



Témalaboratórium beszámoló

Távközlési és Médiainformatikai Tanszék

Készítette:	Vörös Vilmos
Neptun-kód:	HW921K
Ágazat:	Médiainformatika
E-mail cím:	v.vilmos10@gmail.com
Konzulens(ek):	Dr. Cinkler Tibor
E-mail címe(ik):	cinkler.tibor@vik.bme.hu

**Téma címe: Felhasználók és bázisállomások
összerendelésének modellezése és optimalizálása**

Feladat

Az adott félévben a feladatomban egy kisebb modell összeállítása volt, aminek célja szemléltetni a felhasználók csatlakoztatását egy 5G mobilhálózathoz. A koncepció egy néhány bázisállomásból és számos felhasználói eszközökből álló rendszer építése volt, amiben a valós hálózatokban felmerülő költségek mellett lehet modellezni a felhasználók optimális elosztását a kiszolgáló bázisállomások között. A kezdeti, magas fokú absztrakciót a félév során egyre több beépített elemmel kellett csökkenteni, melyek elméleti háttérének megismerése is követelmény volt. A feladat különböző fázisaiban tesztelni kellett, hogy a modellhez éppen hozzáadott részlet vagy egyéb szempont megfelelően, az elvárt módon befolyásolja-e az összerendeléseket. Végül célkitűzés volt a felépített rendszer igényes, könnyen átlátható ábrázolása.

2023/2024. 1. félév

1. A laboratóriumi munka környezetének ismertetése, a munka előzményei és kiindulási állapota

1.1 Bevezető

A laboratóriumi munkám első fele a feladathoz szükséges elméleti ismeretek megismeréséből állt. A téma címében is említett felhasználó-bázisállomás összerendelések modellezéséhez a mobilhálózatok alapvető működésével és az ehhez kapcsolódó legújabb trendekkel kellett megismerkedni, amihez segítségemre volt több, kapcsolódó cikk is [1] [2]. Az itt szerzett ismeretek adták meg az alapját, hogy a felépítendő rendszert a mobilhálózati működés szempontjából leglényegesebb elemekkel bővítsem. A feladat másik célkitűzése a felhasználók elosztásának optimalizálása volt. Ehhez el kellett sajátítanom a lineáris programozás alapjait, majd át kellett ültetnem a kijelölt feladatot egy lineáris programozási feladatba. Ehhez jó alapot adott az IBM CPLEX Optimization Studio¹ súgója, ahol részletes leírást találtam a különböző use case-ekről, alkalmazásokról.[3]

1.2 Elméleti összefoglaló

Az optimalizálási feladatom megoldása teljes egészében a lineáris programozásra (LP) épül. A lineáris programozás egy matematikai módszertan, ami az egyik legelterjedtebb eszköz optimalizálási problémák megoldására. A módszer a lineáris egyenletrendszerekre épül, ami lehetővé teszi egy döntési problémánál az optimális döntés meghozatalát különböző előre definiált korlátok mellett.

A technika három elemre épül: különböző döntési változókra, ezen döntési változókra megfogalmazott kényszerekre, valamint egy, a döntési változókat tartalmazó célfüggvényre. Egy probléma megoldása a döntési változóknak történő értékadásból áll olyan módon, hogy a megadott értékek mellett a célfüggvény minimális vagy maximális értéket vegyen fel. Az optimális értékek meghatározására többféle algoritmus létezik. A módszer nevében álló lineáris szó arra utal, hogy a feladatban megfogalmazott kényszerek és maga a célfüggvény is valamilyen lineáris összefüggése/függvénye a döntési változóknak és egyéb opcionális változóknak.

1.3 A munka állapota, készültségi foka a félév elején

A félévben végzett munkámnak nem volt előzménye.

¹ <https://www.ibm.com/products/ilog-cplex-optimization-studio>

2. Az elvégzett munka és eredmények ismertetése

2.1 Bemelegítés

A félév elején, a feladat körvonalazódásakor megismertem az IBM CPLEX Optimization Studio nevű szoftvert, és átnéztem néhány tutorial kódot, hogy egyfajta áttekintést kapjak a különféle optimalizációs problémákról. A példakódok vizsgálata és futtatása után saját kódot is készítettem, immár a kitűzött feladatomat előkészítve. Az Optimization Studio egy ún. OPL (Optimization Programming Language) nyelvet használ, mellyel könnyen lehet megfogalmazni optimalizációs modelleket. Ez hasonlít a C nyelvre, és könnyen olvasható. Míg az Optimization Studio hatékony algoritmusokat tartalmaz az optimalizációs problémák megoldására, az OPL nyelv nem tartalmazott olyan könyvtárat, ami segített volna a modell kirajzolásában/megjelenítésében.

