

Kodolányi János Egyetem

SZAKDOLGOZAT

ANGYAL JÁNOS

ÜZEMMÉRNÖK-INFORMATIKUS

ALAPKÉPZÉSI SZAK

Budapest

2024.11.15.

**Kodolányi János Egyetem
Informatika Tanszék**

MI-alapú döntéstámogatás informatikai projekteken

AI-based decision support in IT projects

Konzulens: Dr. Pitlik László

Készítette: Angyal János

**ÜZEMMÉRNÖK-INFORMATIKUS
ALAPKÉPZÉSI SZAK**

Budapest

2024.11.15.

Tartalomjegyzék

1. Bevezetés	4
1.1. Kiindulási állapot.....	4
1.2. Motiváció.....	5
1.3. Feladat ismertetés röviden	6
1.4. Célcsoportok.....	7
1.5. Hasznosság	8
1.6. A dolgozat szerkezetéről.....	9
2. Szakirodalmi háttér.....	12
2.1. Rendszertervezés	12
2.1.1. A rendszertervezés és a dolgozat kapcsolata	14
2.1.2. A rendszertervezés és a tantárgyak kapcsolata	15
2.2. Rendszermodellezés	15
2.2.1. A rendszermodellezés és a dolgozat kapcsolata	16
2.2.2. A rendszermodellezés és a tantárgyak kapcsolata	16
2.3. Döntéstámogatás.....	17
2.3.1. Döntéstámogatás és a dolgozat kapcsolata	17
2.3.2. Döntéstámogatás, a mesterséges intelligencia és a tantárgyak kapcsolata.....	17
2.4. Mesterséges intelligencia.....	18
2.4.1. A mesterséges intelligencia szerepe a dolgozatban	19
2.4.2. ChatGPT	20
2.5. Programozás	20
2.5.1. A programozás és a dolgozat kapcsolata	22
2.5.2. A programozás és a tantárgyak kapcsolata	22
2.6. Vállalati ismeretek.....	22
2.6.1. Vállalati ismeretek és a dolgozat kapcsolata	26
2.6.2. Vállalati ismeretek és a tantárgyak kapcsolata	26
2.7. Jogi ismeretek.....	26
2.7.1. A jogi ismeretek és a dolgozat kapcsolata	27
2.7.2. A jogi ismeretek és a tantárgyak kapcsolata	28
2.8. Hálózatok.....	28
2.8.1. A hálózatok és a dolgozat kapcsolata	29
2.8.2. A hálózatok és a tantárgyak kapcsolata	29
2.9. Információs többletérték becslése	29

3. Szerverek ár-teljesítmény vizsgálata	30
3.1. Adatgyűjtés	30
3.2. Adatok feldolgozása	31
3.2.1. Objektumok	31
3.2.2. Attribútumok.....	32
3.2.3. Módszerek	34
3.2.4. COCO STD	34
3.2.5. COCO Y0	35
3.2.6. COCO STEP IX.....	35
3.3. Elemzés - 1	36
3.3.1. Sorszámozás	36
3.3.2. Naiv becslés.....	37
3.3.3. Adatok feldolgozása mesterséges intelligenciával	38
3.3.4. Eredmény, konklúzió	39
3.4. Elemzés - 2	40
3.4.1. Alternatív megoldás.....	40
3.4.2. Sorszámozás	43
3.4.3. Adatok feldolgozása mesterséges intelligenciával	43
3.4.4. Végeredmény, konklúzió	44
4. Informatikai rendszerterv-variánsok vizsgálata	47
4.1. Adatgyűjtés	47
4.2. Adatok feldolgozása	48
4.2.1. Objektumok	48
4.2.2. Attribútumok.....	48
4.2.3. Módszerek	52
4.3. Elemzés - 1	52
4.3.1. Sorszámozás	52
4.3.2. Naiv becslés.....	52
4.3.3. Adatok feldolgozása mesterséges intelligenciával	53
4.3.4. Eredmény, konklúzió	54
4.4. Elemzés - 2	55
4.4.1. Alternatív megoldás.....	55
4.4.2. Sorszámozás	57
4.4.3. Adatok feldolgozása mesterséges intelligenciával	58
4.4.4. Végeredmény, konklúzió	58

4.5. COCO STEP IX+	60
4.6. Informatikai rendezőszekrény tervezése, bemutatása	65
4.6.1. A rack szekrény méretének és típusának kiválasztása	66
4.6.2. Elhelyezés és hozzáférés	66
4.6.3. Terhelhetőség és rögzítés.....	66
4.6.4. Hűtés és szellőzés	66
4.6.5. Áramellátás és UPS	66
4.6.6. Kábelkezelés.....	67
4.6.7. Biztonság	67
4.6.8. Dokumentáció.....	67
4.6.9. Skálázhatóság	67
4.6.10. Költségvetés	67
4.6.11. Gyakorlati tippek	67
5. Szoftver bemutatása.....	71
5.1. Bevezetés.....	71
5.2. Módszertan és tervezés.....	71
5.3. Kódleírás.....	73
5.4. Programkód	74
5.5. Önenellenőrzés, tesztelés és eredmények.....	77
5.6. Összegzés és jövőbeli fejlesztések	79
6. Összefoglalás	85
6.1. Rezümé - HU	85
6.2. Rezümé - EN	86
6.3. Vita.....	87
6.4. Jövőkép.....	88
7. Források.....	89
8. Mellékletek	92
8.1. Rövidítések jegyzék.....	92
8.2. Definíciók jegyzék.....	93
8.3. Ábrajegyzék.....	95

1. Bevezetés

A bevezetés célja, hogy bemutassa a kutatás alapját és indíttatását, ismertesse annak célját, valamint magyarázatot adjon a dolgozat felépítésére. Részletezésre kerül, hogy egyes részterületek (pl. manuális modellezés, szoftverfejlesztés, gyakorlati alkalmazás) miért kerültek kiemelésre vagy mellőzésre, miért az adott arányban szerepelnek a dolgozatban, és miért az adott formában lettek kidolgozva. Továbbá, a bevezetés rávilágít a kutatás során alkalmazott módszerekre és azok indoklására, valamint felvázolja a dolgozat főbb kérdéseit és célkitűzéseit. Ismertetésre kerülnek az elméleti keretek (vö. szakirodalmi alapok), amelyek a kutatás alapját képezik, illetve bemutatásra kerül a téma relevanciája és aktuális jelentősége is. A dolgozat ezen része felvázolja a kutatás előzményeit, illetve a témában eddig elért eredményeket, hogy a jelen munka újdonságait és hozzájárulásait kontextusba helyezze. Emellett röviden kitér a dolgozat által vizsgált problémák gyakorlati jelentőségére és alkalmazhatóságára is, úgymint (vállalati és/vagy köz-) beszerzés és informatikai rendszertervezés.

1.1. Kiindulási állapot

Dolgozatom közvetlen célja, egy fajta mesterséges intelligencia (továbbiakban: MI) alapú manuális modellezés bemutatása, a hasonlóságelemzés módszertanának alkalmazásával. Ennek keretein belül, nagyvállalati szerverek- és informatikai rendszerterv-variánsok ár-teljesítményét vizsgálom, a Knuth-i elvet követve: „*Tudás/tudomány az, ami forráskódba átírható.*” Donald Knuth (1995.) *A=B* Előszó; Marko Petkovsek, Herbert S Wilf, Doron Zeilberger könyvében. Letöltve: <https://www.amazon.com/B-Marko-Petkovsek/dp/1568810636?asin=1568810636&revisionId=&format=4&depth=1> (2024.06.05) doi: ISBN 978-1568810638. Ezen alapulva elengedhetetlenül szükséges vizsgálni a közgazdasági vonatkozást is (vö. beszerzés), illetve az informatikai rendszertervezés (vö. rendszermodellezés) témakör elméleti és gyakorlati alkalmazása is előtérbe kerül a későbbiekben, valós nagyvállalati gyakorlati példákon keresztül, anonim módon. Mindezekon felül, bemutatásra kerül egy saját fejlesztésű szoftver is, amely félig automatizált megoldást kínál a modellek (vö. objektum-attribútum mátrix, továbbiakban OAM) vizsgálatára, quasi helyettünk elvégez bizonyos lépéseket (vö. a nyers adattábla értékeivel műveleteket végez, majd továbbítja az új adattáblát az MI felé), végül pedig visszaadja az eredményt, ami viszont már nem kerül automatikusan kiértékelésre. Az indirekt cél tehát kettős: egyrészt megvizsgálni és feltárni a téma elmélet és gyakorlati vonatkozását, másrészt a manuális alapozással megnyitni az utat (bárki számára) a teljes automatizálás felé, a részleges automatizálás lehetőségeinek felmutatása alapján.

Napjainkban szinte minden egyes cégnél, intézménynél jelen van az informatikai beszerzés/közbeszerzés, nem beszélve arról, hogy magánszemélyként is vásárlunk számítástechnikai eszközöket (pl. notebook, PC, router, okostelefon stb.), de szinte biztos, hogy soha nem objektív döntéseket hozunk - lévén ennek módszertana egyszerűsége ellenére nem része a közoktatásnak (még). Evolúciósan hajlamosak vagyunk érzelmek alapján dönteni, vagy marketinges meggyőző technikákra hallgatni, s ilyenkor minden esetben szubjektív döntés születik, mert más nem tud születni. Ezt elkerülve, két modellezési módszerrel bemutatom, hogyan tudunk hatékonyabban (vö. objektívebben) döntéseket hozni, az eszközbeszerzés és a rendszerterv-variánsok ár/teljesítmény-értékelése során.

1.2. Motiváció

Esetemben, korábbi egyetemi projekteken már vizsgáltam otthoni routerek ár-teljesítményét (vö. https://miau.my-x.hu/bprof/2023/osz/rapid/router_price_v1.xlsx). Illetve Google Trends alapú, tehát a keresések trendjét kimutató adatvagyon alapján router márkák népszerűségét is kutattam a magyarországi vármegyékre vetítve (vö. https://miau.my-x.hu/bprof/2023/osz/rapid/megyek_halozatieszkozok_v2.xlsx). Mindezekon felül, 2024. áprilisában, a török ISPEC IKSAD Institute által szervezett tudományos konferencián is részt vettem a „Szerverek ár-teljesítmény vizsgálata.” c. házi dolgozatommal (vö. <https://miau.my-x.hu/miau/311/server/ppt-server-price-performance-analysis.pptx>). Az alapötlet is innen származik, egyrészt jelenlegi munkám révén érdekelt vagyok a nagyvállalati informatikai beszerzésben, aminek elengedhetetlen része -lenne- az ár/teljesítmény vizsgálat, -elsődlegesen a számítástechnikai eszközök kapcsán. Ezen felül mind egyszerű- és komplex informatikai rendszertervek tervezése, elkészítése és beszerzése is feladataim közé tartozik. Felmerül tehát: **Mi lehet az a módszer, amivel objektívebb lehetne beszerzési döntéseket hozni? Mi lehet az a megoldás, ami mellőzi a rendszertervezési, rendszermodellezési szakkönyvek szómágikus és napjainkban elavult állításait, mint például: „A legtöbb szervezet nagyon fontosnak tartja, hogy a rendszertervezési folyamat általánosan elfogadott módszertanának megfelelően alakítsa ki információs rendszerét”.** (<https://www.webler.hu/tananyag/informaciosrendszerek.pdf> - 2024.06.07. - Információs rendszerek tervezésének módszertana - Komló Csaba)

Az agilis szempontokon túl az elméleti megközelítés is fontos szerepet játszik, amely során felmerül a kérdés, mennyire általánosítható, automatizálható és kontextusfüggetlen-e a szakdolgozat keretében vizsgált megoldás. Más szavakkal fogalmazva, azt vizsgálom, hogy az általam ismertetett megoldás alkalmas lehet-e keretrendszerként szolgálni hasonló

(ár/teljesítmény-értékelési) problémák kezelésére a jövőben az informatikai képzések során, a Knuth-féle szintet elérve/meghaladva. Ennek célja, hogy az emberi tudás forráskódba való átírásával egyre több területen legyen képes a gép (MI) olyan feladatokat ellátni, amelyekre korábban csak az ember volt képes.

A dolgozat tudományos háttere tehát összetett:

- A Knuth-i irányelv vonatkozásában („*Tudás/tudomány az ami forráskódba átírható.*” - vö. https://miau.my-x.hu/miau2009/index_tki.php3?_filterText0=*knuth) valamilyen módszerrel megoldást találni, esetemben a legjobb szerver és rendszerterv-variáns kiválasztására.
- Minél inkább „*context free*”, az az tartalomfüggetlen módszert találjunk (vö. GPS= general problem solving).
- Az önmagukban hatékony megoldások közül olyan lehetőségeket keresünk, amelyek magasabb hatékonyságot és információs többletértéket garantálnak, illetve amelyeket lehetőség szerint később automatizálni lehet.

1.3. Feladat ismertetés röviden

A COCO (Component-based Object Comparison for Objectivity) nevű online szakértői rendszer/modellező robot, azaz MI segítségével (vö. <https://miau.my-x.hu/myx-free/coco/index.html>), a hasonlóságelemzést felhasználva összehasonlítok 10 darab Dell felsőkategóriás nagyvállalati szervert, illetve 10 darab nagyvállalati informatikai rendszerterv-variánst. A vizsgálat során megállapítom a győztest, tehát mindkét esetben a legjobb ár-teljesítményű objektum (szervergép, ill. rendszerterv-variáns) kerül meghatározásra. Az összehasonlítás során valós adatokkal dolgozom. Továbbá a vizsgálat elvégzése után bemutatom a Python nyelvben íródott grafikus felülettel rendelkező Excel-OAM-COCO konverter szoftveremet is, amely a már korábban említett félig automatizált megoldást kínál a modellezési számítások megkönnyítéséhez.

Amennyiben felvetjük a kérdést, hogy lehet-e minden objektum másként egyforma, úgy kijelenthető, hogy naivnak tekinthető módszerekkel nem adható objektív válasz erre a kérdésre: gondolok itt például a súlyozott átlag számításokra vagy a nyerspontszámok/rangsorok alapján (vö. lineáris regresszió) kapott eredményekre. Ezekkel csak szubjektív válaszokat kapunk, mivel a lineáris regresszió becslése során a kapott értékek/adatok pontfelhőjére illesztünk egy egyenest, így a becslés tökéletlen. Bár ez névlegesen nem zárja ki a Knuth-féle elvet, a

kontextusfüggetlenséget és az automatizálási célokat, a szubjektivitás erősen veszélyeztetheti az információs többletérték létezését az objektív hasznosság hiánya miatt, ezért esett a választásom a hasonlóságelemzésre. Itt megjegyzem, hogy hasonló kérdéseket már vizsgált korábban például Bartók Patrik Róbert, korábbi üzemmérnök-informatikus hallgató „*Objektív döntésoptimalizálás mesterséges intelligencia bevonásával.*” című dolgozatában. Ő távoli elérést biztosító szoftvereket vizsgált a biztonság szempontjából, illetve egy iskolának tervezett belső hálózati infrastruktúrát, ahol olcsóbb kategóriás routereket helyezett előtérbe, mindkét a döntésoptimalizáció jegyében. Dolgozataink között nincs direkt kapcsolat, számomra a témaválasztás már azelőtt megszületett, hogy kapcsolatban kerültem volna Bartók Patrik Róbert témájával/dolgozatával. Az én dolgozatom egyrészt sokkal komplexebb modellezéseket tartalmaz, illetve a beszerzési aspektust is vizsgálja és ár-teljesítmény vizsgálatokat helyez előtérbe, nem mellőzve a nagyvállalati informatikai rendszertervezés, rendszermodellezés elméleti és gyakorlati példáját sem. A Bartók-féle dolgozat elméleti aspektusai is mások, mint ezen szakdolgozat esetén (vö. https://miau.my-x.hu/bprof/2021/ertekek_matematikaja_mathematics_of_values.docx), vagyis ott a fő kérdés, hogy vajon objektivizálható-e a „jó” fogalma. Azért döntöttem pontosan ezen témák vizsgálata mellett, mert jelenlegi munkám során fentiek kapcsolata elkerülhetetlen az informatikai projekteknél. (vö. beszerzés/közbeszerzés, ár-teljesítmény vizsgálat, rendszertervezés, objektív döntéshozatal).

1.4. Célcsoportok

Minden vállalati és cégforma, ahol informatikai beszerzést/közbeszerzést végeznek és szem előtt kívánják tartani az objektív döntéshozatalt, illetve a beszerzésért felelős személyek, ahol quasi érdekük/feladatuk lenne az ár-teljesítmény arányt (vö. ár-érték arányt) figyelembe venni. Továbbá magánszemélyek, akiket csak szimplán érdekelnek a szerverek, és az informatikai rendszerterv-variánsok, esetlegesen döntési módszerek és a megszokottól eltérő módon szeretnének döntést hozni.

Mindezekből következik az is, hogy a későbbiekben akár egy startup-potenciál is lehetne a teljes automatizáció, egy beszerzési döntéstámogató szoftver vonatkozásában. A program bekérhetné az adatokat, elvégezné a számításokat majd a végeredményt akár elemzi is a felhasználó számára, a manuális modellezési módszerrel ellentétben, ahol a kiértékelés a felhasználó feladata. Erre való törekvések már vannak, két évfolyamtársam is dolgozik árbecslő-, illetve film szimulátoron, tehát látható egyfajta jövőkép, valamint úgynevezett „megrendelői potenciál” a kutatásom iránt.

1.5. Hasznosság

Jelen esetben inkább a vállalkozásokat és a cégeket segíti ez a dolgozat, hiszen a vizsgált szerverek és rendszerterv-variánsok jellemzően közép- és nagyvállalatok informatikai beszerzői (vö. non-BOM beszerzők) paraméterezik, majd vásárolják még a cég számára, vélhetően az előnyöket-hátrányokat figyelembe véve (vö. ár-érték arány). Azonban, ha elvonatkoztatunk a szerverektől, és a rendszerterv-variánsoktól és egyfajta általános beszerzéstámogató megoldásként tekintünk a dolgozatra, akkor a KKV-k illetve a magánszemélyek számára is vélelmezhetően hasznos lehet, hiszen ez a „*context free*”, azaz tartalomfüggetlen megoldás lényege.

Az árelőny léte teszi legitimmé a feladatot, hiszen amíg a piaci ár nem arányos, azaz nem minden szerver vagy rendszerterv-variáns kerül(het) annyiba, mint amennyit teljesítmény oldalon ígér, addig mindig érdemes lesz feltárni az árelőnyt, tehát ami többet ad, quasi „jobbán megéri a pénzt”.

Tisztázó jelleggel, az árelőny mibenléte azonban objektum és attribútum függő, így tehát minden egyes OAM-ra más-más megoldás állhat elő.

Korábban már születtek hasonló kérdéskört vizsgáló szakdolgozatok és házi dolgozatok:

- Csík Balázs Béla. (2021.). Beruházási/vásárlási döntéseket segítő modellezés. URL: <https://miau.my-x.hu/miau/281/egypereggy.docx>
- Györgydeák Balázs Péter. (2021.). A Szoftverfejlesztés gazdaságtana. URL: https://miau.my-x.hu/miau/276/kje/szd1/a_szoftverfejleszttes_gazdasagtana.pdf
- Kerepesi Adrián. (2013.). Gyártástervezés és termelésirányítás folyamatainak optimalizálása az Országos Villamostávvezeték Zrt. gödi acélszerkezet-gyárában. URL: http://miau.my-x.hu/miau/182/etdk2013_kerepesi_adrian.pdf
- Kreidl Frigyes. (2016.). Döntéstámogatás a mezőgazdasági gépvásárlás területén. URL: http://miau.my-x.hu/miau/213/Kreidl_Frigyes_2016.pdf
- Troll Gergely. (2010.). Közbeszerzési folyamatok automatizálása (a kiírástól az eredményhirdetésig) hasonlóságelemzéssel. URL: <https://miau.my-x.hu/miau2009/index.php3?x=e0&string=troll.g>
- Varga-Dudás József. (2021.). A processzorok zavaros világa letisztult robotszemmel. URL: <https://miau.my-x.hu/miau/279/proc.docx>

1.6. A dolgozat szerkezetéről

A publikáció alapstruktúráját a Kodolányi János Egyetem által közölt szakdolgozat minta adja: https://www.kodolanyi.hu/konyvtar/images/tartalom/File/Honlapra/Feltoltes/szakedoli_minta.pdf

A terjedelmi korlát a konzulensemmel, Dr. Pitlik Lászlóval folytatott konzultáció alapján került meghatározásra.

A dolgozatban felhasznált internetes elérési címek, a dolgozat leadásának napján tesztelésre kerültek és ekkor mindegyik működött. Arra viszont nem tudok működési garanciát vállalni, hogy ezek meddig lesznek elérhetőek a jövőben, hiszen ezek mind ún. harmadik fél által üzemeltetett internetes honlapok, linkek vagy források.

A szakdolgozatban szekunder és primer kutatást egyaránt felhasználtam. A dolgozat első fejezetében bemutatom a szakdolgozat kiindulási állapotát, ismertetem a feladatot és a motivációt, továbbá szót ejtek a célcsoportról és a dolgozat hasznosságáról. Ennek a néhány oldalnak a célja, hogy az Olvasó megismerje a kutatásom célját, múltját, illetve a szakdolgozat szempontjából fontosabb témákat is megértse.

A dolgozat második fejezetében ismertetni kívánom a szakirodalmi háttérrel, a dolgozat tantárgyakhoz való kapcsolatát, illetve kötődését a mesterséges intelligenciához. Ebben a fejezetben főként a reál informatikai tantárgyak kerültek előtérbe, mivel a humán jellegű tárgyak csak kisebb részben kapcsolódnak a dolgozatomhoz.

A dolgozat harmadik fejezetében (saját kutatásként/fejlesztésként) ismertetem a szerver árteljesítmény vizsgálat lépéseit és eredményeit. A dolgozathoz csatolt összes Excel fájlban, a hivatkozott képletek angolul szerepelnek. Az Exceleekben lévő táblázatokban a számokat balra kellett volna rendezni, de ettől eltekintettem mert az kevésbé jól átlátható az esetemben, így a középre rendezést használtam. A tizedesjegyeket csak ott tüntettem fel, ahol az az eredmény/végeredmény szempontjából releváns, a többi helyen (a kerekítésből adódó, és nem releváns) tizedeshelyeket eltávolítottam, a jobb átláthatóság érdekében. Az ábrák tekintetében különös figyelmet fordítottam a különböző elvek tételes betartására. Az Excel táblázatos ábrák nagyrésze színezett/csíkozott, ezzel biztosítva a jobb átláthatóságot. Továbbá, minden ábra jelmagyarázattal ellátott.

Ebben a fejezetben a felhasznált módszerek részletes bemutatása mellett, a „naív”, tehát a nem optimalizált és a „gépi”, azaz az optimalizált eredmények is bemutatásra kerülnek, sok-sok ábrával, annak érdekében, hogy az Olvasó maradéktalanul megérthesse az összefüggéseket. A fejezet egyfajta felvezetést készít elő a következő fejezethez, ahol egy még részletesebb kutatást, illetve gyakorlati megvalósítást is be fogok mutatni.

A dolgozat negyedik fejezetében, ami egyben a legkomplexebb fejezet (saját kutatásként/fejlesztésként) ismertetem a rendszerterv-variánsok ár-teljesítmény vizsgálat lépéseit és eredményeit. Ebben a fejezetben a felhasznált módszerek már nem kerülnek részletes bemutatásra, hiszen az előző fejezetben ez megtörténik, viszont itt is elengedhetetlennek tartom a részletes, ábrákkal való bemutatást, ami mind-mind továbbra is az Olvasó számára biztosítja a megértést. Illetve ehhez szorosan kapcsolódóan bemutatom egy nagyvállalati informatikai rendezőszekrény gyakorlati megvalósítását. A kutatásban közben már felmerült, hogyan lehetne automatizálni az adatok begyűjtést „*context free*” (vö. tartalomfüggetlen) módon. Megpróbáltam a számomra releváns információk begyűjtését az internetről, jelen esetben szerverek és rendszertervek tulajdonságait és árait, és ezzel kapcsolatosan egyfajta automatizációs megoldáson elgondolkodni. Az nem kérdés, hogy mindenképpen automatizálni szükséges, azonban több kérdés, illetve probléma merült fel. Az egyik probléma az esetemben, hogy nem kaptam releváns Google keresési találatokat, továbbá megpróbáltam az adatokat beszerezni a legnagyobb magyar árösszehasonlító oldalról is (vö. www.arukereso.hu), de számomra nem volt biztosított egy felépített adatbázis a témában. Nagy valószínűség szerint a cégeknek nem is érdeke egy ilyet ingyen rendelkezésre bocsátani vagy közzétenni. Esetemben csak akkor gyűjthetőek be az adatok, ha direktbe felkeresem a hardver gyártókat, illetve a forgalmazó partnereket. A kutatásaimhoz szükséges adatokhoz olyan módon lehet csak hozzájutni, amikor én határozom meg, hogy milyen és mennyi objektummal és attribútummal kérjem be az adatokat, például egy intelligens kérdőív segítségével. Ezt például a Google Forms-sal el lehet készíteni, és a releváns partnereknek kiküldeni, így biztosan olyan adatokat kapunk vissza, ami utána azonnal felhasználható lesz. A világot abba az irányba kellene tolni, hogy az adatbázisok OAM-má alakíthatóak legyenek, tehát OAM képesek outputként rendelkezésre álljanak. Ezek tehát azon kérdések, amik elvezetnek az adatokhoz, de ennek mélyebb vizsgálata nem része a dolgozatomnak, ez már önmagában is egy önálló szakdolgozati téma lehetne.

A dolgozat ötödik fejezetében ismertetem az általam Python nyelven programozott szoftvert. Az Olvasó számára itt ki kell derülnön, hogy a szoftver viszonylag egyszerű, és nem fő témája

a dolgozatnak, csupán előkészít egy potenciális automatizálási lehetőséget. A szoftver lehetne többnyelvű, dolgozhatna egyszerre több fájlal, vagy adatbázis támogatást is kaphatna, de jelenleg csak egy viszonylag egyszerűbb problémára ad megoldást, a jövőkép azonban annál nagyobb benne. Illetve annak ellenére, hogy a programozás nem volt egy ideális szakdolgozati téma számomra, mindenképpen létre akartam hozni egy egyszerűbb programot, bizonyítva ezzel, hogy az oktatás során ezzel kapcsolatban is maradéktalanul el tudtam sajátítani a kellő ismereteket, mint például a strukturált gondolkodás vagy a minőségbiztosítási eljárások megismerése.

A hatodik fejezet tartalmazza az összefoglalást, a vitát és a jövőképet néhány oldalban. Rövid és tömörre terveztem ezt a fejezetet, hiszen a korábbi eredmények összefoglalása a cél, illetve az ezzel kapcsolatosan az Olvasóban felmerülő végső kérdések tisztázása is cél.

A hetedik és nyolcadik fejezet tartalmazza a dolgozat során felhasznált forrásokat, illetve a rövidítés-, definíciós- és ábrajegyzéket. Egyaránt felhasználtam magyar és külföldi forrásokat, a saját források mellett a legnagyobbbrészt a MY-X Team, Dr. Pitlik László, Pitlik Marcell, a konzulensem és csapata forrásait helyeztem előtérben, mivel ők is sok-sok éve foglalkozik a hasonlóságelemzés témakörével. A szakdolgozat ábrajegyzékében az átlagnál több, összesen 48db ábra található, amelyek célja a kutatás eredményeinek vizuális bemutatása, illetve az Olvasó számára a könnyebb érthetőség biztosítása.

2. Szakirodalmi háttér

Ebben a fejezetben bemutatom a dolgozat kulcsszavainak szakirodalmi előzményeit/vonatkozásait, valamint kifejtem, hogyan kapcsolódik a dolgozatban felhasznált tudás az egyetemi tanulmányokhoz és a tantárgyakhoz. Továbbá, részletezem a mesterséges intelligencia hatását és alkalmazását a dolgozatban, a BPROF szak követelményeinek megfelelően, mint például a klasszikus és induktív szakértői rendszerek- vagy a gépi tanulás szerepe az IT biztonságban, mint elsajátítandó ismeretanyag. Az induktív szakértői rendszer, akárcsak minden modell, egy olyan eljárás, amely lehetővé teszi a különféle tényezők közötti kapcsolatok, és az azokból következő állapotváltozások összefüggéseinek számítógépes modellezését. Ez a rendszer elsősorban szabályalapú, azaz logikai műveletekre épít, és jellemzően nem metrikus skálakon megjelenített, hanem átalakított és elvonatkoztatott adatokat használ az elemzéshez.

2.1. Rendszertervezés

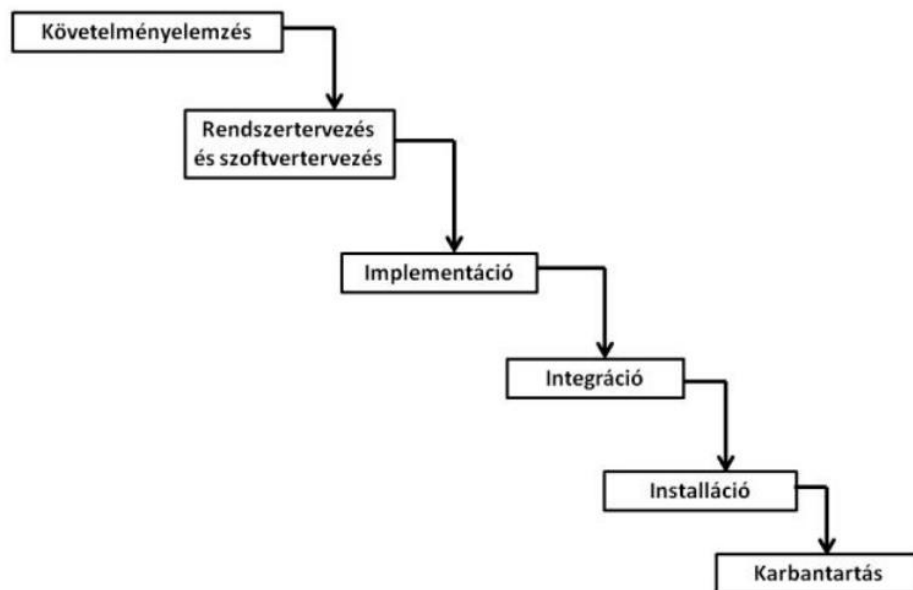
Az informatikai rendszertervezésről a ChatGPT az alábbiakat írta le: *„Az informatikai rendszertervezés kritikus szerepet játszik a közép- és nagyvállalatok működésében, mivel lehetővé teszi a különböző üzleti folyamatok hatékony és integrált működését. A rendszertervezés célja olyan informatikai infrastruktúra kialakítása, amely megfelel a vállalat stratégiájának, támogatja az üzleti célok elérését, és rugalmasságot biztosít a változó piaci környezethez való alkalmazkodásban. Az első lépés a rendszertervezésben az igényfelmérés, amely során felméri a vállalat jelenlegi és jövőbeli szükségleteit. Ez magában foglalja az üzleti folyamatok elemzését, a felhasználói követelmények azonosítását és a kulcsfontosságú teljesítménymutatók (KPI) meghatározását. A következő lépés a rendszerarchitektúra megtervezése, amely az informatikai infrastruktúra magas szintű struktúráját határozza meg, beleértve a hardver-, szoftver-, hálózati- és adatbázis-rendszereket. A rendszertervezés során különös figyelmet kell fordítani a biztonságra és a kockázatkezelésre. A vállalatoknak ki kell alakítaniuk olyan biztonsági mechanizmusokat, amelyek védelmet nyújtanak a külső és belső fenyegetésekkel szemben. Emellett fontos a megfelelőség biztosítása a jogszabályi és iparági előírásoknak, például az adatvédelmi törvényeknek. A rendszerintegráció szintén kulcsfontosságú elem, mivel a cégek gyakran több különböző rendszert használnak, amelyeknek együtt kell működniük. Az integráció lehetőséget nyújt a hatékony adatáramlásra és a folyamatok automatizálására, csökkentve ezzel a manuális munka és a hibák számát. Mindezekon felül, az agilis módszertanok (SCRUM) alkalmazása egyre népszerűbb a*

rendszertervezésben, mivel lehetővé teszi a gyors és rugalmas fejlesztést. Az iteratív tervezési ciklusok és a folyamatos visszajelzések révén a rendszer fokozatosan fejleszthető, így jobban igazodik a felhasználói igényekhez és a piaci változásokhoz. Végezetül, a rendszertervezés része a folyamatos karbantartás és fejlesztés is, amely biztosítja, hogy a rendszer hosszú távon is megfeleljen a vállalat igényeinek. A rendszerfrissítések, hibajavítások és a felhasználói támogatás mind hozzájárulnak a rendszer megbízhatóságához és hatékonyságához.” (<https://chatgpt.com/c/9adb115d-8838-411c-b524-28d4c494354e> - 2024.06.06 - Informatikai rendszertervezés nagyvállalati környezetben - ChatGPT összefoglalója alapján).

Itt fontos megjegyezni, hogy csak elméleti aspektusból kaptunk információt a ChatGPT-től. Egyetértek azzal, hogy a legelső lépés az igényfelmérés, mivel nagyon fontos a skálázhatóság egy vállalat tekintetében, tehát hogy a jövőre nézve is kiszolgálja a céget az adott informatikai rendszer, quasi legyen benne tartalék vagy bővíthető legyen, lehetőleg az életciklus végéig. Dolgozatomban a szerverek és az informatikai rendszerterv-variánsok tekintetében én is kiemeltem szem előtt tartom a skálázhatóságot. A ChatGPT szerint fontos még a teljesítménymutatók, azaz a KPI-k meghatározása is. Én személy szerint ez nem tudom ehhez a témához kapcsolni, nem tartom idevalónak ezt megemlíteni. Ez egy mérhető értéket jelent, amivel meg lehet határozni, hogy hogyan haladunk a kitűzött célunk felé, ennek pl. egy termelő cégnél van értelmezhető jelentése, ha csak nem arra gondol az MI, hogy az adott informatikai rendszer képes lesz-e kiszolgálni a termelést. Ezt viszont már nem boncolgattam tovább, ilyen mélységben ez nem fő témája a dolgozatnak.

A gyakorlati alkalmazást tekintve a korábbi tanulmányaim- és a szakirodalom szerint is, a rendszertervezés régóta a vízesés modellen alapszik, hiszen egyben ez az első publikált rendszertervezési/rendszerfejlesztési modell. (1. ábra). Erről Komló Csaba az alábbiakat írja: „A vízesés modell az első publikált rendszerfejlesztési modell, a hetvenes években Winston W. Royce publikálta „Managing the Development of Large Software Systems” címen. A vízesésmodell egy szekvenciális fejlesztési modell, amely jól elkülönülő lépésekre osztja a rendszertervezés egymás utáni lépcsőit (innen származik az elnevezés, bár az eredeti publikációban még nem így nevezték). A modell lényege, hogy a következő lépcsőfokra nem léphetünk, amíg az előző fázis be nem fejeződött. A gyakorlatban természetesen van lehetőség az előző lépcsőfokra visszatérni (ha funkcionálisan nem kielégítő az eredmény), illetve a lépcsőfokok részben átfedhetik egymást. A lépcsőfokok végét többek között a dokumentáció elkészülte is jelzi. Ha vissza kell térni egy korábbi lépcsőfokhoz, az gyakran az adott lépcsőfok (és ebből adódóan az összes többi) tervezésének újrakezdését jelentheti.”

(<https://www.webler.hu/tananyag/informaciosrendszerek.pdf> - 2024.06.07. - Információs rendszerek tervezésének módszertana - Komló Csaba).



1. ábra - Vizes modell

Forrás: <https://www.webler.hu/tananyag/informaciosrendszerek.pdf>

Meglátásom szerint a modell már eléggé elavult és önmagában nem is használják, hiszen az utóbbi években -ahogy a ChatGPT is említi, már előtérbe kerültek az agilis módszertanok (vö. SCRUM, Kanban), amelyek segítségével teljesen más szemlélettel állnak a rendszertervezéshez. Ezt támasztja alá Székely Blanka publikációja is: „Kutatási munkámban a két legelterjedtebb projekt módszertant (vizes és agilis) vizsgálom és konkrét szempontok alapján elemzem a különbségeket az általam korábban vezetett projekteken keresztül. Eddigi tapasztalataim alapján elmondható, hogy nem lehet csak az egyiket vagy csak a másikat alkalmazni, szükség van széles látókörre és mindkét módszertan alapos ismeretére ahhoz, hogy bármilyen projekt során a tudásunk szerinti legjobb lépést tegyük meg vagy döntést hozzuk meg egy-egy szituációban.”

