



A MISKOLCI EGYETEM HABILITÁCIÓS FÜZETEI
INFORMATIKAI TUDOMÁNYOK TUDOMÁNYÁGI HABILITÁCIÓS BIZOTTSÁG

Tudományos munkásság áttekintő összefoglalása

Írta:

Dr. Hornyák Olivér

egyetemi docens

aki az Informatikai Tudományok Tudományágban „dr. habil.” cím elnyerésére pályázik

Miskolc

2025.

Tartalomjegyzék

Tartalomjegyzék	2
1 Előszó	4
2 A Blockchain technológián alapuló tudományos eredmények	6
2.1 Blokklánc technológia	6
2.2 Az okos városok - nyílt adatok koncepciója blokklánc-technológia alapokon.....	7
2.3 Az okos szerződések jelentősége az autópárhánban.....	8
2.4 Tézisek.....	10
2.5 A témához kapcsolódó publikációk	10
2.6 Felhasznált irodalom.....	11
3 Automatizált feladatértékelő rendszer online informatikai kurzusokhoz	13
3.1 Online informatikai kurzusok	13
3.2 Az automatikus feladatkiértékelés kihívásai.....	13
3.3 Programozási feladat kiértékelés megvalósítása a MeMOOC rendszerben	14
3.4 Tézisek.....	18
3.5 A témához kapcsolódó publikációk	19
3.6 Felhasznált szakirodalom	19
4 Szelektált rajzi elemek automatikus felismerése	21
4.1 A feladat megfogalmazása	21
4.2 Az irodalomban ismert módszerek áttekintése	22
4.3 A kidolgozott, saját fejlesztésű, szabadalmaztatott eljárás	23
4.3.1 Címkezett izomorf algráf-keresés visszalépéses heurisztikával.....	23
4.4 Tézisek.....	26
4.5 A témához kapcsolódó publikációk	26
4.6 Felhasznált irodalom.....	26
4.7 Az eredetiség kutatás eredménye	27
5 Termelésütemezési feladatok megoldása	29
5.1 Diszkrét gyártási folyamatok.....	29

5.2	A flowshop ütemezési feladat	29
5.3	Bakteriális memetik algoritmus (DBMEA)	31
5.3.1	Az algoritmus bemutatása	32
5.3.2	A bakteriális mutációs eljárás	33
5.3.3	Gén-traszfer művelet	34
5.3.4	A Monte-Carlo fakesés algoritmus	35
5.3.5	Hibrid DBMEA algoritmus	37
5.3.6	Futási eredmények	37
5.4	Kiterjesztett flowshop ütemezési feladat	38
5.5	Többcélű heurisztikus ütemezés	39
5.6	Tézisek	40
5.7	A tézishoz kapcsolódó publikációk	41
5.8	Felhasznált irodalom	41
6	<i>Az állapotfelügyelet AdaBoost algoritmussal és az alkalmazási korlátok feltárása</i>	43
6.1.1	Osztályozási feladatok	43
6.1.2	Állapotfelügyelet és karbantartás	43
6.2	Az AdaBoost algoritmus	44
6.3	Élettartam tervező és meghibásodás előrejelző komplex döntéstámogató rendszer, facility management szolgáltatás kialakításához	45
6.4	Tézisek	46
6.5	A tézishoz kapcsolódó publikációk	46
6.6	Felhasznált irodalom	47
7	<i>Summary</i>	49
8	<i>Köszönetnyilvánítás</i>	51

1 Előszó

1997-ben a Miskolci Egyetem Gépészmérnöki Karán végeztem okleveles gépészmérnökként informatika szakirányon. Ezután doktoranduszként az egyetem Alkalmazott Informatikai Tanszékére kerültem. A 1997-2000 között végzett doktori tanulmányom lezárásaként az „Esztergálási műveletek kiterjesztett számítógépes szimulációja intelligens módszerek alkalmazásával” című PhD értekezésemet 2003-ban védtem meg „summa cum laude” minősítéssel. A fokozatszerzés után folytattam a tudományos tevékenységet az alkalmazott informatika szerteágazó területein.

A termelésinformatikai területek sokszínűsége és multidiszciplináris jellege folyamatosan új kihívások elé állítottak. Ez visszaköszön a habilitációs téziszfüzet sokszínűségében. Az elvégzett kutatások nemcsak tudományos értékkel bírnak, hanem gyakorlati alkalmazásuk révén gazdasági előnyöket is teremtettek, több megfogalmazott tézis konkrét ipari kutatások során csiszolódott ki. A termelésinformatikai megoldások képesek növelni a gyártási folyamatok hatékonyságát, csökkenteni a selejtarányt, optimalizálni az erőforrás-felhasználást, és megbízhatóbbá tenni a karbantartási stratégiákat. Az informatika versenyelőnye abban rejlik, hogy lehetőséget biztosít az adatok gyors és intelligens feldolgozására, valós idejű döntéshozatalra és az ipari folyamatok automatizálására. A korszerű algoritmusok és prediktív modellek alkalmazásával a vállalatok nemcsak költségeiket csökkenthetik, hanem piaci rugalmasságukat is növelhetik, hiszen az adatvezérelt működés révén gyorsabban és pontosabban reagálhatnak a változó piaci igényekre. Az ilyen informatikai fejlesztések tehát közvetlenül hozzájárulhatnak egy vállalat versenyképességéhez, és hosszú távon fenntartható gazdasági növekedést eredményezhetnek.

Az elmúlt több, mint húsz év során elvégzett tudományos tevékenységemet az alábbi területeken foglalom össze:

1.) A blokklánc technológián alapuló tudományos eredmények

Rávilágítottam arra, hogy a blokklánc-technológia jelentős szerepet játszhat az okos városok működésének fejlesztésében. A blokklánc-alapú nyílt innováció egy olyan megközelítés, amely a blokklánc-technológia decentralizált, átlátható és biztonságos adattárolási és hitelesítési képességeit használja fel az innovációs folyamatok hatékonyabbá tételére. A lényege, hogy a vállalatok, kutatóintézetek, önkormányzatok és egyéb szereplők megosztják egymással az új ötleteket, technológiákat és fejlesztéseket, így felgyorsítva az innovációs ciklusokat és elősegítve az együttműködésen alapuló fejlődést. A decentralizált rendszerek alkalmazása révén növelhető az adminisztráció hatékonysága, fokozható a közszolgáltatások elfogadottsága, és javítható azok minősége.