Ebből kifolyólag feladatomat Python nyelven írtam, és a docplex nevű csomagot² használtam, amin keresztül elértem a CPLEX Optimization Studio funkcionalitását. Ezen az API-n keresztül definiáltam a modelletem és fogalmaztam meg az LP problémám változóit, kényszereit és célfüggvényét. Az optimalizáción túl az adatfeldolgozáshoz és ábrázoláshoz is tudtam egy-egy Python könyvtárat használni.

2.2 A LP probléma megfogalmazása

2.2.1 Döntési változók

A modellezési feladat két típusú objektumokból áll: bázisállomásokból (BS) és felhasználókból (users). A feladat egyszerűsítése érdekében 90 darab felhasználót és 9 darab bázisállomást helyeztem el egy 600x600-as Descartes-féle koordináta-rendszerben. A BS-ek a modellezett területen egyenletesen, egymástól egyenlő távolságra helyezkednek el, míg a userek a kijelölt területen belül mozognak egy átalakított mobilitási modell szerint. A BS-user összerendeléseket egy 90x9-es méretű mátrix ábrázolja:

$$x_{u,g} \in \{0,1\} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\}$$

$$x_{u,g} = 1, \text{ ha az } u. \text{ felhasználó hozzá van kapcsolva a } g. \text{ gNB-hez.}$$

A bázisállomásokat három mérték jellemzi: a működési állapot, az energiaellátás szerinti preferencia és a kapacitás. Az első két tulajdonságot egy-egy 9 darab elemből álló sorvektor határozza meg. Működési állapotok közül 2 (0-kikapcsolt, 1-bekapcsolt), energiaellátás szerinti preferenciák közül 10 különböző szintet különböztetünk meg. (1- legjobb, 10-legrosszabb). A szimuláció során egy BS működési állapota változhat, de az előre beállított preferencia nem. A működési állapotot reprezentáló sorvektor:

$$y_g \in \{0,1\} \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\}$$

$$y_g = 1, \text{ ha a } g. \text{ gNB bekapcsolt állapotban van.}$$

A felhasználókat helyzetük mellett sáv szélesség-igényük jellemzi, ami azonban állandó marad a szimuláció során. Mivel a modellezési feladat szempontjából lényeges, hogy a már

² <https://pypi.org/project/docplex/>

meglévő összerendeléstől való eltérés extra költségekkel jár, ezért be kellett vezetni olyan döntési változókat, amikkel nyomon lehet követni két megoldás/összerendelés közötti eltérést. Ezt a célt szolgálják az alábbi sorvektorok:

$$\begin{aligned} dy_g &\in \{0,1\} \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\} \\ dx_{u,g} &\in \{0,1\} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \end{aligned}$$

$dy_g = 1$, ha a g . gNB működési állapota megváltozott az optimalizáció után.

$dx_{u,g} = 1$, ha az u . felhasználó és a g . gNB közötti összerendelés az optimalizáció után.

Végül a felhasználók kapcsolatait a hálózathoz egy-egy sorvektor írja le. Az alsó indexek jelölik az adott kapcsolat fontosságát:

$$\begin{aligned} z1_u &\in \{0,1\} \quad \forall u \in \{0, \dots, N_{\text{users}}\} \\ z2_u &\in \{0,1\} \quad \forall u \in \{0, \dots, N_{\text{users}}\} \\ z3_u &\in \{0,1\} \quad \forall u \in \{0, \dots, N_{\text{users}}\} \end{aligned}$$

$z1_u = 0$, ha a u . felhasználó rendelkezik egy elsőszámú kapcsolattal.

$z2_u = 0$, ha a u . felhasználó rendelkezik egy másodlagos kapcsolattal.

$z3_u = 0$, ha a u . felhasználó rendelkezik egy harmadlagos kapcsolattal.

2.2.2 Egyéb változók

A döntési változók mellett egyéb változókra is szükség volt, hogy a valós hálózatokban felmerülő költségeket és egyéb befolyásoló tényezőket jobban lehessen modellezni, illetve ezek hatását vizsgálni.