(https://www.researchgate.net/publication/361326842_Projektmenedzsment_avagy_ami_sikerre_segiti_a_tanacsadot_vizeses_vagy_agilis_modszert_an_esetleg_mas - 2024.06.07. - Projektmenedzsment, avagy ami sikerre segíti a tanácsadót - vizes és vagy agilis módszertan, esetleg más? - Székely Blanka).

2.1.1. A rendszertervezés és a dolgozat kapcsolata

A dolgozat következő fejezeteiben szervereket és informatikai rendszerterv-variánsokat hasonlítok össze egy mesterséges intelligencia segítségével. Ezeknél a feladatoknál a hardverek

ismerete nélkülözhetetlen, figyelembe véve a kompatibilitási tényezőket, a skálázhatóságot vagy a különböző szabványok ismeretét. A dolgozat további részében az általam fejlesztett szoftvert is bemutatom, ami szintén a témához kapcsolható, hiszen habár csak egy egyszerűbb szoftverről van szó, de rendszerszinten ez is különböző részekre bontható. Egyik ilyen lehet a végrehajtó állomány (vö. executable), amely biztosítja a belépési pontot az alkalmazásba, illetve a programkönyvtárak is ide sorolhatóak (vö. library). amelyek az adott funkcionálisok gyűjteménye.

2.1.2. A rendszertervezés és a tantárgyak kapcsolata

A témában kiemelkedő segítséget nyújtott a „Rendszertervezés” tantárgy, amelynek keretében széleskörű ismeretekre tettem szert, mint például a rendszertervezési projektek menedzselése vagy az üzemeltetési tervek készítése, tehát elméleti- és gyakorlati téren egyaránt sokat segített az ismeretanyagok elsajátítása.

2.2. Rendszermodellezés

Az informatikai rendszermodellezés kulcsfontosságú a szakdolgozatom szempontjából, mivel különböző modell variánsokat fogok vizsgálni, illetve egy szoftvert is elkészítettem. Véleményem szerint a rendszermodellezés nem csupán informatikai aspektusban értendő, hiszen az élet számos egyéb területén is találkozhatunk modellekkel. Ezt igazolja a Kritikus Rendszerek Kutatócsoport 2024-es, tehát egészen friss publikációja is: *„Mind a természetben, mind az ember alkotta rendszerekben fellelhetők bizonyos szabályszerűségek: egyesek a rendszer elemei közötti kapcsolatokat, míg mások magukat az elemeket jellemzik. Bár sok mindenben eltér egy közlekedési hálózat, egy számítógép-hálózat, egy épület vagy egy város felépítése, a strukturális modellezéseszköztárával olyan „nyelvet” kapunk, amivel ezeket hasonlóan modellezhetjük. Egy rendszert gyakran úgy jellemezhetünk a legjobban, ha bizonyos elemeit meg- különböztetjük és leírjuk az ezek közötti kapcsolatokat.”* (<https://ftsrg.mit.bme.hu/remo-jegyzet/rendszermodellezes-mobile.pdf> - 2024.06.07. - Rendszermodellezés - Bergmann Gábor, Darvas Dániel, Molnár Vince, Szárnyas Gábor, Tóth Tamás).

Itt mindenképpen érdemes beszélni a „nyelv” szóról, amely idézőjelbe került. Informatikai vonatkozásában programozási nyelvekről beszélhetünk, ilyen például a C vagy a Python. Én a Python nyelvet használtam fel a szakdolgozatomhoz kapcsolt szoftver megalkotásához.

Véleményem szerint programozásnál először fejben kell összerakni a programot (vö. modellt), és meg kell határozni, hogy mit szeretnénk, bár ennek ellenére én is beleütköztem olyan problémába, amikor a programom nem azt csinálta amit szerettem volna, ezért kijavításra kényszerültem. Egy Python programozásról szóló könyv is foglalkozik ezzel: „*A programozáshoz szükséged van egy mentális modellre arról, hogy működik a program. Ha írtál egy programot, amely nem azt csinálja, mint amit elvársz tőle, akkor nagyon gyakran nem a programmal van a gond. A mentális modelled rossz. A legjobb módja a fejünkben lévő modell kijavításának az, hogy darabokra szedjük a programot (rendszerint ezek a függvények és a metódusok), és minden egyes komponenst függetlenül tesztelünk. Ha egyszer megtalálod az eltérést az elképzelésed és a valóság között, meg tudod oldani a problémát.*” (https://mtmi.unideb.hu/pluginfile.php/554/mod_resource/content/3/thinkcs3py3.pdf - 2024.06.07. - Hogyan gondolkodj úgy, mint egy informatikus: Tanulás Python 3 segítségével - Peter Wentworth, Jeffrey Elkner, Allen B. Downey; Chris Meyers).

Kérdésként felmerülhet, hogy a mai mesterséges intelligenciák valóban úgymond gondolkodnak, és elsőre hibátlan programot írnak? Személyes tapasztalatom alapján egyáltalán nem, csak akkor tudta maradéktalanul megírni a programot amit kértem tőle, amikor már teljesen körülírtam mit szeretnék, tehát amikor a „jó mentális modell” már elkészült. Itt kell megjegyeznem, hogy nem tudhatjuk, mikor is kapunk kellően helyes választ. A ChatGPT-vel való együttműködés tehát szakértőket érdemben támogat, laikusokat akár félre is vezethet.

2.2.1. A rendszermodellezés és a dolgozat kapcsolata

A rendszermodellezéssel, modellezéssel kapcsolatos tudás nagyon hasznos volt az OAM-ok vizsgálatánál, mivel a matematikai összefüggések megértése segíti a típusproblémák rutinszerű felismerését és a megoldási alternatívák kiértékelését, amelyek mind a szakmai önképzést segíti elő. A tervezés során szintén kiemelkedő fontossággal bírtak az előzőekben már említett rendszertervezési ismeretek.

2.2.2. A rendszermodellezés és a tantárgyak kapcsolata

A témában kiemelkedő segítséget nyújtott a „Rendszermodellezés” tantárgy, amelynek keretében széleskörű ismeretekre tettem szert, például az informatikai rendszerekkel szembeni követelmények szabatos megfogalmazására, működési környezetük és architektúrájuk modellezésére irányulón.

Felmerülhet az Olvasóban a kérdés, hogy mi a különbség a rendszertervezés és a rendszermodellezés között. A rendszertervezés a rendszer működésének és elemeinek konkrét kialakítására és megvalósítására koncentrál, míg a rendszermodellezés a rendszer logikai struktúrájának, folyamatai kapcsolatainak és működésének absztrakt ábrázolása, amely segíti a tervezést és elemzést. A rendszertervezés és a rendszermodellezés szorosan kapcsolódnak egymáshoz. A modellezés az első lépés, amelyben a rendszer struktúráját és logikáját térképezzük fel, és ez az alap szolgál a tervezéshez, ahol a rendszer részleteit és működését konkrétan kidolgozzák.

2.3. Döntéstámogatás

Ahogy a szakdolgozat címe is árulkodik, mesterséges intelligencia alapú döntéstámogató rendszer létrehozása volt a célom. Meglátásom szerint döntéstámogató rendszerekre azért van nagy szükség, hogy valóban objektív döntéseket tudjunk hozni, segítséggel, de a végső döntést továbbra is az ember végzi. A döntéstámogatásra döntéstámogató rendszereket veszünk igénybe. Erről Dr. Pitlik László konzulensem is ír: *„A döntéstámogató rendszer olyan számítógépre alapozott rendszer, mely adatok keresésével, rendszerezésével, mesterséges intelligencia használatával segíti a döntések meghozatalát. De nem képes önálló döntések meghozatalára, a döntéshozatal továbbra is emberi feladat marad.”* (<https://miau.myx.hu/mediawiki/index.php/Döntéstámogatás> - 2024.06.08. - Döntéstámogatás – Dr. Pitlik László, Pzsombor).

2.3.1. Döntéstámogatás és a dolgozat kapcsolata

A dolgozat célja egy korábban említett döntésoptimalizációs rendszer és módszer kidolgozása, ahogy azt a cím is egyértelműen jelzi. A rendszer biztosítja a keretet és a működési alapot, míg a módszer az adott probléma megoldására szolgáló lépéseket határozza meg.

2.3.2. Döntéstámogatás, a mesterséges intelligencia és a tantárgyak kapcsolata

A döntéstámogatással kapcsolatos kérdésekben véleményem szerint az MI alkalmazása egyértelműen a legegyszerűbb és leghatékonyabb választás. Képes hatalmas mennyiségű adatot rövid idő alatt elemezni és feldolgozni, ami lehetővé teszi a gyorsabb és pontosabb döntéshozatalt. Az MI rendszeresen frissíti az adatait, ezáltal mindig naprakész információk alapján hoz döntéseket, ami különösen fontos a dinamikusan változó környezetekben. Alkalmazása során az adatok elemzéséhez és a mintázatok felismeréséhez fejlett algoritmusokat és gépi tanulási technikákat használ. Ezek a technológiák lehetővé teszik, hogy

az MI rendszerek prediktív elemzéseket végezzenek, ami tovább növeli a döntéshozatal pontosságát és hatékonyságát. Az MI ezen képességei különösen értékesek lehetnek a vállalati környezetben, ahol a gyors és megalapozott döntéshozatal versenyelőnyt jelenthet. Az emberi szakértelem és az MI technológiák kombinációja biztosítja a legjobb eredményeket, mivel az emberi intuíció és kreativitás kiegészíti az MI rendszerek adatfeldolgozási képességeit.

Ugyanakkor fontos ismét hangsúlyozni, hogy a mesterséges intelligencia rendszerek önmagukban nem tévedhetetlenek. Az MI által generált eredmények és javaslatok mindig emberi felülvizsgálatot igényelnek annak érdekében, hogy biztosítsák az optimális döntések meghozatalát. Ezért elengedhetetlen, hogy az adatok gyűjtését és kezelését emberi szakértők végezzék, akik rendelkeznek alapvető számítástechnikai és adatbázis-kezelési ismeretekkel. Esetemben az „Adatbázisok I-II.” és a „Matematikai alapok” tantárgyak nagy segítséget nyújtottak ehhez.

2.4. Mesterséges intelligencia

Folytatva a gondolatmenetet, megjegyzem, hogy az MI már sokkal korábban létezett, mint ahogy azt egy átlag ember gondolná. Erről Dr. Pitlik László konzulensem is említést tesz az alábbiakban: *„A mesterséges intelligencia a számítástechnikának azon ága, amely különböző funkciókat betöltő, képességekkel rendelkező (pl.: érzékelés, gondolkodás stb.) és emberi értelmet igénylő gépek megalkotásával foglalkozik. A mesterséges intelligencia önálló diszciplínaként az 1950-es évek végén jött létre.”* (https://miau.my-x.hu/mediawiki/index.php/Mesterséges_intelligencia - 2024.06.08. - Mesterséges intelligencia – Dr. Pitlik László).

Vajon mondhatom-e tehát, hogy az emberi tudás átültetése egy robotba az MI? A témával Boros Tamás és Filippov Gábor az alábbiakat fogalmazta meg: *“Több mint húsz éve, hogy az IBM számítógépe, a Deep Blue megverte az aktuális sakkvilágbajnokot, Garri Kaszparovot. Ma már a mobiltelefonon futó sakkalkalmazások is könnyedén le tudnak győzni nagymestereket. Alig húsz évvel a Deep Blue diadala után, 2016-ban a Google AlphaGo alkalmazása megverte I Szedol dél-koreai gó nagymestert, akit a megelőző öt évben senkinek nem sikerült. (Szedol fel is hagyott profi gó pályafutásával.) Deep Blue még irgalmatlan géperejére támaszkodott: titka az óriásmemóriában rejlett, amely képessé tette arra, hogy minden lépés után lehetséges helyzetek millióit értékelje ki. Az AlphaGo ennél is többet tudott: a neurális hálózatok és a gépi tanulásmódszerével gondolkodni kezdett, és olyan játékokban nyert könnyedén, amelyet mi, büszkeemberek az intuíció és a kreativitás bevehetetlen terepének tartottunk. Az azóta eltelt négy év pedig ennek a „nagymesternek” a sokszoros meghaladásáról szólt: az AlphaGo*

„utódai”, az AlphaGo Zero és az AlphaZero már az embert tönkreverő ősök felett arattak elsőprőgyőzelmeket, alig néhány napnyi gépi tanulás után. A 2019 végén bemutatott MuZero pedig már úgy diadalmaskodott összes elődje felett, hogy előzetesen még a játékszabályokat sem tanították meg neki -mindenfajta készen kapott tudás nélkül képes kiszámítani a győzelemhez vezető lépéseket. Rég túl vagyunk tehát azon a kérdésen, hogy a gép felülmúlhatja-e az embert- a gépek évente múlják felül újra és újra az embert felülmúló gépeket.” (<https://www.szaktars.hu/osiris/view/boros-tamas-filippov-gabor-szerk-magyarorszag-2030-jovokep-a-magyaroknak-2020/?pg=0&layout=1> - 2024.06.08 - Magyarország 2030 - Jövőkép a magyaroknak - Boros Tamás, Filippov Gábor).

A mesterséges intelligencia a mindennapi életünk szerves részévé vált, többek között a személyre szabott ajánlások, a digitális asszisztensek, és az intelligens otthoni eszközök révén. Ez számos előnnyel jár, például növeli a hatékonyságot, megkönnyíti a rutinfeladatok elvégzését, és javítja a felhasználói élményt azáltal, hogy gyors és pontos információkat és szolgáltatásokat nyújt. Azt is fontos megjegyezni, hogy jelenleg nem minden MI rendelkezik naprakész adatokkal (2. ábra), ilyenek főként az ingyenes szolgáltatások (vö. ChatGPT 3.5).

Mikori a legfrissebb adatbázisod a mai napon? (2024.06.08.)



Az én adatbázisom szeptember 2023-ban frissült utoljára, így nem rendelkezem azutáni friss információkkal. Ha konkrét és aktuális adatokra van szükséged, javaslom, hogy keress megbízható forrásokat vagy híroldalakat, amelyek naprakész információkkal szolgálnak a kívánt témában.

🔊 📄 ↺ 🔍 ↻



2. ábra - ChatGPT 3.5 adatbázisának dátuma

Forrás: <https://www.chatgpt.com/>

2.4.1. A mesterséges intelligencia szerepe a dolgozatban

A dolgozatom során több alkalommal is igénybe vettem a mesterséges intelligencia segítségét. A szerverek ár-teljesítmény vizsgálatánál az adatbegyűjtés után megkérdeztem az OpenAI féle ChatGPT 3.5-öt, a rendszerterv-variánsok vizsgálatánál pedig már rendelkezésre állt az OpenAI féle ChatGPT 4o. Érdeklődésem tárgya az volt, hogy „ő” hogyan paraméterezne szervereket illetve rendszerterveket nagyvállalati környezetben, illetve egyetért-e az általam létrehozott variánsokkal, annak tekintetében, hogy azok nagyvállalati kategóriába sorolhatóak-e. Ennek

eredményét, a szakdolgozathoz szorosan kapcsolódó „*szerver_ar_telj_oam.xlsx*” és az „*it_rendszerterv_oam.xlsx*” Excel fájlok „*ChatGPT*” elnevezésű munkalap tartalmazza. A letöltési hivatkozás a 30. oldalon, a harmadik fő fejezet második bekezdésében található.

Korábbi egyetemi hallgatók is foglalkoztak már hasonló kérdéskörrel, és számos dolgozat és beadandó született, ahol ár/teljesítmény vizsgálatra került sor. Ezeket az alábbi linken lehet összegyűjtve megtekinteni: <https://miau.my-x.hu/miau2009/index.php3?x=e0&string=price>.

Mindezekon felül a szoftverem grafikus felület megalkotásában is részben segítségemre volt az OpenAI féle ChatGPT 3.5, mivel grafikus felület létrehozásában még nincs elegendő tapasztalatom.

Az adatbegyűjtés során létrehozott adattáblákhoz és a programozáshoz is elengedhetetlenek voltak az olyan alapismeretek, amelyeket a „Programozási alapelvek és módszertanok” illetve a „Hálózatok és számítógép architektúrák” tantárgyak hallgatása során szereztem meg. Ezen belül is kifejezett figyelmet fordítottam az alapvető matematikai és fizikai elvekre és módszerekre, illetve az adatbázisalapú rendszerek felépítésének megismerésére.

2.4.2. ChatGPT

Az OpenAI féle ChatGPT szerintem jelenleg a legismertebb és legelterjedtebb mesterséges intelligencia. Ezt támasztja alá a Wikipedia szócikke is: „*A ChatGPT 2022. november 30-án jelent meg prototípusként, és bemutatását követően rövid időn belül nagy érdeklődést váltott ki, szenzációvá vált. Alapvetően lelkes fogadtatásban részesítette mind a szakma, mind a közönség. Generált szövegét „lenyűgözően alaposnak” és „emberekéhez hasonlíthatónak”, sőt, „egy jó tanuló tudásszintjével felérőnek” nevezték. Magát a ChatGPT-t sokan „a valaha volt legjobb MI chatbotként” méltatták. Elon Musk szerint „a ChatGPT ijesztően jó, már nem vagyunk messze a fényes erő MI-től.” A tesztelés során generált részletes válaszainak és számos tudástartományt átfogó artikulált feleleteinek köszönhetően bemutatkozásának jelentőségét sokan az iPhone 2007-es debütálásához hasonlították. Ugyanakkor a válaszokból kitűnő tényyszerű pontosság terén előforduló hiányosságokat is szóvá tette több szakember.*” (<https://hu.wikipedia.org/wiki/ChatGPT> - 2024.06.08 - Wikipedia, ChatGPT című szócikk).

2.5. Programozás

Az előzőekben érintett szakirodalmi témák nagy segítségemre voltak abban, hogy a számítógépes programom elkészüljön. A szoftverem összességében egy döntéstámogató rendszer (vö. MI) egyfajta manuális modellezése, csupán néhány részfolyamatot fél

automatizál, ilyen funkciók például az átalakítás, a feltöltés, illetve a beküldés. Ettől függetlenül, véleményem szerint a szoftver tökéletes alap a teljes automatizáció megteremtéséhez, a későbbiekben.

De mi is az a programozás? Én a legegyszerűbben úgy tudnám leírni, hogy olyan, az ember által is értelmezhető kód, amivel a számítógép egy adott feladatot elvégez. Ezzel a megfogalmazással a következő szakirodalomban is találkozhatunk: *„A számítógép számára a feladat meghatározását programozásnak nevezzük. A programozás valamilyen programozási nyelv segítségével történik. A programozási nyelv a számítástechnikában használt olyan, az ember által is olvasható és értelmezhető utasítások sorozata, amivel közvetlenül, vagy közvetve (például gépi kódra fordítás után) közölhetjük a számítógéppel egy adott feladat elvégzésének módját.”* (<https://gyires.inf.unideb.hu/KMITT/b02/ch09s03.html> - 2024.06.08. - Programozás alapjai - Dr. Nyakóné dr. Juhász Katalin, Dr. Terdik György, Biró Piroska, Dr. Kátai Zoltán).

Megemlíteném, hogy a programnyelvek is fejlődtek az idők során, jelenleg hat generációra bonthatóak a programnyelvek. Sajnos a szakirodalom általában csak a negyedik generációig foglalkozik a témával, a többi annyira új, hogy a szakdolgozat írás pillanatában nemtaláltam komoly publikációt a témában. Illetve, ami még érdekes kérdés, hogy hová sorolják a Python nyelvet, amit én is használtam. Egyesek szerint harmadik (3GL), míg mások szerint a negyedik (4GL) generációba sorolható. Ezt említi az University of Essex egyik hallgatója 2019-ben készült szakdolgozatában: *„Fourth-generation programming languages (4GLs) are DSLs that provide a higher level of abstraction of the computer hardware than the the third-generation programming languages they replaced. C, C++, C#, Java are examples of 3GLs. Some languages like Python are considered to be advanced third-generation languages because they can be extended with 4GL libraries to perform additional tasks. 4GLs offer more semantic properties and greater implementation power.”* (https://repository.essex.ac.uk/32131/1/A_Syntactical_Reverse_Engineering_Approach_to_4_GLs_redacted.pdf 2024.06.08. - A Syntactical Reverse Engineering Approach to Fourth-Generation Programming Languages Using Formal Methods - Majd Abdullatif Zohri Yafi).

Az ötödik és hatodik generáció az leginkább az MI-hez köthető, amikor quasi a fejlesztő emberi nyelven beszélget az MI-vel és gyakorlatilag instrukciókat adva létrejön a program, vagy a program egy részlete, illetve több dimenziós vizualizáció jön létre úgynevezett „low-code/no-code”, tehát kevés kód/kód nélküli alapon. Meglátásom szerint a jövő mindenképpen ez, és bár nem jelenthetjük ki egyelőre, hogy az MI elveszi a programozók munkáját, de egy-egy

részfeladat vagy kisebb szoftverek/weboldalak/blog oldalak elkészítéséhez már nem feltétlen kell egy programozót megfizetnünk.

2.5.1. A programozás és a dolgozat kapcsolata

A szakdolgozat egyik része a korábban már említett Python nyelven megírt program, amelyet egy későbbi fejezetben fogok részletesen bemutatni.

2.5.2. A programozás és a tantárgyak kapcsolata

A program megírásához elengedhetetlen segítség volt a „Programozási alapelvek és módszertanok”, és a „Programozás I-II-III.” tárgyakon elhangzottak. Nekem legfőképp a strukturált gondolkodás megtanulásához volt kiemelkedő segítség.

2.6. Vállalati ismeretek

A dolgozat szempontjából azt gondolom, hogy elengedhetetlen a közgazdasági vonatkozását is megvizsgálni, itt legfőképp a beszerzési- és az alapvető vállalati ismereteket szükséges szem előtt tartani. Egy nagyvállalati környezetben ezek elengedhetetlenek, főképp olyan informatikus munkakörben dolgozók számára, akik informatikai beszerzést/közbeszerzést is végeznek. Én személy szerint ilyen munkakörben dolgozom egy nagyvállalatnál és már több éve, így nem idegen számomra a téma, az évek alatt nagyon sok tapasztalatot gyűjtöttem. A következőkben a „normál” vállalati beszerzéssel foglalkozom, a közbeszerzéssel nem, mivel nincs direkt kapcsolata a dolgozatommal, illetve ez a beszerzés egy speciális ága, amely állami tulajdonú cégeknél jellemző. A vállalati beszerzésről Vörösmarty Gyöngyi és Tátrai Tünde így vélekedik:

„A vállalati beszerzés céljai általában a következők:

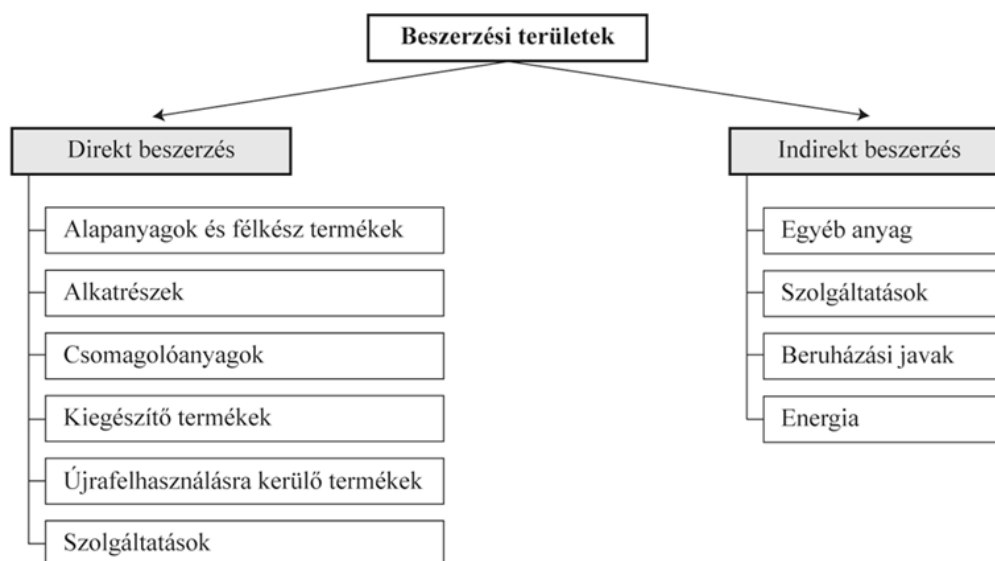
- *Költségek minimalizálása.*
- *Ellátási lánc kockázatainak minimalizálása.*
- *Belső igénylők/vásárlók elégedettségének maximalizálása.*
- *Beszerzési folyamatok optimalizálása.*
- *Beszállítói lehetőségek racionalizálása.*

A beszerzés teljesítményének mérhetőnek kell lenni. A vállalati beszerzés normál értelmezése az ellátásbiztonságra való törekvés. Megfelelő időben, mennyiségben, helyen, minőségben a megfelelő áron szemponttal kiegészítve az adott beszerzés megfelelőségére segítséget ad a

mutatók meghatározásában. Így például a minőség, a szállítási pontosság, az ár mérésére, ha nem is általános érvényű, de az adott szituációhoz alkalmazható mutatókat könnyű találni.”

Vörösmarty Gyöngyi-Tátrai Tünde (2019): Beszerzés. Wolters Kluwer Hungary Kft. <https://doi.org/10.55413/9789632958453> Letöltve: https://mersz.hu/dokumentum/YOV1723_15/#YOV1723_13_p2 (2024. 06. 08.).

Beszerzési területeket tekintve megkülönböztetünk direkt és indirekt beszerzést. Egyszerűen megfogalmazva egy nagyvállalati gyártó cégre vetítve, a direkt beszerzés az a terület, ami a termeléshez szükséges, az indirekt pedig minden más. Ezt a 3. ábra szemlélteti.



3. ábra - Beszerzési területek

Forrás: <https://mersz.hu/vorosmarty-tatrai-beszerzes-2017>

Megkülönböztetünk még lokális és globális beszerzést is, ami nem azt jelenti, hogy egy cég külföldről vásárol, sokkal inkább multinacionális vállalatokra jellemző stratégia. Ezt Vörösmarty Gyöngyi és Tátrai Tünde az alábbiakban taglalja: „A globális beszerzés (global sourcing) ennek értelmében a beszerzési tárgyak közös részének, a folyamatoknak, a technológiának és a beszállítóknak az integrálását és koordinációját jelenti. A lokális beszerzés azokat a feladatokat jelenti, amelyeket a vállalat nem centralizált, a helyi beszerzési szervezet (nem egyszer a helyi beszállítói lehetőségekre építve) lát el.” Vörösmarty Gyöngyi-Tátrai Tünde (2019): Beszerzés. Wolters Kluwer Hungary Kft. <https://doi.org/10.55413/9789632958453> Letöltve: https://mersz.hu/dokumentum/YOV1723_60/#YOV1723_58_p1 (2024. 06. 08.).

A beszerzések tekintetében kiemelt feladat az árazás, ár-teljesítmény arány és ezeknek az elemzése. A dolgozat bevezetésében már említettem, hogy általában szubjektív döntés születik, és erre keresem a megoldást. **Vajon lehet-e minden objektum másképp egyforma?** (vö. Mindenki másképp egyforma - Mérő László).

A felhasznált szakirodalom tárgyalja az ár értelmezését befolyásoló tényezőket, illetve az árelemzésről is szó esik. Itt mindenképpen kiemelném, hogy a szakirodalom is említ modelleket az árelemzés vonatkozásában, csakúgy ahogy én is utaltam már erre a dolgozat legelején (vö. 1.1. Kiindulási állapot). Bár hagyományos értelemben véve a két említés nem ugyanolyan modellezési eljárást takar, viszont mindenképpen összefüggésben vannak (vö. modellezés, rendszermodellezés). Erről röviden, így ejt néhány szót Vörösmarty Gyöngyi és Tátrai Tünde: *„Az árelemzés hagyományos módszerei az árképzés verseny alapú és költség alapú modelljéhez kötődnek. Ezek értelmében a beszerzést értékelhetjük más árakhoz viszonyítva, illetve költség alapon.”* Vörösmarty Gyöngyi-Tátrai Tünde (2019): Beszerzés. Wolters Kluwer Hungary Kft.. <https://doi.org/10.55413/9789632958453> Letöltve: https://mersz.hu/dokumentum/YOV1723_128/#YOV1723_126_p1 (2024. 06. 08.).

A beszerzés egyik leglényegesebb aspektusa az életciklus költségelemzés. Ez azt jelenti, hogy a beszerzett termék (pl. szerver) teljes életciklusa alatt jelentkező minden költséget igyekszik számszerűsíteni. Könyvelési szempontból a nagyobb értékű számítástechnikai eszközök tárgyi eszköznek minősülnek, amelyeket általában 3 év az értékcsökkenési leírása, tehát ez azt jelenti, hogy a költsége a cégnél több év alatt jelenik meg (vö. nem a beszerzés évében). Sőt, amennyiben nem tárgyévi elején szereztük be, még arányosítani is szükséges. Minden egyéb, ami nem tárgyi eszköz, az „cost” azaz költség. Ennek a nyilvántartása úgymond nem szigorú számadású, tehát lazábban lehet kezelni (vö. nem kell leltározni). Az életciklus költségelemzés egy bonyolult eljárás, amelynél több szempontot kell figyelembe venni. Legegyszerűbb példa, amikor vásárol a vállalat egy szerver, viszont azt használja a gyártás, a raktár és az iroda is, melyeknek külön-külön költséghelye van, jellemzően itt arányos szétosztást kell alkalmazni. A szakirodalomban Vörösmarty Gyöngyi és Tátrai Tünde ezt a következőképpen határozza meg:

„Az életciklus-költségelemzés lépései általában a következők:

- *Az összes érintett költségcsoport meghatározása.*
- *Az egyes költségcsoportok időszakonkénti nagyságának meghatározása. (Ez, amennyiben nem rendelkezünk korábbi időszakok adataival, számos nehézségbe*

ütközhet, kérhetünk referenciákat, használhatunk - igaz, alkalmanként bizonytalansággal - katalógus adatokat.)

- Az egyes költségek azonos időszakra diszkontálása.
- Az adatok összevetése, az alternatívák (például eltérő energiaár-emelkedési adatokkal számított változatok) összehasonlítása.”

Vörösmarty Gyöngyi-Tátrai Tünde (2019): Beszerzés. Wolters Kluwer Hungary Kft.. <https://doi.org/10.55413/9789632958453> Letöltve: https://mersz.hu/dokumentum/YOV1723_134/#YOV1723_132_p6 (2024. 06. 08.).

Egy gyakorlati példát említve, a jelenlegi munkahelyemen az SAP integrált vállalatirányítási rendszer, illetve minden hozzákapcsolódó támogatási költség, a globális és egyben központosított (vö. headquarter - HQ általi) beszerzés által menedzselt. Nagyjából a magyar vállalatra az összeköltség ~7%-a esik, amit a cég globális központja kiszámláz felénk, így vállalaton belüli fizetési tranzakció történik. Ez a példa még az egyszerűbbek közül való, hiszen itt egyszerű, létszamarányos számítás alapján történik a felosztás. A fent említett serveres példa azért az sokkal bonyolultabb számításokat igényel egy vállalatnál.

Végezetül az úgynevezett elektronikus beszerzési rendszerekre is kitérek. A legismertebb az SAP Business Network (vö. korábban ARIBA). Az ehhez hasonló szoftverek már régen túllépnek a „normál” beszerzés folyamatán, és foglalkozik a fizetéssel, leltározással, katalógusokkal, tervezéssel és elemzéssel is. Kiemelem a katalógusokat, amellyel Vörösmarty Gyöngyi és Tátrai Tünde a szakirodalomban is foglalkozik: *„Legyen szó egy multinacionális vállalat centralizált beszerzéseiről eredményeként létrejövő keretszerződésekről, a munkavállalók vagy leányvállalatok által az adott termékeket, árukat elérve a katalógus nemcsak az összehasonlításra, hanem a megrendelésre, a közvetlen információhoz jutásra, hanem olyan beszerzési statisztikák generálására is képes lehet, amely további visszacsatolást képes nyújtani a katalóguson kívüli keretszerződések megkötésére, a beszerzés lebonyolítására. Ezek eredményeként pedig a beszerzett termékek katalógusban történő elhelyezése, felhasználóbarát kínálata tovább képes javítani a beszerzés hatékonyságát.”* Vörösmarty Gyöngyi-Tátrai Tünde (2019): Beszerzés. Wolters Kluwer Hungary Kft. <https://doi.org/10.55413/9789632958453> Letöltve: https://mersz.hu/dokumentum/YOV1723_173/#YOV1723_171_p11 (2024. 06. 08.).

Fentiekből következik, hogy az informatikai projektek döntéstámogatása nem egy egyszerű feladat, azonban a mesterséges intelligencia itt talán kevésbé van jelen. A dolgozat készítésekor

nem áll rendelkezésemre információ arról, hogy a világ vezető vállalatirányítási szoftverét fejlesztő SAP vállalat rendelkezne konkrét, kifejezetten beszerzéstámogató megoldással. Habár, egyéb MI alapú integrált megoldások már jelen vannak, de ezek jellemzően nem konkrétan az objektív döntéstámogatást segítik. A dolgozat írásakor az alábbi weboldalon szemlélteti az SAP, hogy milyen támogatást nyújt a beszerzéseket illetően. SAP (2024) <https://www.sap.com/products/artificial-intelligence/procurement.html>.

2.6.1. Vállalati ismeretek és a dolgozat kapcsolata

Összegzésként megállapítom, hogy az alapvető beszerzési és vállalati ismeretek elengedhetetlenek a rendszertervezés és rendszermodellezés során, mivel ezek az ismeretek biztosítják a szervezet működési folyamatai közötti összefüggések pontos megértését, valamint az optimális megoldások kialakítását.

2.6.2. Vállalati ismeretek és a tantárgyak kapcsolata

A témában kiemelkedő segítséget nyújtottak a „Vállalati gazdaságtan” és a „Vezetési és vállalkozási ismeretek” tantárgyak, amelynek keretében megtanultam munkahelyem erőforrásaival gazdálkodni, illetve megismertem a vállalatok funkcionális területeit és az azok közötti összefüggéseket.

2.7. Jogi ismeretek

A korábbiakban megismert szakirodalomon túl véleményem szerint szükség lehet alapvető jogi ismeretekre is. Amennyiben egy közép- és nagyvállalat szemszögéből közelítem meg, egy informatikai beszerzőnek a következőkben felsorolt törvényekkel kell tisztában lennie. Mindenhol indoklom, hogy miért gondolom ezt.

Fontos a munkajogi szabályozás (2012. évi I. törvény, a munka törvénykönyve) ismerete, mivel a beszerzési folyamatokban gyakran szükség van a munkavállalók bevonására és azok jogainak biztosítására.

Az adatvédelmi szabályozás (2011. évi CXII. törvény, az információs önrendelkezési jogról és az információszabadságról) ismerete is elengedhetetlen, különösen az érzékeny és személyes adatok kezelése során.

A számviteli törvény (2000. évi C. törvény) ismerete szintén fontos, mivel ez a beszerzések pénzügyi tervezéséhez és elszámolásához kapcsolódó előírásokat tartalmazza.

Tisztában kell lenni a fogyasztóvédelmi törvénnyel (1997. évi CLV. törvény) is, amely a beszerzett termékek és szolgáltatások minőségével és a vevői jogokkal kapcsolatos előírásokat tartalmazza.

Elengedhetetlen a szerzői jogi törvény ismerete (1999. évi LXXVI. törvény), különösen a szoftverlicenckre és az iparjogvédelemre vonatkozó rendelkezéseket, hogy a beszerzéseknél megfelelően kezelje az esetleges szellemi tulajdonjogokat, itt legfőképp a szoftverekre kell gondoljunk és azon belül is kiemelten fontos a szoftverlicenz szerződés (EULA). Budai Balázs Benjámin, az egyik legújabb, 2024-es könyvében ír erről: *„A szoftverek szellemi termékek, szerzői jogi védelem alatt állnak. Illegális beszerzésük tehát etikai és büntetőjogi vétség. A magyar jogszabályok szerint egy szoftver akkor legális, ha rendelkezünk a szoftver gyártójától származó engedéllyel, amelyben a gyártó feljogosít a szoftver használatára, és tudjuk igazolni annak származását, például számlával, szerződéssel, átvételi elismervénnyel. A szoftver licencszerződéssel jogosítja fel a szerzői jog tulajdonosa szellemi termékének használatára a vásárlót. Egy adott szoftver esetében a végfelhasználói licencszerződés (EULA) határozza meg a szerzői jog tulajdonosa által megengedett szoftverhasználat feltételeit (használható példányszámot, időtartamot, felhasználói kört, jogokat, korlátozásokat, jogorvoslati lehetőségeket).”* Budai Balázs Benjámin (szerk.) (2024): Digitális készségfejlesztés. Budapest: Akadémiai Kiadó - Ludovika Egyetemi Kiadó. <https://doi.org/10.1556/9789634549611> Letöltve: https://mersz.hu/dokumentum/m1151digk_54/#m1151digk_52_p14 (2024. 06. 08.).