Az ellátási láncok átláthatósága és hatékonysága is növelhető a blokklánc által biztosított, megváltoztathatatlan adatnyilvántartásokkal, lehetővé téve az alkatrészek eredetének, gyártásának és szállításának pontos nyomon követését. Az okos szerződések

automatizálhatják a hulladékgazdálkodást és optimalizálhatják az újrahasznosítási folyamatokat, ezáltal mérsékelve például az autóipar környezeti terhelését.

- 2.) Az online programozás oktatásának automatizált feladatkiértékelési megoldásait mutatja be a második téziscsoport. A kifejlesztett kiértékelő modul figyelembe veszi az informatika oktatásban megjelenő programozási feladatok sajátosságait, amikor különböző aspektusokból vizsgálva értékeli a hallgatói feladatot. A kidolgozott módszer feladatgenerálásra is alkalmas, és online kurzusanyagok kidolgozásában hasznosítható.
- 3.) A szelektált rajzi elemek automatikus felismerése feladatkörben szabadalmaztatott eljárást egy többfelhasználós interaktív alkalmazás formájában fejlesztettem ki kollégáimmal. A feladatot ipari igények alapján oldottuk meg azzal a céllal, hogy műszaki rajzok kiválasztott elemeiből elvégezze a jellemzően műszaki-technológiai adatok kinyerését. Az újdonságértéke két a szelekciós és a felismerés alappilléreken nyugszik. Hatékony heurisztikus algoritmussal képes címkézett részgráfokat keresni (szelekció), és a speciálisan kifejlesztett karakter-felismerő algoritmus képes a műszaki rajzokon használatos jelölések tetszőleges orientációjú adatainak kezelése (felismerés).
- 4.) A tudományos kutatásaim egy része a termelésütemezés témakörébe esik. A kutatások célja a gyártási folyamatok hatékonyságának növelése volt, különféle ütemezési modellek és optimalizálási technikák alkalmazásával. A vizsgált feladatok jellegzetessége, hogy a mögöttük álló rendszerek komplexek és a különböző tényezők hatásáról függ az elérhető teljesítmény. A diszkrét bakteriális memetik algoritmusok (DBMEA) alkalmazhatóságát vizsgáltam a termelésütemezés terén, és rámutattam arra, hogy különböző vizsgált benchmark problémákban, amelyek valós gyártási környezeteket modelleztek, a DBMEA képes volt az eddig ismert megoldásoknál jobb megoldásokat előállítani. Az elért eredmények alapján kimutatható, hogy a DBMEA nemcsak megállja a helyét a jelenlegi optimalizálási technikákkal szemben, hanem bizonyos esetekben előnyöket is kínál, amelyek hozzájárulnak a termelési ütemezési problémák hatékonyabb megoldásához.
- 5.) Az AdaBoost algoritmuson alapuló állapotfelügyelet tématerületen egy kutatási pályázat megvalósításának keretei között dolgoztam. Ezek a prediktív modellezés területén végzett kutatásaim több aspektust vizsgálnak, különös tekintettel az algoritmus alkalmazhatóságára és korlátaira. Rávilágítottam arra, hogy az AdaBoost bizonyos adattípusok esetén gyenge eredményeket adhat, ami befolyásolja az ipari rendszerekben való felhasználhatóságát. Bemutattam, hogyan alkalmazható az algoritmus adatvezérelt motorállapot-felügyeletben, illetve hogyan támogatja a maradék hasznos élettartam (Remaining Useful Life, RUL) becslését, amely kulcsfontosságú a prediktív karbantartásban. Továbbá áttekintettem a sorozat-előrejelzési problémák kiértékelési módszereit, amelyek az AdaBoost teljesítményének mérésében is relevánsak.

2 A Blockchain technológián alapuló tudományos eredmények

2.1 Blokklánc technológia

Az elmúlt évtizedekben a blokklánc technológia (Blockchain technology, BT) jelentős fejlődésen ment keresztül, és egyre inkább a digitális transzformáció meghatározó elemévé vált. Ez a technológia, amely először a 2000-es évek végén jelent meg, mint a legismertebb kriptovaluta, a Bitcoin alaparchitektúrája, mára túllépett eredeti célján. Hasonlóan az internethez és más forradalmi technológiákhoz, a blokklánc technológia ma már az élet számos területén képes alapvető változásokat előidézni. Különösen a pénzügyi szektorban mutatja meg átalakító erejét, miközben jelentős hatást gyakorol a digitális gazdaság és a társadalom más aspektusaira is.

2025-re a blokklánc technológia nemcsak kriptovalutákra korlátozódik, hanem számos iparágban elterjedt, beleértve az ellátási lánc menedzsmentet, az egészségügyet, a jogi rendszereket és a szellemi tulajdon (Intellectual Property, IP) kezelését. A blokklánc képes forradalmasítani a nyílt innováció (Open Innovation, OI) területét is, mivel decentralizált, biztonságos és átlátható platformokat kínál, amelyek lehetővé teszik az együttműködést és az értékek megosztását a szereplők között.

Napjainkban a Bitcoin népszerűsége és értéke továbbra is meghatározó a kriptovaluták piacán: 2024-ben a piaci kapitalizációja már meghaladja a 600 milliárd amerikai dollárt, és ára tartósan 30 000 USD fölött mozog. A kriptovaluták ökoszisztémája jelentősen bővült az elmúlt években, a blokklánc alapú eszközök között olyan digitális valuták, mint az Ethereum (ETH), a Cardano (ADA), a Binance Coin (BNB) és a Solana (SOL), szintén jelentős piaci kapitalizációt értek el.

Ezeket a kriptovalutákat egyre szélesebb körben fogadják el fizetőeszközként, beleértve globális e-kereskedelmi platformokat és számos más nemzetközi vállalatot. A Bitcoin és más digitális valuták népszerűsége az infláció elleni védelem és a decentralizált, biztonságos tranzakciók lehetősége miatt tovább növekszik, ami a blokklánc technológia általános elfogadottságát is erősíti.

A blokklánc technológia azonban sokkal többre képes, mint pusztán kriptovaluták támogatása. Az Ethereum platform és a kapcsolódó virtuális valuta, az Ether (ETH) 2015-ös bevezetésével lehetővé vált blokklánc-alapú alkalmazások fejlesztése szinte bármilyen szektor számára. Ezek az alkalmazások okosszerződések segítségével bármilyen értékű tranzakció rögzítésére használhatók, legyen szó hagyományos pénznemekről, árucikkekről (például aranyról vagy olajról), energiáról, ingatlanszerződésekről vagy akár a nyílt innovációban központi szerepet játszó szellemi tulajdonjogokról.