- Legutolsó összerendeléseket reprezentáló mátrix x :

$$\begin{aligned} previous_state_x_{u,g} &\in \{0,1\} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \\ previous_state_x_{u,g} &= 1, \text{ ha az } u. \text{ felhasználó hozzá volt kapcsolva} \\ &\text{a } g. \text{ gNB-hez a legutóbbi optimalizálás megoldásában.} \end{aligned}$$

- Bázisállomások legutolsó állapotait reprezentáló sorvektor y :

$$\begin{aligned} previous_state_y_g &\in \{0,1\} \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\} \\ previous_state_y_g &= 1, \text{ ha az } g. \text{ gNB bekapcsolt állapotban} \\ &\text{volt a legutolsó optimalizálás megoldásában.} \end{aligned}$$

- b_u : Egy felhasználó sávszélesség-igénye.
- c : Egy bázisállomás kapacitása, azaz az összes átvihető forgalom sávszélessége Mbps mértékegységben. Ez az érték az összes BS-t jellemzi.
- $power_cost_g$: Egy bázisállomás preferenciája energiaellátásának szempontjából.
- $power_cost_weight$: Egy bekapcsolt bázisállomás energiaellátásának költségének súlyozása a célfüggvényben.
- $switching_cost$: Egy bázisállomás állapotváltozásának költsége.
- $distance_weight$: Egy felhasználó és egy bázisállomás közötti távolság súlyozása a

célfüggvényben.

- $distance_{u,g}$: Számított érték, amely leírja egy bázisállomás és egy felhasználó közötti Euklideszi távolságot.
- $handover_cost$: Egy handover költsége.
- $distance_limit$: Maximum megengedett távolság egy összerendelt bázisállomás és felhasználó között.
- $blocking_cost$: Egy felhasználó "elvesztésének" a költsége, azaz amikor nem rendeljük egyetlen bázisállomáshoz sem.
- $second_connection_cost$: Annak a költsége, amikor egy felhasználót nem tudunk egy második bázisállomáshoz rendelni.
- $third_connection_cost$: Annak a költsége, amikor egy felhasználót nem tudunk egy harmadik bázisállomáshoz rendelni.

2.2.3 Kényszerek

Az optimalizációs modellben alkalmazott kényszerek:

- A bázisállomások állapotváltozásait leíró kényszerek:

$$\begin{aligned} y_g - \text{previous_state_}y_g &\leq dy_g \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\} \\ \text{previous_state_}y_g - y_g &\leq dy_g \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\} \\ y_g + \text{previous_state_}y_g &\geq dy_g \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\} \\ 2 - y_g - \text{previous_state_}y_g &\geq dy_g \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\} \end{aligned}$$

- Az összerendelés-változásokat, azaz a handovereket leíró kényszerek:

$$\begin{aligned} x_{u,g} - \text{previous_state_}x_{u,g} &\leq dx_{u,g} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \\ \text{previous_state_}x_{u,g} - x_{u,g} &\leq dx_{u,g} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \\ x_{u,g} + \text{previous_state_}x_{u,g} &\geq dx_{u,g} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \\ 2 - x_{u,g} - \text{previous_state_}x_{u,g} &\geq dx_{u,g} \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \end{aligned}$$

- Minden felhasználót maximum három bázisállomáshoz rendelünk hozzá:

$$\sum_{g=0}^{N_{\text{gNBs}}} x_{u,g} = 3 - z1_u - z2_u - z3_u \quad \forall u \in \{0, \dots, N_{\text{users}}\}$$

- A megengedettnél nagyobb távolságra lévő bázisállomásokat és felhasználókat nem rendelhetjük egymáshoz:

$$\begin{aligned} x_{u,g} &= 0, \text{ ha } distance_{u,g} > distance_limit \\ &\forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\} \end{aligned}$$

- Minden felhasználó csak bekapcsolt állapotban lévő bázisállomáshoz rendelhető hozzá:

$$x_{u,g} \leq y_g \quad \forall u \in \{0, \dots, N_{\text{users}}\}, g \in \{0, \dots, N_{\text{gNBs}}\}$$