Gyakorlati tapasztalatom szerint egyébként a beszerzések egyre jobban automatizáltak, például a korábban már említett SAP vállalatirányítási rendszer is képes rá, hogy bizonyos fogysokat figyelve utánrendelést adjon le automatikusan a beszállítók felé. Ez az automatizáció is rávilágít arra, hogy egyre jobban teret fog nyerni az MI, ahogy az előző fejezetben már kitértem erre az SAP-t illetően.

2.7.1. A jogi ismeretek és a dolgozat kapcsolata

A szakdolgozat és a jogi ismeretek között szoros kapcsolat lehet, esetemben legfőképp a szerzői jog és a szellemi tulajdonjog. A dolgozatnak plágiummentesnek kell lennie, és minden felhasznált irodalom, ábra és idézet pontos hivatkozásokat kíván, sőt kötelező a szakdolgozat szabályzat szerint.

2.7.2. A jogi ismeretek és a tantárgyak kapcsolata

A témában nagy segítséget nyújtottak a „Szakterületi jogi ismeretek” tárgy keretében elsajátított elméleti anyagok, megértettem a szakmai etikai elveit és vonatkozásait, illetve a napi munkavégzésem során törekszem ezek betartására és betartatására.

2.8. Hálózatok

Az informatikai hálózatok alapvető ismerete elengedhetetlen a szakdolgozatom negyedik fejezet szempontjából, ahol informatikai rendszerterveket vizsgállok, ami elég sok hálózati eszközt tartalmaz. De hogy miért is van szükség informatikai hálózatokra? Ezt a szegedi Krudy Gyula Szakképző Iskola weblapján talált forrás nagyon jól leírja: *„Egy cégnél például, ahol nagyszámú számítógépekkel dolgoznak, előbb utóbb megjelenik az igény arra, hogy a fizikailag különböző helyen lévő (irodákban, raktárban, stb.), elszigetelt gépeket összekössék egymással. Itt lehet szó erőforrás-megosztásról (re-source sharing), a cél pedig az, hogy minden program, eszköz és legfőképpen adat mindenki számára elérhető legyen a hálózaton, tekintet nélkül az erőforrás és a felhasználó fizikai helyére. Egy nyilvánvaló példa erre, amikor irodai dolgozók egy csoportja megosztva használ egy nyomtatót. Egyiküknek sincs szüksége saját nyomtatóra, és egy nagy kapacitású hálózati nyomtató gyakran olcsóbb, gyorsabb, és könnyebben karbantartható, mint nagyszámú egyedi nyomtató.”* (https://szaszak.krudy-szeged.hu/static/kruszam/szoftuz/halozati_ismeretek1.pdf - 2024.06.08. - Hálózati ismeretek).

Itt érdemes az erőforrás azaz a „re-source sharing” szót megvizsgálni milyen kontextusban értelmezhető? Jelen esetben az erőforrás megosztása alatt több mindent értünk. Ez lehet a szerver, lehet a nyomtató, de adott esetben bizonyos hálózati eszközök is, például a vezeték nélküli internet hozzáférés is. Mindezekon felül az erőforrás lehet elsődleges és másodlagos is. Erről Dr. Pitlik László is publikált az alábbiak szerint: *"Az informatika elsődleges erőforrásai az adatok, (pl. matematikai, szervezési) módszerek, valamint a technológia (az IT). Másodlagos erőforrásai az egyén (az ún. „szakértői” tudás) és az emberek szerveződése (kooperáció). Mint látható az IT csak egy (bár karakterisztikus) szelete az informatikának. Az informatika tehát egyfajta világnézet, mely értelmében a rendelkezésre álló nem IT-jellegű erőforrásokat és az információs technológiákat úgy kell kapcsolatba hozni egymással, hogy adott célok esetén a végeredmény hatékonyabb (jelen esetben számvitel-szervezési megoldást) jelentsen, mint IT eszközök nélkül.”* (https://miau.my-x.hu/mediawiki/index.php/Informatikai_eroforrasok - 2024.06.08. - Informatikai erőforrások – Dr. Pitlik László).

2.8.1. A hálózatok és a dolgozat kapcsolata

A dolgozat negyedik fejezetében a rendszerterv-variánsok összeállításánál, konfigurálásánál elengedhetetlen tisztában lenni az alapvető informatikai hálózati ismeretekkel és szabványokkal, hiszen e nélkül akár inkompatibilis vagy alultervezett tervvariáns is előállítható.

2.8.2. A hálózatok és a tantárgyak kapcsolata

Szerverek és informatikai rendszerterv-variánsok összeállításánál elengedhetetlenek a „Hálózatok és számítógép architektúrák”, „Operációs rendszerek” és az „Elektronika fizika alapjai” tantárgyak alapos ismeretei.

2.9. Információs többletérték becslése

„Az információ a vállalat életében erőforrásként van jelen, mint erőforrást ezt is felhasználja a termelésben, melynek célja a haszon, profitszerzés. Az információ feldolgozásánál fontos, hogy megfelelő helyre jusson a megfelelő időben, megfelelő minőségben, mennyiségben és megfelelő árért, azaz a lehető legjobban hasznosítsuk. Tökéletes körülmények között az erőforrások input- és output értéke között az elérhető legnagyobb különbség jelentkezik. Ez a különbség az információs többletérték.” (https://miau.my-x.hu/mediawiki/index.php/Információs_többletérték - 2024.06.08. - Információs többletérték - Sztanko, Durdam, Dr. Pitlik László).

A dolgozat későbbi példáin keresztül láthatjuk, hogy a nyers adatok alapján nem biztos, hogy előre meg tudtuk volna állapítani a legjobb ár-érték arányú objektumokat. A felhasznált módszer azonban segített megállapítani az összefüggéseket a nyers adatokban, ami az emberi szem számára nem mindig megállapítható, így egy pontosabb eredményhez jutottam hozzá.

3. Szerverek ár-teljesítmény vizsgálata

Ebben a fejezetben bemutatásra kerül az adatgyűjtés folyamata, az adatok feldolgozásának menete, illetve az objektumok és attribútumok ismertetésére is sor kerül. Majd különböző elemzési módszerekkel vizsgálatokat végzek, amellyel kialakítok egy stabil és igazolt végeredményt.

A vizsgálat során „*Saját forrás*”-ként fogok hivatkozni az alábbi internetes tárhelyre feltöltött „*szerver_ar_telj_oam.xlsx*” fájlra: <https://miau.my-x.hu/miau/311/server/>. Ez a komplex, több munkalappal rendelkező Microsoft Excel programmal készült fájl tartalmazza a fejezetben bemutatásra kerülő vizsgálatok alapadatait, OAM táblázatait és a különböző mesterséges intelligenciával elvégzett modellezési számításokat, továbbá egyéb adatokat mint például a ChatGPT-vel folytatott beszélgetést. Ezen a tárhelyen egyébként megtalálható a szakdolgozat szempontjából releváns összes csatolmány.

3.1. Adatgyűjtés

A releváns adatok megszerzésére kizárólagosan a www.dell.com honlapot, azaz a Dell Incorporated, amerikai hardvergyártó- és tervező vállalat központi internetes információbázisát használtam. (vö. a weblapokat általában azért készítik, hogy a mögötte álló céget vagy személyt megtalálják az interneten és információt szerezzenek róla a honlapján). Az elemzéshez a Dell PowerEdge R960 szervereket választottam ki, amelyet a közép- és nagyvállalati szegmensbe pozicionáljuk. Dell (2024) https://www.dell.com/en-us/shop/dell-poweredge-servers/poweredge-r960-rack-server/spd/poweredge-r960/pe_r960_16718_vi_vp. Azért esett erre a választásom, mert az adott szerver modell nagyban kapcsolódik a jelenlegi munkahelyemen használt szerverekhez, és érdemesnek láttam objektív vizsgálatot végezni ebben a szegmensben.

Az adatok begyűjtését 2024.03.27-én végeztem. E miatt lehetséges, hogy a szakdolgozat készítésekor a www.dell.com honlapon közölt árakban változás következett be. A dolgozat „*context free*”, azaz minél inkább tartalomfüggetlen jellegét tekintve ez nem okoz problémát. Ez azt jelenti, hogy amennyiben a szakdolgozat írásának napján szerezném be az adatokat, a vizsgálat akkor is eredményesen elvégezhető lenne, a már esetlegesen különböző adatokkal.

3.2. Adatok feldolgozása

Az adatok feldolgozásához Solver típusú elemzést-, illetve az úgynevezett a COCO modellezőt (vö. COCO online szakértői rendszer/online modellező robot/MI) használtam. A Solver egy olyan algoritmus vagy szoftvereszköz, amely problémák megoldására szolgál, különösen matematikai, logikai vagy optimalizálási feladatok esetén. A Solver feladata, hogy egy adott probléma bemeneti adatait felhasználva kiszámítsa a megoldást, vagy megtalálja a legjobb (optimális) eredményt bizonyos feltételek mellett. A COCO a 2000-es évek elején lett kifejlesztve Magyarországon, a Szent István Egyetem kutatói és az INNOREG társulásában. Ez az online szakértői rendszer megmutatja, hogy a vizsgált objektumok (pl. országok, városok, termékek, piacok, cégek stb.), a vizsgálatba bevont tulajdonságaik alapján, egymáshoz viszonyítva alul- vagy fölülértékeltek.

Ebben a fejezetben az elemzést csak a Dell PowerEdge R960 típusú szerverek vonatkozásában végeztem, de ez bármilyen más informatikai eszköz, rendszerterv vagy rendszermodell vonatkozásában kivitelezhető, amennyiben találunk megfelelő objektumokat és azok attribútumait fel tudjuk írni számszerűsítve, vagy egyéb módszerekkel (vö. igen/nem = 1;0).

Esetemben az összes attribútumra (vö. hardver elem) jellemző, hogy minél drágább annál jobb, illetve minél jobb egy-egy tulajdonság, annál drágább lesz az objektum (vö. szerver) is. Ennek az irányvektorok megadásánál lesz majd jelentősége. Dr. Pitlik László ezt így fogalmazza meg: *„Az irányvektor a hasonlóságelemzés során az egymással összevetésre kerülő objektumokat leíró attribútumok (X_i) és a következményváltozók közötti ceteris paribus alakzatok irányultságának meghatározása, mely lehet pl. egyenes arányosságot követő, fordított arányosságok követő, optimum jellegű, monoton, egyéb.”* ([Hasonlóságelemzés alapok.doc](#) - 2024.06.08. - Hasonlóságelemzési alapok - Dr. Pitlik László).

3.2.1. Objektumok

Az objektumok jelen esetben a Dell PowerEdge R960 közép- és nagyvállalati szerverek különböző hardver elemű variánsai lesznek. Összesen 10db variánst vizsgálók. A variánsok összeállításánál törekedtem egy alapmodell, és a létező legjobb/legdrágább modell vizsgálatára is, a kettő között pedig különböző igényeknek megfelelő, és valóságon alapuló variánsokat konfiguráltam.

3.2.2. Attribútumok

Az attribútumok esetünkben a szerverek hardver elemei. A hardver a fizikai komponensek összessége, amelyeket egy számítógépes rendszer felépítéséhez használnak. A későbbiekben különböző jelöléseket fogok használni az OAM modell fejléceiben, a szerverek „ $O(n)$ -el, az ár „ $Y=\acute{A}r$ ”-ral lesz jelölve, a hardver elemek pedig „ $X(An)$ ”-el.

A jelenleg vizsgált Dell PowerEdge R960 szervert, a következő hardver elemekkel lehet jellemezni, illetve a hardver elemek összessége megadja a szerver árát. Ezekből az adatokból készül el az OAM táblázat:

- **Szerver ára ($Y=\acute{A}r$):** A szerver hardver elemeinek kiválasztása után egy kész „számítógépet” kapunk, quasi 1db-os mennyiségben, adott kikalkulált áron. Mértékegysége: USD (Amerikai dollár).
- **Processzor típus/ mennyiség $X(A1)$:** A nagy teljesítményű processzorok típusa az Intel gyártón belül a Xeon termékcsalád köré korlátozódnak. Ezek nagyon erős, nagy számítási kapacitással és nagy gyorsítótárral rendelkező komponensek. Ezek a processzorok megtalálhatóak még adatközpontokban, illetve nemrég került bevezetésre egyes processzorokban az Intel AI technológia is. Mértékegység: Db (Darab).
- **Processzor magok száma $X(A2)$:** A processzor magok száma általában páros számokkal írható le. Minél több processzormag van a processzoron belül, annál hatékonyabb az egység, egy-egy mag akár különböző feladatok elvégzésére is alkalmas. Mértékegysége: Db (Darab).
- **Processzor órajele $X(A3)$:** A CPU frekvenciája, Hertz-ben (Hz) mérve, az a sebesség, amellyel működik. A múltban a gyorsabb frekvencia jobb teljesítményt jelentett. Ez már nem feltétlenül van így. Egyes esetekben a CPU infrastruktúrája miatt egy alacsonyabb frekvencián futó CPU valóban jobban teljesít, mint egy magasabb frekvencián futó processzor. Mértékegysége: Mhz (Megahertz).
- **RAM memória mérete $X(A4)$:** A RAM, vagyis a Random Access Memory (véletlen-, vagy közvetlen hozzáférésű memória) a számítógépek által használt memória elsődleges formája. A RAM célja, hogy aktív, nagy sebességű memóriában tartsa mindazt, amit a processzor éppen futtat, amin éppen dolgozik. Mértékegysége: GB (Gigabájt).
- **RAID vezérlő $X(A5)$:** A RAID egy tárolási technológia, mely segítségével az adatok elosztása vagy replikálása több fizikailag független merevlemezen, egy logikai lemez létrehozásával lehetséges. A RAID vezérlő jellemzően minden szerver

alapfelszereltségébe beletartozik, azonban sokszor szükség lehet egy plusz vezérlőre is, jelen esetben is ez került beépítésre. Mértékegysége: 1; 0 (Van/Nincs).

- **NVME SSD meghajtó X(A6):** Az NVME SSD meghajtó jelenleg a létező leggyorsabb írás/olvasási sebességgel rendelkező mozgó alkatrészt nem tartalmazó adattároló egység, melyen általában az operációs rendszerek foglalnak helyet. Manapság már annyira kedvező az áruk, hogy minden egyéb adatok tárolására is előszeretettel ezt használják. Mértékegysége: TB (Terrabájt).
- **SATA SSD meghajtó X(A7):** A SATA SSD meghajtó egy 2.5” -os mozgó alkatrész nélküli 2,5” -os adattároló egység, mely fizikailag sokkal nagyobb, mint az NVME SSD meghajtó, illetve lényegesen lassabb írás/olvasási sebességre képes. Nagyon kedvező ára miatt kiváló ár-érték aránnyal bír. Mértékegysége: TB (terrabájt).
- **SAS SSD meghajtó X(A8):** A SAS SSD-k abban különböznek a SATA SSD-ktől, hogy csatolófelületük más, általában HOT-PLUG (vö. gyorsan cserélhető) kerettel szerelik őket, sebességük és megbízhatóságuk lényegesen nagyobb, mint a normál SSD meghajtóké. Mértékegysége: TB (Terrabájt).
- **DRAC management X(A9):** A Dell Remote Access Controller a Dell Inc. saját fejlesztésű távmanagement hardver egysége. Ez a megoldás lehetővé teszi a távoli szerver kezelését, az összeomlott rendszer helyreállítását, Dell PowerEdge TM hardver teljesítmény szabályozást. Különböző előfizetési csomaggal vásárolható. Mértékegysége: 1; 2; 3. (csomagok „jósága” szerint).
- **Optikai csatoló X(A10):** Az optikai csatoló gyakorlatilag egy nagyon gyors hálózati kártya (akár 25GbE sebességű), amely üvegszálas kábellel való összeköttetést tesz lehetővé különböző eszközök között. Ezt általában „storage”-hoz használják, tehát a számítógépen (vö. szerveren) kívüli adattároló eszköz összeköttetését biztosítja. Mértékegysége: Db (Darab).
- **Tápegység X(A11):** A tápegység a szerver tápellátásáért felel. A mai szerverekben általában QUAD típusú tápegység van, ezek redundáns egységek, amennyiben az egyik tápegység meghibásodik, a másik azonnal átveszi annak szerepét. Jellemzően túláram, túlfeszültség és egyéb ehhez hasonló védelemmel vannak ellátva. Mértékegysége: W (Watt).

A következőkben a kiindulási adattábla látható (4. ábra), az objektumok és attribútumok függvényében:

	A	B	C	D	E	F	G	H	I	J	K	L	M
	PowerEdge R960 Szerver / Tulajdonság	X(A1) CPU Xeon (darab)	X(A2) CPU magok száma (darab)	X(A3) CPU órajel (MHz)	X(A4) RAM memória (GB)	X(A5) RAID vezérlő (db)	X(A6) NVME SSD meghajtó (TB)	X(A7) SATA SSD meghajtó (TB)	X(A8) SAS HDD meghajtó (TB)	X(A9) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3)	X(A10) Optikai csatló külső storage-hoz (db)	X(A11) Quad tápegység (Watt)	Y = Ár (USD)
1													
2	Server1 - O1	2	16	3700	64	1	4	1	1.2	1	0	1400	24734
3	Server2 - O2	2	32	2900	128	1	4	1	1.2	1	0	1400	28778
4	Server3 - O3	2	56	2000	256	1	4	2	2.4	1	0	1800	42778
5	Server4 - O4	2	80	2200	512	1	4	4	4.8	1	0	1800	73210
6	Server5 - O5	2	96	2100	1024	1	4	8	9.6	2	0	1800	112185
7	Server6 - O6	4	112	2000	1024	1	8	8	9.6	2	1	1800	105512
8	Server7 - O7	4	160	2200	2048	1	8	16	9.6	2	1	1800	192820
9	Server8 - O8	4	192	2100	3096	1	16	16	9.6	2	1	2400	275005
10	Server9 - O9	4	240	1900	4096	1	32	16	9.6	3	2	2800	361161
11	Server10 - O10	4	240	1900	16384	1	128	16	9.6	3	2	2800	1115893
12													
13	irány	0	0	0	0	0	0	0	0	0	0	0	
14	korreláció	0.566	0.761	-0.407	0.998	-	0.987	0.646	0.492	0.758	0.776	0.792	
15													
16	O(n) = Objektum	X(An) = Attribútum											

4. ábra - OAM

Saját forrás: „Adattábla (nyers)” munkalap A14-M14 tartománya / Jelmagyarázat: a kiindulási adattáblán láthatóak a sor és az oszlopfejlécek, illetve a mértékegységek is. Ezen kívül az irányokat és a korrelációt is feltüntettem.

Ezen hardveres jellemzők (vö. attribútumok) közül, munkám során egy beszerzéskor általában három nagyon fontos hardver elemre koncentrálok, amelyek a jövőben nehezen vagy egyáltalán nem bővíthetők. Ezt a jövőbeli skálázhatóság miatt tartom fontosnak.

Jellemzően ez a processzor, a tápegység és a szerver háza. Ezeknek a bővítése kevés kivételtől eltekintve csak az alkatrész teljes cseréjével oldható meg, bővítési lehetőség szinte nincs (kivéve, ha az alaplapon több a processzor foglalát, mint a megvásárolt processzor(ok) száma).

3.2.3. Módszerek

Ahhoz, hogy az adatokat össze tudjuk hasonlítani, és ez által optimalizált választ kapjunk arra, hogy melyik szerver éri meg legjobban az árát, egy OAM-ot (vö. tanulási mintát) kellett létrehozni. Az attribútumokat ezután rangsorolni kell (vö. SORSZÁM függvény), ahol a legjobb érték fogja a legkisebb sorszámot kapni. Ezután a COCO (vö. hasonlóságelemzés) elnevezésű modellező robot, azaz MI segítségével különböző modellezési számítások eredményeit kapjuk meg, amiből stabil és valósághű következtetéseket vonhatunk el, quasi eredményt hirdethetünk. A modellek stabilitását is vizsgálnunk kell menet közben.

3.2.4. COCO STD

A COCO modul alapváltozata, amely objektumok ár-teljesítmény számításra képes (vö. árelőny-termelési függvény generátor), Y-ként minden esetben szükséges valós árat megadni.

„A COCO-STD modelljében rejlő lehetőség, hogy valós Y-változót (pl. árat) az X-változók lépcsős függvényeként közelítjük, vagyis minden X változóhoz meghatározunk egy-egy lépcsős függvényt pl. ár/érték-arány-elemzés érdekében, ahol az X-tömb csak pozitív egész számokat, azaz rangsor-számokat tartalmaz. A COCO-STD modell esetében a lépcsőkre vonatkozó megkötése: a következő lépcső legyen kisebb egyenlő, mint az azt megelőző. Az Y (oszlop) vektor az attribútumokat (O sor vektorok) leíró paraméterek közül az egyetlen kitüntetett (pl.: ár).”

Forrás: Pitlik Marcell: Térfogati égés optikai vizsgálata és mesterséges intelligencia alapú elemzése. In: http://www.combustioninstitute.hu/tdk_dij/2021/PitlikMarcell_TDK2020.pdf (Letöltve: 2024.06.15.)

3.2.5. COCO Y0

A COCO modul Y0 változata, amely teljesítmény elemzésre használható, minél nagyobb a kapott pont, annál jobb a teljesítmény. Az Y mindig fiktív és azonos.

„Az anti-diszkriminációs számítások-- más néven az ideálkereső, mesterséges fogalom-alkotó modell jelölése --, ahol minden X esetén az idealitás irányába ható irány megadása után optimalizálás keretében keressük az átlagtól leginkább eltérő objektumot úgy, hogy az optimalizálás célja mindvégig az objektumok azonosságának kikényszeríteni akarása (vö. lehet-e minden objektum másként egyforma - azaz létezik-e az adott fogalom ábrázolási skálája: vö. gázelegy-összetétel idealitás-index). Y0 modell esetén a lépcsők nagysága minimum 1 egység, vagyis a következő lépcső értéke legyen minimum 1 egységgel kisebb, mint az azt megelőző. Az Y vektor ez esetben konstans pl.: 1000 egység. (Tapasztalati érték: Y értéke legyen min. 2 nagyságrenddel nagyobb, mint az objektumok száma - a jelenleg alkalmazott LP-motor specialitásait figyelembe véve.)”

Forrás: Pitlik Marcell: Térfogati égés optikai vizsgálata és mesterséges intelligencia alapú elemzése. In: http://www.combustioninstitute.hu/tdk_dij/2021/PitlikMarcell_TDK2020.pdf (Letöltve: 2024.06.15.)

3.2.6. COCO STEP IX

Olyan COCO modul, amely a hasonlóságelemzés eredményeként egyensúlyi állapotokat mutató OAM esetén feloldja a legjobb ár-teljesítmény-viszony látszólagos hiányának dilemmáját. Hosszabb elnevezése COCO STEP mIn/maX, itt vegyük észre, hogy a „mIn/maX” a minimum és maximum szavak rövidítése, és az I és X betű adja meg a modul neve végét.

*„A modul célja, hogy egy, a hasonlóságelemzés eredményeként egyensúlyi állapotokat mutató OAM esetén feloldja a legjobb ár/teljesítmény-viszony látszólagos hiányának dilemmáját. A számításmenet módszertani szempontból roppant egyszerű: Venni kell egy-egy tényár helyett egy irracionálisan magas (pl. *100), ill. irracionálisan alacsony (pl. 0) árat (egyenként, azaz futtatásonként pontszerűen változtatva a kiindulási/egyensúlyi OAM-t). Majd az így végzett futtatás eredményéből következő ABS (hibák) összegével korrigálni kell az irracionális árat a megfelelő irányba (annyiszor, míg a hiba újra nulla nem lesz.) Az így megtalált ponttól minden további eltérés lefelé, illetve felfelé egyensúlyvesztést enged kimutatni, vagyis található lesz egy vagy több olyan objektum, mely a megváltozott árú objektumhoz képest ellentétes ár/teljesítmény kondíciókat mutat. A STEP IX futtatások eredménye tehát egy mátrix, mely sorai az objektumok, oszlopai pedig: a megtalált minimális és maximális egyensúlyi ár, ill. ezen árak tényárra vonatkoztatott aránya (min/tényleges és max/tényleges).*

Az objektumok rangsorolásának elve: ha több mint egy olyan objektum van, mely min./tényleges értéke 1 (ill. azonos), akkor ezek közül az a jobb, melynek a max/tény értéke nagyobb. Vagyis: a gyengébb min/tény érték önmagában is gyengébb rangsort kell, hogy jelentsen a max/tény értéktől függetlenül.” Forrás: MIAU (2024) https://miau.my-x.hu/myx-free/index_e8.php3?x=stp4

3.3. Elemzés - 1

Ebben a fejezetben bemutatom az elemzési folyamat lépéseit. Miután elkészült az OAM mátrix (4. ábra), ezt kiegészítettem egyéb adatokkal is, ami később hasznos lehet, az irányokkal és a korrelációval. Az irány 0, ha egy érték: „minél nagyobb annál jobb”, illetve 1, ha: „minél kisebb, annál jobb”. A korreláció jelzi két tetszőleges érték közötti lineáris kapcsolat nagyságát és irányát (avagy ezek egymáshoz való viszonyát). A „CPU órajel” azaz a D oszlop -0,407 értékkel korrelál az „Ár”, azaz M oszloppal. Ez az érték így kakukktojásnak tekinthető, viszont nem befolyásolta a végeredményt, amennyiben eltávolítottam az OAM-ból. Továbbá, ahogy fentebb már említettem, a processzoroknál nem feltétlen a nagyobb órajel jelenti a nagyobb teljesítményt.

3.3.1. Sorszámozás

Ezt követően az OAM-ban meglévő adatokat rangsoroltam egy ehhez hasonló táblázatban (5. ábra). Ebből sok hasznos információt meglehetett állapítani nagyon könnyen, összemérve versenytársaikkal, az adott adatnál melyik szerver részegység paramétere volt az első, vagy

utolsó helyen. Azért is nagyon hasznos még ez a táblázat elkészítése, mert ez által megkapjuk a szükséges számokat a COCO STD modell (árelőny-termelési függvény) elkészítéséhez/bebecsléséhez. A citromsárgával jelölt X(A5) oszlop, azaz eredeti attribútumként „RAID vezérlő +1” nevű oszlop jelen esetben elhanyagolható és törölhető a táblázatból, mivel az értékek egyformák.

	A	B	C	D	E	F	G	H	I	J	K	L	M
	PowerEdge R960 Server / Tulajdonság	X(A1) CPU Xeon (sorszám)	X(A2) CPU magok száma (sorszám)	X(A3) CPU órajel (sorszám)	X(A4) RAM memória (sorszám)	X(A5) RAID vezérlő (sorszám)	X(A6) NVME SSD meghajtó (sorszám)	X(A7) SATA SSD meghajtó (sorszám)	X(A8) SAS HDD meghajtó (sorszám)	X(A9) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (sorszám)	X(A10) Optikai csatoló külső storage-hoz (sorszám)	X(A11) Quad tápegység (sorszám)	Y = Ár (USD)
1													
2	Server1 - O1	6	10	1	10	1	6	9	9	7	6	9	24734
3	Server2 - O2	6	9	2	9	1	6	9	9	7	6	9	28778
4	Server3 - O3	6	8	7	8	1	6	8	8	7	6	4	42778
5	Server4 - O4	6	7	3	7	1	6	7	7	7	6	4	73210
6	Server5 - O5	6	6	5	5	1	6	5	1	3	6	4	112185
7	Server6 - O6	1	5	7	5	1	4	5	1	3	3	4	105512
8	Server7 - O7	1	4	3	4	1	4	1	1	3	3	4	192820
9	Server8 - O8	1	3	5	3	1	3	1	1	3	3	3	275005
10	Server9 - O9	1	1	9	2	1	2	1	1	1	1	1	361161
11	Server10 - O10	1	1	9	1	1	1	1	1	1	1	1	1115893

5. ábra - Rangsorolás (OAM)

Saját forrás: „Rangsorok (OAM)” munkalap A1-M11 tartománya / Jelmagyarázat: a sárga X(A5)-el jelölt oszlop minden értéke megegyezik, ezért törölhető a táblázatból, mert a végeredményt nem befolyásolja, hiszen azonos rangsorszámokat tartalmaz. Mivel ez az oszlop törölve lesz, így az attribútumok száma 11-ről 10-re csökken. Ez azt eredményezi, hogy az X(A6) oszlop eggyel előrébb kerül és megkapja az X(A5) elnevezést. Ugyanez lesz érvényes a fennmaradó többi attribútumra is, a többi oszlop fejlécének X(An) elnevezése is egy eggyel csökken, ami X(An-1) képlettel írható le.

3.3.2. Naiv becslés

Elkészítettem egy naiv becslési táblázatot összehasonlítási céllal (6. ábra). Ebben az esetben a "naiv értékelés" nem negatív értelemben használandó, csupán azt jelenti, hogy az értékelési módszer nem optimalizált. Ez szolgál majd kontrasztként a robot modell számításainak bemutatásához. Kiszámoltam a sorszámok átlagát (vö. ÁTLAG függvény), majd ezek előnyösségi sorrendjét. Ezeken kívül kiszámoltam a sorszámok átlagának inverzét, naiv ár-teljesítmény arányt, továbbá ezek előnyösségi sorrendjét.

	A	N	O	P	Q	R
1	PowerEdge R960 Szerver / Tulajdonság	Sorszámok átlaga	Sorszám- átlagok előnyössége (sorrend)	Sorszámok átlagának inverze (átlag)	Naiv ár/teljesítmény- arány (USD/sorszámátlag; Q=M/P)	Naiv ár/teljesítmény- arányok előnyössége (sorrend)
2	Server1 - O1	6.7	10	3.3	7558	1
3	Server2 - O2	6.6	9	3.4	8556	2
4	Server3 - O3	6.3	8	3.7	11477	3
5	Server4 - O4	5.5	7	4.5	16435	5
6	Server5 - O5	4.4	6	5.6	19904	6
7	Server6 - O6	3.5	5	6.5	16347	4
8	Server7 - O7	2.6	4	7.4	26185	7
9	Server8 - O8	2.5	3	7.5	36446	8
10	Server9 - O9	1.9	2	8.1	44638	9
11	Server10 - O10	1.7	1	8.3	134888	10

6. ábra - Naiv becslések

Saját forrás: „Rangsorok (OAM)” munkalap A1-A11 és N1-R11 tartománya / Jelmagyarázat: Minél zöldebb annál jobb, minél pirosabb annál rosszabb. Különböző átlag számításokkal -naiv becslésként- megkapjuk, hogy a „Sorszám átlagok előnyössége” oszlopa szerint a legdrágább szerver a legjobb, sorszámokkal illusztrálva. A „Naiv ár/teljesítmény arány” oszlop szerint a legolcsóbb szerver a legjobb, illetve a legdrágább a legrosszabb.

Ebből következőképpen ugyanezt mutatja meg a „Naiv ár/teljesítmény arányok előnyössége” oszlop, csak sorszámokkal. A „Sorszámok átlaga”, azaz az N oszlop értékei, az attribútumok rangsorszámának átlagát mutatják meg. Az attribútumokat nem tudtam feltüntetni az ábrán, mert a táblázat túl széles lenne. Ez az 5. ábrán megtekinthető.

3.3.3. Adatok feldolgozása mesterséges intelligenciával

Következő és egyik legfontosabb része a feladatnak maga a COCO (vö. Component-based Object Comparison for Objectivity) online modell elkészítése (az ár-érték-becslés levezetése) volt, mivel ezzel meg lehetett adni a végeredményét az egész feladatnak. Az online alkalmazás dolgozta fel a rangsorolási táblázat adatait, és abból vezette le optimalizáció keretében az eredményt.

A 7. ábrán szemléltetem a lényegét, mivel itt vannak a tényleges becslés-komponensek, ami megmutatja, hogy mennyi az ár-érték aránya az adott szervernek attribútumonkénti és adott esetekben a helyezésenkénti lebontásban. Ezen kívül még olyan hasznos információval is szolgál, hogy mennyivel térnek el az alapárhoz képest a Delta/Tény résznél.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	COCO:STD	X(A1) CPU Xeon (USD)	X(A2) CPU magok száma (USD)	X(A3) CPU órajel (USD)	X(A4) RAM memória (USD)	X(A5) NVME SSD meghajtó (USD)	X(A6) SATA SSD meghajtó (USD)	X(A7) SAS HDD meghajtó (USD)	X(A8) iDRAC (Ex.-1, Ent.-2, Ent. Adv.-3) (USD)	X(A9) Optikai csatoló külső storage-hoz (USD)	X(A10) Quad tápegység (USD)	Becsülés (USD; L=SUM(B:K))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; O=N/M)
43															
44	Server1 - O1	0	0	24734	0	0	0	0	0	0	0	24734	24734	0	0
45	Server2 - O2	0	0	24734	4044	0	0	0	0	0	0	28778	28778	0	0
46	Server3 - O3	0	0	0	42778	0	0	0	0	0	0	42778	42778	0	0
47	Server4 - O4	0	0	6673	66537	0	0	0	0	0	0	73210	73210	0	0
48	Server5 - O5	0	0	6673	105512	0	0	0	0	0	0	112185	112185	0	0
49	Server6 - O6	0	0	0	105512	0	0	0	0	0	0	105512	105512	0	0
50	Server7 - O7	0	0	6673	186147	0	0	0	0	0	0	192820	192820	0	0
51	Server8 - O8	0	0	6673	268332	0	0	0	0	0	0	275005	275005	0	0
52	Server9 - O9	0	0	0	361161	0	0	0	0	0	0	361161	361161	0	0
53	Server10 - O10	0	0	0	1115890	0	0	0	0	0	0	1115890	1115893	3	0

7. ábra - COCO elemzés (STD eredmény)

Saját forrás: COCO STD (manuális eredmény)” munkalap A43-O53 tartománya / Jelmagyarázat: a lépcsősfüggvény standard (STD) változatában nincs szigorú monotonitás mint az Y0-ban, tehát különböző teljesítmény rangsorszintekhez azonos csere-dollár érték tartozhat, például X(A3). A különböző háttérszámítások mellett látható, hogy a „Becsülés” oszlopban megkaptuk az új árakat. Összevetve, az „Y=Ár (USD)” oszloppal látható, hogy nincs eltérés. A „Becsülés” oszlop=B-től összeadjuk az összegeket K-ig. Az O10-nél egy picit zaj megfigyelhető, de ez elhanyagolható. Minden érték mértékegysége USD, kivéve a „Delta/Tény” oszlopot, ami %.

A COCO STD elemzést csak úgy lehet elvégezni, ha minden objektumhoz tartozik egy ár (Y), és ezeknél az objektumoknál minimum kettő azonos attribútum mentén összehasonlíthatónak kell lenniük. Az összehasonlított attribútumoknál nem szabad hiányoznia semmilyen adatnak sem, mivel akkor nem elvégezhető a modell-számítás.

3.3.4. Eredmény, konklúzió

A 8. ábrán látható a COCO STD elemzés végeredménye, a végbemenő háttérszámítások nélkül.

	A	L	M	N	O
	COCO:STD	Becsülés (USD; L=SUM(B:K))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; O=N/M)
43					
44	Server1 - O1	24734	24734	0	0
45	Server2 - O2	28778	28778	0	0
46	Server3 - O3	42778	42778	0	0
47	Server4 - O4	73210	73210	0	0
48	Server5 - O5	112185	112185	0	0
49	Server6 - O6	105512	105512	0	0
50	Server7 - O7	192820	192820	0	0
51	Server8 - O8	275005	275005	0	0
52	Server9 - O9	361161	361161	0	0
53	Server10 - O10	1115890	1115893	3	0

8. ábra - A tényleges és a becsült árak távolsága, a háttérszámítások nélkül. (USD)

Saját forrás: „COCO STD (manuális eredmény)” munkalap A43-A53 és L43-N53 tartománya / Jelmagyarázat: Az O10-nél egy picit zaj megfigyelhető, de ez, az értéket látva elhanyagolható. Minden érték mértékegysége USD, kivéve a „Delta/Tény” oszlopot, mely %.

A végeredmény azt mutatja, hogy minden szerver korrektül van árazva, gyakorlatilag nem mutatható ki érdemi eltérés. Egy minimális zaj megfigyelhető az utolsó konfigurációnál, viszont ez nem számottevő, bőven hibahatáron belüli. A validitás ellenőrzésére inverzet nem volt értelme készíteni, mivel nem kaptunk eltéréseket.

Az inverz azt jelenti, hogy a legnagyobb sorszámhoz hozzáadunk egyet, majd ebből a számból kivonjuk a meglévő sorszámokat, majd így is lefuttatjuk a modellt. A validitás ellenőrzése akkor sikeres, ha ugyanolyan eredményeket kapunk mint az eredeti modell esetében, csak ellentétes előjelekkel, (vö. +/-, pozitív/negatív).