A blokklánc technológia egyik előnye, hogy azonnali, decentralizált és biztonságos tranzakciókat tesz lehetővé közvetítők, például brókerek vagy ügynökök nélkül. Az innováció egyszerre zseniális és széles tömegek számára vonzó, fejlődése pedig megállíthatatlannak tűnik.

Számos blokklánc-alapú alkalmazás lépte túl a megvalósíthatósági tanulmány szintjét és bevezetésre került. Ezek közül sokat úgynevezett ICO-k (Initial Coin Offering) vagy tokeneladások finanszíroznak, amelyek során értéktokeneket bocsátanak ki és osztanak szét más virtuális vagy hagyományos pénznemekkel való fizetés után. A tokenek a befektető számára részesedést nyújtanak az alkalmazás mögött álló vállalatban, vagy lehetőséget biztosítanak a tokenek kereskedelmére kriptovaluta-tőzsdéken. Ezen a módon jelentős összegeket sikerült összegyűjteni. Az ebbe az új iparágba áramló tőkeösszegek a technológia fontosságát és folyamatos fejlesztését jelzik.

A BT által előidézett paradigmaváltást felismerték a pénzintézetek is. Néhányan közülük partnerségeket alakítottak ki blokklánc-vállalatokkal, saját fejlesztéseket indítottak, és szabadalmi bejelentéseket is tettek BT-innovációkra. A blokklánc-technológia témakörében jelentős mennyiségű szellemi tulajdon jön létre ahogy ezt a BT-hez kapcsolódó szabadalmi és védjegybejelentések növekedése is mutatja. Jelenleg, 2024-ben, az Európai Szabadalmi Hivatal (EPO) Espacenet adatbázisában több, mint 10 000 „blockchain” és több, mint 1000 „bitcoin” említésű szabadalmi bejegyzést találhatunk. Évek múlva ezek az áttörést jelentő szabadalmak értékesnek bizonyulhatnak. Emellett a milliányi megírt kódsor is jelentős szellemi tulajdont képezhet, amely szerzői jogvédelem, adatbázis-védelem alá kerülhet, vagy titokban tartható, annak ellenére, hogy a projektek többsége nyílt forráskódúként jelenik meg.

A BT egyik legérdekesebb aspektusa, hogy képes átalakítani az innovációs folyamatokat. A BT használata segíthet radikálisan újragondolni az üzleti folyamatokat és a versenysztratégiákat, ami egybevág az Open Innovation, OI céljaival.

A BT megértéséhez érdemes átgondolni, hogy sem az üzleti világ, sem a kormányok nem működnek elszigetelten. Mindannyian részei egy üzleti hálózatnak, ahol az eszközök tulajdonjoga folyamatosan átruházódik, amit szerződések szabályoznak. Jelenleg a hálózat résztvevői külön főkönyveket vezetnek, amelyekben rögzítik a tulajdonukat, és frissítik az adataikat, amikor a tulajdon jog változik. Bár ez a rendszer bevált, gyakran nagyon költséges és kevésbé hatékony.

A BT ezzel szemben lehetővé teszi a tranzakciók érvényességének közös elfogadását, a tulajdonjogok időbeli auditálását, egy közös, manipulációtól mentes főkönyv vezetését, valamint a tranzakciók véglegességének garantálását. Emellett a kormányzati felügyelet, megfelelőség és auditálás is integrálható ebbe a hálózatba.

2.2 Az okos városok - nyílt adatok koncepciója blokklánc-technológia alapokon

Az okos városok (Smart cities) koncepciója a modern városok gazdasági, társadalmi és környezeti problémáinak hatékonyabb kezelésére törekszik, az információs és kommunikációs technológiák (ICT) alkalmazásával. A cél városi szolgáltatások fejlesztése, hatékonyság,

biztonság, környezetvédelem, gazdaság figyelembevételével. A nyílt innováció kulcsszerepet játszik az ilyen technológiák összekapcsolásában és egymás előnyeinek kiaknázásában.

A városok működésében óriási adatmennyiség keletkezik, amelyek kezelése, átláthatósága és biztonsága kulcsfontosságú. A blokklánc-technológia képes egy semleges, decentralizált és biztonságos adatbázist létrehozni, amely elősegíti az átláthatóságot és csökkenti a korrupció kockázatát. Ez az alapja lehet a nyílt innováció további fejlődésének az okos városokban.

Az okos városok a digitalizációra és a hiper-összekapcsoltságra épülnek, amelyet az Internet, az IoT (dolgok internete), a mesterséges intelligencia és a robotika támogat. Az új társadalmi elvárások – például a nagyobb átláthatóság, hatékonyabb közigazgatás és szélesebb hozzáférés a közadatokhoz – együtt a nagy adathalmazok (big data) és a nyílt adatok megjelenésével átalakítják a kormányzatokat és a közszolgáltatásokat.

A blokklánc-technológia szerepe az okos városokban a következőképpen foglalható össze:

- Adatmegosztás és adatvédelem: A BT lehetővé teszi, hogy a szereplők biztonságosan osszanak meg adatokat egymással, ezáltal elősegítve az innovációt nyílt adatalapú csatornákon keresztül.
- Időbélyegzés és digitális aláírások: A BT időbélyegzési funkciója különösen hasznos lehet az IoT-alapú tranzakciókban és a B2B (üzlet-üzlet között) folyamatokban.
- Okos szerződések (Smart Contracts): Automatizálják és költséghatékonyá teszik a tranzakciókat, miközben biztosítják az adatok integritását.

Az okos városok fejlődésében a BT integrációja új típusú „kripto városokat” (Crypto City) hozhat létre, amelyek decentralizált adat-infrastruktúrákra épülnek. Ezek az infrastruktúrák lehetővé teszik a helyi közszolgáltatások civil irányítását, valamint gazdasági és társadalmi vállalkozások támogatását az OI elősegítése érdekében.

Az OI számára nélkülözhetetlen a nyers adatok és az előzetesen validált adatok elérése. A BT biztosítja ezek elérhetőségét és a biztonságos adatcserét, azonban szabványokra van szükség a rendszer-összehangoltság érdekében.