- Egy adott bázisállomáshoz csatlakoztatott felhasználók összsávszélesség-igénye nem haladhatja meg a bázisállomás kapacitását:

$$\sum_{u=0}^{N_{\text{users}}} x_{u,g} \times b_u \leq c \quad \forall g \in \{0, \dots, N_{\text{gNBs}}\}$$

2.2.4 Célfüggvény

Az optimalizációs modell célja a következő függvény minimalizálása:

$$\begin{aligned} \min \quad & \left(\sum_{g=0}^{N_{\text{gNBs}}} \sum_{u=0}^{N_{\text{users}}} \text{distance_weight} \times x_{u,g} \times \text{distance}_{u,g} \right. \\ & + \sum_{g=0}^{N_{\text{gNBs}}} \text{power_cost_weight} \times y_g \times \text{power_cost}_g + \sum_{g=0}^{N_{\text{gNBs}}} \text{switching_cost} \times dy_g \\ & + \frac{1}{2} \sum_{g=0}^{N_{\text{gNBs}}} \sum_{u=0}^{N_{\text{users}}} \text{handover_cost} \times dx_{u,g} + \sum_{u=0}^{N_{\text{users}}} \text{blocking_cost} \times z1_u \\ & \left. + \sum_{u=0}^{N_{\text{users}}} \text{second_connection_cost} \times z2_u + \sum_{u=0}^{N_{\text{users}}} \text{third_connection_cost} \times z3_u \right) \end{aligned}$$

2.3 Szimuláció

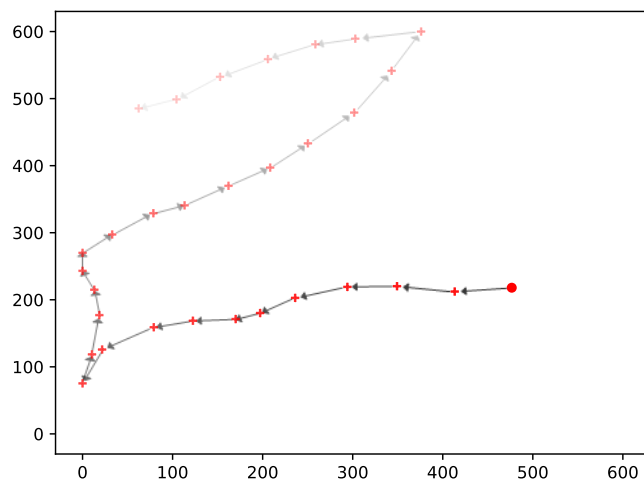
A felépített rendszer működése egyszerű: a bázisállomások fix helyen vannak elosztva a modellezett területen, míg a felhasználók a területen belül mozognak. Utóbbiak 0.5 másodpercenként lépnek egyet. 10 lépésenként megtörténik minden felhasználó valamely bázisállomáshoz rendelése a bevezetett költségek minimalizálása mellett. Ezek az összerendelések fennmaradnak újabb 10 lépésen keresztül, majd újból kiszámításra kerülnek már a felhasználók új helyzetét figyelembe véve.

A felhasználók mozgása egy előre meghatározott modell szerint történik, amire több példát is láttam egy csak mobilitással foglalkozó cikkben [4]. Minden egyes felhasználó rendelkezik egy mozgási iránnyal és sebességgel, és ezen mértékek alapján kerül kiszámításra az adott user új pozíciója a koordináta-rendszerben. A valóságot jobban tükröző mobilitás elérése érdekében egy lépés megtétele után nem sorsolódik egy új, véletlen irány és sebesség, hanem a korábban fennálló értékek kerülnek kombinálásra véletlen értékekkel. Az új irány számításának esetében az új érték tartománya úgy van meghatározva, hogy az eredeti iránytól egy kis, maximum 45 fokban térjen el, ezzel is utánozva egy ember viszonylag egyenes vonalban történő haladását.

$$\begin{aligned} \phi_u &= 0.4 \cdot \text{random} \left(\phi_{u,\text{last}} - \frac{\pi}{4}, \phi_{u,\text{last}} + \frac{\pi}{4} \right) + 0.6 \cdot \phi_{u,\text{last}} \\ v_u &= 0.4 \cdot \text{random}(1,20) + 0.6 \cdot v_{u,\text{last}} \end{aligned}$$

Amikor egy felhasználó a kijelölt terület széléhez ér, mozgási iránya úgy változik, hogy az elért szélétől elfele, a terület közepe fele kezdjen el mozogni a következő lépésben. Például, ha a user a terület bal széléhez ér, mozgási irányi így módosul:

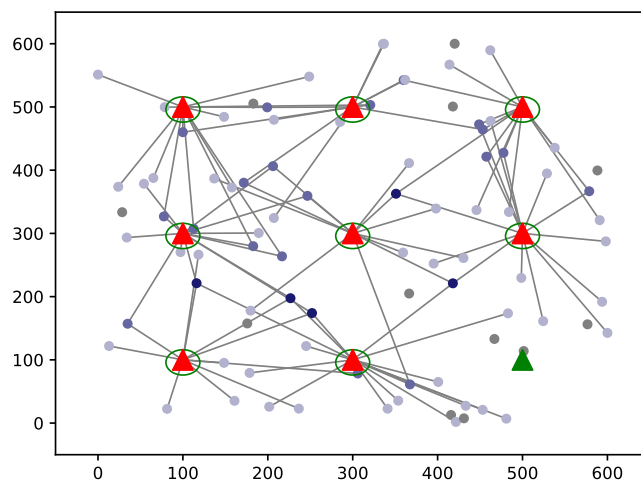
if $user_x_u = 0$: then $\phi_{u,last} = \text{random}(0, \pi)$



1. ábra. Felhasználók mozgásának modellezése

2.4 Ábrázolás

A feladat lényegi része volt a felhasználók és bázisállomások összerendelésének és a bázisállomások működési állapotának szemléltése is a szimuláció közben. Ezt a Matplotlib nevű Python könyvtárral³ valósítottam meg, ami egy egyszerű és jól értelmezhető interfészt biztosított a kijelölt területet reprezentáló koordináta-rendszer kirajzolásához, a bázisállomások és felhasználók helyzetének, állapotának megjelenítéséhez.



2. ábra. Összerendelések és állapotok ábrázolása

A háromszögek jelölik a bázisállomásokat, míg a pontok a felhasználókat. A piros színű bázisállomások bekapcsolt, a zöld színűek kikapcsolt állapotban vannak. A színes kör a bázisállomás legutóbbi állapotára utal: a zöld kör jelzi, ha az adott állomás éppen bekapcsolt, a piros kör pedig azt, ha éppen kikapcsolt állapotba került. Ahol nincs kör, ott nem történt állapotváltozás.

A felhasználókat kisebb pontok reprezentálják: minél sötétebb kék egy pont, annál több bázisállomással van kapcsolatban az adott user. Ha egyetlen állomáshoz sem lett hozzárendelve, akkor szürkére vált a pont. Magukra a konkrét összerendelésekre a vékony szürke vonalak utalnak.

2.5 Tesztelés változó költségekkel

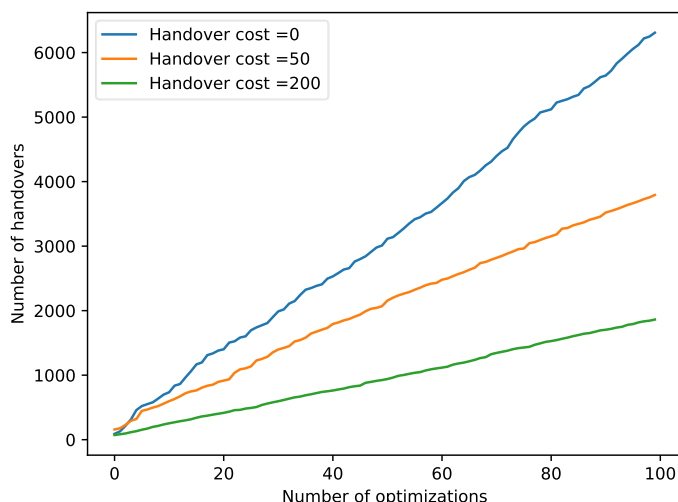
Az elkészült modell helyes működését végül tesztelni kellett, azaz megvizsgálni, hogy az optimalizálás eredménye hogyan változik a különböző költségek függvényében. Minden ilyen vizsgálatnál volt egy vagy két kijelölt költség, melyek különböző értékeinek beállításával mértem/számoltam egy olyan mértéket, ami nagy mértékben függött a vizsgált költségektől. A méréseket egy előre meghatározott, 1000 felhasználói lépésből álló szimuláción végeztem, 10 lépésenként egy optimalizációt végezve, melynek eredménye egy új, optimális összerendelés volt. A felhasználónkénti 1000 lépés meghatározásakor előálló, valamint a mérési adatokat a pandas nevű könyvtárral⁴ végeztem.