De mit tudunk tenni ilyenkor? Hiszen nekem feladatom megállapítani, hogy melyik szervernek milyen az ár-teljesítmény aránya (vö. ár-érték aránya).

3.4. Elemzés - 2

Amennyiben a COCO STD modell eredménye az, hogy minden szerver korrektül van árazva, alternatív megoldást kell keresni.

3.4.1. Alternatív megoldás

Amennyiben a fenti eredményt kapjuk, egy új megoldással kell előállnunk, melynek neve: COCO STEP IX.

Először egy OAM sorozatot készítünk, ahol megállapítjuk az alsó és felső határokat. Mindezt úgy, hogy veszünk egy irreálisan alacsony (vö. min.) és magas (vö. max.) árat. Mindebből dupla annyi táblázat készül ahány objektumunk (vö. szerver) van, tehát jelen esetben 20db. A 9. ábrán látható milyen OAM táblázatok készültek a „Server1 - O1” elnevezésű objektumra vetítve (vö. Objektum 1). Ezt meg kell ismételni az összes objektumra nézve.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	O1_alsó	X(A1) CPU Xeon (sorszám)	X(A2) CPU magok száma (sorszám)	X(A3) CPU órajel (sorszám)	X(A4) RAM memória (sorszám)	X(A5) NVME SSD meghajtó (sorszám)	X(A6) SATA SSD meghajtó (sorszám)	X(A7) SAS HDD meghajtó (sorszám)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (sorszám)	X(A9) Optikai csatoló külső storage-hoz (sorszám)	X(A10) Quad tápegység (sorszám)	Y = Ár (USD)	COCO STD becsült alsóhatár ár (USD)	Eredeti ár (USD)
1														
2	Server1 - O1	6	10	1	10	6	9	9	7	6	9	0	3337	24734
3	Server2 - O2	6	9	2	9	6	9	9	7	6	9	28778		
4	Server3 - O3	6	8	7	8	6	8	8	7	6	4	42778		
5	Server4 - O4	6	7	3	7	6	7	7	7	6	4	73210		
6	Server5 - O5	6	6	5	5	6	5	1	3	6	4	112185		
7	Server6 - O6	1	5	7	5	4	5	1	3	3	4	105512		
8	Server7 - O7	1	4	3	4	4	1	1	3	3	4	192820		
9	Server8 - O8	1	3	5	3	3	1	1	3	3	3	275005		
10	Server9 - O9	1	1	9	2	2	1	1	1	1	1	361161		
11	Server10 - O10	1	1	9	1	1	1	1	1	1	1	1115893		
12														
	O1_felső	X(A1) CPU Xeon (sorszám)	X(A2) CPU magok száma (sorszám)	X(A3) CPU órajel (sorszám)	X(A4) RAM memória (sorszám)	X(A5) NVME SSD meghajtó (sorszám)	X(A6) SATA SSD meghajtó (sorszám)	X(A7) SAS HDD meghajtó (sorszám)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (sorszám)	X(A9) Optikai csatoló külső storage-hoz (sorszám)	X(A10) Quad tápegység (sorszám)	Y = Ár (USD)	COCO STD becsült felsőhatár ár (USD)	Eredeti ár (USD)
13														
14	Server1 - O1	6	10	1	10	6	9	9	7	6	9	9999999	10000000	24734
15	Server2 - O2	6	9	2	9	6	9	9	7	6	9	28778		
16	Server3 - O3	6	8	7	8	6	8	8	7	6	4	42778		
17	Server4 - O4	6	7	3	7	6	7	7	7	6	4	73210		
18	Server5 - O5	6	6	5	5	6	5	1	3	6	4	112185		
19	Server6 - O6	1	5	7	5	4	5	1	3	3	4	105512		
20	Server7 - O7	1	4	3	4	4	1	1	3	3	4	192820		
21	Server8 - O8	1	3	5	3	3	1	1	3	3	3	275005		
22	Server9 - O9	1	1	9	2	2	1	1	1	1	1	361161		
23	Server10 - O10	1	1	9	1	1	1	1	1	1	1	1115893		

9. ábra - STEP IX sorozat OAM táblázatai, alsó-felső határ (Objektum 1)

Saját forrás: „STEP IX OAM sorozat” munkalap A1-N23 tartománya / Jelmagyarázat: a sárgával kijelölt cellákban láthatjuk az irreálisan alacsony- illetve magas árat, amivel újból elvégezzük a számításokat, tehát beküldjük a COCO STD online hasonlóságelemző robotba, amely optimalizálás keretében számításokat végez, a hasonlóságelemzés módszerét felhasználva.

Az OAM táblázatokat a COCO STD online modellező robotba egyesével beküldtem, majd a végeredményeket (10. és 11. ábra) kimásoltam a Saját forrás „STEP IX COCO STD” munkalapjába.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	X(A1) CPU Xeon (USD)	X(A2) CPU magok száma (USD)	X(A3) CPU órajel (USD)	X(A4) RAM memória (USD)	X(A5) NVME SSD meghajtó (USD)	X(A6) SATA SSD meghajtó (USD)	X(A7) SAS HDD meghajtó (USD)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (USD)	X(A9) Optikai csatoló külső storage-hoz (USD)	X(A10) Quad tápegység (USD)	Becsülés (USD; L=SUM(B:K))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; O=N/M)
43														
44	0	0	3337	0	0	0	0	0	0	0	3337	0	-3337	0
45	0	14389	3337	11053	0	0	0	0	0	0	28778	28778	0	0
46	0	21389	0	21389	0	0	0	0	0	0	42778	42778	0	0
47	0	36605	3337	33269	0	0	0	0	0	0	73210	73210	0	0
48	0	52756	3337	52756	0	0	0	0	0	0	108849	112185	3336	3.0
49	0	52756	0	52756	0	0	0	0	0	0	105512	105512	-0.1	0
50	0	52756	3337	93074	0	43654	0	0	0	0	192820	192820	-0.1	0
51	0	52756	3337	175259	0	43654	0	0	0	0	275005	275005	-0.2	0
52	0	95834	0	221673	0	43654	0	0	0	0	361161	361161	-0.2	0
53	0	95834	0	976404	0	43654	0	0	0	0	1115892	1115893	0.8	0

10. ábra - STEP IX COCO STD elemzés eredménye, alsó határ (Objektum 1)

Saját forrás: „STEP IX COCO STD” munkalap A43-O53 tartománya / Jelmagyarázat: a sárgával kijelölt cellában láthatjuk az irreálisan alacsony árat, a különböző elvégzett háttérszámítások mellett. A „Becsülés” oszlop árai, amik USD-ban vannak megadva, a B-től a K oszlopig szereplő árak összeadva.

Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
COCO:STD	X(A1) CPU Xeon (USD)	X(A2) CPU magok száma (USD)	X(A3) CPU órajel (USD)	X(A4) RAM memória (USD)	X(A5) NVME SSD meghajtó (USD)	X(A6) SATA SSD meghajtó (USD)	X(A7) SAS HDD meghajtó (USD)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (USD)	X(A9) Optikai csatoló külső storage-hoz (USD)	X(A10) Quad tápegység (USD)	Becslés (USD; L=SUM(B:K))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; AE=AD/AC)
Server1 - O1	0	0	10000000	0	0	0	0	0	0	0	10000000	9999999	-1	0
Server2 - O2	0	0	28778	0	0	0	0	0	0	0	28778	28778	0	0
Server3 - O3	0	0	0	42778	0	0	0	0	0	0	42778	42778	0	0
Server4 - O4	0	0	6673	66537	0	0	0	0	0	0	73210	73210	0	0
Server5 - O5	0	0	6673	105512	0	0	0	0	0	0	112185	112185	0	0
Server6 - O6	0	0	0	105512	0	0	0	0	0	0	105512	105512	0	0
Server7 - O7	0	0	6673	186147	0	0	0	0	0	0	192820	192820	0	0
Server8 - O8	0	0	6673	268332	0	0	0	0	0	0	275005	275005	0	0
Server9 - O9	0	0	0	361161	0	0	0	0	0	0	361161	361161	0	0
Server10 - O10	0	0	0	1115890	0	0	0	0	0	0	1115890	1115893	3	0

11. ábra - STEP IX COCO STD elemzés eredménye, felső határ (Objektum 1)

Saját forrás: „STEP IX COCO STD” munkalap Q43-AE53 tartománya / Jelmagyarázat: a sárgával kijelölt cellában láthatjuk az irreálisan magas árat, a különböző elvégzett háttérszámítások mellett. A „Becslés” oszlop árai, amik USD-ban vannak megadva, az R-től az AA oszlopig szereplő árak összeadva.

Az itt újonnan megkapott alsóhatár és felsőhatár becslési értékeket (vö. 10. és 11. ábrán sárgával jelölt cellák) bemásoltam a 9. ábrán látható Y=Ár (USD) oszlop megfelelő celláiba, tehát a Saját forrás: „STEP IX OAM sorozat” munkalap L2 cellájába és a Saját forrás: „STEP IX OAM sorozat” munkalap L14 cellájába.

Az alsó és felső határok megállapítása után elkészült a STEP IX OAM tábla (12. ábra). A táblázat tartalmaz még különböző arány számításokat és fedezet számítást is, illetve az irányok is feltüntetésre kerülnek, továbbá az Y0 fiktív változó (vö. pontszám/jószágpont) is bevezetésre került.

	A	B	C	D	E	F	G	H	I	J	K
1						Irány (minél...annál előnyösebb)	0	0	1	1	
2	STEP IX	Alsóhatár min. (USD)	Tényleges ár (USD)	Felsőhatár max (USD)		STEP IX OAM	X(A1) min%	X(A2) max%	X(A3) lent/fent-arány	X(A4) fedezet %	Y0 (pontszám)
3	Server1 - O1	3337	24734	10000000		Server1 - O1	13%	40430%	0.21%	0.25%	100000
4	Server2 - O2	3337	28778	5033756		Server2 - O2	12%	17492%	0.51%	0.57%	100000
5	Server3 - O3	0	42778	2959999		Server3 - O3	0%	6919%	1.47%	1.45%	100000
6	Server4 - O4	21408	73210	3505035		Server4 - O4	29%	4788%	1.51%	2.10%	100000
7	Server5 - O5	22453	112185	3148854		Server5 - O5	20%	2807%	2.95%	3.59%	100000
8	Server6 - O6	42616	105512	3318986		Server6 - O6	40%	3146%	1.96%	3.22%	100000
9	Server7 - O7	59214	192820	3596331		Server7 - O7	31%	1865%	3.93%	5.45%	100000
10	Server8 - O8	88619	275005	4093475		Server8 - O8	32%	1489%	4.88%	6.87%	100000
11	Server9 - O9	137284	361161	5557946		Server9 - O9	38%	1539%	4.31%	6.66%	100000
12	Server10 - O10	247428	1115893	10000000		Server10 - O10	22%	896%	9.78%	11.44%	100000

12. ábra - STEP IX OAM (min./max.) Y0 elemzéshez

Saját forrás: „Eredmények OAM” munkalap A1-K12 tartománya / Jelmagyarázat: Az A1-D12 táblázatban látható a korábban kiszámolt minimum árak, a tényleges/eredeti árak, illetve a korábban kiszámolt maximum árak is, a 10. és 11. ábra szerint, ezek mértékegysége USD. Az F1-K12 táblázatban különböző matematikai számítások eredményei látható, melynek mértékegysége %, illetve egy úgynevezett Y0 fiktív változó/pontszám, ami bevezetésre került. Továbbá, feltüntettem az irányokat is. Ezek szükséges, hogy a későbbiekben átalakított táblázat feldolgozható legyen a mesterséges intelligencia által, ez a 13. ábrán lesz látható.

3.4.2. Sorszámozás

Ezek a számítások elősegítik, hogy egy olyan rangsorolt OAM táblát (13. ábra) készítsünk, amit a COCO Y0 anti-diszkriminációs, hasonlóságelemző online modellezőbe be tudunk tölteni. Ahogy már korábban említettem, az Y0 változó/pontszám fiktív és kellően/megfelelően nagy érték ahhoz, hogy a modellező robotnak teret adjunk.

	M	N	O	P	Q	R
1						
2	STEP IX OAM	X(A1) min% (sorszám)	X(A2) max% (sorszám)	X(A3) lent/fent-arány (sorszám)	X(A4) fedezet % (sorszám)	Y0 (pontszám)
3	Server1 - O1	8	1	1	1	100000
4	Server2 - O2	9	2	2	2	100000
5	Server3 - O3	10	3	3	3	100000
6	Server4 - O4	5	4	4	4	100000
7	Server5 - O5	7	6	6	6	100000
8	Server6 - O6	1	5	5	5	100000
9	Server7 - O7	4	7	7	7	100000
10	Server8 - O8	3	9	9	9	100000
11	Server9 - O9	2	8	8	8	100000
12	Server10 - O10	6	10	10	10	100000

13. ábra - Rangsorolás (OAM)

Saját forrás: „Eredmények OAM” munkalap M2-R12 tartománya / Jelmagyarázat: A táblázatban különböző matematikai számítások eredményeinek rangsorszámait láthatóak, illetve egy úgynevezett Y0 fiktív változó/pontszám. Ezek szükségesek ahhoz, hogy táblázat feldolgozható legyen a mesterséges intelligencia által.

Tisztázó jelleggel megjegyzem, hogy minden Y0 képes STD jelleggel árelőny-termelési függvényt generálni, viszont egyetlen STD modell sem alkalmas Y0 kérdések megoldására.

3.4.3. Adatok feldolgozása mesterséges intelligenciával

Az alábbi összefoglaló ábrán látható a COCO Y0 modell eredménye a feldolgozás után. A táblázatot úgy értelmezzük, hogy minél több „Becsles” pontot kap az adott objektum, annál „értékesebb”, tehát győztest hirdethetünk.

	A	B	C	D	E	F	G	H	I
	COCO:Y0	X(A1) (pontszám)	X(A2) (pontszám)	X(A3) (pontszám)	X(A4) (pontszám)	Becslés (pontszám)	Y0 (pontszám)	Delta	Delta/Tény (pontszám/pontszám; I=H/G)
43									
44	Server1 - O1	49986.9	49998.9	9	9	100003.8	100000	-3.8	0
45	Server2 - O2	49985.9	49997.9	8	8	99999.8	100000	0.2	0
46	Server3 - O3	49984.9	49996.9	7	7	99995.8	100000	4.2	0
47	Server4 - O4	99990.8	7	6	6	100009.3	100000	-9.3	-0.01
48	Server5 - O5	99983.3	4	4	4	99995.3	100000	4.7	0
49	Server6 - O6	99994.8	5	5	5	100009.8	100000	-9.8	-0.01
50	Server7 - O7	99991.8	3	3	3	100000.8	100000	-0.8	0
51	Server8 - O8	99992.8	1	1	1	99995.8	100000	4.2	0
52	Server9 - O9	99993.8	2	2	2	99999.8	100000	0.2	0
53	Server10 - O10	99989.8	0	0	0	99989.8	100000	10.2	0.01

14. ábra - COCO elemzés (Y0 eredmény)

Saját forrás: „Y0” munkalap A43-I53 tartománya / Jelmagyarázat: A „Becslés” oszlopban a mesterséges intelligencia által kapott becslés pontszámok összessége látható, a B oszloptól az E oszlopig összeadva. Minél több pontot kap egy objektum, annál értékesebb, tehát a legtöbb pontszámot kapott objektum a legjobb, illetve ez alapján felállítható egy végső sorrend is.

3.4.4. Végeredmény, konklúzió

Ez a végeredmény kellően stabil ahhoz, hogy kihirdessem az első három helyezettet, illetve a legrosszabb ár-érték arányú szerveret (15. ábra). Jelen esetben mivel kevés objektumunk van, könnyedén ki tudjuk választani a győztest, viszont amennyiben sokkal több objektumunk lenne akkor egy függvényt használnák a legnagyobb érték megállapítására (vö. MAX függvény). Amint megkaptuk a legnagyobb értéket a holtversenyt is célszerű lehet ellenőriznünk, szintén függvénnyel (vö. DARABTELI). Jelen esetben holtverseny nem állt fenn.

	M	N	O	P	Q	R	S	T	U
1									
2	STEP IX OAM	X(A1) min% (sorszám)	X(A2) max% (sorszám)	X(A3) lent/fent-arány (sorszám)	X(A4) fedezet % (sorszám)	Y0 (pontszám)	Becslés (pontszám)	Validitás (1=igen, 0=nem)	Konklúzió (helyezettek)
3	Server1 - O1	8	1	1	1	100000	100003.8	1	harmadik legjobb
4	Server2 - O2	9	2	2	2	100000	99999.8	1	
5	Server3 - O3	10	3	3	3	100000	99995.8	1	
6	Server4 - O4	5	4	4	4	100000	100009.3	1	második legjobb
7	Server5 - O5	7	6	6	6	100000	99995.3	1	
8	Server6 - O6	1	5	5	5	100000	100009.8	1	legjobb ár/érték arány
9	Server7 - O7	4	7	7	7	100000	100000.8	1	
10	Server8 - O8	3	9	9	9	100000	99995.8	1	
11	Server9 - O9	2	8	8	8	100000	99999.8	1	
12	Server10 - O10	6	10	10	10	100000	99989.8	1	legrosszabb ár/érték arány

15. ábra - COCO elemzés (becslés-validitás-konklúzió)

Saját forrás: „Y0” munkalap M1-U12 tartománya / Jelmagyarázat: A táblázatban különböző matematikai számítások eredményeinek rangsorszámai láthatóak, illetve egy úgynevezett Y0 fiktív változó/pontszám. Ezek szükségesek ahhoz, hogy táblázat feldolgozható legyen a mesterséges intelligencia által. A „Becslés pontszám” oszlopban a mesterséges intelligencia által kapott becslés pontszámok láthatóak. Minél több pontot kap egy objektum, annál értékesebb, tehát a legtöbb pontszámot kapott objektum a legjobb, illetve ez alapján felállítható egy végső sorrend is. Minél zöldőbb annál jobb, minél pirosabb annál rosszabb, tehát a kialakult végeredmény látható, illetve a validitás ellenőrzése is megtörtént.

Immár eredményként megkapjuk, hogy vajon lehet-e minden objektum másképp egyforma? Avagy melyik a leginkább ideális objektum? (16. ábra).

	B
10	1. Közép. és nagyvállalati szerverek ár-teljesítmény vizsgálata. (2024.03.27-i árlista alapján, a nem egyedi (non-CTO) szerverkonfigurációkat tekintve)
11	1. helyezett: Server6
12	2. helyezett: Server4
13	3. helyezett: Server1
14	Legrosszabb: Server10

16. ábra - Végeredmény (helyezések)

Saját forrás: „Alapadatok” munkalap B10-B14 tartománya / Jelmagyarázat: minél zöldebb annál jobb helyezés, a halvány piros színnel jelölt szerver a legrosszabb ár-érték arányú.

A validitás ellenőrzésére inverzet is készítettem (17. ábra). Ahogy korábban már kifejtettem, az inverz azt jelenti, hogy a legnagyobb sorszámmal hozzáadunk egyet, majd ebből a számból kivonjuk a meglévő sorszámmat, majd így is lefuttatjuk a modellt.

	M	N	O	P	Q	R
	COCO:Y0	X(A1) min% (inverz sorszám)	X(A2) max% (inverz sorszám)	X(A3) lent/fent- arány (inverz sorszám)	X(A4) fedezet (inverz sorszám)	Y0 (pontszám)
7						
8	Server1 - O1	3	10	10	10	100000
9	Server2 - O2	2	9	9	9	100000
10	Server3 - O3	1	8	8	8	100000
11	Server4 - O4	6	7	7	7	100000
12	Server5 - O5	4	5	5	5	100000
13	Server6 - O6	10	6	6	6	100000
14	Server7 - O7	7	4	4	4	100000
15	Server8 - O8	8	2	2	2	100000
16	Server9 - O9	9	3	3	3	100000
17	Server10 - O10	5	1	1	1	100000

17. ábra - Inverz OAM táblázat

Saját forrás: „Y0” munkalap M7-R17 tartománya / Jelmagyarázat: A táblázatban látható sorszámmal a 13. ábrán látható táblázatban szereplő sorszámmal -11-ből kivont- inverz eredményei. Illetve, az úgynevezett Y0 fiktív változó/pontszám is feltüntetésre kerül. Ezek szükségesek ahhoz, hogy az inverz táblázat feldolgozható legyen a mesterséges intelligencia által.

A 18. ábrán a „normál” és inverz OAM táblázatunk COCO Y0 modell végeredményei láthatóak (vö. bal oldal - normál, jobb oldal - inverz), ahol is tisztán látszik, hogy az értéke megegyeznek, csak az előjelük ellentétes, tehát a végeredmény validnak/érvényesnek tekinthető.

	A	B	C	D	E	F	G	H				T	U	V	W	X	Y	Z	AA	AB
	COCO:Y0	X(A1) (pontszám)	X(A2) (pontszám)	X(A3) (pontszám)	X(A4) (pontszám)	Becslés (pontszám)	Y0 (pontszám)	Delta	Delta/Tény (pontszám/pontszám; 1:H/G)			COCO:Y0	X(A1) (pontszám)	X(A2) (pontszám)	X(A3) (pontszám)	X(A4) (pontszám)	Becslés (pontszám)	Y0 (pontszám)	Delta	Delta/Tény (pontszám/pontszám; AB-AA/Z)
43	Server1 - O1	49986.9	49986.9	9	9	100003.8	100000	-3.8	0	43	Server1 - O1	99996.2	0	0	0	99996.2	100000	3.8	0	
44	Server2 - O2	49985.9	49997.9	8	8	99999.8	100000	0.2	0	44	Server2 - O2	99997.2	1	1	1	100000.2	100000	-0.2	0	
45	Server3 - O3	49984.9	49996.9	7	7	99995.8	100000	4.2	0	45	Server3 - O3	99998.2	2	2	2	100004.2	100000	-4.2	0	
46	Server4 - O4	99990.8	7	6	6	100009.3	100000	-9.3	-0.01	46	Server4 - O4	49988.1	49996.6	3	3	99990.7	100000	9.3	0.01	
47	Server5 - O5	99983.3	4	4	4	99995.3	100000	4.7	0	47	Server5 - O5	49995.6	49999.1	5	5	100004.7	100000	-4.7	0	
48	Server6 - O6	99994.8	5	5	5	100009.8	100000	-9.8	-0.01	48	Server6 - O6	49984.1	49998.1	4	4	99990.2	100000	9.8	0.01	
49	Server7 - O7	99991.8	3	3	3	100000.8	100000	-8.8	0	49	Server7 - O7	49987.1	50000.1	6	6	99999.2	100000	0.8	0	
50	Server8 - O8	99992.8	1	1	1	99995.8	100000	4.2	0	50	Server8 - O8	49986.1	50002.1	8	8	100004.2	100000	-4.2	0	
51	Server9 - O9	99993.8	2	2	2	99999.8	100000	0.2	0	51	Server9 - O9	49985.1	50001.1	7	7	100000.2	100000	-0.2	0	
52	Server10 - O10	99989.8	0	0	0	99989.8	100000	10.2	0.01	52	Server10 - O10	49989.1	50003.1	9	9	100010.2	100000	-10.2	-0.01	

18. ábra - Validitás ellenőrzés táblázatai (ellentétes előjelek)

Saját forrás: „Y0” munkalap A43-AB53 tartománya / Jelmagyarázat: a két táblázat egyfajta ellenőrzésre szolgál, amennyiben a „Delta” oszlopban a számok megegyeznek, de előjelük ellentétes, úgy a validitás igazolásra került. A 14. ábra jelmagyarázata bővebb információval szolgál a táblázatról.

A 15. ábrán látható a „validitás” elnevezésű oszlop, ahol 1-es értékek szerepelnek (valid = 1, nem valid = 0). Ezt a „HA” függvényvel ellenőriztem, az alábbi képlettel: =HA(Y0!H44*Y0!AA44<=0;1;0).

A példában mindenképpen be szerettem volna mutatni a szélsőséges eseteket, tehát az „ideált”, a lehető legjobb szervert is. A végeredményt tekintve ez lett a legrosszabb ár-teljesítmény arányú objektum. **Itt felmerülhet a kérdés, hogy van-e értelme ideáltól való eltérést vizsgálni?** Ennél a feladatnál nincs értelme, hiszen szervereknél fontos a skálázhatóság, és egy adott beszerzésre/közbeszerzésre általában véges a költségvetésünk. Ennek megfelelően kötelező a számunkra ideálist beszerezni, minden paramétert figyelembe véve.

A harmadik fejezetben tehát különböző modellezési számításokkal és mesterséges intelligencia bevonásával szerverek ár-teljesítményét vizsgáltam.

4. Informatikai rendszerterv-variánsok vizsgálata

Az előző fejezethez képest egy komplexebb és gyakorlatiasabb megközelítésből mutatom be, hogyan lehet segítségünkre a mesterséges intelligencia az informatikai rendszertervezésben, rendszerterv-variánsok objektív vizsgálatakor. Cél a legjobb ár-teljesítmény arányú rendszerterv-variáns vizsgálata, a költségvetés figyelembevételével.

A feladat komplexitását az adja, hogy a rendszerterv-variánsok összeállításánál tisztában kell lenni különböző informatikai ismeretekkel, úgymint a funkcionális és nem-funkcionális követelményekkel, hálózati infrastruktúrával, operációs rendszerekkel és a virtualizációval, továbbá az alapvető biztonsági fogalmakkal, illetve adatbázis tervezési ismeretekkel is rendelkezünk kell. Ez adja quasi a feladat alapbonyolultságát. Egyszerűbben fogalmazva, minden mindennel összefügg, tehát nem választhatunk találomra részelemeket, részegységeket a rendszertervünkhöz.

A korábbiakhoz hasonlóan bemutatásra kerül az adatgyűjtés folyamata, az adatok feldolgozásának menete, illetve az objektumok és attribútumok ismertetésére is sor kerül. Majd különböző elemzési módszerekkel vizsgálatokat végzek, amellyel kialakítok egy stabil és igazolt végeredményt. A gyakorlati megközelítést fokozva pedig bemutatok egy valós informatikai rendezőszekrény tervet (anonim módon), amely megvalósításra került egy multinacionális vállalatnál a közelmúltban.

A vizsgálat során „*Saját forrás2*”-ként fogok hivatkozni az alábbi internetes tárhelyre feltöltött „*it_rendszerterv_oam.xlsx*” fájlra: <https://miau.my-x.hu/miau/311/server/>. Ez a komplex, több munkalappal rendelkező Microsoft Excel programmal készült fájl tartalmazza a fejezetben bemutatásra kerülő vizsgálatok alapadatait, OAM táblázatait és a különböző mesterséges intelligenciával elvégzett modellezési számításokat, továbbá egyéb adatokat, mint például a ChatGPT-vel folytatott beszélgetést. Ahogy korábban már említettem, ezen a tárhelyen megtalálható a szakdolgozat szempontjából releváns összes csatolmány.

4.1. Adatgyűjtés

A releváns adatok megszerzésére a www.dell.com, a www.3bnetwork.hu, a www.senetic.hu, a www.energom.hu és a www.checkfirewalls.com honlapokat használtam. Az elemzéshez az előzőekben megismert Dell PowerEdge R960 szerverből választottam ki kettőt, X-Tech informatikai rendezőszekrényből is kettőt, HP Aruba switchből (vö. kapcsoló) pedig hármat, Checkpoint tűzfalból kettőt, HP Aruba access pointből (vö. hozzáférési pont) hármat, Cover

illetve Riello szünetmentes tápegységből kettőt, továbbá Legrand márkájú patch panelből szintén kettőt választottam ki. Így összesen 16db attribútumom (vö. hardver elem) lett, 10db objektumhoz (vö. rendszerterv-variánshoz).

Az adatok begyűjtését 2024.05.05-én végeztem. E miatt lehetséges, hogy a szakdolgozat készítésekor a közölt árakban változás következett be. A dolgozat „*context free*”, azaz minél inkább tartalomfüggetlen jellegét tekintve ez nem okoz problémát, ahogyan korábban erre már a 3.1-es fejezetben utaltam.

4.2. Adatok feldolgozása

Az adatok feldolgozása az előzőkéhez hasonlóan történt, amit a 3.2. Adatok feldolgozása c. fejezetben már részleteztem.

4.2.1. Objektumok

Az objektumok jelen esetben közép- és nagyvállalati rendszertervek különböző hardver elemű variánsai. Összesen 10db variánst vizsgálok. A variánsok összeállításánál törekedtem reális és valóban előforduló rendszertervek összeállítására. Viszont a korábban már említett „*context free*”, azaz tartalomfüggetlen elvet követve, a kiindulási táblázat (19. ábra) bármilyen egyéb releváns adattal kitölthető, azaz más komponensekkel és más darabszámmal is elvégezhető lenne a vizsgálat.

4.2.2. Attribútumok

Az attribútumok esetünkben a rendszerterv-variánsok hardver elemei. Az előzőkhez hasonlóan különböző jelöléseket fogok használni az OAM modell fejléceiben, a rendszertervek „ $O(n)$ -el, az ár „ $Y=\acute{A}r$ ”-ral lesz jelölve, a hardver elemek pedig „ $X(An)$ ”-el.

A jelenleg vizsgált informatikai rendszerterv-variánsokat a következő hardver elemekkel lehet jellemezni, illetve a hardver elemek összessége megadja a rendszertervek árát. Ezekből az adatokból készül el az OAM táblázat:

- **Rendszerterv ára ($Y=\acute{A}r$):** A hardver elemek kiválasztása és darabszámuk megadása után egy „kulcsrakész”, több részegységből álló komplett rendszertervet kapunk, adott kikalkulált áron. Mértékegysége: USD (Amerikai dollár).
- **Szerver1 $X(A1)$:** A korábban már vizsgált Dell PowerEdge R960 szerver egyik variánsa. Mértékegység: Db (Darab).

- **Szerver2 X(A2):** A korábban már vizsgált Dell PowerEdge R960 szerver másik variánsa. Mértékegysége: Db (Darab).
- **Rackszekrény1 X(A3):** A rackszekrény, más néven rack, egy szabványosított keret vagy szekrény, amely különféle elektronikai, számítógépes és hálózati berendezések elhelyezésére szolgál. Ezeket a szekrényeket gyakran használják adatközpontokban, szervertermekben és egyéb informatikai infrastruktúrákban. Mértékegysége: db.
- **Rackszekrény2 X(A4):** A rackszekrény, más néven rack, egy szabványosított keret vagy szekrény, amely különféle elektronikai, számítógépes és hálózati berendezések elhelyezésére szolgál. Ezeket a nagyobb méretű szekrényeket gyakran használják adatközpontokban, szervertermekben és egyéb informatikai infrastruktúrákban. Mértékegysége: Db (Darab).
- **„Core” switch X(A5):** A core switch egy olyan nagy teljesítményű hálózati switch, amely az adatközpontok, nagyvállalati hálózatok, illetve szolgáltatói hálózatok központi részeként szolgál. A core switch a hálózati infrastruktúra gerincében helyezkedik el, és kritikus szerepet játszik a hálózati forgalom kezelésében, irányításában és továbbításában. Mértékegysége: Db (Darab).
- **Normál switch1 X(A6):** Egy "normál" switch, amelyet gyakran egyszerűen csak switch-nek neveznek, egy hálózati eszköz, amely az adatkapcsolati rétegben (Layer 2) működik és az eszközök közötti adatforgalmat irányítja egy helyi hálózaton (LAN). A switch-ek az adatokat (csomagokat) az eszközök MAC címei alapján továbbítják, és különböznek a routerektől, amelyek IP címek alapján irányítják a forgalmat a hálózatok között. Mértékegysége: Db (Darab).
- **Normál switch2 X(A7):** Egy "normál" switch, amelyet gyakran egyszerűen csak switch-nek neveznek, egy hálózati eszköz, amely az adatkapcsolati rétegben (Layer 2) működik és az eszközök közötti adatforgalmat irányítja egy helyi hálózaton (LAN). A switch-ek az adatokat (csomagokat) az eszközök MAC címei alapján továbbítják, és különböznek a routerektől, amelyek IP címek alapján irányítják a forgalmat a hálózatok között. Ez nagyobb teljesítményű, mint az előző. Mértékegysége: Db (Darab).
- **Tűzfal1 X(A8):** A tűzfal egy hálózatbiztonsági rendszer, amely figyeli és irányítja a bejövő és kimenő hálózati forgalmat, és szabályokat alkalmaz annak meghatározására, hogy mely forgalom engedélyezett vagy tiltott. A tűzfalak kulcsfontosságú eszközök a számítógépes hálózatok és rendszerek védelmében, mivel segítenek megakadályozni a

jogosulatlan hozzáférést és a különböző típusú támadásokat. Mértékegysége: Db (Darab).

- **Tűzfal2 X(A9):** A tűzfal egy hálózatzbiztonsági rendszer, amely figyeli és irányítja a bejövő és kimenő hálózati forgalmat, és szabályokat alkalmaz annak meghatározására, hogy mely forgalom engedélyezett vagy tiltott. A tűzfalak kulcsfontosságú eszközök a számítógépes hálózatok és rendszerek védelmében, mivel segítenek megakadályozni a jogosulatlan hozzáférést és a különböző típusú támadásokat. Ez a tűzfal nagyobb teljesítménnyel rendelkezik mint az előző. Mértékegysége: Db (Darab).
- **Wifi AP1 X(A10):** A WiFi AP, azaz WiFi Access Point (vö. vezeték nélküli hozzáférési pont), egy olyan eszköz, amely lehetővé teszi a vezeték nélküli eszközök számára, hogy csatlakozzanak egy vezetékes hálózathoz. A WiFi AP a vezetékes hálózatot (pl. Ethernet) alakítja át vezeték nélküli hálózattá, így a WiFi-képes eszközök (pl. okostelefonok, laptopok, tabletek) csatlakozhatnak az internethez és más hálózati erőforrásokhoz. Mértékegysége: Db (Darab).
- **Wifi AP2 X(A11):** A WiFi AP, azaz WiFi Access Point (vö. vezeték nélküli hozzáférési pont), egy olyan eszköz, amely lehetővé teszi a vezeték nélküli eszközök számára, hogy csatlakozzanak egy vezetékes hálózathoz. A WiFi AP a vezetékes hálózatot (pl. Ethernet) alakítja át vezeték nélküli hálózattá, így a WiFi-képes eszközök (pl. okostelefonok, laptopok, tabletek) csatlakozhatnak az internethez és más hálózati erőforrásokhoz. Ez az eszköz nagyobb teljesítménnyel rendelkezik és drágább is mint az előző. Mértékegysége: Db (Darab).
- **Wifi AP3 X(A12):** A WiFi AP, azaz WiFi Access Point (vö. vezeték nélküli hozzáférési pont), egy olyan eszköz, amely lehetővé teszi a vezeték nélküli eszközök számára, hogy csatlakozzanak egy vezetékes hálózathoz. A WiFi AP a vezetékes hálózatot (pl. Ethernet) alakítja át vezeték nélküli hálózattá, így a WiFi-képes eszközök (pl. okostelefonok, laptopok, tabletek) csatlakozhatnak az internethez és más hálózati erőforrásokhoz. Ez az eszköz még nagyobb teljesítménnyel rendelkezik és még drágább is mint az előző. Mértékegysége: Db (Darab).
- **UPS1 X(A13):** Az UPS (Uninterruptible Power Supply) vagy szünetmentes tápegység egy olyan eszköz, amely védelmet nyújt az elektromos hálózat kimaradásai és zavarai ellen azáltal, hogy folyamatos áramellátást biztosít a csatlakoztatott eszközöknek rövid időre, amikor az elektromos hálózat nem áll rendelkezésre. Az UPS-ek különösen fontosak a számítógépes rendszerek, adatközpontok, orvosi berendezések és más

kritikus eszközök számára, ahol az áramkimaradás adatvesztést, rendszerleállást vagy károsodást okozhat. Mértékegysége: Db (Darab).