2.3 Az okos szerződések jelentősége az autóiparban

Az okos szerződések a hagyományos szerződések digitális, automatizált megfelelői, amelyek a feltételeket és a rendelkezéseket kód formájában rögzítik. A felek közötti megállapodásokat először kódolják, majd aláírják, és feltöltik a blokkláncra. Ezek a szerződések a blokkláncon tárolt elosztott kódreszletekként működnek, és önállóan végrehajthatók, amint a meghatározott feltételek és szabályok teljesülnek. Az okos szerződések önálló működése lehetővé teszi, hogy külső beavatkozás nélkül, megbízhatóan hajtsák végre a szerződésben foglaltakat.

Bár az okos szerződések fogalma nem kizárólagosan a blokklánc technológiához kötött, általában automatizált, önállóan végrehajtható tranzakciókkal azonosítják. Jogi szempontból

ezek két típusra oszthatók. Az egyik esetben a szerződés rugalmasan módosítható - alacsony költséggel -, míg más esetekben a módosítás költsége olyan magas lehet, hogy gyakorlatilag nem megvalósíthatónak tekinthetjük. Az okos szerződések rugalmassága érdekében léteznek frissítési mechanizmusok, amelyek jogi vagy technikai változtatásokhoz igazodhatnak. Például a szerződés frissítése lehetséges nyilvános blokklánc adatok vagy API-k segítségével. A szerződések másik fajtája esetén a szerződési feltételek lehetnek módosíthatatlanok is, ezzel garantálva, hogy a felek ne változtathassák meg azokat egyoldalúan.

Az okos szerződések technológiai alapja az, hogy önálló működésük során pontosan végrehajtsák a kódolt feltételeket, minimalizálva az emberi hibák vagy szándékos manipuláció lehetőségét. Ezzel az innovációval a szerződéskötés gyorsabbá, hatékonyabbá és biztonságosabbá válik.

Az okos szerződések számos előnnyel rendelkeznek a hagyományos szerződésekkel szemben, digitális, automatizált természetüknek köszönhetően. Ezek a számítógépes programok pontosan rögzítik a jogi feltételeket és kitételeket, majd biztosítják azok betartását, függetlenül az állami jogi környezettől. Az okos szerződések kiküszöbölik az emberi félreértésekből adódó problémákat, és garantálják az érintett felek közötti megállapodások pontos végrehajtását.

Az okos szerződések egyik kiemelkedő előnye, hogy digitális természetük biztosítja a szerződés végleges változatát, amelyet számítógépek pontosan hajtanak végre az előírt feltételek teljesülésekor. Ezáltal a papírentes működés lehetővé válik, elkerülve a hagyományos szerződések megsemmisülésének, károsodásának vagy többszörözésének problémáit. Az összes feltétel és kitétel egyértelműen kódolva van, így elkerülhetők a félreértelmezések. A blokkláncon való tárolás biztosítja, hogy minden szükséges adat részletesen rögzítésre kerüljön, így nincs szükség további tárolási vagy biztonsági mentési megoldásokra.

Az okos szerződések gyorsan és pontosan hajtják végre a tranzakciókat, így nincs késlekedés a feldolgozás során. Biztosítják a résztvevő felek közötti bizalmat, mivel az adatok és a tranzakciók átláthatók és biztonságosak. A blokklánc alapú szerződések különösen előnyösek az olyan iparágakban, mint az autóipar, mivel automatizálják a meglévő folyamatokat, új iparágakat és piacokat hoznak létre, valamint megkönnyítik az „ha-akkor” típusú utasítások és az értékátvitel kódolását.

A blokklánc digitális, biztonságos és ellenálló környezetet biztosít, amely részletes és változtathatatlan tranzakciós előzményeket tart fenn, ezáltal ideális platformot nyújt az okos szerződések számára. Az ezekkel járó automatizáció és megbízhatóság alapvetően változtathatja meg a különböző iparágak működését és növelheti hatékonyságukat.

A blokklánc technológia rendkívül ígéretes megoldásként jelenik meg az autóipar számára, amely képes biztonságossá tenni az autók használatával kapcsolatos adatokat, támogatni a pénzügyi tranzakciókat, biztosítani az alkatrészek nyomonkövethetőségét az ellátási láncban, és automatizálni a járműelosztás különböző fázisait.

Az autóiipari cégek számára kulcsfontosságú, hogy azonosítsák azokat a területeket, ahol a blokklánc a legnagyobb előnyöket kínálja. Az értékteremtési lánc minden szakaszában lehetőség van a blokklánc-alapú megoldások bevezetésére, beleértve a gyártási fázist, az ellátási lánc menedzsmentet, valamint a járművek nyomon követését és újrahasznosítását.

Az okos szerződések szerepe különösen fontos a járművek adatainak biztonságos kezelése, a fizetési folyamatok egyszerűsítése és az autóregisztráció automatizálása terén. Az okos szerződéseket használhatják a tulajdonjog átruházására vagy az alkatrészek nyomon követésére az ellátási láncban. Ehhez a területhez tartozik még digitális járműazonosítás, a járműtörténet és adatok nyomon követése, az autonóm járművek és gépek fizetési rendszerei, a biztonságos mobil kereskedelem, valamint a használat alapú mobilitási árazás és biztosítási rendszerek.

Az autóiipar jövője egyre inkább a mobility-as-a-service modell irányába tolódik, ahol a digitális adatok és megoldások kulcsszerepet játszanak. Az okos szerződések és a blokklánc együttesen átláthatóbbá, biztonságosabbá és hatékonyabbá tehetik az autóiipar működését, új piacokat teremthetnek, és jelentősen csökkenthetik a tranzakciós költségeket. Ez a technológia hosszú távon alapvetően formálhatja át az autóiipart, miközben növeli az ágazat fenntarthatóságát és hatékonyságát.

2.4 Tézisek

T2.1 A blokklánc-technológia jelentős potenciállal rendelkezik az okos városok működésének javításában, különösen a nyílt innováció támogatásában. Az Open Innovations erősítésével és a decentralizált rendszerek bevezetésével a blokklánc technológia hozzájárulhat az adminisztráció hatékonyságának növeléséhez, az elfogadottság növeléséhez, a közszolgáltatások minőségének javításához. Az új technológiák alkalmazása azonban még kísérleti fázisban van, és hosszabb távú integrációt igényel.

