2016%20elfogadott_mod_20180801_web.pdf. [Hozzáférés dátuma: 2. október 2018.].

[4] „BME VIK BSc szakdolgozat, záróvizsga, oklevél szabályzat a BME,” 07 06 2017. [Online]. Available: https://www.vik.bme.hu/document/1343/original/BSc-ZV-170607.pdf. [Hozzáférés dátuma: 23 10 2018].

[5] Pinedo és L. Michael, Scheduling: Theory, Algorithms, and Systems, 2016..

[6] ITC 2019 Discussion Forum, „ITC 2019: International Timetabling Competition,” [Online]. Available: https://www.itc2019.org/home. [Hozzáférés dátuma: 24. október 2018.].

[7] „PATAT Conferences,” [Online]. Available: http://patatconference.org/index.html. [Hozzáférés dátuma: 24. október 2018.].

[8] A. SCHAERF, „A Survey of Automated Timetabling,” p. 87, 1999.

[9] R. Wijgers és J. Hoogeveen, „Solving the examination timetabling problem,” 2007.

[10] S. M. Al-Yakoob, H. D. Sherali és M. Al-Jazzaf, „A mixed-integer mathematical modeling approach to exam timetabling,” *Computational Management Science,* 2010.

[11] B. Kochaniková és H. Rudová, „Student Scheduling for Bachelor State Examinations,” 2013.

[12] L. K. Bergmann, K. Fischer és S. Zurheide, „A linear mixed-integer model for realistic examination timetabling problems," *10th International Conference of the Practice and Theory of Automated Timetabling,* 2014.

[13] V. Ivancevic, M. Knezevic és I. Lukovic, „A Course Exam Scheduling Approach based on Data Mining," *Frontiers in Artificial Intelligence and Applications,* 2014.

[14] E. ASLAN, T. ŞİMŞEK és A. KARKACIER, „A Binary Integer Programming Model For Exam Scheduling Problem With Several Departments," *13th International Conference on Knowledge, Economoy and Management,* 2017.

[15] S. Erdős és B. Kővári, „Algorithms for final exam scheduling," 2018.. [Online]. Available: https://github.com/BenceKovari/Schedule/blob/master/Fitness.md.

[16] O. T. Arogundade, A. T. Akinwale és O. M. Aweda, „A Genetic Algorithm Approach for a Real-World University Examination Timetabling Problem," *International Journal of Computer Applications (0975 – 8887),* 2010.

[17] N. Kokash, „An introduction to heuristic algorithms".

[18] „Javító út keresés páros gráfokban, magyar módszer," [Online]. Available: http://www.math.u-szeged.hu/~hajnal/courses/MSc_Grafelmelet/grafelmelet/magyar.htm. [Hozzáférés dátuma: 3. október 2018.].

[19] J. J. Grefenstette, „Genetic algorithms for changing environments," *PPSN,* pp. 137-144, 1992.

[20] W. Carsten, „Population Size vs. Runtime of a Simple Evolutionary Algorithm," *Theoretical Computer Science,* pp. 104-120, 2008..

[21] G. B. Dantzig, Linear Programming and Extensions, Princeton, New Jersey: Princeton University Press, 1963..

[22] G. Dantzig, „Origins of the simplex method," *A History of Scientific Computing,* 1987.

[23] R. Anand, D. Aggarwal és V. Kumar, „A comparative analysis of optimization solvers," *Journal of Statistics and Management Systems,* pp. 623-635, 2017.

[24] „https://www.gurobi.com/,” Gurobi Optimization, LLC, 2007. [Online]. [Hozzáférés dátuma: 2019].

[25] B. McCarl és T. Spreen, „Linear Programming Modeling: Nonlinearities and Approximation,” 1997.