- **UPS2 X(A14):** Az UPS (Uninterruptible Power Supply) vagy szünetmentes tápegység egy olyan eszköz, amely védelmet nyújt az elektromos hálózat kimaradásai és zavarai ellen azért, hogy folyamatos áramellátást biztosít a csatlakoztatott eszközöknek rövid időre, amikor az elektromos hálózat nem áll rendelkezésre. Az UPS-ek különösen fontosak a számítógépes rendszerek, adatközpontok, orvosi berendezések és más kritikus eszközök számára, ahol az áramkimaradás adatvesztést, rendszerleállást vagy károsodást okozhat. Ez az UPS nagyobb akkumulátor kapacitással és teljesítménnyel rendelkezik mint az előző. Mértékegysége: Db (Darab).
- **Patch panel optika1 X(A15):** Az optikai patch panel (más néven optikai elosztó panel) egy olyan eszköz, amely az optikai kábelek kezelésére és csatlakoztatására szolgál. Az ilyen panelek elengedhetetlenek az optikai hálózatokban, mert rendezett és szervezett módon teszik lehetővé az optikai szálak csatlakoztatását, elosztását és átkonfigurálását. Mértékegysége: Db (Darab).
- **Patch panel réz1 X(A16):** A réz patch panel egy olyan eszköz, amely a rézalapú hálózati kábelek, például az Ethernet kábelek (Cat5e, Cat6, Cat6a, Cat7 stb.) csatlakoztatására és kezelésére szolgál. Ezek a panelek lehetővé teszik, hogy a hálózati eszközök, mint például szerverek, switchek és munkaállomások, rendezett és szervezett módon csatlakozhassanak egy hálózathoz. Mértékegysége: Db (Darab).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	Rendszerterv / Részegységek	X(A1) Szerver1 (Szerver0*) (db)	X(A2) Szerver2 (Szerver4*) (db)	X(A3) Rackszekrény1 (db)	X(A4) Rackszekrény2 (db)	X(A5) "Core" switch1 (db)	X(A6) Normál switch1 (db)	X(A7) Normál switch2 (db)	X(A8) Tűzfal1 (db)	X(A9) Tűzfal2 (db)	X(A10) Wifi AP1 (db)	X(A11) Wifi AP2 (db)	X(A12) Wifi AP3 (db)	X(A13) UPS1 (db)	X(A14) UPS2 (db)	X(A15) Patch panel optika1 (db)	X(A16) Patch panel réz1 (db)	Y = Ár (USD)
1	Rendszerterv1 - O1	1	0	1	0	0	4	0	1	0	10	0	0	1	0	3	3	147238
2	Rendszerterv2 - O2	2	0	1	0	0	4	1	1	0	15	5	0	1	0	4	4	259458
3	Rendszerterv3 - O3	1	0	2	0	1	3	0	1	0	20	3	3	1	0	5	2	155869
4	Rendszerterv4 - O4	1	0	1	0	1	2	2	2	0	0	14	2	1	0	2	2	184330
5	Rendszerterv5 - O5	2	0	2	0	1	3	0	2	0	0	24	4	0	1	4	6	304036
6	Rendszerterv6 - O6	0	1	1	1	1	0	4	0	1	0	10	20	0	1	5	4	164355
7	Rendszerterv7 - O7	0	1	0	2	1	0	3	0	2	0	5	15	0	1	3	4	199070
8	Rendszerterv8 - O8	0	2	0	1	1	0	6	0	1	0	20	20	0	1	2	3	245539
9	Rendszerterv9 - O9	0	1	0	2	1	0	5	0	2	0	10	30	0	1	3	3	217975
10	Rendszerterv10 - O10	0	2	0	2	1	0	4	0	2	0	3	40	0	1	6	4	296430
11																		
12																		
13	irány	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	korreláció	0.219	0.339	-0.159	0.207	0.130	-0.114	0.172	0.041	0.207	-0.333	0.452	0.348	-0.463	0.463	0.206	0.668	
15																		
16																		
17	Ár (USD)	106879	74159	572	863	3175	1369	1830	27210	43196	203	451	871	4348	12010	166	75	
18																		
19	O(n) = Objektum X(A)n = Attributum																	

19. ábra - OAM

Saját forrás2: „Adattábla (nyers)” munkalap A1-R19 tartománya / Jelmagyarázat: a kiindulási adattáblán láthatóak a sor és az oszlopfejlécek, illetve a mértékegységek is. Ezen kívül az irányokat és a korrelációt is feltüntettem. Az Ár (USD) mezőben a hardver elemek egységárai láthatóak, a táblázatban pedig a darabszámuk, ebből adódik az R oszlop, azaz az össz. ár, amit a továbbiakban csak Y=ÁR (USD)-vel jelölök.

4.2.3. Módszerek

A módszerek, a 3.2.3. Módszerek c. fejezetben már bemutatottakkal teljesen megegyeznek.

4.3. Elemzés - 1

Ebben a fejezetben az előzőkhez hasonlóan bemutatom az elemzési folyamat lépéseit. Miután elkészült az OAM (19. ábra), ezt kiegészítettem egyéb adatokkal is, ami később hasznos lehet, az irányokkal és a korrelációval. Az irány 0, ha egy érték: „minél nagyobb annál jobb”, illetve 1, ha: „minél kisebb, annál jobb”. A korreláció jelzi két tetszőleges érték közötti lineáris kapcsolat nagyságát és irányát (avagy ezek egymáshoz való viszonyát).

4.3.1. Sorszámozás

Ezt követően az OAM-ban meglévő adatokat rangsoroltam egy ehhez hasonló táblázatban (20. ábra). Ebből sok hasznos információt meglehetett állapítani nagyon könnyen, összemérve versenytársaikkal, az adott adathoz melyik rendszerterv részegység paramétere volt az első, vagy utolsó helyen. Ahogy az előzőekben, itt is azért nagyon fontos ez a táblázat elkészítése, mert ez által megkapjuk a szükséges számokat a COCO STD modell (árelőny-termelési függvény) elkészítéséhez/becsléséhez.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	Rendszerterv / Részegységek	X(A1) Szerver1 (sorszám)	X(A2) Szerver2 (Sorver4) (sorszám)	X(A3) Rackszekrény1 (sorszám)	X(A4) Rackszekrény2 (sorszám)	X(A5) "Core" switch1 (sorszám)	X(A6) Normál switch1 (sorszám)	X(A7) Normál switch2 (sorszám)	X(A8) Tűzfal1 (sorszám)	X(A9) Tűzfal2 (sorszám)	X(A10) Wifi AP1 (sorszám)	X(A11) Wifi AP2 (sorszám)	X(A12) Wifi AP3 (sorszám)	X(A13) UPS1 (sorszám)	X(A14) UPS2 (sorszám)	X(A15) Patch panel optika1 (sorszám)	X(A16) Patch panel réz1 (sorszám)	Y = Ár (USD)
1	Rendszerterv1 - O1	3	6	3	6	9	1	8	3	6	3	10	9	1	7	6	6	147238
2	Rendszerterv2 - O2	1	6	3	6	9	1	7	3	6	2	6	9	1	7	4	2	259458
3	Rendszerterv3 - O3	3	6	1	6	1	3	8	3	6	1	8	7	1	7	2	9	155869
4	Rendszerterv4 - O4	3	6	3	6	1	5	6	1	6	4	3	8	1	7	9	9	184330
5	Rendszerterv5 - O5	1	6	1	6	1	3	8	1	6	4	1	6	5	1	4	1	304036
6	Rendszerterv6 - O6	6	3	3	4	1	6	3	6	4	4	4	3	5	1	2	2	164355
7	Rendszerterv7 - O7	6	3	7	1	1	6	5	6	1	4	6	5	5	1	6	2	199070
8	Rendszerterv8 - O8	6	1	7	4	1	6	1	6	4	4	2	3	5	1	9	6	245539
9	Rendszerterv9 - O9	6	3	7	1	1	6	2	6	1	4	4	2	5	1	6	6	217975
10	Rendszerterv10 - O10	6	1	7	1	1	6	3	6	1	4	8	1	5	1	1	2	296430

20. ábra - Rangsorolás (OAM)

Saját forrás2: „Rangsorok (OAM)” munkalap A3-R13 tartománya / Jelmagyarázat: a táblázat rangsorszámokat tartalmaz. A piros csillag azt jelenti, hogy az adott szerverek a korábbi server_price_oam.xlsx táblázatból lettek kiválasztva.

4.3.2. Naiv becslés

Az előzőkhez hasonlóan itt is elkészítettem egy naiv becslési táblázatot összehasonlítási céllal (21. ábra). Ebben az esetben a "naiv értékelés" nem negatív értelemben használandó, csupán azt jelenti, hogy az értékelési módszer nem optimalizált. Ez szolgál majd kontrasztként a robot modell számításainak bemutatásához. Kiszámoltam a sorszámok átlagát (vö. ÁTLAG függvény), majd ezek előnyösségi sorrendjét. Ezeken kívül kiszámoltam a sorszámok átlagának inverzét, naiv ár-teljesítmény arányt, továbbá ezek előnyösségi sorrendjét.

	A	S	T	U	V	W
	Rendszerterv / Részegységek	Sorszámok átlaga	Sorszám- átlagok előnyössége (sorrend)	Sorszámok átlagának inverze (átlag)	Naiv ár/teljesítmény- arány (USD/sorszámátlag; $V=R/U$)	Naiv ár/teljesítmény- arányok előnyössége (sorrend)
1						
2	Rendszerterv1 - O1	5.5	10	4.5	32392	3
3	Rendszerterv2 - O2	4.4	7	5.6	46033	9
4	Rendszerterv3 - O3	5.0	8	5.0	31174	2
5	Rendszerterv4 - O4	5.4	9	4.6	39757	6
6	Rendszerterv5 - O5	3.6	2	6.4	47777	10
7	Rendszerterv6 - O6	3.6	2	6.4	25827	1
8	Rendszerterv7 - O7	4.3	5	5.7	34758	4
9	Rendszerterv8 - O8	4.3	5	5.7	42872	7
10	Rendszerterv9 - O9	3.9	4	6.1	35787	5
11	Rendszerterv10 - O10	3.5	1	6.5	45288	8

21. ábra - Naiv becslések

Saját forrás2: „Rangsorok (OAM)” munkalap A1-A11 és S1-RW11 tartománya /Jelmagyarázat: Minél zöldebb annál jobb, minél pirosabb annál rosszabb. Különböző átlag számításokkal -naiv becslésként- megkapjuk, hogy a „Sorszám átlagok előnyössége” oszlopa szerint a legdrágább szerver a legjobb, sorszámokkal illusztrálva. A „Naiv ár/teljesítmény arány” oszlop szerint a legolcsóbb szerver a legjobb, illetve a legdrágább a legrosszabb.

Ebből következően ugyanezt mutatja meg a „Naiv ár/teljesítmény arányok előnyössége” oszlop, csak sorszámokkal. A „Sorszámok átlaga”, azaz az S oszlop értékei, az attribútumok rangsorszámának átlagát mutatják meg. Az attribútumokat nem tudtam feltüntetni az ábrán, mert a táblázat túl széles lenne. Ez az 20. ábrán megtekinthető.

4.3.3. Adatok feldolgozása mesterséges intelligenciával

A korábbiakhoz hasonlóan a legfontosabb része a feladatnak maga a COCO (vö. Component-based Object Comparison for Objectivity) online modell elkészítése (az árelőny-becslés levezetése) volt, mivel ezzel meg lehetett adni a végeredményét az egész feladatnak. Az online alkalmazás dolgozta fel a rangsorolási táblázat adatait, és abból vezette le optimalizáció keretében az eredményt.

A 22. ábrán van a lényeg, mivel itt vannak a tényleges becslés-komponensek, ami megmutatja, hogy mennyi az ár-érték aránya az adott szervernek attribútumonkénti és adott esetekben a helyezésenkénti lebontásban. Ezen kívül még olyan hasznos információval is szolgál, hogy mennyivel térnek el az alapárhoz képest a Delta/Tény résznél.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	COCO:STD	X(A1) Server1 (USD)	X(A2) Server2 (USD)	X(A3) Racksekrény1 (USD)	X(A4) Racksekrény2 (USD)	X(A5) "Core" switch1 (USD)	X(A6) Normál switch1 (USD)	X(A7) Normál switch2 (USD)	X(A8) Tűzfal1 (USD)	X(A9) Tűzfal2 (USD)	X(A10) WiFi AP1 (USD)	X(A11) WiFi AP2 (USD)	X(A12) WiFi AP3 (USD)	X(A13) UPS1 (USD)	X(A14) UPS2 (USD)	X(A15) Patch panel optikai (USD)	X(A16) Patch panel véri (USD)	Becsülés (USD; R=SUM(B:Q))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; U=S/R)
43	Rendszerterv1 - O1	0	0	0	0	0	0	0	0	0	147238	0	0	0	0	0	0	147238	147238	0	0
45	Rendszerterv2 - O2	0	0	0	0	0	0	103589	0	0	155869	0	0	0	0	0	0	259458	259458	0	0
46	Rendszerterv3 - O3	0	0	0	0	0	0	0	0	0	155869	0	0	0	0	0	0	155869	155869	0	0
47	Rendszerterv4 - O4	0	0	0	0	0	0	103589	0	0	0	80741	0	0	0	0	0	184330	184330	0	0
48	Rendszerterv5 - O5	0	0	0	0	0	0	0	0	0	0	304036	0	0	0	0	0	304036	304036	0	0
49	Rendszerterv6 - O6	0	0	0	0	0	0	164355	0	0	0	0	0	0	0	0	0	164355	164355	0	0
50	Rendszerterv7 - O7	0	0	0	34715	0	0	164355	0	0	0	0	0	0	0	0	0	199070	199070	0	0
51	Rendszerterv8 - O8	0	0	0	0	0	0	164798	0	0	0	80741	0	0	0	0	0	245539	245539	0	0
52	Rendszerterv9 - O9	0	0	0	34715	0	0	164798	0	0	0	0	18462	0	0	0	0	217975	217975	0	0
53	Rendszerterv10 - O10	0	0	0	34715	0	0	164355	0	0	0	0	97360	0	0	0	0	296430	296430	0	0

22. ábra - COCO elemzés (STD eredmény)

Saját forrás2: COCO STD (manuális eredmény) munkalap A43-U53 tartománya / Jelmagyarázat: a lépcsősfüggvény standard (STD) változatában nincs szigorú monotonitás mint az Y0-ban, tehát különböző teljesítmény rangsorszintekhez azonos cseredollár érték tartozhat, például X(A7). A különböző háttérszámítások mellett látható, hogy a „Becsülés” oszlopban megkaptuk az új árakat, összehasonlítva az „Y=Ár (USD)” oszloppal látható, hogy nincs eltérés. „Becsülés” oszlop=B-től összeadjuk az összegeket Q-ig. Minden érték mértékegysége USD, kivéve a „Delta/Tény” oszlopot, mely %.

Ismétlésként megjegyzem, hogy a COCO STD elemzést csak úgy lehet elvégezni, ha minden objektumhoz tartozik egy ár (Y), és ezeknél az objektumoknál minimum kettő azonos attribútum mentén összehasonlíthatónak kell lenniük. Az összehasonlított attribútumoknál nem szabad hiányoznia semmilyen adatnak sem, mivel akkor nem elvégezhető a modell-számítás.

4.3.4. Eredmény, konklúzió

A 23. ábrán látható a COCO STD elemzés eredménye.

	A	R	S	T	U
	COCO:STD	Becsülés (USD; R=SUM(B:Q))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; U=S/R)
43					
44	Rendszerterv1 - O1	147238	147238	0	0
45	Rendszerterv2 - O2	259458	259458	0	0
46	Rendszerterv3 - O3	155869	155869	0	0
47	Rendszerterv4 - O4	184330	184330	0	0
48	Rendszerterv5 - O5	304036	304036	0	0
49	Rendszerterv6 - O6	164355	164355	0	0
50	Rendszerterv7 - O7	199070	199070	0	0
51	Rendszerterv8 - O8	245539	245539	0	0
52	Rendszerterv9 - O9	217975	217975	0	0
53	Rendszerterv10 - O10	296430	296430	0	0

23. ábra - A tényleges és a becsült árak távolsága, a háttérszámítások nélkül (USD)

Saját forrás2: „COCO STD (manuális eredmény)” munkalap A43-A53 és R43-U53 tartománya / Jelmagyarázat:

Minden érték mértékegysége USD, kivéve a „Delta/Tény” oszlopot, mely %.

A végeredmény azt mutatja ennél az elemzésnél is, hogy minden informatikai rendszerterv-variáns korrektül van árazva, gyakorlatilag nem mutatható ki eltérés. A validitás ellenőrzésére inverzet nem volt értelme készíteni. Az inverzről már korábban tettem említést a 3.3.4. Eredmény, konklúzió c. fejezetben.

Tovább kell menjünk, hiszen nekem célom megállapítani, hogy melyik rendszerterv-variánsnak milyen az ár-teljesítmény viszonya.

4.4. Elemzés - 2

Amennyiben a COCO STD modell eredménye az, hogy minden rendszerterv-variáns korrektül van árazva, ismét alternatív megoldást kell keresni.

4.4.1. Alternatív megoldás

Amennyiben a fenti eredményt kapjuk, ismét a korábbi megoldással kell előállnunk, melynek neve emlékeztül: COCO STEP IX.

Ismét egy OAM sorozatot készítünk, ahol megállapítjuk az alsó és felső határokat. Mindezt úgy, hogy veszünk egy irreálisan alacsony és magas árat. Mindebből dupla annyi táblázat készül ahány objektumunk (vö. tervvariáns) van, tehát jelen esetben ismét 20db, ahogy korábban. A 24. ábrán látható milyen OAM táblázatok készültek a „Rendszerterv1 - O1” elnevezésű objektumra vetítve (vö. Objektum 1). Ezt meg kell ismételni az összes objektumra.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	O1_alsó	X(A1) Szerver1 (sorszám)	X(A2) Szerver2 (sorszám)	X(A3) Rackszekrény1 (sorszám)	X(A4) Rackszekrény2 (sorszám)	X(A5) "Core" switch1 (sorszám)	X(A6) Normál switch1 (sorszám)	X(A7) Normál switch2 (sorszám)	X(A8) Tűzfal1 (sorszám)	X(A9) Tűzfal2 (sorszám)	X(A10) Wifi AP1 (sorszám)	X(A11) Wifi AP2 (sorszám)	X(A12) Wifi AP3 (sorszám)	X(A13) UPS1 (sorszám)	X(A14) UPS2 (sorszám)	X(A15) Patch panel optika1 (sorszám)	X(A16) Patch panel réz1 (sorszám)	Y = Ár (USD)	COCO STD becsült alsóhatár ár (USD)	Eredeti ár (USD)
1	Rendszerterv1 - O1	3	6	3	6	9	1	8	3	6	3	10	9	1	7	6	6	0	0	147238
2	Rendszerterv2 - O2	1	6	3	6	9	1	7	3	6	2	6	9	1	7	4	2	259458		
3	Rendszerterv3 - O3	3	6	1	6	1	3	8	3	6	1	8	7	1	7	2	9	155869		
4	Rendszerterv4 - O4	3	6	3	6	1	5	6	1	6	4	3	8	1	7	9	9	184330		
5	Rendszerterv5 - O5	1	6	1	6	1	3	8	1	6	4	1	6	5	1	4	1	304036		
6	Rendszerterv6 - O6	6	3	3	4	1	6	3	6	4	4	4	3	5	1	2	2	164355		
7	Rendszerterv7 - O7	6	3	7	1	1	6	5	6	1	4	6	5	5	1	6	2	199070		
8	Rendszerterv8 - O8	6	1	7	4	1	6	1	6	4	4	2	3	5	1	9	6	245539		
9	Rendszerterv9 - O9	6	3	7	1	1	6	2	6	1	4	4	2	5	1	6	6	217975		
10	Rendszerterv10 - O10	6	1	7	1	1	6	3	6	1	4	8	1	5	1	1	2	296430		
11																				
12	O1_felső	X(A1) Szerver1 (sorszám)	X(A2) Szerver2 (sorszám)	X(A3) Rackszekrény1 (sorszám)	X(A4) Rackszekrény2 (sorszám)	X(A5) "Core" switch1 (sorszám)	X(A6) Normál switch1 (sorszám)	X(A7) Normál switch2 (sorszám)	X(A8) Tűzfal1 (sorszám)	X(A9) Tűzfal2 (sorszám)	X(A10) Wifi AP1 (sorszám)	X(A11) Wifi AP2 (sorszám)	X(A12) Wifi AP3 (sorszám)	X(A13) UPS1 (sorszám)	X(A14) UPS2 (sorszám)	X(A15) Patch panel optika1 (sorszám)	X(A16) Patch panel réz1 (sorszám)	Y = Ár (USD)	COCO STD becsült felsőhatár ár (USD)	Eredeti ár (USD)
13	Rendszerterv1 - O1	3	6	3	6	9	1	8	3	6	3	10	9	1	7	6	6	999999	629728	147238
14	Rendszerterv2 - O2	1	6	3	6	9	1	7	3	6	2	6	9	1	7	4	2	259458		
15	Rendszerterv3 - O3	3	6	1	6	1	3	8	3	6	1	8	7	1	7	2	9	155869		
16	Rendszerterv4 - O4	3	6	3	6	1	5	6	1	6	4	3	8	1	7	9	9	184330		
17	Rendszerterv5 - O5	1	6	1	6	1	3	8	1	6	4	1	6	5	1	4	1	304036		
18	Rendszerterv6 - O6	6	3	3	4	1	6	3	6	4	4	4	3	5	1	2	2	164355		
19	Rendszerterv7 - O7	6	3	7	1	1	6	5	6	1	4	6	5	5	1	6	2	199070		
20	Rendszerterv8 - O8	6	1	7	4	1	6	1	6	4	4	2	3	5	1	9	6	245539		
21	Rendszerterv9 - O9	6	3	7	1	1	6	2	6	1	4	4	2	5	1	6	6	217975		
22	Rendszerterv10 - O10	6	1	7	1	1	6	3	6	1	4	8	1	5	1	1	2	296430		

24. ábra - STEP IX sorozat OAM táblázatai, alsó-felső határ (Objektum 1)

Saját forrás2: „STEP IX OAM sorozat” munkalap A1-T23 tartománya / Jelmagyarázat: a sárgával kijelölt cellákban láthatjuk az irreálisan alacsony- illetve magas árat, amivel újból elvégezzük a számításokat, tehát beküldünk a COCO STD online modellező robotba, amely optimalizálás keretében számításokat végez, a hasonlóságelemzés módszerét felhasználva.

Az OAM táblázatokat a COCO STD online robotba egyesével beküldtem, majd a végeredményeket (25. és 26. ábra) kimásoltam a Saját forrás2 „STEP IX COCO STD” munkalapjára.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	COCO:STD	X(A1) Szerver1 (Szerver6*) (USD)	X(A2) Szerver2 (Szerver4*) (USD)	X(A3) Rackszek rény1 (USD)	X(A4) Rackszek rény2 (USD)	X(A5) "Core" switch1 (USD)	X(A6) Normál switch1 (USD)	X(A7) Normál switch2 (USD)	X(A8) Tűzfal1 (USD)	X(A9) Tűzfal2 (USD)	X(A10) Wifi AP1 (USD)	X(A11) Wifi AP2 (USD)	X(A12) Wifi AP3 (USD)	X(A13) UPS1 (USD)	X(A14) UPS2 (USD)	X(A15) Patch panel optika1 (USD)	X(A16) Patch panel réz1 (USD)	Becslés (USD; R=SUM(B:Q))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; U=5/R)
43	Rendszerterv1 - O1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	Rendszerterv2 - O2	0	0	0	0	0	0	103589	0	0	155869	0	0	0	0	0	0	259458	259458	0	0
45	Rendszerterv3 - O3	0	0	0	0	0	0	0	0	0	155869	0	0	0	0	0	0	155869	155869	0	0
46	Rendszerterv4 - O4	0	0	0	0	0	0	103589	0	0	0	80741	0	0	0	0	0	184330	184330	0	0
47	Rendszerterv5 - O5	0	0	0	0	0	0	0	0	0	0	304036	0	0	0	0	0	304036	304036	0	0
48	Rendszerterv6 - O6	0	0	0	0	0	0	164355	0	0	0	0	0	0	0	0	0	164355	164355	0	0
49	Rendszerterv7 - O7	0	0	0	34715	0	0	164355	0	0	0	0	0	0	0	0	0	199070	199070	0	0
50	Rendszerterv8 - O8	0	0	0	0	0	0	164798	0	0	0	80741	0	0	0	0	0	245539	245539	0	0
51	Rendszerterv9 - O9	0	0	0	34715	0	0	164798	0	0	0	0	18462	0	0	0	0	217975	217975	0	0
52	Rendszerterv10 - O10	0	0	0	34715	0	0	164355	0	0	0	0	97360	0	0	0	0	296430	296430	0	0

25. ábra - STEP IX COCO STD elemzés eredménye, alsó határ (Objektum 1)

Saját forrás2s: „STEP IX COCO STD” munkalap A43-U53 tartománya / Jelmagyarázat: a sárgával kijelölt cellában láthatjuk az irreálisan alacsony árat, a különböző elvégzett háttérszámítások mellett. A „Becslés” oszlop árai, amik USD-ban vannak megadva, a B-től a Q oszlopig szereplő árak összeadva.

	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
	COCO:STD	X(A1) Szerver1 (Szerver6*) (USD)	X(A2) Szerver2 (Szerver4*) (USD)	X(A3) Rackszek rény1 (USD)	X(A4) Rackszek rény2 (USD)	X(A5) "Core" switch1 (USD)	X(A6) Normál switch1 (USD)	X(A7) Normál switch2 (USD)	X(A8) Tűzfal1 (USD)	X(A9) Tűzfal2 (USD)	X(A10) Wifi AP1 (USD)	X(A11) Wifi AP2 (USD)	X(A12) Wifi AP3 (USD)	X(A13) UPS1 (USD)	X(A14) UPS2 (USD)	X(A15) Patch panel optika1 (USD)	X(A16) Patch panel réz1 (USD)	Becslés (USD; AN=SUM(X:AM))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; AQ=AO/AN)
43	Rendszerterv1 - O1	0	0	25787	0	0	551794	0	0	0	52148	0	0	0	0	0	0	629728	999999	370271	37
44	Rendszerterv2 - O2	0	0	25787	0	0	551794	0	0	0	52148	0	0	0	0	0	0	629728	259458	-370270	-143
45	Rendszerterv3 - O3	0	0	25787	0	77935	0	0	0	52148	0	0	0	0	0	0	0	155869	155869	0.1	0
46	Rendszerterv4 - O4	0	0	25787	0	77935	0	0	14231	0	0	66379	0	0	0	0	0	184330	184330	0.1	0
47	Rendszerterv5 - O5	0	0	25787	0	77935	0	0	14231	0	0	181841	0	0	4243	0	0	304036	304036	0.2	0
48	Rendszerterv6 - O6	0	0	25787	0	77935	0	56391	0	0	0	0	0	0	4243	0	0	164355	164355	0.1	0
49	Rendszerterv7 - O7	0	0	0	60502	77935	0	56391	0	0	0	0	0	0	4243	0	0	199070	199070	0.1	0
50	Rendszerterv8 - O8	0	39910	0	0	77935	0	57074	0	0	0	66379	0	0	4243	0	0	245539	245539	-0.4	0
51	Rendszerterv9 - O9	0	0	0	60502	77935	0	57074	0	0	0	0	18223	0	4243	0	0	217975	217975	-0.4	0
52	Rendszerterv10 - O10	0	39910	0	60502	77935	0	56391	0	0	0	0	57451	0	4243	0	0	296430	296430	-0.4	0

26. ábra - STEP IX COCO STD elemzés eredménye, felső határ (Objektum 1)

Saját forrás2: „STEP IX COCO STD” munkalap W43-AQ53 tartománya / Jelmagyarázat: a sárgával kijelölt cellában láthatjuk az irreálisan magas árat, a különböző elvégzett háttérszámítások mellett. A „Becslés” oszlop árai, amik USD-ban vannak megadva, az X-től az AM oszlopig szereplő árak összeadva.

Az itt újonnan megkapott alsóhatár és felsőhatár becslési értékeket (vö. 25. és 26. ábrán sárgával jelölt cellák) bemásoltam a 24. ábrán látható Y=Ár (USD) oszlop megfelelő celláiba, tehát a Saját forrás2: „STEP IX OAM sorozat” munkalap R2 cellájába, és a Saját forrás2: „STEP IX OAM sorozat” munkalap R14 cellájába.

Az alsó és felső határok megállapítása után elkészült a STEP IX OAM tábla (27. ábra). A táblázat tartalmaz még különböző arány számításokat és fedezet számítást is, illetve az irányok is feltüntetésre kerülnek, továbbá az Y0 fiktív változó is bevezetésre került.

	A	B	C	D	E	F	G	H	I	J	K
1						irány (minél...annál előnyösebb)	0	0	1	1	
2	STEP IX	Alsóhatár min. (USD)	Tényleges ár (USD)	Felsőhatár max (USD)		STEP IX OAM	X(A1) min%	X(A2) max%	X(A3) lent/fent-arány	X(A4) fedezet %	Y0 (pontszám)
3	Rendszerterv1 - O1	0	147238	629728		Rendszerterv1 - O1	0%	428%	30.52%	23.38%	100000
4	Rendszerterv2 - O2	73619	259458	888794		Rendszerterv2 - O2	28%	343%	29.53%	31.83%	100000
5	Rendszerterv3 - O3	0	155869	999998		Rendszerterv3 - O3	0%	642%	18.47%	15.59%	100000
6	Rendszerterv4 - O4	0	184330	807814		Rendszerterv4 - O4	0%	438%	29.56%	22.82%	100000
7	Rendszerterv5 - O5	0	304036	999999		Rendszerterv5 - O5	0%	329%	43.69%	30.40%	100000
8	Rendszerterv6 - O6	0	164355	686390		Rendszerterv6 - O6	0%	418%	31.48%	23.94%	100000
9	Rendszerterv7 - O7	0	199070	595087		Rendszerterv7 - O7	0%	299%	50.27%	33.45%	100000
10	Rendszerterv8 - O8	0	245539	999999		Rendszerterv8 - O8	0%	407%	32.54%	24.55%	100000
11	Rendszerterv9 - O9	28520	217975	770984		Rendszerterv9 - O9	13%	354%	34.26%	29.36%	100000
12	Rendszerterv10 - O10	17358	296430	999999		Rendszerterv10 - O10	6%	337%	39.67%	30.17%	100000

27. ábra - STEP-IX OAM (min./max.) Y0 elemzéshez

Saját forrás: „Eredmények OAM” munkalap A1-L12 tartománya / Jelmagyarázat: Az A1-D12 táblázatban látható a korábban kiszámolt minimum árak, a tényleges/eredeti árak, illetve a korábban kiszámolt maximum árak is, a 25. és 26. ábra szerint, ezek mértékegysége USD. Az F1-K12 táblázatban különböző matematikai számítások eredményei látható, melynek mértékegysége %, illetve egy úgynevezett Y0 fiktív változó/pontszám, ami bevezetésre került. Továbbá, feltüntettem az irányokat is. Ezek szükséges, hogy a későbbiekben átalakított táblázat feldolgozható legyen a mesterséges intelligencia által, ez a 28. ábrán lesz látható.

4.4.2. Sorszámozás

Ezek a számítások ismét elősegítik, hogy egy olyan rangsorolt OAM táblát (28. ábra) készítsünk, amit a COCO Y0 anti-diszkriminációs, hasonlóságelemző online modellezőbe be tudunk tölteni. Ahogy már korábban említettem, az Y0 változó fiktív és kellően/megfelelően nagy érték ahhoz, hogy a modellező robotnak teret adjunk.

	M	N	O	P	Q	R
1						
2	STEP IX OAM	X(A1) min% (sorszám)	X(A2) max% (sorszám)	X(A3) lent/fent-arány (sorszám)	X(A4) fedezet % (sorszám)	Y0 (pontszám)
3	Rendszerterv1 - O1	4	3	4	3	100000
4	Rendszerterv2 - O2	1	7	2	9	100000
5	Rendszerterv3 - O3	4	1	1	1	100000
6	Rendszerterv4 - O4	4	2	3	2	100000
7	Rendszerterv5 - O5	4	9	9	8	100000
8	Rendszerterv6 - O6	4	4	5	4	100000
9	Rendszerterv7 - O7	4	10	10	10	100000
10	Rendszerterv8 - O8	4	5	6	5	100000
11	Rendszerterv9 - O9	2	6	7	6	100000
12	Rendszerterv10 - O10	3	8	8	7	100000

28. ábra - Rangsorolás (OAM)

Saját forrás: „Eredmények OAM” munkalap M2-R12 tartománya / Jelmagyarázat: A táblázatban különböző matematikai számítások eredményeinek rangsorszámjai láthatóak, illetve egy úgynevezett Y0 fiktív változó/pontszám. Ezek szükségesek ahhoz, hogy táblázat feldolgozható legyen a mesterséges intelligencia által.

Ismét megjegyzem, ahogy már korábban is, hogy minden Y0 képes STD jelleggel árelőny-termelési függvényt generálni, viszont egyetlen STD modell sem alkalmas Y0 kérdések megoldására.

4.4.3. Adatok feldolgozása mesterséges intelligenciával

Az alábbi összefoglaló ábrán látható a COCO Y0 modell eredménye a feldolgozás után. A táblázatot úgy értelmezzük, hogy minél több „Becslés” pontot kap az adott objektum, annál „értékesebb”, tehát győztest hirdethetünk.

	A	B	C	D	E	F	G	H	I
	COCO:Y0	X(A1) (pontszám)	X(A2) (pontszám)	X(A3) (pontszám)	X(A4) (pontszám)	Becslés (pontszám)	Y0 (pontszám)	Delta	Delta/Tény (pontszám/pontszám; I=H/G)
43									
44	Rendszerterv1 - O1	6	99984	6	8	100004	100000	-4	0
45	Rendszerterv2 - O2	16	99980	8	1	100005	100000	-5	-0.01
46	Rendszerterv3 - O3	6	99986	9	10	100011	100000	-11	-0.01
47	Rendszerterv4 - O4	6	99985	7	9	100007	100000	-7	-0.01
48	Rendszerterv5 - O5	6	99978	1	3	99988	100000	12	0.01
49	Rendszerterv6 - O6	6	99983	5	7	100001	100000	-1	0
50	Rendszerterv7 - O7	6	99977	0	0	99983	100000	17	0.02
51	Rendszerterv8 - O8	6	99982	4	6	99998	100000	2	0
52	Rendszerterv9 - O9	15	99981	3	5	100004	100000	-4	0
53	Rendszerterv10 - O10	14	99979	2	4	99999	100000	1	0

29. ábra - COCO elemzés (Y0 eredmény)

Saját forrás2: „Y0” munkalap A43-I53 tartománya / Jelmagyarázat: A „Becslés” oszlopban a mesterséges intelligencia által kapott becslés pontszámok összessége látható, a B oszloptól az E oszlopig összeadva. Minél több pontot kap egy objektum, annál értékesebb, tehát a legtöbb pontszámot kapott objektum a legjobb, illetve ez alapján felállítható egy végső sorrend is.

4.4.4. Végeredmény, konklúzió

Ez a végeredmény már szintén kellően stabil ahhoz, hogy kihirdessem az első három helyezettet, illetve a legrosszabb ár-érték arányú szerveret (30. ábra). Jelen esetben mivel kevés objektumunk van, könnyedén ki tudjuk választani a győztest, viszont amennyiben sokkal több objektumunk lenne akkor egy függvényt használnák a legnagyobb érték megállapítására (vö. MAX függvény). Amint megkaptuk a legnagyobb értéket a holtversenyt is célszerű lehet ellenőriznünk, szintén függvényrel (vö. DARABTELI). Jelen esetben holtverseny nem állt fenn.

	M	N	O	P	Q	R	S	T	U
1									
2	STEP IX OAM	X(A1) min% (sorszám)	X(A2) max% (sorszám)	X(A3) lent/fent-arány (sorszám)	X(A4) fedezet % (sorszám)	Y0 (pontszám)	Becslés (pontszám)	Validitás (1=igen, 0=nem)	Konklúzió (helyezettek)
3	Rendszerterv1 - O1	4	3	4	3	100000	100004	1	
4	Rendszerterv2 - O2	1	7	2	9	100000	100005	1	harmadik legjobb
5	Rendszerterv3 - O3	4	1	1	1	100000	100011	1	legjobb ár/érték arány
6	Rendszerterv4 - O4	4	2	3	2	100000	100007	1	második legjobb
7	Rendszerterv5 - O5	4	9	9	8	100000	99988	1	
8	Rendszerterv6 - O6	4	4	5	4	100000	100001	1	
9	Rendszerterv7 - O7	4	10	10	10	100000	99983	1	legrosszabb ár/érték arány
10	Rendszerterv8 - O8	4	5	6	5	100000	99998	1	
11	Rendszerterv9 - O9	2	6	7	6	100000	100004	1	
12	Rendszerterv10 - O10	3	8	8	7	100000	99999	1	

30. ábra - COCO elemzés (becslés-validitás-konklúzió)

Saját forrás2: „Y0” munkalap M1-U12 tartománya / Jelmagyarázat: A táblázatban különböző matematikai számítások eredményeinek rangsorszámai láthatóak, illetve egy úgynevezett Y0 fiktív változó/pontszám. Ezek szükségesek ahhoz, hogy táblázat feldolgozható legyen a mesterséges intelligencia által. A „Becslés pontszám” oszlopban a mesterséges intelligencia által kapott becslés pontszámok láthatóak. Minél több pontot kap egy objektum, annál értékesebb, tehát a legtöbb pontszámot kapott objektum a legjobb, illetve ez alapján felállítható egy végső sorrend is. Minél zöldebb annál jobb, minél pirosabb annál rosszabb, tehát a kialakult végeredmény látható, illetve a validitás ellenőrzése is megtörtént.