T2.2 A blokklánc és az okos szerződések hozzájárulnak a fenntartható autóiiparhoz ideértve a gyártási, kereskedelmi, használati, karbantartási és újrahasznosítási szakaszokat. Az ellátási láncok átláthatóbbá és hatékonyabbá válnak a blokklánc által biztosított változtathatatlan adatnyilvántartásokkal. az alkatrészek eredete, gyártása és szállítása nyomon követhető, csökkentve a selejtarányt és az illegális tevékenységeket. Az okos szerződések lehetővé teszik az újrahasznosítási folyamatok optimalizálását, automatizálják a hulladékgazdálkodást, ezzel mérsékelve az iparág környezeti terhelését.

2.5 A témához kapcsolódó publikációk

[P2.1] De La Rosa, J. L., Torres-Padrosa, V., El-Fakdi, A., Gibovic, D., Hornyák, O., Maicher, L., & Miralles, F. (2017. december). A survey of blockchain technologies for open innovation. In *Proceedings of the 4th Annual World Open Innovation Conference* pp. 1-27.

[P2.2] Hornyák, Olivér & Alkhoury, George Farid. Smart contracts in the automotive industry. In: *Vehicle and automotive engineering*. Singapore: Springer Singapore, 2020. pp. 148-157., SJR: Q4

2.6 Felhasznált irodalom

- [1] Bartoletti, M., & Pompianu, L. (2017). An empirical analysis of smart contracts: Platforms, applications, and design patterns. In *International Conference on Financial Cryptography and Data Security* (pp. 494–509). Springer, Cham.
- [2] Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access*, 4, 2292–2303.
- [3] Clack, C. D., Bakshi, V. A., & Braine, L. (2016). Smart contract templates: Foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*.
- [4] Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. (2016). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International Conference on Financial Cryptography and Data Security* (pp. 79–94). Springer, Berlin, Heidelberg.
- [5] Dolgui, A., Ivanov, D., Potryasaev, S., Sokolov, B., Ivanova, M., & Werner, F. (2020). Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain. *International Journal of Production Research*, 58(7), 2184–2199.
- [6] Fraga-Lamas, P., & Fernández-Caramés, T. M. (2019). A review on blockchain technologies for an advanced and cyber-resilient automotive industry. *IEEE Access*, 7, 45201–45218.
- [7] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 839–858).
- [8] Li, C., Palanisamy, B., & Xu, R. (2019). Scalable and privacy-preserving design of on/off-chain smart contracts. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)* (pp. 7–12).
- [9] Ortega, V., Bouchmal, F., & Monserrat, J. F. (2018). Trusted 5G vehicular networks: Blockchains and content-centric networking. *IEEE Vehicular Technology Magazine*, 13(2), 121–127.
- [10] Sharma, P. K., Kumar, N., & Park, J. H. (2019). Blockchain-based distributed framework for automotive industry in a smart city. *IEEE Transactions on Industrial Informatics*, 15(7), 4197–4205.
- [11] Singh, M., & Kim, S. (2017). Blockchain based intelligent vehicle data sharing framework. *arXiv preprint arXiv:1708.09721*.
- [12] Wöhrer, M., & Zdun, U. (2018). Design patterns for smart contracts in the Ethereum ecosystem. In *2018 IEEE International Conference on Internet of Things (iThings) and*

- IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 1513–1520).
- [13] Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2020). An overview of blockchain technology: Architecture, consensus, and future trends. *IEEE International Conference on Big Data (Big Data)*, 2020, 1178–1187.
- [14] Wang, S., Ding, D., & Li, J. (2021). A survey on blockchain-based smart contracts: Applications, opportunities, and challenges. *IEEE Transactions on Services Computing*, 14(4), 1–17.
- [15] Liu, Y., & Liu, Y. (2022). Formal verification of smart contracts: A survey. *ACM Computing Surveys*, 54(9), 1–36.
- [16] Li, J., Huang, J., & Shen, J. (2021). A comprehensive survey on smart contract construction and execution: Paradigms, tools, and systems. *IEEE Access*, 9, 13843–13860.
- [17] Zhang, P., White, J., Schmidt, D. C., & Lenz, G. (2020). Applying software patterns to address interoperability in blockchain-based healthcare apps. *Blockchain in Healthcare Today*, 3, 1–10.

3 Automatizált feladatértékelő rendszer online informatikai kurzusokhoz

3.1 Online informatikai kurzusok

A MOOC (Massive Open Online Courses) kurzusok jelentősége az elmúlt években rohamosan növekedett, különösen a pandémiás időszak alatt. Ezek az online oktatási platformok lehetővé teszik, hogy a hallgatók földrajzi helytől és időbeosztástól függetlenül hozzáférjenek magas színvonalú egyetemi vagy egyéb szintű tananyagokhoz, így széles körben támogatva tanulási folyamatot. A COVID-19 okozta lezárások és a személyes oktatás korlátozása során vált igazán nyilvánvalóvá, hogy az online tanulási lehetőségek létjogosultsága nemcsak átmeneti, hanem hosszú távon fenntartható és fenntartandó megoldást is kínál az oktatás számára.

A MOOC kurzusok felépítése általában több komponensre épül, amelyek célja a hallgatók önálló tanulásának támogatása. Ezek a kurzusok videóelőadásokat, szöveges tananyagot, tesztek és feladatokat tartalmaznak, amelyek biztosítják az átadott ismeretek könnyen elérhető és strukturált formában történő feldolgozását. Emellett interaktív fórumok és közösségi funkciók is rendelkezésre állnak, amelyek lehetőséget nyújtanak a hallgatók közötti kommunikációra és együttműködésre, elősegítve a közösségi tanulást és a tudásmegosztást.

A programozás oktatása azonban speciális megközelítést igényel a MOOC keretében. Míg más tantárgyaknál elég lehet az elméleti tudás átadása, addig a programozás elsajátítása gyakorlati készségeket, algoritmikus gondolkodást és problémamegoldó képességet is megkövetel. Ennek érdekében a programozási MOOC-ok gyakorlati kódolási feladatokat és azonnali visszajelzést biztosító rendszereket kínálnak, hogy a hallgatók aktívan alkalmazhassák az elsajátított ismereteket. Fontos, hogy ezek a rendszerek képesek legyenek a programkód automatikus kiértékelésére, ezzel támogatva a hallgatók folyamatos fejlődését és a hibák gyors felismerését, amely elengedhetetlen a hatékony programozási készség kialakulásához.