³ <https://matplotlib.org/>

⁴ <https://pypi.org/project/pandas/>

2.5.1 Handoverek számának mérése a *handover_cost* költség függvényében

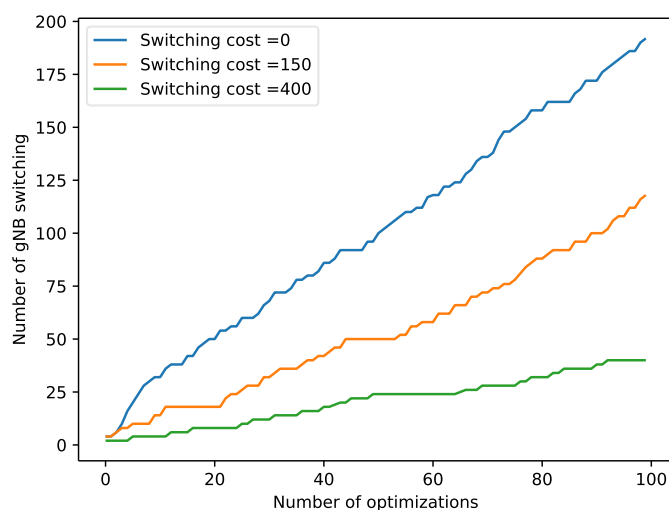
Ebben a mérésben a handover költség 3 különböző értékeinél számoltam össze az egész futás során, azaz 1000 lépés alatt a lezajlott handoverek számát. A modellben az számított handovernek, amikor egy felhasználóhoz rendelt bázisállomások száma megváltozott.



3. ábra. Handoverek száma a megfelelő költség függvényében

2.5.2 Bázisállomás be- és kikapcsolásinak mérése a *switching_cost* költség függvényében

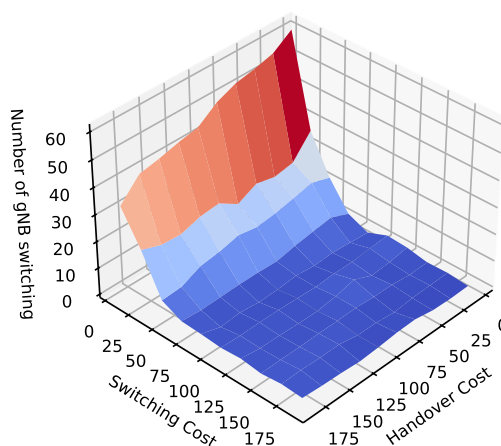
Az alábbi mérésben egy bázisállomás működési állapotának megváltozásához rendelt költség 3 különböző értékeinél számoltam össze a futás alatt történt összes állapotváltozást.



4. ábra. A bázisállomások ki-és bekapcsolásainak száma a megfelelő költség függvényében

2.5.3 Bázisállomás be- és kikapcsolásinak mérése a *switching_cost* és *handover_cost* költség függvényében

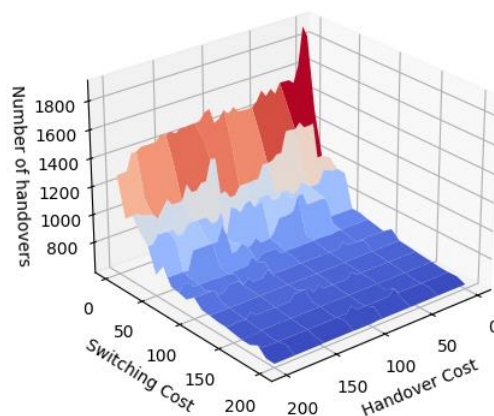
Az alábbi mérésben egy bázisállomások működési állapotainak megváltozásához rendelt költség 40 különböző értékénél, valamint a handover költség 20 különböző értékénél számoltam össze a futás alatt történt összes BS állapotváltozást. A szebb megjelenítés érdekében 2x4-es ablakmérettel simítottam a felületet, vagyis az eredményként kapott pontokat 8-as csoportokban átlagoltam és az átlagot jelenítettem meg.