Immár eredményként megkapjuk, hogy vajon lehet-e minden objektum másképp egyforma? Avagy melyik a leginkább ideális objektum? (31. ábra).

	B	C	D
19	1. Közép. és nagyvállalati informatikai rendszertervek ár-teljesítmény vizsgálata. (2024.05.05-i árlista alapján)		
20	1. helyezett: Rendszerterv3		
21	2. helyezett: Rendszerterv4		
22	3. helyezett: Rendszerterv2		
23	Legrosszabb: Rendszerterv7		

31. ábra - Végeredmény (helyezések)

Saját forrás: „Alapadatok” munkalap B19-B23 tartománya / Jelmagyarázat: minél zöldebb annál jobb helyezés, a halvány piros színnel jelölt szerver a legrosszabb ár-érték arányú.

A validitás ellenőrzésére inverzet is készítettem (32. ábra). Ahogy korábban már kifejtettem, az inverz azt jelenti, hogy a legnagyobb sorszámhoz hozzáadunk egyet, majd ebből a számból kivonjuk a meglévő sorszámokat, majd így is lefuttatjuk a modellt.

	M	N	O	P	Q	R
7	COCO:Y0	X(A1) min% (inverz sorszám)	X(A2) max% (inverz sorszám)	X(A3) lent/fent- arány (inverz sorszám)	X(A4) fedezet (inverz sorszám)	Y0 (pontszám)
8	Rendszerterv1 - O1	7	8	7	8	100000
9	Rendszerterv2 - O2	10	4	9	2	100000
10	Rendszerterv3 - O3	7	10	10	10	100000
11	Rendszerterv4 - O4	7	9	8	9	100000
12	Rendszerterv5 - O5	7	2	2	3	100000
13	Rendszerterv6 - O6	7	7	6	7	100000
14	Rendszerterv7 - O7	7	1	1	1	100000
15	Rendszerterv8 - O8	7	6	5	6	100000
16	Rendszerterv9 - O9	9	5	4	5	100000
17	Rendszerterv10 - O10	8	3	3	4	100000

32. ábra - Inverz OAM táblázat

Saját forrás2: „Y0” munkalap M7-R17 tartománya / Jelmagyarázat: A táblázatban látható sorszámok a 28. ábrán látható táblázatban szereplő sorszámok -11-ből kivont- inverz eredményei. Illetve, az úgynevezett Y0 fiktív változó/pontszám is feltüntetésre kerül. Ezek szükségesek ahhoz, hogy az inverz táblázat feldolgozható legyen a mesterséges intelligencia által.

A 33. ábrán a „normál” és inverz OAM táblázatunk COCO Y0 modell végeredményei láthatóak (vö. bal oldal - normál, jobb oldal - inverz), ahol is tisztán látszik, hogy az értéke megegyeznek, csak az előjelük ellentétes, tehát a végeredmény validnak/érvényesnek tekinthető.

A								B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
COCO:Y0								X(A1) (pontszám)	X(A2) (pontszám)	X(A3) (pontszám)	X(A4) (pontszám)	Becslés (pontszám)	Y0 (pontszám)	Delta	Delta/Tény (pontszám/pontszám; I=H/G)	COCO:Y0	X(A1) (pontszám)	X(A2) (pontszám)	X(A3) (pontszám)	X(A4) (pontszám)	Becslés (pontszám)	Y0 (pontszám)	Delta	Delta/Tény (pontszám/pontszám; AB=AA/Z)										
43	Rendszerterv1 - O1								6	99984	6	8	100004	100000	-4	0	44	Rendszerterv1 - O1								49996	49995	3	2	99996	100000	4	0	
44	Rendszerterv2 - O2								16	99980	8	1	100005	100000	-5	-0.01	45	Rendszerterv2 - O2								49986	49999	1	9	99995	100000	5	0.01	
45	Rendszerterv3 - O3								6	99986	9	10	100011	100000	-11	-0.01	46	Rendszerterv3 - O3								49996	49993	0	0	99989	100000	11	0.01	
46	Rendszerterv4 - O4								6	99985	7	9	100007	100000	-7	-0.01	47	Rendszerterv4 - O4								49996	49994	2	1	99993	100000	7	0.01	
47	Rendszerterv5 - O5								6	99978	1	3	99988	100000	12	0.01	48	Rendszerterv5 - O5								49996	50001	8	7	100012	100000	-12	-0.01	
48	Rendszerterv6 - O6								6	99983	5	7	100001	100000	-1	0	49	Rendszerterv6 - O6								49996	49996	4	3	99999	100000	1	0	
49	Rendszerterv7 - O7								6	99977	0	0	99983	100000	17	0.02	50	Rendszerterv7 - O7								49996	50002	9	10	100017	100000	-17	-0.02	
50	Rendszerterv8 - O8								6	99982	4	6	99998	100000	2	0	51	Rendszerterv8 - O8								49996	49997	5	4	100002	100000	-2	0	
51	Rendszerterv9 - O9								15	99981	3	5	100004	100000	-4	0	52	Rendszerterv9 - O9								49987	49998	6	5	99996	100000	4	0	
52	Rendszerterv10 - O10								14	99979	2	4	99999	100000	1	0	53	Rendszerterv10 - O10								49988	50000	7	6	100001	100000	-1	0	

33. ábra - Validitás ellenőrzés táblázatai (ellentétes előjelek)

Saját forrás2: „Y0” munkalap A43-AB32 tartománya / Jelmagyarázat: a két táblázat egyfajta ellenőrzésre szolgál, amennyiben a „Delta” oszlopban a számok megegyeznek, de előjelük ellentétes, úgy a validitás igazolásra került. A 29. ábra jelmagyarázata bővebb információval szolgál a táblázatról.

A 30. ábrán látható a „validitás” elnevezésű oszlop, ahol 1-es értékek szerepelnek (valid = 1, nem valid = 0). Ezt a „HA” függvénnyel ellenőriztem, az alábbi képlettel: =HA(Y0!H44*Y0!AA44<=0;1;0).

A korábbiakhoz hasonlóan, ideáltól való eltérést nincs értelme vizsgálni.

4.5. COCO STEP IX+

A STEP IX+ módszer, a korábban a 3.2.6. COCO STEP IX c. fejezetben bemutatott (vö. rövidített/gyors módszer) lassabb (vö. teljes) változata. Célja, hogy a gyorsított eljárás potenciális kockázataira rámutasson. A módszer egyben növeli a validitást is, tehát

önellenőrzési körnek is nevezhetjük. Fenti gondolatmenet a COCO output-ok szabályelvű értelmezési világának is része.

Lényege, hogy a [Saját forrás](#) és [Saját forrás2](#) „STEP IX OAM sorozat” munkalapok objektum-attribútum mátrixait tovább vizsgáljuk. Az eddig bemutatott esetekben vettünk egy irreálisan alacsony és irreálisan magas árat, ezekből újabb és újabb OAM táblákat hoztunk létre (9. és 24. ábra), majd ezt vizsgáltuk a mesterséges intelligenciával. Az itt kapott eredményekkel, illetve egyéb számításokat elvégezve egy újabb OAM összeállítás után (12. és 27. ábra), a fiktív változót (vö. $Y0 = \text{pontszám/jóspont}$) hozzáadva kialakítottunk egyfajta végeredményt (15. és 30. ábra). Visszatérve a normál STEP IX OAM sorozatokhoz, bemutatom a szerver-ár teljesítmény vizsgálat „Server - O1” (Objektum 1) alsó határára nézve, hogyan kell a STEP IX+ módszert alkalmazni.

A kiindulási táblázatunk a 34. ábrán látható, amit már korábban ismertettem. Az L oszlopban (vö. Y) árakat láthatunk, melyek mértékegysége továbbra is USA dollár. Jelen esetben, a „Server1 - O1” (Objektum 1) esetében vettem a legkisebb irreális árat (vö. $Y=0$ USD), ez az ábrán sárgával kiemelve látható.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	O1_alsó	X(A1) CPU Xeon (sorszám)	X(A2) CPU magok száma (sorszám)	X(A3) CPU órajel (sorszám)	X(A4) RAM memória (sorszám)	X(A5) NVME SSD meghajtó (sorszám)	X(A6) SATA SSD meghajtó (sorszám)	X(A7) SAS HDD meghajtó (sorszám)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (sorszám)	X(A9) Optikai csatoló külső storage-hoz (sorszám)	X(A10) Quad tápegység (sorszám)	Y = Ár (USD)	COCO STD becsült alsóhatár ár (USD)	Eredeti ár (USD)
1														
2	Server1 - O1	6	10	1	10	6	9	9	7	6	9	0	3337	24734
3	Server2 - O2	6	9	2	9	6	9	9	7	6	9	28778		
4	Server3 - O3	6	8	7	8	6	8	8	7	6	4	42778		
5	Server4 - O4	6	7	3	7	6	7	7	7	6	4	73210		
6	Server5 - O5	6	6	5	5	6	5	1	3	6	4	112185		
7	Server6 - O6	1	5	7	5	4	5	1	3	3	4	105512		
8	Server7 - O7	1	4	3	4	4	1	1	3	3	4	192820		
9	Server8 - O8	1	3	5	3	3	1	1	3	3	3	275005		
10	Server9 - O9	1	1	9	2	2	1	1	1	1	1	361161		
11	Server10 - O10	1	1	9	1	1	1	1	1	1	1	1115893		

34. ábra - STEP IX sorozat OAM táblázata, alsó határ (Objektum 1)

Saját forrás: „STEP IX OAM sorozat” munkalap A1-N11 tartománya / Jelmagyarázat: az ábrán különböző szerver objektumok tulajdonságai láthatóak, ahol a tulajdonságok rangsorszámot kaptak, ezáltal összehasonlíthatóak lesznek egymással, és a mesterséges intelligencia fel tudja dolgozni a táblázatot. Az L oszlop árakat tartalmaz, melyek a különböző szerver konfigurációk ára, USA dollárban megadva. A sárgával jelölt L2 cellában egy irreálisan alacsony ár ($Y=0$ USD) található. Ez az ábra megegyezik a 9. ábra első felével.

Tehát, a 34. ábrán látható a táblázatot beküldjük a COCO STD online robotba, majd a kapott eredményt tovább hasznosítjuk. Az Objektum 1 (vö. Server1 - O1) kapott becslési értékét a 35. ábrán láthatjuk, narancssárgával kiemelve.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	COCO:STD	X(A1) CPU Xeon (USD)	X(A2) CPU magok száma (USD)	X(A3) CPU órajel (USD)	X(A4) RAM memória (USD)	X(A5) NVME SSD meghajtó (USD)	X(A6) SATA SSD meghajtó (USD)	X(A7) SAS HDD meghajtó (USD)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (USD)	X(A9) Optikai csatoló külső storage-hoz (USD)	X(A10) Quad tápegység (USD)	Becsítés (USD; L=SUM(B:K))	Y = Ár (USD)	Delta (USD)	Delta/Tény (USD/USD; O=N/M)
43															
44	Server1 - O1	0	0	3337	0	0	0	0	0	0	0	3337	0	-3337	0
45	Server2 - O2	0	14389	3337	11053	0	0	0	0	0	0	28778	28778	0	0
46	Server3 - O3	0	21389	0	21389	0	0	0	0	0	0	42778	42778	0	0
47	Server4 - O4	0	36605	3337	33269	0	0	0	0	0	0	73210	73210	0	0
48	Server5 - O5	0	52756	3337	52756	0	0	0	0	0	0	108849	112185	3336	3.0
49	Server6 - O6	0	52756	0	52756	0	0	0	0	0	0	105512	105512	-0.1	0
50	Server7 - O7	0	52756	3337	93074	0	43654	0	0	0	0	192820	192820	-0.1	0
51	Server8 - O8	0	52756	3337	175259	0	43654	0	0	0	0	275005	275005	-0.2	0
52	Server9 - O9	0	95834	0	221673	0	43654	0	0	0	0	361161	361161	-0.2	0
53	Server10 - O10	0	95834	0	976404	0	43654	0	0	0	0	1115892	1115893	0.8	0

35. ábra - STEP IX sorozat OAM táblázata, alsó határ becslés (Objektum 1)

Saját forrás: „STEP IX COCO STD” munkalap A43-O53 tartománya / Jelmagyarázat: a sárgával kijelölt cellában láthatjuk az irreálisan alacsony árat, a különböző elvégzett háttérszámítások mellett. A „Becsítés” oszlop árai, amik USD-ban vannak megadva, a B-től a K oszlopig szereplő árak összeadva. Az ábra megegyezik a 10. ábrával.

A kapott eredményt, ami a „Server - O1” elnevezésű objektum (vö. Objektum 1) új becsült értéke, tehát jelen esetben a 3337 USD-t, visszahelyettesítjük az eredeti kiindulási táblázatunkba a 36. ábrán látható módon, citromsárga háttérrel jelölve, tehát az előző (Y=0 USD) ár helyére.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	O1_alsó	X(A1) CPU Xeon (sorszám)	X(A2) CPU magok száma (sorszám)	X(A3) CPU órajel (sorszám)	X(A4) RAM memória (sorszám)	X(A5) NVME SSD meghajtó (sorszám)	X(A6) SATA SSD meghajtó (sorszám)	X(A7) SAS HDD meghajtó (sorszám)	X(A8) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (sorszám)	X(A9) Optikai csatoló külső storage-hoz (sorszám)	X(A10) Quad tápegység (sorszám)	Y = Ár (USD)	COCO STD becsült alsóhatár ár (USD)	Eredeti ár (USD)
1														
2	Server1 - O1	6	10	1	10	6	9	9	7	6	9	3337	3337	24734
3	Server2 - O2	6	9	2	9	6	9	9	7	6	9	28778		
4	Server3 - O3	6	8	7	8	6	8	8	7	6	4	42778		
5	Server4 - O4	6	7	3	7	6	7	7	7	6	4	73210		
6	Server5 - O5	6	6	5	5	6	5	1	3	6	4	112185		
7	Server6 - O6	1	5	7	5	4	5	1	3	3	4	105512		
8	Server7 - O7	1	4	3	4	4	1	1	3	3	4	192820		
9	Server8 - O8	1	3	5	3	3	1	1	3	3	3	275005		
10	Server9 - O9	1	1	9	2	2	1	1	1	1	1	361161		
11	Server10 - O10	1	1	9	1	1	1	1	1	1	1	1115893		

36. ábra - STEP IX sorozat OAM táblázata, alsó határ, új értékkel (Objektum 1)

Saját forrás: „STEP IX OAM sorozat” munkalap A1-N11 tartománya / Jelmagyarázat: A 34. táblázatot felhasználva, az L2 cella eredeti értékét (Y=0 USD), visszahelyettesítjük a mesterséges intelligencia által korábban kiszámolt -és a 35. ábrán is látható- L2 cella 3337 USD értékére. Így egy újabb OAM táblázatot kapunk, amit ismét feldolgozunk a mesterséges intelligencia segítségével. Tesszük ezt mindaddig, amíg a Delta, tehát az eltérés értéke 0 (azaz nulla) nem lesz, az Objektum 1-re nézve. Ez a COCO STEP IX+ módszer.

Az új értékkel kiegészített OAM táblánkat ismét beküldjük a COCO STD online robotba, majd a kapott eredményt tovább használjuk, tehát visszahelyettesítjük az eredeti OAM „Server1 - O1” objektum (vö. Objektum 1) árak helyére, mindaddig amíg 0 eltérést kapunk (vö. Delta értéke = 0).

Ennek a részletesebb levezetése a szakdolgozathoz csatolt „szerver_ar_telj_stepix_gyorsított_teljes21.xlsx” fájlban látható, amelyre a továbbiakban [Saját forrás3](#) néven fogok hivatkozni. A fájlban megtalálható az összes lefutott OAM „végeredménye”, azonban a visszakapott COCO eredmények csak a „Server1 - O1”, azaz Objektum 1 alsó és felső határaitra készült el (vö. „STEP IX+ (telj.,o1_also)” és „STEP IX+ (telj.,o1_felső)” munkalapok).

A 37. ábrán a „Server1 - O1”, tehát az Objektum 1 alsó és felső határainak végeredménye látható, mindaddig amíg az eltérés értéke 0 nem lett. Összefoglalva tehát, ez egy ismétlődő új ár képzési ciklus, amíg az utolsó modell hibája nem tekinthető nullának. Hogy ezt mennyi lépésből kapjuk meg, az nagyon változó.

Vegyük észre, hogy a „Server - O1” tehát az Objektum 1 alsó határok új becslési értékeit tekintve a végeredmény, azaz a 6672 USD és az első nem nulla ár, azaz a 3337 USD között -a kerekítési hibahatáron belül nézve- kétszeres eltérés van.

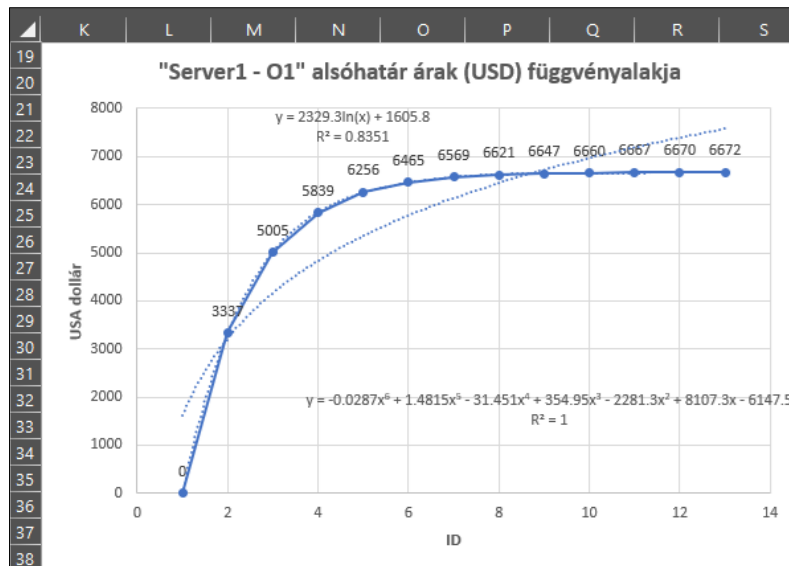
	H	I
19	ID	Server 1 - O1
20		Alsóhatár ár (USD)
21	1	0
22	2	3337
23	3	5005
24	4	5839
25	5	6256
26	6	6465
27	7	6569
28	8	6621
29	9	6647
30	10	6660
31	11	6667
32	12	6670
33	13	6672

	H	I
49	ID	Server 1 - O1
50		Felsőhatár ár (USD)
51	1	9999999
52	2	10000000

37. ábra - STEP IX sorozat OAM táblázata, alsó és felső határ becslések (Objektum 1)

Saját forrás3: „STEP IX+ 2;1 (telj., végeredm)” munkalap H19-I33 és H49-I52 tartománya / Jelmagyarázat: A „Server1 - O1” azaz Objektum 1-re nézve irreálisan alacsony és magas árakat véve (vö. 0 illetve 9999999 USD), -tehát az alsó és felső határok- mesterséges intelligenciával számolt végeredményei láthatóak két táblázatba sűrítve. A mesterséges intelligenciába beküldjük az újonnan kapott adatokat, amelyek számát ID-val jelöltem (vö. új árakat). A feldolgozást mindaddig folytatjuk, amíg az eltérés értéke 0, azaz nulla nem lesz. Összefoglalva tehát, ez egy ismétlődő új ár képzési ciklus, amíg az utolsó modell hibája nem tekinthető nullának. A behelyettesítés jól látható a 34. és a 36. ábrán.

Itt meg kell vizsgálni a kapott értékek függvényalakját (38. ábra). A kérdés innentől az, hogy minden egyes gyors számítás ugyanerre a függvényalakra vezetne-e.

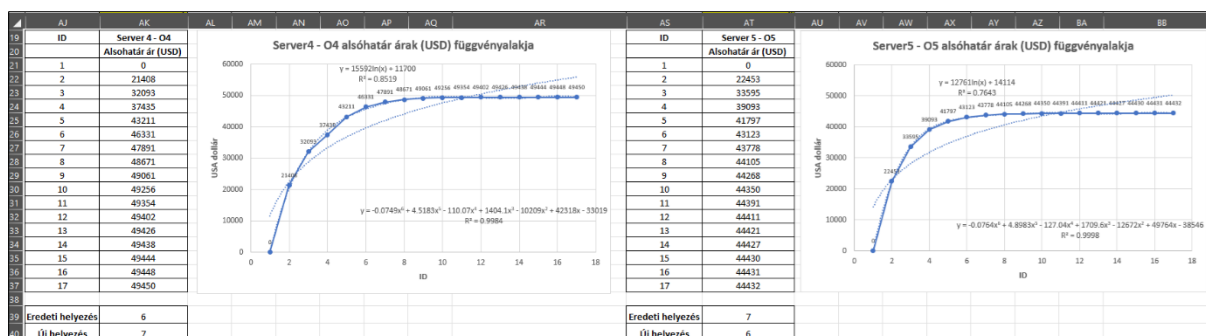


38. ábra - Objektum 1 alsó határ, függvényalak

Saját forrás3: „STEP IX+ 2;1 (telj., végeredm)” munkalap K19-S38 tartománya / Jelmagyarázat: a 37. ábrán látható „Server1 - O1” azaz Objektum 1-re mesterséges intelligencia által végeredményül kapott alsó határ árakból diagrammot készítettem. A diagram függőleges tengelye árakat jelöl, USA dollárban megadva. A vízszintes tengely pedig úgynevezett ID-nak van jelölve (vö. azonosító), ami a mesterséges intelligencia által számolt új ár képzési ciklusok során megkapott újabb és újabb árakat mutatja meg, mindaddig amíg a számolt eltérés 0 nem lesz.

Amennyiben az összes objektumra előállítottuk az alsó és felső határok új becslési értékeit, akkor különböző számításokkal (vö. MAX és SORSZÁM függvény), meghatározzuk a sorrendet és összevetjük a korábbi, normál STEP IX-ben kapott végső sorrendekkel, hogy van-e az Y0 záró inputokban helyezésváltozás.

Jelen esetben az alsóhatárokat nézve, amennyiben az összes objektumra, tehát szerverre előállítottam az OAM táblázatokat, majd abból a mesterséges intelligenciával kiszámoltam az új árakat (mindaddig amíg az eltérés értéke 0, azaz nulla nem lett), a „Server4 - O4” és „Server5 - O5”, tehát az Objektum 4 és 5-nél állt elő változás. A 39. ábra alján látható, hogy az eredetileg 6. helyezett „Server4 - O4”, tehát az Objektum 4 a 7. helyre került, míg az eredetileg 7. helyezett „Server5 - O5”. tehát az Objektum 5 előrébb, a 6. helyre került a STEP IX+ módszert alkalmazva.



39. ábra - Objektum 4 és 5 alsó határ COCO STEP IX+ árdiagrammok

Saját forrás3: „STEP IX+ 2;1 (telj., végeredm)” munkalap AJ19-BB40 tartománya / Jelmagyarázat: A „Server4 - O5” és a „Server5 - O5” azaz Objektum 4 és 5-re nézve irreálisan alacsony árakat véve (vö. 0 USD), -tehát az újonnan behelyettesített alsó határok- mesterséges intelligenciával számolt végeredményei láthatóak két táblázatba sűrítve. A mesterséges intelligenciába beküldjük az újonnan kapott adatokat, amelyek számát ID-val jelöltem (vö. új árakat). A feldolgozást mindaddig folytattam, amíg az eltérés értéke 0, azaz nulla nem lesz. Összefoglalva tehát, ez egy ismétlődő új ár képzési ciklus, amíg az utolsó modell hibája nem tekinthető nullának. A behelyettesítés jól látható a 34. és a 36. ábrán. Továbbá, láthatóak a végeredményül kapott alsó határ árakból elkészült diagrammok, függvényalakok.

A végső konklúzió tehát, hogy ha minden objektumra a maximum kapcsán is előállnak az iterációk, a görbék és az arányok (vö. min: 1:2), akkor ott is kialakulhat sorszám-anomália. Ha minden Y0 input megvan, akkor ezen anomáliák végső hatásai is tetten érhetőek.

Ennél részletesebb elemzésbe a szakdolgozatban most nem bocsátkoznék a COCO STEP IX+ módszert illetően, mert a bonyolultságot is tekintve ez nem része a dolgozatomnak, ez inkább már MsC szint, viszont fontosnak tartottam megemlíteni ezt a módszert is, amivel még pontosabb eredményeket kaphatunk.

4.6. Informatikai rendezőszekrény tervezése, bemutatása

Ebben a fejezetben egy informatikai rendezőszekrény (vö. rack szekrény) gyakorlati tervét és általános megvalósítását mutatom be. Ez szorosan kapcsolódik a rendszerterv-variánsok vizsgálatához, hiszen az ott bemutatott hardver elemek nagyrésze szükséges egy komplett rack szekrény tervezéséhez és megvalósításához. A gyakorlati példában két darab, egymással sorolt rack szekrényt mutatok be, amely egy multinacionális nagyvállalati szerver szobájában kivitelezésre került 2022. év elején, és a mai napig változtatások nélkül üzemel. A bemutatást anonim módon teszem.

A megvalósítás alapos előkészítést és figyelmet igényel, hogy biztosítsa a hálózati eszközök, a szerverek és egyéb releváns hardver elemek megfelelő elhelyezését, működését és karbantarthatóságát. Íme néhány fontos szempont és lépés a rack szekrény tervezéséhez.

4.6.1. A rack szekrény méretének és típusának kiválasztása

- Magasság: A rack szekrények magasságát "U" egységekben mérik (1U = 1.75 hüvelyk). A leggyakoribb méretek 42U, 45U és 48U.
- Szélesség: A standard szélesség 19 hüvelyk, de vannak 23 hüvelykes szekrények is speciális alkalmazásokhoz.
- Mélység: A mélység a beépített eszközök mélysége alapján választható. Általában 600 mm, 800 mm, 1000 mm vagy 1200 mm.

4.6.2. Elhelyezés és hozzáférés

- Helyiség kiválasztása: Gondoskodni kell arról, hogy a helyiség megfelelő szellőzéssel, klímaberendezéssel és biztonsággal rendelkezzen.
- Hozzáférhetőség: Biztosítani kell a könnyű hozzáférést a karbantartáshoz, kábelkezeléshez és hibaelhárításhoz. A rack szekrénynek mind az első, mind a hátsó ajtaja nyitható legyen.

4.6.3. Terhelhetőség és rögzítés

- Súlyterhelés: A szekrény terhelhetőségét ellenőrizni kell, hogy képes legyen elbírní a beépített eszközök súlyát.
- Rögzítés: Fontos, hogy a szekrény biztonságosan rögzítve legyen a padlóhoz vagy falhoz, hogy megakadályozza a baleseteket.

4.6.4. Hűtés és szellőzés

- Légáramlás: Biztosítani kell a megfelelő légáramlást a szekrényben, hogy elkerülje a hőmérséklet túlmelegedését. Használható perforált ajtók, szellőzőpanelek és ventilátorok.
- Hőmérséklet-monitorozás: Hőmérséklet-érzékelők és monitorozó rendszerek alkalmazása ajánlott a hőmérséklet folyamatos nyomon követésére.

4.6.5. Áramellátás és UPS

- PDU: A megfelelő PDU kiválasztása fontos, hogy minden eszköz áramellátása biztonságos és megbízható legyen.

- UPS: Szükséges a szünetmentes tápegység használata a kritikus eszközök áramellátásának biztosításához áramkimaradás esetén.

4.6.6. Kábelkezelés

- Kábelvezetők: Használjunk kábelvezetőket, kábelcsatornákat és kábelkötegeket a rendezett és átlátható kábelkezelés érdekében.
- Címkézés: Minden kábelt és csatlakozót címkézzünk fel a könnyű azonosítás érdekében.

4.6.7. Biztonság

- Zárhatóság biztosítása: A szekrény ajtóit zárral kell ellátni, hogy megakadályozzuk az illetéktelen hozzáférést.
- Fizikai védelem: A szekrények elhelyezése biztonságos, korlátozott hozzáférésű helyiségekben ajánlott.

4.6.8. Dokumentáció

- Helyfoglalási terv: Készítsünk részletes helyfoglalási tervet, amely tartalmazza a szekrénybe telepítendő összes eszköz helyét és a csatlakozási pontokat.
- Hálózati diagramok: Dokumentáljuk a hálózati kapcsolatokat és az áramellátási elosztást.

4.6.9. Skálázhatóság

- Jövőbeli bővítések: Tervezzünk úgy, hogy a jövőbeni bővítések számára is legyen hely a szekrényben. Számoljunk a további eszközök, kábelek és áramforrások igényével.

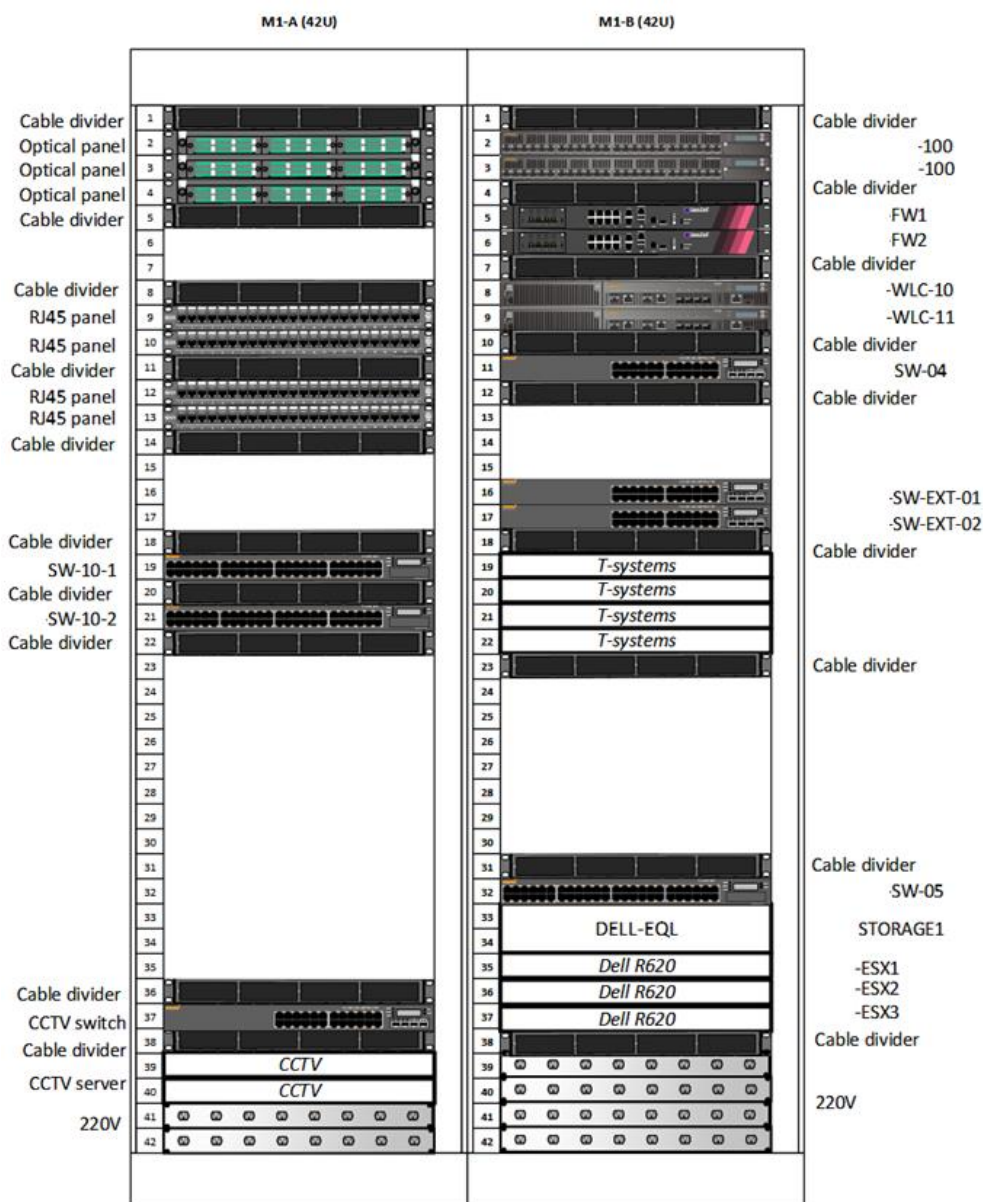
4.6.10. Költségvetés

- Költségtervezés: Vegyük figyelembe a szekrény, a hűtési rendszerek, a kábelkezelő rendszerek, a PDU-k és egyéb kiegészítők költségeit.

4.6.11. Gyakorlati tippek

- Rendeljünk tartalék U egységeket: Mindig hagyjunk üres helyeket a bővítések és a könnyebb légáramlás érdekében.
- Gyakori ellenőrzés: Rendszeresen ellenőrizzük a hőmérsékletet, a légáramlást és az áramellátást.

4.6.12. Rack szekrény diagram bemutatása



40. ábra - Nagyvállalat informatikai rendezőszekrény ábrája (szerver szoba)

Forrás: Anonim vállalat engedélyével / Jelmagyarázat: a képen két álló informatikai rendezőszekrény szerepel egymás mellett, benne különböző hardver elemek, amelyek a későbbiekben bemutatásra kerülnek.

A 40. ábrán egy magyarországi nagyvállalt szerver szobájában elhelyezkedő két darab, egymással sorolt rack szekrény látható. A helyiség biztonságosan zárható, beléptető rendszerrel felszerelt, illetve jól szellőző és légkondicionált. A helyiség tűzbiztos, tűzgátló bejárati ajtóval és automata porral oltó rendszerrel felszerelt. A helyiségben egy 3 fázisos szünetmentes tápegység is helyet kapott. A szekrények zárhatóak, üres helyekkel még bőven rendelkeznek. Beton alapra csavarozottak, illetve védőföldeléssel ellátottak. A rack szekrényekbe futó kábelek

kábelrendezőkben és kábelcsatornáknakban tagolt. Mindezekből következik, hogy minden a követelményeknek megfelelő.

Az anonimitás érdekében a képről néhány szó törlésre került, mivel abból megállapítható lenne a cég. A feliratok angolul készültek, mivel egy multinacionális cégről van szó.

A bal oldali rack szekrényben felülről-lefelé haladva a következőket találhatjuk:

- Cable divider (kábelrendező): az UTP (vö. árnyékolatlan csavart érpáras kábeltípus) és optikai kábelek elvezetésére, rendszerezésére szolgálnak
- Optical panel (optikai patch panel): az optikai kábelek helyiségek vagy épületek közötti összeköttetésére szolgál
- RJ45 panel (réz patch panel): helyi hálózat végpontjainak összegyűjtésére szolgál
- CÉGNÉV-SW-10-1/2 (switch): switchek (vö. kapcsoló), amely a helyi informatikai hálózat adatáramlását biztosítja
- CCTV switch: a kamerarendszer hálózatának adatáramlását biztosítja
- CCTV server: a kamerarendszer központi számítógépe (vö. szerver)
- 220V: rack szekrénybe szerelhető földelt konnektorok

A jobb oldali rack szekrényben felülről-lefelé haladva a következőket találhatjuk:

- Cable divider (kábelrendező): az UTP és optikai kábelek elvezetésére szolgálnak
- FW1/FW2 (firewall): tűzfalak, amelyek az informatikai hálózatot védik, azért van kettő darab, mivel az egyik a fővonalhoz, a másik a tartalék internetvonalhoz kapcsolódik
- CÉGNÉV-WLC-10/11 (WireLess Controller): vezeték nélküli hálózat vezérlő egysége, ehhez kapcsolódnak az access pointok (vö. hozzáférési pont), a könnyebb kezelhetőség miatt
- CÉGNÉV-SW-EXT-1/2 (external switch): ezek a switchek virtuális hálózatok összeköttetésére szolgálnak, például telephelyek között, országok között vagy akár földrészek közötti VPN kapcsolatok kialakítására használatosak
- CÉGNÉV-SW-4/5 (switch): switchek (vö. kapcsoló), amely a helyi informatikai hálózat adatáramlását biztosítja
- CÉGNÉV-STORAGE1: helyi nagy kapacitású SSD tároló egység amely az adatok tárolására szolgál
- CÉGNÉV-ESX1/2/3: fizikai szerverek, amelyeken a VMWARE virtuális szerverek futnak

- 220V: rack szekrénybe szerelhető földelt konnektorok

Összességében a rack szekrény tervezése alapos előkészületeket és átgondolt megvalósítást igényel, hogy biztosítsa a hálózati eszközök hatékony és megbízható működését, valamint a könnyű karbantarthatóságot és bővíthetőséget.