3.2 Az automatikus feladatkiértékelés kihívásai

A programozási MOOC-ok kiértékelési rendszereinek fejlesztése során számos funkcionális és nemfunkcionális szempontot kell figyelembe venni, amelyek biztosítják a hatékony és biztonságos működést. Ezek a rendszerek nemcsak a diákok kódjainak tesztelését végzik, hanem egyúttal stabil és megbízható környezetet - integrált LMS-t (Learning Management System) is biztosítanak, amelyben a felhasználók akár nagy létszámban is problémamentesen dolgozhatnak. Az LMS olyan digitális platform, amely lehetővé teszi az oktatási anyagok szervezését, a tanulók tevékenységének nyomon követését, az értékelési folyamatok lebonyolítását, valamint az oktatók és diákok közötti kommunikációt. Az alábbiakban összefoglalt alapelvek szükségesek ahhoz, hogy a MOOC platformok hosszú távon is fenntartható megoldásként szolgáljanak a programozás oktatásában:

1. A kiértékelő rendszernek képesnek kell lennie a kód tényleges futtatására; egyszerű szöveges összehasonlítás nem elég a kód teljes ellenőrzéséhez.

2. Fontos, hogy a kiértékelő minden diák kódját elkülönítve futtassa, hogy azok ne zavarják vagy befolyásolják egymást.
3. A rendszernek képesnek kell lennie kezelni a kártékony vagy hibás kódot anélkül, hogy az a MOOC platform működésére negatív hatással lenne. Például, ha egy diák végtelen ciklust tartalmazó kódot tölt fel, a platformnak továbbra is stabilan kell működnie. Hasonlóképpen, ha valaki nagy mennyiségű memóriát foglalna le vagy a merevlemezt próbálná törölni, azt a rendszernek meg kell akadályoznia.
4. A rendszer legyen képes nem teljesen kidolgozott kódot (pl. adatstruktúrákat) kezelni.
5. A hatékonyság kiemelten fontos, a rendszert arra kell felkészíteni, hogy a feladatbeadás határidejekor extrém nagy lesz a terhelés. Ezért a rendszernek sorba kell tudnia rendezni a beérkező kéréseket, és gyors válaszokat kell biztosítani, úgy, hogy egyidejűleg akár több ezer diák beküldött kódját is kezelni tudja.
6. A kiértékelőnek támogatnia kell a virtuális végrehajtó környezetet, különösen, ha az oktatás a gazdarendszertől eltérő operációs rendszerre van kidolgozva, például Windowsra összpontosít, de a MOOC rendszer UNIX alapon fut.
7. A biztonság kulcsszerepet játszik: mivel a diákok kreditpontokat szerezhetnek a kurzus teljesítésével, nem szabad megengedni, hogy módosítsák a beküldött kódjukra kapott értékelést.
8. A kiértékelő rendszernek világos visszajelzést kell adnia arról, hogy a kód elfogadható-e, és ha szükséges, iránymutatást kell nyújtania a hibák javításához.
9. Az anonimitást meg kell őrizni a kiértékelés során, a rendszernek nem szabad tudnia, hogy a kód személy szerint kihez tartozik.

3.3 Programozási feladat kiértékelés megvalósítása a MeMOOC rendszerben

A MeMOOC projekt a Miskolci Egyetem online oktatási kezdeményezése, amely 2017-ben indult. A projekt célja az volt, hogy nyílt és zárt kurzusok keretében az informatika alapjait, programozást és alkalmazásfejlesztést tanítson, valamint felkészítse a hallgatókat a Mérnök-informatikus Felsőoktatási Szakképzés (FOSZ) vizsgáira és kreditjeinek megszerzésére. A pandémiás időszakban a MeMOOC szerepe felértékelődött. A MeMOOC rendszer az open edX platformon alapul, amely egy nyílt forráskódú, nagy mértékben testreszabható online tanulási környezet.

Ehhez a környezethez fejlesztettünk kiértékelő rendszer, amely technikailag edX platformhoz kapcsolódó külső modul és önálló szolgáltatásként működik, így akár az edX-től függetlenül is alkalmazható. Kidolgoztunk egy üzenetsor alapú aszinkron interfészt (XQueue) Az LMS ezen keresztül kommunikál ezzel a kiértékelővel: elküldi a diák megoldását, majd aszinkron módon várja a kiértékelés eredményét, amelyet aztán visszajuttat az LMS-be. A beküldött megoldások egy üzenetsorban tárolódnak addig, amíg a kiértékelő nem kérdezi le őket feldolgozás céljából. Így működik az aszinkron kiértékelő rendszer, amely az üzeneteket sorból lekérdezi, és a válaszokat oda tárolja (1. ábra).



1. ábra A külső kiértékelő modul elvi működése

Biztonságos és költséghatékony megoldást dolgoztunk ki a hallgatói kódok tárolására. Alapesetben az edX a nagyobb fájlokat Amazon S3 tárhelyre tölti fel, és az URL-eket a lokális adatbázisban tárolja. Több ezer beküldött fájl esetén ez a megközelítés nagyon hatékony. A hátránya az, hogy az Amazon S3 kereskedelmi szolgáltatás, ami növelheti a fenntartási költségeket, és külső függést okoz a szolgáltatásban. Ezért a MeMOOC-ban egy lokális MySQL adatbázist használtunk a beküldött fájlok tárolására. Kidolgoztunk egy második üzenetsort is (RabbitMQ), ami a következőképpen működik:

1. Amikor a hallgatók egy új feladatot nyújtanak be a MeMOOC rendszerben, az XQueue sor automatikusan értesítést kap a beküldésről. Ez az értesítés indítja el a feladatkiértékelési folyamatot, és jelzi, hogy egy új megoldás érkezett be a rendszerbe, amelyet a kiértékelő modulnak fel kell dolgoznia.
2. A RabbitMQ, mint üzenetközvetítő rendszer, az adatbázisból kiolvassa a beérkezett feladatokhoz kapcsolódó adatokat és továbbítja azokat a kiértékelő rendszer felé.
3. Szerializációval gondoskodunk arról, hogy az objektumokat és a leíró adatstruktúrát olyan formátumba alakítsuk át, amely könnyen tárolható vagy továbbítható.
4. A MeMOOC rendszer küld egy üzenetet a kiértékelő rendszernek és várakozik a válaszra.
5. A külső kiértékelő inicializálja a virtuális fordító környezetet.
6. A külső kiértékelő lefordítja a beküldött programkódokat.
7. A külső kiértékelő rendszer a beküldött kód ellenőrzése érdekében előre kidolgozott egységtesztet futtat, amelyek különböző szempontok alapján vizsgálják a kód működését és helyességét. Ezek az egységtesztet átfogó elemzést nyújtanak, például ellenőrzik a kód szintaktikai helyességét, a logikai működését, és azt, hogy a várt eredményeket adja-e vissza.