5. ábra. A bázisállomások működési állapotváltozásainak száma a két megjelölt költség függvényében

2.5.4 Handoverek számának mérése a *switching_cost* és *handover_cost* költség függvényében

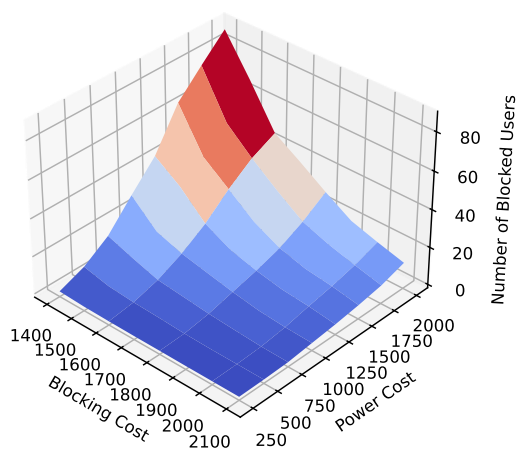
Az alábbi mérésben egy bázisállomás működési állapotának megváltozásához rendelt költség 41 különböző értékénél, valamint a handover költség 21 különböző értékénél számoltam össze a futás alatt történt összes handovert.



6. ábra. Handoverek száma a két megjelölt költség függvényében

2.5.5 „Elvesztett” felhasználók számának mérése *power_cost* és a *blocking_cost* költségek függvényében

Az alábbi mérésben a futás során „elvesztett” felhasználókat számoltam össze a kapcsolatvesztés büntetésének (*blocking_cost*) és a bázisállomások energiaellátásának költségének függvényében. Egy user elvesztettnek minősült, ha az optimalizálás nem rendelte egyetlen bázisállomáshoz sem.



7. ábra "Elvesztett" felhasználók száma a két megjelölt költség függvényében

2.2 Összefoglalás

Az idei félévben munkám fókuszra volt betekintést nyerni a felhasználók elosztásának feladatába egy mobilhálózatban, valamint megfogalmazni és különböző eszközökkel megoldani egy optimalizációs feladatot ebben a témakörben. Céлом egy, a valóságot minél jobban tükröző modell megfogalmazása, felépítése és ábrázolása volt, ami lehetőséget ad elmélyülni és megérteni a felhasználók bázisállomásokhoz való csatlakoztatásának komplex feladatát. Az új generációs mobilhálózatokban egyre fontosabbá válik a különböző típusú bázisállomások használata, mivel egyre sűrűbben kell lefedni kis területeket, ahol hatékonyan, takarékosan kell felhasználni az energiát, valamint megoldani egy eszköz több bázisállomáshoz való egyidejű csatlakozását. Emiatt releváns annak vizsgálata, hogy milyen jutalmakat és büntetéseket alkalmazunk, amikor értékeljük a felhasználók hálózathoz való csatlakoztatásának minőségét és hatékonyságát.

Az optimalizációs feladatot lineáris programozási módszerekkel oldottam meg. Ehhez az IBM CPLEX Optimization Studio által biztosított Python API-t használtam. Az adatok feldolgozásához és az ábrák kirajzolásához is Python könyvtárakat használtam. A tesztek során kirajzolt felületek megjelenítéséhez az ablakolás/simítás módszerét használtam a kiugró értékek eltüntetésére.

Nehézségek, eredmények
Következtetések?

3. Irodalom, és csatlakozó dokumentumok jegyzéke

A tanulmányozott irodalom jegyzéke:

- [1] Ericsson. (2020). *5G Wireless Access: An Overview*. Letöltve: 2023.9.28. Forrás: <https://www.ericsson.com/en/reports-and-papers/white-papers/5g-wireless-access-an-overview>.
- [2] A. Fayad, T. Cinkler and J. Rak, *5G Millimeter Wave Network Optimization: Dual Connectivity and Power Allocation Strategy*, in *IEEE Access*, vol. 11, pp. 82079-82094, 2023, doi: 10.1109/ACCESS.2023.3300960.
- [3] IBM Corporation. (2021). CPLEX User's Manual. IBM CPLEX Optimization Studio, Version 20.1.0. Letöltve: 2023.09.23. Forrás: <https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-users-manual>
- [4] T. Camp, J. Boleng and V. Davies, *A survey of mobility models for ad hoc network research*, *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483-502, 2002.
- [5]