A negyedik fejezetben tehát különböző modellezési számításokkal és mesterséges intelligencia bevonásával informatikai rendszerterv-variánsok ár-teljesítményét vizsgáltam. Továbbá, bemutattam egy validátást növelő módszert is, illetve betekintést nyújtottam egy rack szekrény rendszerterv megvalósításába, egy valós gyakorlati példán keresztül.

5. Szoftver bemutatása

Az ötödik fejezetben bemutatom a Python nyelven írt saját fejlesztésű Excel-OAM-COCO STD és Excel-OAM-COCO Y0 konverter programot. Mivel a programok viszonylag egyszerűek, és nem fő témája a szakdolgozatnak, így kevésbé részletesen, de kellően informatívan fogom demonstrálni a lényegét. A szakdolgozat mellékletét képezi a két szoftver, a felhasználói kézikönyv, a programkódok és a tesztelés, illetve a validálás teszt esetei. A melléklet az alábbi helyről letölthető: <https://miau.my-x.hu/miau/311/server/szoftver.zip>.

5.1. Bevezetés

Célkitűzés és funkciók:

A konverter szoftverem célja a Microsoft Excelben létrehozott OAM konverziója és automatikus beküldése a COCO STD és a COCO Y0 online modellező robotokba, amit már a szakdolgozatom korábbi fejezeteiben részletesen bemutattam. A szoftverem lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan dolgozhassanak nagy mennyiségű adattáblákkal, automatizálva a rangsorolási folyamatot és a beküldést, illetve az eredmény kiíratását.

Kontextus és motiváció:

A szoftver fejlesztése mögötti motiváció a manuális munka csökkentése és az elemzési folyamatok gyorsítása nagy mennyiségű adatok esetén, ez nagyban segítette a szakdolgozati munkámat. Ismétlésképpen, a COCO STD mesterséges intelligencia használata elősegíti az árelőny-termelési függvény generálást, a COCO Y0 pedig az anti-diszkriminatív elemzéseket, amelyek fontosak a döntéshozatali folyamatok támogatásában, ahogyan a szakdolgozat harmadik és negyedik fejezetében már korábban bemutatam.

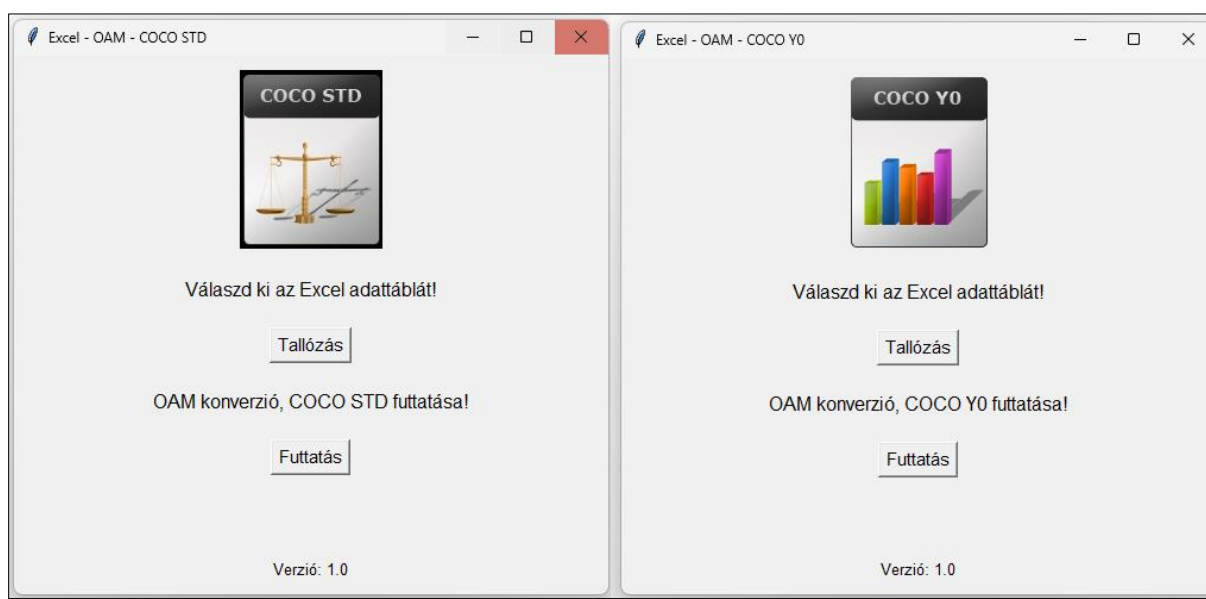
5.2. Módszertan és tervezés

Architektúra és felépítés:

A szoftver architektúrája három fő részre oszlik: a grafikus felhasználói felület (a továbbiakban: GUI), az adatkezelés és konverziós logika, valamint az adatok beküldését végző online kommunikációs rész.

GUI (graphical user interface):

A GUI, azaz a grafikus felhasználói felület felelős a felhasználói interakciók kezeléséért és a vizuális visszajelzések megjelenítéséért. A grafikus megjelenítésért felelős modult (Tkinter könyvtár) használva a grafikus felületen megjelenítünk egy képet, címkéket és gombokat a fájl kiválasztásához és a konverzió elindításához. A felhasználó ezen keresztül választhatja ki az Excel fájlt, amelyet konvertálni szeretne (41. ábra). Ezután a GUI vezeti át a felhasználót a konverziós folyamaton, és jelzi az eredményeket.



41. ábra - Grafikus felület (COCO STD, COCO Y0)

Forrás: saját készítés / Jelmagyarázat: az ábrán két szoftver grafikus felülete látható egy mellett.

Adatkezelés és konverziós logika:

Az adatkezelés részfeladata az Excel fájl beolvasása (Openpyxl könyvtár), az objektum-attribútum mátrix konverziója (vö. rangolás, SORSZÁM függvénnyel) és az adatok előkészítése az online robotba való beküldésre. Az adatkezelés logikája magában foglalja az Excel fájl validációját, az objektum-attribútum mátrix rangsorolását és az előkészített adatok formázását a COCO STD és COCO Y0 online robot számára. Az Excel fájlnak megfelelő struktúrában kell elkészülnie, az ehhez szükséges minta megtalálható a programkönyvtárban, *adattabla_minta_std.xlsx* és *adattabla_minta_y0.xlsx* néven.

Online kommunikációs rész:

Ez a rész felelős az előkészített adatok elküldéséért a COCO STD és a COCO Y0 online robot számára. Az URL kezelő (Urllib könyvtár) modult használva történik az adatok POST kéréssel történő elküldése a szerverre, majd az ott kapott választ HTML fájl formájában letöltjük és elmentjük a helyi rendszerre. Lehetett volna cURL kéréssel is visszakapni az eredményt, például egy TXT fájlban, de választásom most nem erre esett.

5.3. Kódleírás

A szoftver a következőképpen fut le, amennyiben hiba nem áll fenn. A hibakeresési funkciókat egy későbbi alfejezetbe mutatom be.

```
main.py
1 workbook = openpyxl.load_workbook(filename=self.file_path)
2     # Az Excel munkafüzetet betöltjük az openpyxl csomag segítségével
3     items = [[param.value for param in row] for row in workbook.active.iter_rows()]
4     # Az aktív munkalapból kigyűjtjük az adatokat,
5     # az items névű 2-dimenziós tömbbe (python-ban ez listák listája)
6     workbook.close()
7     # Nincs többé szükségünk az Excel munkafüzetre, mivel az adatokat kigyűjtöttük belőle.
8     # Az items névű Python adatstruktúrával dolgozunk tovább
9
10    for j in range(1, len(items[0]) - 1):
11        # Oszloponként dolgozzuk fel az adatokat, "j" változó az oszlopindex
12        # Az első és az utolsó oszlopokban nem feldolgozandó adatokat tartolunk.
13        items_to_rank = [items[i][j] for i in range(1, len(items))]
14        # A j-edik oszlopban kigyűjtjük az adatokat, csak az első (0 index) sort hagyjuk ki (fejlec).
15        reverse = not items[0][j].endswith('1')
16        # Megállapítjuk első (0 index) fejlecsorol a sorrendet.
17        ranks = list(scipy.stats.rankdata(items_to_rank, method='max' if reverse else 'min'))
18        if reverse:
19            ranks = [len(ranks) - rank + 1 for rank in ranks]
20        # A Scipy segítségével kiszámítjuk a "rank"-okat
21        for i in range(1, len(items)):
22            items[i][j] = ranks[i-1]
23        # Az eredeti értékeket felülírjuk a "rank"-okkal
24        # (Csak a mi adatstruktúránkban, az Excel munkafüzetben nem)
25
26    form = {
27        'matrix': '\r\n'.join(['\t'.join(map(str, items[i][1:])) for i in range(1, len(items))]) + '\r\n',
28        'job': '',
29        'modell': 'STD',
30    }
31    # A "rank"-okat tartalmazó táblázatot átalakítjuk olyan formára,
32    # amit a webalkalmazás fel tud dolgozni:
33    # soron belül az értékek TAB-bal ("\t") elválasztva,
34    # a sorok pedig Windows-on használt sortöréssel ("\r\n") elválasztva
35
36    data = urllib.parse.urlencode(form).encode('latin-1')
37    req = urllib.request.Request('https://miau.my-x.hu/myx-free/coco/engine3.php', data=data)
38
39    output_file_path = os.path.splitext(self.file_path)[0] + '_cocostd_eredmeny.html'
40
41    with urllib.request.urlopen(req) as resp:
42        with open(output_file_path, 'w', encoding='utf-8') as f:
43            f.write(resp.read().decode('latin-1'))
44    # A webalkalmazástól kapott választ fájlba írjuk.
45
```

42. ábra - Excel-OAM-COCO STD konverter; a programkód egy részlete

Forrás: saját programkód / Jelmagyarázat: az ábrán a saját fejlesztésű Python programkód egy részlete látható.

Először az adatokat tartalmazó Excel táblázatot betöltjük az Openpyxl csomag segítségével.

Majd a munkalapról kigyűjtjük az adatokat az `items` nevű két dimenziós tömbbe (vö. „i” és „j” változók, sor/oszlop). Ennek lényege, hogy nem az eredeti Excel táblával dolgozunk, abban semmilyen módosítás nem történik.

Harmadik lépésként a program bezárja az Excel fájlt, hiszen többé nincs rá szükségünk, alapszabály, hogy amit nem használunk zárjuk is be.

Ezután az `items` nevű adatstruktúrával dolgozunk tovább, és oszloponként dolgozzuk fel az adatokat. Az első (vö. objektumok) és utolsó oszlopban (vö. ár vagy fiktív változó) nem feldolgozandó adatokat tárolunk. A „j” változó az oszlop index. A ciklus az összes oszlopon végigmegy, amelyben kigyűjtendő adatunk van.

Majd megállapítjuk a fejlécből a SORSZÁM függvény irányát, aminek megadása kötelező, a fejléc szöveg után a „szóköz” után zárójelben (0) vagy (1) értékekkel.

A következő lépésben a Scipy modullal kiszámoljuk a sorszámokat, az irányok figyelembevételével. Majd az `items` táblázat béli értékeket felül írjuk azon értékek sorszámaival.

Ezek után a rangsorolt adatokat tartalmazó táblázatot átalakítjuk olyan formára, amit a COCO online mesterséges intelligencia is fel tud dolgozni. A sorokon belül az értékeket TAB-bal elválasztjuk, a sorokat pedig sortöréssel tördeljük.

Az utolsó előtti lépésben a feldolgozható táblázatot betöltjük a COCO online mesterséges intelligenciába.

Végül a futtatás után visszkapott eredményt kiíratjuk egy fájlba, melynek neve az eredeti Excel fájl neve, illetve ezt kiegészítjük az alábbi szöveggel: „_cocostd_eredmeny.html”. Így tehát ha van egy „szerver_oam.xlsx” nevű fájlunk, az eredmény fájl a következő: „szerver_oam_cocostd_eredmény.html”. Értelemszerűen az Y0 modellnél pedig: „szerver_oam_cocoy0_eredmény.html”.

5.4. Programkód

A következőkben a teljes programkódot feltüntetem a szakdolgozatban. Jelen esetben a COCO STD konverter programkódját másolom be, ez 99,9%-ban megegyezik a COCO Y0 kódjával, ezért a minimális eltérések miatt csak egy programkód kerül bemutatásra.

| PROGRAMKÓD KEZDETE |

```
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import openpyxl
import scipy.stats
import urllib
import urllib.request
import os

VERSION = "1.0" # Verziószám hozzáadása

class ExcelToHtmlConverter:
    def __init__(self, master):
        self.master = master
        master.title("Excel - OAM - COCO STD")

        # Ablak méretének beállítása
        master.geometry("500x450")

        # Kép hozzáadása a felülethez
        image = Image.open("std.jpg")
        image = image.resize((120, 150))
        photo = ImageTk.PhotoImage(image)

        self.image_label = tk.Label(master, image=photo)
        self.image_label.image = photo
        self.image_label.pack(pady=10)

        # Címke
        self.label = tk.Label(master, text="Válaszd ki az Excel adattáblát!", font=("Helvetica", 12))
        self.label.pack(pady=10)

        # Tallózási gomb
        self.select_button = tk.Button(master, text="Tallózás", command=self.select_file, font=("Helvetica", 11))
        self.select_button.pack(pady=10)

        # Címke2
        self.label2 = tk.Label(master, text="OAM konverzió, COCO STD futtatása!", font=("Helvetica", 12))
        self.label2.pack(pady=10)

        # Konvertálási gomb
        self.convert_button = tk.Button(master, text="Futtatás", command=self.convert_to_html, font=("Helvetica", 11))
        self.convert_button.pack(pady=10)

        # Eredmény címke
        self.result_label = tk.Label(master, text="", font=("Helvetica", 10))
        self.result_label.pack(pady=10)

        # Verziószám címke hozzáadása az ablak aljához
        self.version_label = tk.Label(master, text=f"Verzió: {VERSION}", font=("Helvetica", 10))
        self.version_label.pack(side=tk.BOTTOM, pady=10)

    def select_file(self):
        file_path = filedialog.askopenfilename(filetypes=[("Excel files", "*.xlsx")])
        if file_path:
            try:
                workbook = openpyxl.load_workbook(filename=file_path)
                sheet = workbook.active
                if sheet.dimensions == "A1":
                    messagebox.showerror("Hiba", "Az Excel adattábla üres!")
                    return
                elif sheet['A1'].value is None:
                    messagebox.showerror("Hiba", "Az Excel fájl nem a megfelelő struktúrában készült!")
                    return
                elif not any(sheet.iter_rows(values_only=True)):
```

```

        messagebox.showerror("Hiba", "Az Excel fájl nem tartalmaz munkalapokat!")
        return
    else:
        self.label.config(text=f"Kiválasztott fájl: {file_path}")
        self.file_path = file_path
    except Exception as e:
        messagebox.showerror("Hiba", f"Hiba történt a fájl beolvasása közben: {str(e)}")

def convert_to_html(self):
    if hasattr(self, 'file_path') and self.file_path:
        try:
            self.convert_and_save_html()
            messagebox.showinfo("Siker", "Kész! Az eredményt megtalálod a kiindulási fájl könyvtárában.")
        except openpyxl.utils.exceptions.InvalidFileException as e:
            messagebox.showerror("Hiba", "Hiba: Az állomány nem érvényes Excel fájl.")
        except urllib.error.URLError as e:
            messagebox.showerror("Hiba", "Hiba: Nem sikerült kapcsolódni az internethez.")
        except Exception as e:
            messagebox.showerror("Hiba", f"Hiba a konvertálás során: {str(e)}")
    else:
        messagebox.showerror("Hiba", "Először válassz ki egy Excel fájlt!")

def convert_and_save_html(self):
    if hasattr(self, 'file_path') and self.file_path:
        try:
            workbook = openpyxl.load_workbook(filename=self.file_path)
            items = [[param.value for param in row] for row in workbook.active.iter_rows()]
            workbook.close()

            for j in range(1, len(items[0]) - 1):
                items_to_rank = [items[i][j] for i in range(1, len(items))]
                reverse = not items[0][j].endswith('(1)')
                ranks = list(scipy.stats.rankdata(items_to_rank, method='max' if reverse else 'min'))
                if reverse:
                    ranks = [len(ranks) - rank + 1 for rank in ranks]
                for i in range(1, len(items)):
                    items[i][j] = ranks[i-1]

            form = {
                'matrix': '\r\n'.join(['\t'.join(map(str, items[i][1:])) for i in range(1, len(items))]) + '\r\n',
                'job': "",
                'modell': 'STD',
            }
            data = urllib.parse.urlencode(form).encode('latin-1')
            req = urllib.request.Request("https://miau.my-x.hu/myx-free/coco/engine3.php", data=data)

            output_file_path = os.path.splitext(self.file_path)[0] + '_cocostd_eredmeny.html'

            with urllib.request.urlopen(req) as resp:
                with open(output_file_path, 'w', encoding='utf-8') as f:
                    f.write(resp.read().decode('latin-1'))
        except Exception as e:
            raise e

if __name__ == "__main__":
    root = tk.Tk()
    app = ExcelToHtmlConverter(root)
    root.mainloop()

```

| PROGRAMKÓD VÉGE |

5.5. Önellenőrzés, tesztelés és eredmények

A szoftver önellenőrzési funkcióinak tesztelése- és a végeredmények helyességének vizsgálata során négy fő területre összpontosítottam. Az Excel fájlok validációjára, az Internet kapcsolat meglétére, a konverziós folyamat helyességére, illetve a visszakapott eredmény ellenőrzésére. Az első három esetben a szoftver válaszüzenetben figyelmeztetett, a negyedik esetben csak a manuális tesztesetek felállításával tudtam meggyőződni a hibátlan működésről.

A következő tesztek kerültek végrehajtásra.

Excel fájl validáció tesztek:

- Üres Excel fájl kezelése: Ellenőriztem, hogy a program helyesen kezeli az üres Excel fájlokat, és megjelenik-e a megfelelő hibaüzenet.
- Megfelelő struktúra ellenőrzése: Teszteltem, hogy a szoftver helyesen észleli-e az Excel fájlok nem megfelelő struktúráját, és megfelelően jelzi-e ezt a felhasználónak.
- Munkalapok ellenőrzése: Ellenőriztem, hogy a szoftver kezeli-e helyesen azokat az Excel fájlokat, amelyek nem tartalmazznak munkalapokat.

Internet kapcsolati teszt:

- Hálózati kapcsolat ellenőrzése: Ellenőriztem, hogy a szoftver megjeleníteni-e a megfelelő hibaüzenetet, ha nincs Internet kapcsolatunk. A beküldés előtt jelzi, ha nincs kapcsolat, hiszen ez elengedhetetlen az online robot használatához.

Konverziós folyamat tesztek:

- Paraméterek rangsorolása: Ellenőriztem, hogy a szoftver helyesen rangsorolta-e a megadott paramétereket a konverzió előtt.
- Adatküldés és eredmény mentése: Teszteltem, hogy a szoftver helyesen küldi-e el az adatokat az online robot szolgáltatásnak.

Végeredmény ellenőrzése:

- Összehasonlítás a manuális módszerrel: A végeredményt több esetben összevettem a manuálisan elkészített teszt esetekkel, illetve a szakdolgozati *Saját forrás* és *Saját forrás2*-ben is felhasználtam a szoftvert és az eredményeket validáltam a manuálisan végzett módszerrel. A validációt az alábbi képlettel végeztem: =HA('COCO STD (konverter eredmény)'!L44<='COCO STD (manuális eredmény)'!L44;1;0). A képletből

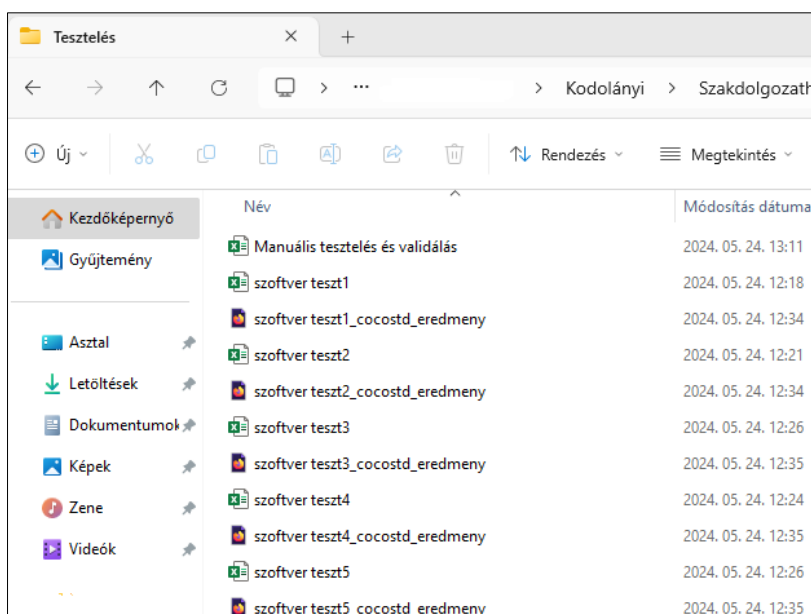
következi, hogy amennyiben az eredmény valid, akkor 1-es számmal jelzett értéket kapunk vissza (43. ábra).

	P	Q
6	Dell PowerEdge R960	Validáció
7	Server1 - O1	1
8	Server2 - O2	1
9	Server3 - O3	1
10	Server4 - O4	1
11	Server5 - O5	1
12	Server6 - O6	1
13	Server7 - O7	1
14	Server8 - O8	1
15	Server9 - O9	1
16	Server10 - O10	1
17		
18		0 = nem valid
19		1 = valid

43. ábra - Manuális és szoftveres végeredmény validációja

Saját forrás: „COCO STD (manuális eredmény)” munkalap P6-Q18 tartománya / Jelmagyarázat: a manuális és a szoftverrel elkészített számítások végeredménye 100%-osan megegyezik, tehát az eredmény validáltnak tekinthető, a szoftver jó működik.

A szakdolgozat mellékletét képezik az alábbi teszteset fájlok és eredmények, ami a 44. ábrán látható.



44. ábra - Manuális tesztesetek, végeredmények és validáció

Forrás: saját készítés / Jelmagyarázat: képernyőkép a Windows Intézőből, itt különböző Excel adattáblák beküldését végeztem el a szoftverbe, majd ezek végeredményei megjelentek a kiindulási mappában. Végül ezekből elkészült a „Manuális tesztelés és validálás.xlsx” Excel fájl, amiben a manuális- és a szoftveres teszt eredményei validálásra kerültek.

A végzett tesztek alapján a szoftver stabilan és megbízhatóan működik.

5.6. Összegzés és jövőbeli fejlesztések

Annak érdekében, hogy a szoftver továbbra is versenyképes maradjon és még jobban kiszolgálja a felhasználói igényeket, számos jövőbeli fejlesztési lehetőség van figyelembe véve, illetve esetlegesen az évfolyamtársaim is tovább gondolhatják ezt a feladatot, mint potenciális szakdolgozati kihívás.

Mindezeken felül az önálló verzióban a következő fejlesztési lehetőségeket látom:

- Magasabb fokú automatizálás, az eredmény kiértékelésének elkészítése automatikusan
- Kiegészítő interaktív funkciók bevezetése, például „drag-and-drop” funkció az Excel fájlok tallózásához.
- Konfigurációs lehetőségek bevezetése a felhasználók számára az elemzések testreszabhatóságának növelése érdekében.
- A két szoftver összevonása egy alkalmazásba.
- További nyelv hozzáadása, nyelv választás lehetősége
- Adatbázis támogatás
- Egyszerre több fájl vagy egy fájlban belül több adattábla feldolgozása

Itt megemlítem, hogy Juhász Krisztián évfolyamtársam dolgozik egy webalkalmazáson, ahol cél a teljes automatizálás. Jelenleg különböző tanulási mintákat vizsgál (vö. adatvagyon) a korábban általam vizsgált szerver-ár teljesítmény OAM-ot felhasználva. Itt feladata megállapítani az adatvagyon kapcsán, hogy vajon **több adat jobb modell bizonyítottságot eredményez-e?**

Juhász Krisztián, Dr. Pitlik László. (dátum nélk.). OAM 99. Letöltés dátuma: 2024.06.24.

Forrás: https://miau.my-x.hu/miau/310/OAM_99_60_1kor.xlsx

A fenti linkről letölthető, megnyitható az adatvagyon. Megnyitva a fájlt láthatjuk, hogy különböző scenáriók vizsgálata a cél, jelen esetben egy úgynevezett „**99**-es OAM” felállítása. Minden scenárió (az elméleti maximum: 99), csak egyetlen egy cella kapcsán tér el a kiindulástól, és az eltérés mindenkor szomszédsági jellegű, tehát ami eddig két szomszédos sorszám volt, abból most egy sorszám lesz, a kisebb (vö. jobb) ismétlődésével. Tehát, mivel 10 objektumunk (vö. szerver) van és 11 attribútumunk (vö. szerverek hardver elemei), így elméletileg/maximum $11 \cdot 10 - 11 = 99$ csere eset lenne lehetséges, mivel egy rangsorszám mindig „fennmarad”, azt önmagára cserélni értelmetlen.

A cserék/összevonások értelme nem más, mint két egymást követő sorszámából a kisebbet megtartani, mindkét mátrix pozícióban. Vagyis, mindig csak egy cellányi információ változik, a többi változatlan. Így nem egy egymásra épülő változtatás lánc alakul ki, hanem mindig egy konkrét cella lesz össze vetve a hozzá a leghasonlóbbal és azzal egyenlővé téve.

A szakdolgozat írásakor Juhász Krisztián még dolgozik a feladaton, tehát csak részeredményünk van.

A következő Microsoft Excel alapú pivot táblázatban (45. ábra) látható, hogy a sorszámcserek milyen hatással voltak a modellekre, quasi érzékenységi vizsgálat történik.

	A	B	C	D	E	F
3	Típus	Mennyiség / Becslés	Átlag / Becslés	Maximum / Becslés	Minimum / Becslés2	Konklúzió adatmennyiség függően
4	10->7	1	999988	999988	999988	
5	10->8	1	999988	999988	999988	
6	10->9	6	999990	1000002	999988	
7	2->1	7	1000039	1000051	1000015	<--masszívan a legjobb
8	3->2	6	1000023	1000043	1000014	<--masszívan jó
9	4->3	6	999996	1000037	999988	
10	5->1	1	999987	999987	999987	
11	5->2	1	999991	999991	999991	
12	5->4	6	999997	1000045	999988	
13	6->5	6	999998	1000046	999988	
14	7->5	1	999991	999991	999991	
15	7->6	6	999959	999988	999812	
16	8->1	1	1000048	1000048	1000048	<--bizonytalanul jó
17	8->7	4	999995	1000014	999988	
18	9->1	1	999999	999999	999999	
19	9->7	2	999988	999988	999988	
20	9->8	4	1000000	1000034	999988	
21	Végösszeg	60	1000000	1000051	999812	

45. ábra - Sorszámok cseréjének hatása (pivot táblázat)

Forrás: https://miau.my-x.hu/miau/310/OAM_99_60_1kor.xlsx, „pivot” munkalap A3-F21 tartománya /

Jelmagyarázat: Juhász Krisztián évfolyamtársam részeredményei, a szerver-ár teljesítmény kiinduló táblázatomat tovább vizsgálva (vö. sorszámcserek hatásának vizsgálata).

Az „A” oszlopban látható, hogy melyik sorszámot mire cseréltük ki. A cserék lényege, hogy ha egy attribútum eredetileg 2-es rangsorszámot kapott, azt lecseréljük az 1-esre, és így tovább, de mindig csak egy változtatás történik és abból egy új OAM tábla jön létre. Továbbá, a csere sorszámok nem mindig egymás melletti szomszédos számok, hiszen a rangsorolás (vö. RANK függvény) azt adja meg, hogy egy szám nagysága alapján hányadik egy számsorozatban, ezért esetünkben 17 féle csere típus jött létre, ahogyan ezt a 45. táblázat A4-A20 celláiban láthatjuk. Azért „csak” 17, mivel amennyiben rangsorszámozásnál több érték sorszáma azonos, akkor ezek közül mindig az elsőt adja vissza a függvény. Minél többször előfordul egy-egy szcenárió,

annál jobban arra kell figyelniük, illetve minél nagyobb az „Átlag”, „Maximum” és „Minimum” értéke. Juhász Krisztián esetében a 2→1 csere típus fordult elő a legtöbbször, ahogyan a 45. ábra A7 cellájában látható, narancssárgával kiemelve.

A „B” oszlopban látható, hogy az adott csere hányszor fordult elő, az összes, azaz **60** csere esetet figyelembe véve. A vizsgálat során arra a kérdésre kell választ kapnunk, hogy lehet-e minden objektum másként egyforma/értékes?

A „C”, „D” és „E” oszlopban pontszámok láthatóak, amelyek a sorszámcserék után létrejött újabb és újabb objektum-attribútum mátrixokból számolt egyfajta végeredmények. Ezek az újonnan létrejött OAM-ok, a korábban már említett COCO Y0 anti-diszkriminációs, hasonlóságelemző online modellezőbe betöltésre kerültek. Az Y0 változó/pontszám fiktív és kellően/megfelelően nagy érték ahhoz, hogy a modellező robotnak teret adjunk. Az irány megadása után optimalizálás keretében keressük az átlagtól leginkább eltérő objektumot A színek azt jelölik, hogy a kapott becült pontszámok hogyan viszonyulnak egymáshoz képest, tehát a legzöldebb a legjobb/legértékesebb, a legpirosabb a legrosszabb/kevésbé értékes.

Itt meg kell vizsgálnunk, hogy hogyan és miből kapták meg ezeket a pontszámokat. A 46. ábrán látható „OAM_99_RESULTS (2)” munkalap táblázatában egy paraméter tömeg van statisztikázva. A B4 cellában jelzett, azaz a 2→1 típusú sorszámcserét vizsgálva értelmezem a táblázatot. Az „A”, azaz „id”-vel jelölt oszlopban a sorszámcserék száma látható. A „B” oszlopban a sorszámcserék láthatóak, tehát hogy miről-mire cseréltünk. A D4 és az E4 cella azt mutatja meg, hogy mekkora az adatvagyon. Ez az adatvagyon egy másik táblázatból származik, ami a 47. ábrán látható. A „D” oszlop, azaz a „magányos S1 adaptált” azt jelenti, hogy az adott OAM1 táblázat S1 lépcsőjében hány darab nem nulla (azaz nem 0) érték szerepel. Az „E” oszlop, azaz a „nem nullás lépcső” azt jelenti, hogy az egész OAM táblázatban, hány darab nem nulla (azaz nem 0) érték szerepel. Jelen esetben ez 13db lesz, ezt úgy kaptuk meg, hogy a 47. ábrán összeszámoltuk, hány darab van az összes nem nulla (azaz nem 0) számból. Az „F” oszlopban szereplő érték, azaz „szum S10”, az OAM1 táblázat S10 sora összeadva, ahogyan ez a 47. ábra S10 sorából kiszámolható, tehát $0+9000+0+0+0+0+0+0+0+0 = 9000$ USD. A G oszlop, azaz az „aranyérem összes” oszlopban látható szám azt mutatja meg, hogy a sorszámcseré esetén a kicserélt sorszám hányszor szerepel az adott OAM1 táblázatban, a 2→1 sorszámcserét vizsgálva az „OAM_1_COCO” munkalapon látható végeredmények tekintetében. A „H” oszlop, azaz „összes egyedi” azt jelenti, hogy hány darab egyedi érték van az „OAM_1_COCO” munkalap A31-L41 táblázat minden egyes oszlopát tekintve, külön-

külön oszloponként. Az „I” oszlop, azaz Y0 fiktív pontszám, pedig az anti-diszkriminatív COCO Y0 elemzéshez szükséges. Mindezekon felül a táblázat 1-es sorában szórást is számoltunk, illetve a leglényegesebb, hogy az irányvektorokat is megadták, azaz „minél nagyobb annál jobb” vagy „minél kisebb annál jobb”. Az „AO”, azaz a „Becsítés” oszlopban, pedig a becslési pontszámokat láthatjuk, amit a „modell” munkalap COCO Y0 végeredményeket figyelembe véve.

	A	B	C	D	E	F	G	H	I	AO	AP	AQ
1			szórás	0.982761013	1.365018987	1161.895004	0.947640543	1.665875518				
2			szabály	minél nagyobb, annál jobb	minél nagyobb, annál jobb	minél kisebb, annál jobb	minél nagyobb, annál jobb	minél nagyobb, annál jobb				
3	id	tipus	attribútum	magányos s1 adaptált	nem nullás lépcső	szum S10	aranyérem összes	összes egyedi	Y0	Becsítés	Validitás	tipus
4	1	2->1	OAM 1	4	13	9000	39	23	1000000	1000015	1	2->1
5	2	5->2	OAM 2	5	5	9000	36	24	1000000	999991	1	5->2
6	3	7->5	OAM 3	5	5	9000	36	24	1000000	999991	1	7->5
7	4	10->7	OAM 4	5	5	9000	36	23	1000000	999988	1	10->7
8	5	2->1	OAM 5	7	7	9000	37	28	1000000	1000051	1	2->1
9	6	3->2	OAM 6	7	7	9000	36	28	1000000	1000043	1	3->2
10	7	4->3	OAM 7	7	7	9000	36	26	1000000	1000037	1	4->3
11	8	5->4	Variáns 8	8	8	9000	36	26	1000000	1000045	1	5->4
12	9	6->5	Variáns 9	8	8	9000	36	27	1000000	1000046	1	6->5
13	10	7->6	Variáns 10	3	3	18000	36	20	1000000	999812	1	7->6
14	11	8->7	Variáns 11	6	6	9000	36	25	1000000	1000014	1	8->7
15	12	9->8	Variáns 12	7	7	9000	36	25	1000000	1000034	1	9->8
16	13	10->9	Variáns 13	6	6	9000	36	23	1000000	1000002	1	10->9
17	14	2->1	Variáns 14	7	7	9000	37	25	1000000	1000042	1	2->1
18	15	3->2	Variáns 15	6	6	9000	36	28	1000000	1000023	1	3->2

46. ábra - Pivot táblázat adatforrásának egy részlete

Forrás: https://miau.my-x.hu/miau/310/OAM_99_60_1kor.xlsx, „OAM_99_RESULTS (2)” munkalap A1-AQ18 tartománya / Jelmagyarázat: Juhász Krisztián évfolyamtársam részeredményei, a server-ár teljesítmény kiinduló táblázatomat tovább vizsgálva (vö. sorszámcsere hatásának vizsgálata). Az ábrán egy pivot táblázat adatforrása látható. Ebben az adatforrásban egy paraméter tömeg van statisztikázva. Ennek részletes magyarázata a 79. oldal lap aljától olvasható.

	A	B	C	D	E	F	G	H	I	J	K	L
31	Lépcsők(2)	X(A1)	X(A2)	X(A3)	X(A4)	X(A5)	X(A6)	X(A7)	X(A8)	X(A9)	X(A10)	X(A11)
32	S1	20000	206000	0	0	9000	0	0	0	5000	0	0
33	S2	0	166000	0	0	0	0	0	0	0	0	0
34	S3	0	126000	0	0	0	0	0	0	0	0	0
35	S4	0	86000	0	0	0	0	0	0	0	0	0
36	S5	0	66000	0	0	0	0	0	0	0	0	0
37	S6	0	46000	0	0	0	0	0	0	0	0	0
38	S7	0	31000	0	0	0	0	0	0	0	0	0
39	S8	0	26000	0	0	0	0	0	0	0	0	0
40	S9	0	16000	0	0	0	0	0	0	0	0	0
41	S10	0	9000	0	0	0	0	0	0	0	0	0

47. ábra - COCO elemzés (STD eredmény) - OAMI, 2→1 sorszámcsere

Forrás: https://miau.my-x.hu/miau/310/OAM_99_60_1kor.xlsx, „OAM_1_COCO” munkalap A31-L41 tartománya / Jelmagyarázat: A 46. ábra D4 cellájának eredménye ennek a táblázatnak a sárgával jelölt adataiból számolandóak, azaz a 4db nem nulla érték. Az ábrán látható számértékek mértékegysége USD, azaz USA dollár. Az S1-S10 jelölés az objektumokat jelenti (vö. szerverek), az X(A1)-X(A11) pedig az attribútumokat (vö. szerverek tulajdonságai). Erről részletesebben a 3.2.1 és a 3.2.2 pontokban írtam korábban.