A kiértékelés után a kód futtatásának eredményét egy fájlban tároljuk, amelynek neve *véletlenszerűen generált karakterekből áll*. Ez a véletlenszerű elnevezés segít elkerülni a fájlütközéseket és növeli a biztonságot, mivel a fájlok egyedileg azonosíthatók és nem kapcsolhatók közvetlenül a beküldőhöz.

A megvalósított kiértékelő rendszer képes különféle szintű kódokat – teljes programokat, metódusokat, struktúrákat, és objektum-orientált nyelvek esetében osztálydeklarációkat tesztelni. A rendszer különböző nyelveken különböző– jellemzően az egységtesztelésen alapuló - módszerekkel működik; például Java esetében a JUnit assertEquals metódusát használva vizsgálja a kód helyességét. A visszaadott válasz információt ad a kód megfelelőségéről, pontszámot tartalmaz, valamint egyedi szöveges visszajelzést ad a megoldáshoz. Részleges

kódok esetén a teszt kód és a beküldött kód egyesítésével, majd futtatásával történik a kiértékelés, ahol a teszt eredményének kimenete szolgál az értékelés alapjául.

A kódolási feladatok kiosztásánál háromféle működést dolgoztunk ki:

- ugyanazt a kódolási feladatot adunk minden felhasználónak,
- ugyanazt a kódolási feladatot adunk véletlenszerűen generált értékelési paraméterekkel,
- véletlenszerűen adunk egy kódolási feladatot.

A MeMOOC rendszerben alkalmazott, új, saját fejlesztésű kiértékelő program a kiértékelő rendszerek a négy aspektusa szerint képes volt a beküldött kódok ellenőrzésére, ezt tézisértékű eredménynek tartom.

A kiértékelő rendszer négy aspektusa a következő:

Szintaktika aspektus: Ellenőrzi, hogy a kód szintaktikailag helyes-e, vagyis lefordítható-e (compiler segítségével) vagy értelmezhető-e (interpreter segítségével) az adott programozási nyelv szabályai szerint. Emellett a kódolási konvenciók betartását is vizsgálja. A modern szoftverfejlesztésben kialakult az a gyakorlat, amely a kód formázására elnevezésére megkötésekkel ír elő. A kidolgozott kiértékelő alkalmas arra, hogy ezeket a szabályokat is ellenőrizze.

Strukturális aspektus: A kód szerkezeti elemeit vizsgálja, például osztályok, mezők és metódusok deklarációját. Ellenőrzi, hogy a kód tartalmazza-e a szükséges struktúrákat és elemeket, mint például egy adott osztályban a szükséges mezők és metódusok meglétét. Egy példa a strukturális aspektusra, amikor a hallgatóknak egy `Vehicle` osztályt kell létrehozniuk, amely különböző járművek alapvető adatait tárolja. A feladat szerint ennek az osztálynak rendelkeznie kell egy `string` típusú `model` mezővel, amely a jármű modelljét tárolja, és egy `double` típusú `fuelLevel` mezővel, amely a jármű üzemanyagszintjét reprezentálja. Az osztálynak tartalmaznia kell egy `Refuel` nevű metódust is, amely a jármű üzemanyagszintjét növeli adott értékkel. A kiértékelő rendszernek a következő struktúrákat kell ellenőriznie:

- Létre lett-e hozva a `Vehicle` osztály?
- Tartalmazza-e a `model` és `fuelLevel` mezőket a megfelelő típusokkal (`string` és `double`)?
- Megtalálható-e a `Refuel` metódus, amely megfelelően növeli a `fuelLevel` adattagot?

A kiértékelés egységtesztelő keretrendszereken alapul, ilyen például Java esetén a `JUnit`, dinamikusan próbálja meg betölteni a `Vehicle` osztályt. Ha nem találja az osztályt, egy `ClassNotFoundException` kivételt dob, amely hibajelzést küld a beküldő hallgatónak. A kiértékelés során továbbá megvizsgálja, hogy a `model` és `fuelLevel` mezők megfelelően létre lettek-e hozva. Ha például a `fuelLevel` mező hiányzik, a teszt hibával zárul. Végül a

kiértékelő ellenőrzi a `Refuel` metódus működését. Ha a metódus nem létezik vagy nem megfelelően frissíti a `fuelLevel` értékét, a kiértékelő ezt is hibaként jelzi. Ez a strukturális ellenőrzés segít biztosítani, hogy a hallgatók megfelelően hozzák létre az osztály szerkezetét és elemeit, így megfeleljenek a feladat követelményeinek.

Működési logika aspektus: A kód logikai működését teszteli annak biztosítására, hogy az a várt eredményeket produkálja. Ezt például Java nyelv esetén szintén `JUnit`-tesztek futtatásával ellenőrizzük: megvizsgáljuk, hogy a kód a meghatározott specifikáció szerint működik-e. A működési logika unit tesztekkel történő ellenőrzéséhez először meghatározzuk a tesztelési feltételeket, hogy az adott funkció minden lehetséges bemeneti és kimeneti állapotban a várt eredményt adja vissza. A tesztelési folyamat során pozitív tesztekkel ellenőrizzük a helyes működést, vagyis azt, hogy a kód megfelelő bemenetek esetén a helyes kimenetet adja. Ezzel párhuzamosan negatív tesztek is végrehajtunk, amelyek biztosítják, hogy a kód hibás bemenetek esetén megfelelően kezelje a hibákat. A határesetek tesztelése is lényeges része a folyamatnak: ezekkel a szélsőséges bemenetekre adott válaszokat vizsgáljuk, hogy biztosítsuk a program stabil működését az összes lehetséges esetben. Összetett logikai folyamatok esetén többféle bemeneti kombinációt tesztelünk annak érdekében, hogy a kód minden döntési ága helyesen működjön.

Grafikus felhasználói interfész (GUI) aspektus: A grafikus interfésszel rendelkező alkalmazások esetén vizsgálja, hogy a GUI elemei megfelelően vannak-e implementálva, például eseménykezelők, gombok és interaktív elemek. Ezt jellemzően mock keretrendszerek (pl.: `Mockito` vagy egyéb, hasonló eszközök) segítségével ellenőrizhetjük. A mockolás olyan tesztelési technika, amelyben az alkalmazás valódi komponensei helyett azok mesterséges, szimulált példányait (mock objektumokat) hozzuk létre. Ezek az objektumok előre meghatározott viselkedést adnak vissza, így lehetővé teszik, hogy a vizsgált komponens működését izoláltan, a külső függőségektől függetlenül ellenőrizzük. Ennek köszönhetően pontosabban mérhető, hogy a GUI elemek megfelelően hívják-e a háttérfunkciókat, a várt események bekövetkeznek-e, és az interakciók helyesen kezelődnek-e.