Folytatva a pivot táblázat elemzését (45. ábra), az „F” oszlopban láthatjuk a végeredményt, amelyek az adatmennyiségek függvényében értékelhetők. A legkisebb hatást a $10 \rightarrow 7$, $10 \rightarrow 8$ és $10 \rightarrow 9$ sorszámcserekkal kaptuk. A sok erős- és valid adatnak köszönhetően, pedig a $2 \rightarrow 1$ -es sorszámcserevel a „masszívan a legjobb” eredményt kaptuk. Az $5 \rightarrow 1$ sorszámcsere kapcsán azt állapíthatjuk meg, hogy a legkevésbé volt hatással az érzékenységi vizsgálat során. A $8 \rightarrow 1$ sorszámcsere ugyan csak egyszer fordult elő, de itt kiemelkedő pontszámot kaptunk, azonban a kevés adat miatt csak „bizonytalanul jó” végeredményben részesülhet. A táblázat 21-es sora, amely világoskékkel jelölt és „Végösszeg” elnevezésű, B21-es cellájában látható a 60-as szám, amely a csere esetek összességét jelenti. Ebben a sorban látható még az „Átlag”, „Maximum” és „Minimum” / „Becslés” eredményei, tehát az összes csere eset kapcsán megkapott pontszámok: átlag-, maximum- és minimum kapott pontszáma.

Mi lehet az összefüggés a fentiekben már megemlített 99-17-60 számok között?

Összefoglalva, mivel 10 objektum és 11 attribútum van ebben az esetben, (vö. $11 \cdot 10 - 11 = 99$, ahol -11 jelzi, hogy egy oszlopon belül a lehetséges n egyedi érték között csak n-1 csere hajtható végre, azaz oszloponként -1 a korrekció, mindösszesen 11) így -a korábban már említettek szerint- **99** elméleti csere eset lenne lehetséges összesen, ha az összes rangsorolt érték egyedi lenne.

A **60** eset jelen vizsgálat során végeredménynek tekinthető (48. ábra), mivel ennyi az elvégezhető csere eset összesen, ez a rangsorszámokból következik. Oszloponként lebontva, amennyiben minden oszlopban összeszámoljuk hány darab egyedi sorszám van, majd ebből kivonunk egyet, akkor megkapjuk az oszloponkénti lehetséges csere eseteket. Ezt az alábbi függvénnyel számoltam ki, a „B” oszlopra: $=\text{SZUM}(\text{HA}(\text{GYAKORISÁG}(\text{B2:B11}; \text{B2:B11}) > 0; 1)) - 1$. Ezt a számítást minden oszlopra el kell végezni, majd végül pedig össze kell adni az összes oszlopra vonatkozóan.

	A	B	C	D	E	F	G	H	I	J	K	L	M
	PowerEdge R940 Server / Tulajdonság	X(A1) CPU Xeon (sorszám)	X(A2) CPU magok száma (sorszám)	X(A3) CPU órajel (sorszám)	X(A4) RAM memória (sorszám)	X(A5) RAID vezérlő (sorszám)	X(A6) NVME SSD meghajtó (sorszám)	X(A7) SATA SSD meghajtó (sorszám)	X(A8) SAS HDD meghajtó (sorszám)	X(A9) iDRAC (Ex.=1, Ent.=2, Ent. Adv.=3) (sorszám)	X(A10) Optikai csatlakozó külső storage-hoz (sorszám)	X(A11) Quad tápegység (sorszám)	Ár (USD)
1													
2	Server1 - 01	10	10	10	10	1	10	10	5	10	1	10	18000
3	Server2 - 02	7	9	9	9	1	9	9	5	8	9	9	25000
4	Server3 - 03	7	8	7	8	1	7	8	5	8	9	8	35000
5	Server4 - 04	7	7	7	7	1	7	7	5	1	1	7	45000
6	Server5 - 05	5	6	6	6	1	6	6	5	1	1	6	60000
7	Server6 - 06	5	5	5	5	1	5	5	5	1	1	5	80000
8	Server7 - 07	2	4	4	4	1	4	4	1	1	1	4	120000
9	Server8 - 08	2	3	3	3	1	3	3	1	1	1	3	160000
10	Server9 - 09	2	2	2	2	1	2	2	1	1	1	2	200000
11	Server10 - 010	1	1	1	1	1	1	1	1	1	1	1	240000
12	Csere esetek	4	9	8	9	0	8	9	1	2	1	9	
13	60												

48. ábra - Kiindulási OAM táblázat, csere esetek száma

Forrás: https://miau.my-x.hu/miau/310/OAM_99_60_1kor.xlsx, „STARTER” munkalap A1-M13 tartománya (egydi szerkesztés) / Jelmagyarázat: Az ábrán a kiindulási server-ár teljesítmény OAM táblázat látható, ami tartalmazza a server objektumokat, az attribútumokat sorszámozva, illetve az árakat, amelyek USA dollárban vannak megadva. Egy függvény segítségével összeszámoltam, hogy oszloponként hány egyedi sorszám fordul elő, majd ebből kivontam 1-et. Az „A13” cellában pedig az összes csere esetet összeadtam, így 60 lett a végeredmény. Ennek lényege, hogy a 11*10-es, tehát 110 elemű táblázatunkból kimutassuk, hogy az elméleti 99 csere esetből (vö. hiszen önmagára nem cserélünk egy számot), a megadott server-ár teljesítményeket vizsgáló OAM táblázat esetében hány csere szcenárió lehetséges.

A **17** fajta csere típus az azt jelenti, hogy ennek a konzisztencia-szcenárió vizsgálatnak az esetében, ennyi féle helyezéscsere történt meg. Ennek oka, hogy a rangsorolás után nem mindig az egymás melletti számok cseréje történt meg, hiszen az attribútumok tekintetében fennáll az egyezőség (vö. egyforma hardver elemek). Vagyis pl. a 46. ábra „B” oszlopában is jelzett cseretípus-címkekből a 60 lehetséges csere (objektum) esetén 17 egyedi cserecímke állt elő (vö. 45. ábra „A” oszlop).

Végső konklúzió tehát hogy, egy modell annál jobb, amennyiben a cserék után minél több adatot kapunk. Minél több változó esik ki, annál rosszabb a modell. Tehát, keresni kell a változások hatására az esetleges hitelesség javulás állapotát, vagy éppen a hitelesség romlás állapotát. Az ideális helyzet az lenne, ha minden cellában számérték szerepelne. Ezt 10-20db objektummal (vö. server) nehéz kikényszeríteni, 1000 objektum variánsnál már nagy valószínűség szerint ideális eseteket kapnánk.

6. Összefoglalás

Ebben a fejezetben a dolgozat összefoglalása található, ezeken felül vita és a jövőkép is említésre kerül.

6.1. Rezümé - HU

A szakdolgozatomban nagyvállalati szerverek és informatikai rendszerterv-variánsok ár-teljesítményét vizsgáltam, többféle manuális modellezési módszerrel. Mindezekon felül a rendszertervezés gyakorlati alkalmazását is bemutattam, egy nagyvállalati informatikai rendezőszekerekény megvalósításával. Továbbá a saját, Python nyelven írt Excel-OAM-COCO STD/Y0 szoftverem is ismertetésre került. Összességében munkámat igen komplexnek értékelem, mivel négy külön fejezetben csak a „core business”-el foglalkoztam, és ennek terjedelme már jóval meghaladja az elvárt szintet.

Az ár-teljesítmény vizsgálatok tekintetében sikerült egy induktív szakértői rendszer segítségével (vö. 2-es fejezet bevezetés), az objektumok meghatározott attribútumainak adatai alapján a mesterséges intelligencia segítségével példákat mutatni az objektív döntéshozatalra. *„Az induktív szakértői rendszer az emberi szakértői tevékenység szimulálása, amely a logikai következtetés szabályai szerint végez műveleteket az adott információkkal.”* (https://miau.my-x.hu/mediawiki/index.php/Induktív_szakértői_rendszer - 2024.06.23. - Induktív szakértői rendszer - Dr. Pitlik László) Bemutatásra kerültek még a naiv, azaz a nem optimalizált eredmények is, azonban ez ellentmondásokhoz vezetett. **A végső következtetés lehetne az is, hogy nem mindig a legdrágább a legjobb, ár/teljesítmény arányt jelentő objektum!** Feltéve, hogy nem áll rendelkezésre végtelen mennyiségű forrás a vásárlásra. További konklúzió, hogy egy olcsó termék is bírhat jó ár-érték aránnyal.

A gyakorlati példát tekintve sikerült bemutatni a rendszertervezés egy lehetséges gyakorlati alkalmazását is, ami szorosan összefügg a rendszerterv-variánsok objektumainak vizsgálatával, hiszen egy-egy objektum szinte önmagában is egy rack szekrény tervnek felel meg, sőt túlmutat azon, hiszen a vezetéknélküli hálózat egyes elemeit is tartalmazta. Megjegyzem, tudatosan csak a fő hardver elemeket vizsgáltam, a kábelezést, az áramellátás kialakítását és a munkadíjat most nem vettem számításba, mivel ezek meghatározására konkrét méretekre, illetve árajánlatokra lenne szükség.

Az Excel-OAM-COCO STD/Y0 szoftver tekintetében úgy érzem, hogy sikerült egy alapot kialakítani egy olyan induktív szakértői rendszer kialakításához, ahol mi magunk adhatjuk meg

az objektumokat és attribútumokat, illetve ami akár már kiértékelést is végez. Jelenleg a szoftver hasznosságát abban látom, hogy a Kodolányi János Egyetemen jelenleg tanuló vagy a jövőben érkező diákok fel tudják használni OAM-ok vizsgálatához, és időt takarítanak meg vele. Vagy akár a későbbi szakdolgozatokhoz is hasznos lehet, ahogyan én is felhasználtam.

6.2. Rezümé - EN

In my thesis, I examined the price-performance ratio of large company servers and IT system plan variants, using several manual modeling methods. In addition to all of this, I also presented the practical application of system design, with the implementation of a large company's IT management cabinet. Furthermore, my own Excel-OAM-COCO STD/Y0 software, written in Python, was also presented. Overall, I consider my work to be very complex, since I only dealt with the "core business" in four separate chapters, and the scope of this already exceeds the expected level.

Regarding the price-performance tests, it was possible to show examples of objective decision-making with the help of an inductive expert system, based on the data of the specific attributes of the objects, with the help of artificial intelligence. *"The inductive expert system is a simulation of human expert activity, which performs operations with the given information according to the rules of logical inference."* (https://miau.my-x.hu/mediawiki/index.php/Induktív_szakértői_rendszer - 23.06.2024. - Inductive expert system - László Pitlik)

Naive, i.e. non-optimized, results were also presented, but this led to contradictions. **The final conclusion could also be that the most expensive object is not always the best object with a price/performance ratio!** Provided you don't have an infinite amount of funds available to purchase. Another conclusion is that even a cheap product can have a good price-value ratio.

Looking at the practical example, it was possible to present a possible practical application of the system design, which is closely related to the examination of the objects of the system design variants, since each object almost corresponds to a rack cabinet design by itself, and even goes beyond it, since it also contained some elements of the wireless network. I note that I consciously examined only the main hardware elements, I did not take into account the cabling, the power supply design and the labor fee, as specific dimensions and price offers would be required to determine these.

With regard to the Excel-OAM-COCO STD/Y0 software, I feel that a foundation has been created for the creation of an expert system where we can enter the objects and attributes

ourselves, and which even performs evaluation. Currently, I see the usefulness of the software in that students currently studying at János Kodolányi University or future students can use it to examine OAMs and save time with it. Or it can even be useful for later theses', as I used it.

6.3. Vita

A szerverek és informatikai rendszerterv-variánsok ár-teljesítmény vizsgálatát tekintve az elemzések megválaszolják, hogy az adott időpontban melyik szerver és melyik rendszerterv-variáns a legideálisabb választás a konkurens objektumokhoz képest. Az idő előrehaladtával egyrészt az árak aktualitásukat veszítik, másrészt már a forint-dollár árfolyamváltozás miatt is belátható időn belül torzabb képet kapunk. Szintén megkerülhetetlen szempont, hogy rövid időtávot tekintve (vö. 1-3 év) az adott hardvereket kivezetik, és helyükbe új modellek lépnek, így a vizsgált hardverek is aktualitásukat veszítik a közeljövőben. Bár megjegyzendő, hogy a vállalati szférában nem annyira gyors a fejlesztés, mint a consumer (vö. fogyasztó) szektorban.

A szerverek vizsgálatakor felmerülhet, miért pont ennyire különböző konfigurációkat vizsgáltam. A kérdés jogos, hiszen beszerzéskor jellemzően már a felmerült igényekhez mérten vizsgálunk objektumokat (vö. szervereket). Ebben a gyakorlati példában mindenképpen be szerettem mutatni a szélsőséges eseteket, tehát az „ideált”, a műszaki szempontból lehető legjobb szervert is számításba akartam venni, hiszen a sokaság és így a benchmark-definíció mindenkor a döntéshozó feladata és felelőssége.

A rendszerterv-variánsok vizsgálatánál szintén felmerülhet a kérdés, miért pont annyi darabszámú attribútumból állt egy-egy rendszerterv-variáns leírása, amennyit éppen vizsgáltam. A kérdés jogos lehet, hiszen elég kicsi az esély rá, hogy éppen pontosan egy ilyen objektumra lenne szüksége egy vállalatnak. Itt is érvényes az általánosítás: ez a dolgozat elsődlegesen nem tartalmi, hanem módszertani, automatizálási oldalról értelmezte az ár/teljesítmény-optimalizálás kihívását. Minden tartalmi döntés mindenkor a döntéshozó felelőssége. Ez felveti, hogy vajon ezen attribútum- és/vagy objektum-definíciós kérdésekben egy MI léphet-e előre az önállóság skáláján: a válasz olyan komplex, hogy egy önálló szakdolgozatot igényel.

Fentiek miatt az ár-teljesítmény vizsgálatokra egy-egy lehetséges példaként célszerű tekinteni, hiszen a dolgozat elején már említett „*context free*”, azaz tartalomfüggetlen megoldás bemutatása volt a célom.

A szoftver tekintetében felmerül, hogy értékes lehet-e egy potenciális megrendelő számára? Meglátásom szerint ebben a formában még nem piacképes a megoldás. A szoftvert egyrészt a saját munkám megkönnyítésére írtam (így az információs többletérték fogalma csak rám, mint megrendelőre és egyben felhasználóra érvényes), másrészt elősegítette a strukturált gondolkodás kialakítását számomra. A szakdolgozat terjedelmi korlátai nem teszik lehetővé egy-egy komplex téma minden részletének tetszőleges mélységű kidolgozását. Jelen dolgozat azonban példa arra, hogy a már kidolgozott és a kidolgozása méltóként felismert részletek hitelesen illene, hogy demonstrálják, hogy a szerző bármely szükséges irányba azonnal képes a munka folytatására. A dolgozat elején (vö. 1.5-ös fejezet) már bemutattam néhány előzmény munkát/szakdolgozatot, amelyek együttesen már elegendőek lehetnek egy piacképes projekt megalapozásához.

6.4. Jövőkép

A jövőben ehhez hasonló elemzések bármilyen (pl.: hardvereket vagy szoftvereket vizsgáló) informatikai beszerzési/közbeszerzési igényre elkészíthetők lesznek (vö. startup-képes core business).

A dolgozatban használt számításmenetek, összefüggések „*context free*” módon továbbra is felhasználhatóak a nagy közönség számára, annak ellenére, hogy az én konkrét példáim tartalmilag (vö. kontextusukat tekintve) néhány éven belül elavulnak.

A teljes automatizáció felé haladva elkészíthető lenne egy úgynevezett árbecslő szimulátor is, tehát az adaptációs lehetőségek adottak. Az egyik évfolyamtársam, Kolonics Zsolt egy IMDB filmszimulátoron dolgozik éppen, ami egy önálló szakdolgozati téma, és azt vizsgálja, hogy egy-egy film bizonyos tartalmi elemei alapján megbecsülhető-e az adott film IMDB pontszáma. A másik évfolyamokon is készül hasonló témájuk szakdolgozat, például www.arukereso.hu adatok alapján ár/teljesítmény-vizsgálat keretrendszer, illetve korábbi hallgatók által több ezzel foglalkozó egyetemi beadandó dolgozat is létrejött, ahogyan korábban a szakirodalmi fejezetben is említettem (vö. 2.4.1 fejezet).

Bízom benne továbbá (lévén záróvizsga szempontból elvileg kötelező), hogy az évfolyamtársaim illetve a következő évfolyamok elolvassák a szakdolgozatomat, illetve megismerik a programomat, akkor tudnak profitálni belőle és tudom segíteni a munkájukat hasonló elemzések vagy akár a teljesen automatizált árbecslő szimulátor elkészítésében.

7. Források

- Andrew S. Tanenbaum, David J. Wetherall. (2010.). Computer Networks (5th Edition) Letöltés dátuma: 2024.06.04. Forrás: <https://csc-knu.github.io/sys-prog/books/Andrew%20S.%20Tanenbaum%20-%20Computer%20Networks.pdf> doi: ISBN 978-0-13212695-3 / 2010
- Angyal János. (2024.03.28). Szerver ár-teljesítmény OAM fájl. Letöltés dátuma: 2024.06.24. Forrás: https://miau.my-x.hu/miau/311/server/szerver_ar_telj_oam.xlsx
- Angyal János. (2024.03.28). IT rendszerterv OAM fájl. Letöltés dátuma: 2024.06.24. Forrás: https://miau.my-x.hu/miau/311/server/it_rendszerterv_oam.xlsx
- Angyal János. (2024.04.20). Török konferencia előadás - Server price performance analysis. Letöltés dátuma: 2024.04.20. Forrás: <https://miau.my-x.hu/miau/311/server/ppt-server-price-performance-analysis.pptx>
- Bartók Patrik. (2023.11.15.). Objektív döntésoptimalizálás mesterséges intelligencia bevonásával. Letöltés dátuma: 2024.03.20. Forrás: https://miau.my-x.hu/miau/307/szakdolgozat_bp.pdf
- Bauer András, Berács József. (2016.). Marketing. Letöltés dátuma: 2024.05.15. Forrás: https://mersz.hu/dokumentum/dj204m__2/#dj204m_impresszum doi: ISBN 978 963 454 007 6 / 2016
- Bányász Péter, Budai Balázs Benjámin, Herendy Csilla, László Gábor, Orbán Anna, Orbók Ákos. (2022.). Digitális készségfejlesztés. Letöltés dátuma: 2024.04.29. Forrás: https://mersz.hu/dokumentum/m1151digk__2/#m1151digk_0 doi: ISBN 978 963 454 961 1 / 2022
- Boros Tamás - Filippov Gábor. (2020.): Magyarország 2030 - Jövőkép a magyaroknak. doi: ISBN: 9789632764160
- Chikán Attila. (2020.). Vállalati gazdaságtan. Letöltés dátuma: 2024.04.28. Forrás: https://mersz.hu/dokumentum/m795valgt__2/#m795valgt_0 doi: ISBN 978 963 454 589 7 / 2020
- Dr. Pitlik László. (dátum nélk.). COCO modulok weblapja. Forrás: <https://miau.my-x.hu/myx-free/coco>
- Dr. Pitlik László. (dátum nélk.). COCO kulcsszóra keresés. Magyar Internetes Agrár / Alkalmazott Informatikai Újság (MIAU). Forrás: <https://miau.my-x.hu/miau2009/index.php3?x=e0&string=coco> doi: ISSN 1419-1652 / 1998

- Dr. Pitlik László. (dátum nélk.). STEP IX leírása. Magyar Internetes Agrár / Alkalmazott Informatikai Újság (MIAU). Forrás: https://miau.my-x.hu/myx-free/index_e8.php3?x=stp4 doi: ISSN 1419-1652 / 1998
- EUROPOL. (2023.). ChatGPT - The impact of Large Language Models on Law Enforcement. Letöltés dátuma: 2024.03.19. Forrás: <https://www.europol.europa.eu/cms/sites/default/files/documents/Tech%20Watch%20Flash%20-%20The%20Impact%20of%20Large%20Language%20Models%20on%20Law%20Enforcement.pdf> doi: ISSN 2811-7719 / 2023
- Gorelov Iván. (2009.). Automatikus döntéshozatal online támogatásának lehetőségei az internetcsomagok közötti választás kapcsán. Országos Tudományos Diákköri Konferencia Debrecen, 2009. Letöltés dátuma: 2024.06.02. Forrás: <http://miau.my-x.hu/miau/128/otdk.ppt>
- Juhász Krisztián. (2024.). Modell-számítások (személyes közlés - munkaanyag)
- Kaufman & Canoles. (2024.). Trends in Software Licensing. Letöltés dátuma: 2024.06.04. Forrás: <https://www.kaufcan.com/news/articles/trends-in-software-licensing>
- Katonka Zsolt. (2017.). A mesterséges intelligencia szerepe a mindennapi életben. Letöltés dátuma: 2024.05.23. Forrás: http://dolgozattar.uni-bge.hu/12581/1/katonka_zsolt_2017j%C3%BAn_publikus.pdf
- Kocsis Gábor. (2020.). Szabadságnylvántartó rendszer tervezése Agilis módszerrel és UX design felhasználásával. Letöltés dátuma: 2024.03.15. Forrás: https://perepo-dolgozat.uni-pannon.hu/id/eprint/25259/1/kocsis%20g%C3%A1bor_2020jan_publikus.pdf
- Mark Lutz. (2009.). Learning Python (Fourth Edition). Letöltés dátuma: 2024.05.22. Forrás: https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf doi: ISBN 978-0-596-15806-4
- Marko Petkovsek, Herbert S Wilf, Doron Zeilberger. (1996.). A=B (First Edition). Letöltés dátuma: 2024.06.02. Forrás: <https://www.amazon.com/B-Marko-Petkovsek/dp/1568810636?asin=1568810636&revisionId=&format=4&depth=1> doi: ISBN 978-1568810638
- My-X Team. (2005.). COCO Y0 Működési logikája. Forrás: <https://miau.my-x.hu/myx-free/index.php3?x=e091>

- NBC NEWS. (2023.). Inside ChatGPT: How AI chatbots work. Letöltés dátuma: 2024.04.22.
Forrás: <https://www.nbcnews.com/data-graphics/chat-gpt-artificial-intelligence-how-chatbot-work-rcna83266>
- Peter Wentworth, Jeffrey Elkner, Allen B. Downey; Chris Meyers. (2019.). Hogyan gondolkozz úgy, mint egy informatikus: Tanulás Python 3 segítségével, 3. kiadás. Letöltés dátuma: 2024.04.22. Forrás: https://mtmi.unideb.hu/pluginfile.php/554/mod_resource/content/3/thinkcspy3.pdf
- Pitlik Marcell. (2020.). Térfogati égés optikai vizsgálata és mesterséges intelligencia alapú elemzése. Tudományos Diákköri Konferencia Budapest, 2020. Letöltés dátuma: 2024.04.03. Forrás: http://www.combustioninstitute.hu/tdk_dij/2021/PitlikMarcell_TDK_2020.pdf
- Stanfordi Egyetem. (dátum nélk.). Stanfordi Egyetem Hivatalos Honlapja. Letöltés dátuma: 2024. 03 25. Forrás: Dr. Donald Knuth profilja: <https://profiles.stanford.edu/donald-knuth?tab=bio>
- Székely Blanka. (2022.). Projektmenedzsment, avagy ami sikerre segíti a tanácsadót - vízesés vagy agilis módszertan, esetleg más? Letöltés dátuma: 2024.06.07. Forrás: https://www.researchgate.net/publication/361326842_Projektmenedzsment_avagy_ami_sikerre_segiti_a_tanacsadot_-_vizeses_vagy_agilis_modszertan_esetleg_mas
doi:10.18531/Studia.Mundi.2022.09.01.91-103
- Vörösmarty Gyöngyi, Tátrai Tünde. (2019.). Beszerzés. Letöltés dátuma: 2024.04.10. Forrás: https://mersz.hu/dokumentum/YOV1723__2/#YOV1723_0 doi: ISBN 978 963 295 844 6 / 2019
- Wikipédia (HU): Korreláció, <https://hu.wikipedia.org/wiki/Korreláció> - 2024.04.19
- Wikipedia (EN): ChatGPT, <https://en.wikipedia.org/wiki/ChatGPT> - 2024.04.20
- Wikipedia (HU): cURL, <https://hu.wikipedia.org/wiki/CURL> - 2024.04.21
- Wikipedia (EN): HTML, <https://en.wikipedia.org/wiki/HTML> - 2024.04.21
- Wikipedia (HU): SCRUM, <https://hu.wikipedia.org/wiki/Scrum> - 2024.04.21
- Wikipedia (EN): Firewall, [https://en.wikipedia.org/wiki/Firewall_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing)) - 2024.04.21

8. Mellékletek

Ebben a fejezetben a dolgozatban található rövidítések, definíciók és ábrák jegyzéke található.

8.1. Rövidítések jegyzék

COCO (Component based Object Comparison for Objectivity): hasonlóságelemzés. Optimalizálás keretében minden tulajdonság esetén minden rangsorszámhoz hozzárendelünk egy konkrét értéket (lépcsőt), amennyivel az adott tulajdonság hozzájárul a függő változó értékéhez. Megmutatja, hogy a vizsgált objektumok, a vizsgálatba bevont tulajdonságaik alapján, egymáshoz viszonyítva alul- vagy fölülértékeltek.

cURL: A cURL egy számítógépes szoftver projekt, amely több protokollon keresztüli fájllelérést biztosít egy parancssori eszköz és egy könyvtár segítségével. A cURL projekt két terméket hoz létre, a libcurl-t és a cURL-t. Először 1997-ben adták ki.

GUI: Graphical User Interface, azaz grafikus felhasználói felület, a számítógép és az ember közötti kapcsolatot biztosító elemek, amelyek a monitoron szöveges és képi formában jelennek meg.

HTML: HyperText Markup Language, amely egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

IMDB: Internet Movie Datavase, azaz a világ legnagyobb filmadatbázisa, és filmértékelő weblapja.

KPI: Key Performance Index, azaz magyarul fő teljesítménymutatót jelent.

MES: Manufacturing Execution System, a MES szoftvermegoldás egy automatizálási elemekkel kombinált informatikai rendszer, amelyet a gyártási folyamatok folyamatos nyomon követésére, állapotukra és a minőségükre vonatkozó információk megszerzésére használnak.

MI: Mesterséges Intelligencia.

OAM: Objektum Attribútum Mátrix, azaz egyfajta tanulási minta.

PDU: Power Distribution Unit, azaz áramelosztó egység.

SAP: System Applications and Products in Data Processing, azaz a világ vezető integrált vállalatirányítási rendszere.

SSD: Solid-state drive. Szilárdtestmeghajtó, egy mozgó alkatrész nélküli tároló eszköz számítógépekhez, laptopokhoz és szerverekhez.

TXT: Azaz text fájl, az informatikában legelterjedtebb szövegfájl formátum, amely nem tartalmaz semmilyen formázást.

UPS: Uninterruptible Power Supply, azaz szünetmentes tápegység. Az eszköz villamos energiát biztosít számítógépeknek és bármilyen más berendezéseknek, akkor is, ha a villamos hálózaton megszűnik a készülékek működéséhez szükséges tápfeszültség.

URL: Más néven webcím (pl. www.google.com).

USD: az Amerikai Egyesült Államok hivatalos pénznemének rövidítése, magyarul általában amerikai dollárnak nevezzük.

WMS: Warehouse Management Software, ez egy olyan szoftver, amely segíti a vállalatok napi raktárműveleteinek kezelésében és ellenőrzésében.

8.2. Definíciók jegyzék

ChatGPT 3.5: az OpenAI fejlesztő cég mesterséges intelligenciájának 3.5-ös verziója. URL: <https://gettalkative.com/info/gpt-models-compared>

ChatGPT 4o: az OpenAI fejlesztő cég mesterséges intelligenciájának 4o-s verziója. URL: <https://gettalkative.com/info/gpt-models-compared>

COCO STD: a COCO modul alapváltozata, amely objektumok ár-teljesítmény számításra képes, Y-ként minden esetben szükséges valós árat megadni. URL: https://miau.myx.hu/miau/189/coco_demo.pdf

COCO Y0: a COCO modul Y0 változata, amely teljesítmény elemzésre használható, minél nagyobb a kapott pont, annál jobb a teljesítmény. Az Y mindig fiktív és azonos. URL: https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://miau.myx.hu/miau/270/y0_alternative_approaches.docx&ved=2ahUKEwigrtX5qJqJAXWAzgIHHZl_EFoQFnoECBoQAQ&usg=AOvVaw30eqv7jdbYV6udH_IBvFOy

Consumer: normál fogyasztó, végfelhasználó. URL: <https://linguajuris.hu/customer-vs-consumer/>

Excel-OAM-COCO konverter: egy saját fejlesztésű szoftver, amely Excel alapú OAM-ok feldolgozására használható. (saját definíció)

Excel-OAM-COCO STD konverter: a saját fejlesztésű szoftverem Standard változata, amely Excel alapú OAM-ok feldolgozására használható. (saját definíció)

Excel-OAM-COCO Y0 konverter: a saját fejlesztésű szoftverem Y0 változata, amely Excel alapú OAM-ok feldolgozására használható. (saját definíció)

Executable: a Windows alapú operációs rendszereknél használatos fájlformátum, amely a szoftverek futtatását teszi lehetővé, tehát végrehajtandó kódot és egyebeket, pl. ikont tartalmaz. URL: <https://hu.wikipedia.org/wiki/EXE>

Google Trends: a Google Trends egy ingyenesen hozzáférhető eszköz, amely lehetővé teszi a különböző keresési kulcsszavak és kifejezések statisztikáinak elemzését. Az adatokat különböző földrajzi területekre bontva jeleníti meg, kezdve a globális szinttől egészen a specifikus országokig, mint például Magyarország. Továbbá, amennyiben elegendő adat áll rendelkezésre, lehetőség van a statisztikai adatok részletesebb bontására is, például Magyarország egyes megyéire vonatkozóan. URL: <https://thepitch.hu/google-trends-hasznalata/>

HQ (headquarter): főhadiszállás, cégeknél az anyavállalatot nevezzük így. URL: <https://www.hrportal.hu/jelentese/hq--20210409.html>

Intel AI: az Intel processzorgyártó cég mesterséges intelligencián alapuló megoldása processzorokhoz. URL: <https://www.intel.com/content/www/us/en/artificial-intelligence/overview.html>

Kibertámadás: olyan támadási kísérlet, amelyben illetéktelen behatolással jogosulatlan hozzáférést próbálnak szerezni számítástechnikai rendszerekhez, adatlopást illetve egyéb károkozást is megvalósíthatnak. URL: <https://www.microsoft.com/hu-hu/security/business/security-101/what-is-a-cyberattack>

Library: könyvtárat jelent, a programozási nyelvben az előre megírt kódok gyűjteménye. URL: <https://www.geeksforgeeks.org/libraries-in-python/>

Microsoft Excel: a Microsoft szoftvergyártó vállalat táblázatkezelő programja. URL: https://en.wikipedia.org/wiki/Microsoft_Excel

Non-BOM: indirekt beszerzés, tehát minden olyan termék vagy szolgáltatás, ami nem az üzemi termeléshez kapcsolódik. URL: https://mersz.hu/dokumentum/wk40__17/

OpenAI: egy amerikai cég, amely mesterséges intelligencia kutatással foglalkozik. URL: <https://en.wikipedia.org/wiki/OpenAI>

Python: általános célú, magas szintű programozási nyelv. URL: [https://hu.wikipedia.org/wiki/Python_\(programoz%C3%A1si_nyelv\)](https://hu.wikipedia.org/wiki/Python_(programoz%C3%A1si_nyelv))

SCRUM: a Scrum egy összetett (komplex) problémák megoldására kifejlesztett általános keretrendszer. A szoftverfejlesztés területén gyakran használják az agilis szoftverfejlesztés egyik fontos összetevőjeként, de ugyanúgy megtaláljuk szolgáltatások fejlesztésénél, szervezetek irányításánál, még iskolákban is. URL: <https://hu.wikipedia.org/wiki/Scrum>

STEP IX: olyan modul, amely a hasonlóságelemzés eredményeként egyensúlyi állapotokat mutató OAM esetén feloldja a legjobb ár-teljesítmény-viszony látszólagos hiányának dilemmáját. URL: https://miau.my-x.hu/myx-free/index_e8.php3?x=stp4

Tűzfal: a tűzfal (vö. angolul firewall) célja a számítástechnikában annak biztosítása, hogy a hálózaton keresztül egy adott számítógépbe ne történhessen illetéktelen behatolás. Szoftver- és hardverkomponensekből áll. URL: [https://en.wikipedia.org/wiki/Firewall_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing))

VMWARE: egy amerikai szoftvercég, amely operációs rendszerek virtualizációjára szakosodott, tehát ezeket az operációs rendszereket nem feltétlen kell fizikai szervereken tárolni. URL: <https://www.techtarget.com/searchvmware/definition/VMware>

VPN: a virtuális magánhálózat lehetővé teszi a felhasználók számára, hogy egy megosztott vagy nyilvános hálózaton keresztül úgy küldjenek és fogadjanak adatokat, mintha számítógépeik közvetlenül kapcsolódnának a helyi hálózathoz. A kapcsolat „magán”-jellegét az adja, hogy a VPN-en keresztül menő adatok a titkosítás miatt nem láthatók az eredeti hálózaton. A titkosítás általánosan használt szolgáltatása a VPN-nek, de titkosítás nélkül is használható a különböző adatfolyamok elkülönítésére vagy a hálózat logikai felépítésének egyszerűsítésére. A VPN-funkciót gyakran beépítik a routerekbe, de sok esetben szoftveresen valósítják meg. URL: <https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn>

8.3. Ábrajegyzék

1. ábra - Vízesés modell	14
2. ábra - ChatGPT 3.5 adatbázisának dátuma.....	19
3. ábra - Beszerzési területek.....	23
4. ábra - OAM.....	34
5. ábra - Rangsorolás (OAM)	37

6. ábra - Naiv becslések.....	38
7. ábra - COCO elemzés (STD eredmény).....	39
8. ábra - A tényleges és a becsült árak távolsága, a háttérszámítások nélkül. (USD).....	39
9. ábra - STEP IX sorozat OAM táblázatai, alsó-felső határ (Objektum 1).....	41
10. ábra - STEP IX COCO STD elemzés eredménye, alsó határ (Objektum 1).....	41
11. ábra - STEP IX COCO STD elemzés eredménye, felső határ (Objektum 1).....	42
12. ábra - STEP IX OAM (min./max.) Y0 elemzéshez.....	42
13. ábra - Rangsorolás (OAM).....	43
14. ábra - COCO elemzés (Y0 eredmény).....	44
15. ábra - COCO elemzés (becslés-validitás-konklúzió).....	44
16. ábra - Végeredmény (helyezések).....	45
17. ábra - Inverz OAM táblázat.....	45
18. ábra - Validitás ellenőrzés táblázatai (ellentétes előjelek).....	46
19. ábra - OAM.....	51
20. ábra - Rangsorolás (OAM).....	52
21. ábra - Naiv becslések.....	53
22. ábra - COCO elemzés (STD eredmény).....	54
23. ábra - A tényleges és a becsült árak távolsága, a háttérszámítások nélkül (USD).....	54
24. ábra - STEP IX sorozat OAM táblázatai, alsó-felső határ (Objektum 1).....	55
25. ábra - STEP IX COCO STD elemzés eredménye, alsó határ (Objektum 1).....	56
26. ábra - STEP IX COCO STD elemzés eredménye, felső határ (Objektum 1).....	56
27. ábra - STEP-IX OAM (min./max.) Y0 elemzéshez.....	57
28. ábra - Rangsorolás (OAM).....	57
29. ábra - COCO elemzés (Y0 eredmény).....	58
30. ábra - COCO elemzés (becslés-validitás-konklúzió).....	59
31. ábra - Végeredmény (helyezések).....	59
32. ábra - Inverz OAM táblázat.....	60
33. ábra - Validitás ellenőrzés táblázatai (ellentétes előjelek).....	60
34. ábra - STEP IX sorozat OAM táblázata, alsó határ (Objektum 1).....	61
35. ábra - STEP IX sorozat OAM táblázata, alsó határ becslés (Objektum 1).....	62
36. ábra - STEP IX sorozat OAM táblázata, alsó határ, új értékkel (Objektum 1).....	62
37. ábra - STEP IX sorozat OAM táblázata, alsó és felső határ becslések (Objektum 1).....	63
38. ábra - Objektum 1 alsó határ, függvényalak.....	64
39. ábra - Objektum 4 és 5 alsó határ COCO STEP IX+ árdiagrammok.....	65

40. ábra - Nagyvállalat informatikai rendezőszekrény ábrája (szerver szoba).....	68
41. ábra - Grafikus felület (COCO STD, COCO Y0).....	72
42. ábra - Excel-OAM-COCO STD konverter, a programkód egy részlete.....	73
43. ábra - Manuális és szoftveres végeredmény validációja	78
44. ábra - Manuális tesztesetek, végeredmények és validáció	78
45. ábra - Sorszámok cseréjének hatása (pivot táblázat)	80
46. ábra - Pivot táblázat adatforrásának egy részlete	82
47. ábra - COCO elemzés (STD eredmény) - OAM1, 2→1 sorszámcsere	82
48. ábra - Kiindulási OAM táblázat, csere esetek száma	84