Egy példa erre, amikor a hallgatói programnak egy bejelentkező felületet kell létrehoznia, amely tartalmaz egy felhasználónév mezőt, egy jelszó mezőt és egy „Bejelentkezés” gombot, valamint implementálnia kell a gomb eseménykezelőjét, amely ellenőrzi a bevitt adatok helyességét. A tesztelés során a `Mockito` framework segítségével a következőket vizsgálhatjuk meg: először is, hogy a `Login` osztály létrehozta-e a felhasználónév és jelszó mezőket, valamint a gombot. Továbbá meg kell győződnünk arról, hogy a „Bejelentkezés” gombhoz tartozó eseménykezelő megfelelően implementálva van-e, azaz a gomb lenyomásakor az eseménykezelő metódus valóban lefut. A tesztet úgy állítjuk be, hogy ellenőrizzük, hogy a három felületkomponens példányosítva lett-e, és hogy a gombhoz tartozó eseménykezelő meghívásra kerül-e. Ehhez a `Mockito` keretrendszer segítségével létrehozunk egy „mock” példányt a „Bejelentkezés” gombra, és figyeljük, hogy a gombhoz tartozó eseménykezelő

valóban aktiválódik-e kattintáskor. Amennyiben valamelyik komponens vagy az eseménykezelő nem létezik, a teszt hibát jelez. Ebben az esetben a teszt program hibáüzenetet küld a hallgatónak, amely jelzi, hogy például a felhasználónév mező vagy a gombhoz tartozó eseménykezelő hiányzik a programból. Ez a módszer lehetővé teszi, hogy a GUI-komponensek és eseménykezelők meglétét és helyes működését automatikusan észleljük és visszajelezzük a hallgatóknak.

3.4 Tézisek

T3.1 A hallgatói programozási feladatok automatikus kiértékelését egy négy aspektust vizsgáló módszer hatékonyan lehetővé teszi.

1. A szintaktikai aspektus biztosítja, hogy a hallgatók által beküldött kód megfeleljen a programozási nyelv szintaktikai szabályainak, ami elengedhetetlen ahhoz, hogy a kód lefordítható vagy értelmezhető legyen.
2. A strukturális aspektus lehetővé teszi, hogy a kód felépítése megfeleljen a kívánt szerkezeti előírásoknak, ellenőrizve a szükséges osztályok, mezők és metódusok deklarációját.
3. A működési logika aspektus a kód logikai működését értékeli, biztosítva, hogy az adott kód megfelelően teljesítse a kijelölt feladatot és a várt eredményeket produkálja.
4. A grafikus felhasználói interfész (GUI) aspektus lehetővé teszi a kód vizsgálatát olyan szempontból, hogy a grafikus elemek helyesen és hatékonyan biztosítsák a felhasználói interakciókat a felületen.

T3.2 A kétszintű eseménysor hatékonyan támogatja a hallgatói feladatok kiértékelését azáltal, hogy elkülöníti a feladatok fogadását és azok tényleges feldolgozását, ezzel növelve a rendszer rugalmasságát és teljesítményét. Az első szintű eseménysor biztosítja, hogy minden beküldött feladat gyorsan értesítést kapjon, és sorba rendeződjön a kiértékeléshez. A második szintű eseménysor pedig lehetővé teszi a tényleges feldolgozást, például a kód futtatását, elemzését és a visszajelzés elkészítését. Ez a megközelítés biztosítja a párhuzamos feldolgozást, csökkenti a várakozási időt, és képes kezelni a nagyszámú, egyidejű feladatbeküldést, így hatékonyan támogatva a hallgatói munkák gyors és megbízható kiértékelését.

T3.3 Kialakítható olyan lokális virtuális futtatókörnyezet, amely lehetővé teszi a hallgatói feladatok biztonságos és elszeparált kiértékelését, így minimalizálva a rendszer működésére és adatbiztonságára gyakorolt kockázatokat. Ez a megoldás biztosítja, hogy minden beküldött kód önálló, izolált környezetben fusson, megakadályozva, hogy hibás vagy kártékony kód befolyásolja a többi feladat értékelését vagy a platform működését. A lokális virtuális környezet így lehetőséget nyújt a hatékony, biztonságos és skálázható kiértékelésre, támogatva a nagyszámú párhuzamos beküldés kezelését.

3.5 A témához kapcsolódó publikációk

[P3.1] Király Sándor, Nehéz Károly, Hornyák Olivér: Some aspects of grading Java code submissions in MOOCs, RESEARCH IN LEARNING TECHNOLOGY 25 Paper: 1945, 16 p. (2017), SJR: Q1

[P3.2] Hornyák Olivér: Programozási feladatok kiértékelése a távoktatásban, ENELKO 2022, XXIII. Energetika-Elektrotechnika – ENELKO és XXXII. Számítástechnika és Oktatás – SzámOkt Multi-konferencia (2022) 210 p. pp. 113-118., 6 p.

[P3.3] Hornyák Olivér: Open edX kurzusok speciális követelményei és lehetőségei, In: Ollé, János (szerk.) I. Nyílt Oktatás Konferencia Absztraktkötet, Eger, Magyarország : EKF Líceum Kiadó, (2015) pp. 15-16. , 2 p.

[P3.4] Hornyák Olivér: Online Informatikai Egyetem, In: Ollé, János (szerk.) I. Nyílt Oktatás Konferencia Absztraktkötet, Eger, Magyarország : EKF Líceum Kiadó, (2015) pp. 15-16.,

3.6 Felhasznált szakirodalom

[1] Arefin, A. S. (2015). *Pedagogy of computer programming: An interactive and collaborative learning approach*. Macquarie University Postgraduate Certificate of Higher Education EDCN 871 Final Project.

[2] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *SIGCSE '02 Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*, 38–42.

[3] Muratet, M., Torguet, P., Viallet, F., & Jessel, J. P. (2010). Experimental feedback on Prog&Play: A serious game for programming practice. In L. Kjeldahl & G. Baronosk (Eds.), *EUROGRAPHICS* (pp. 1–8).

[4] Staubitz, T., Pfeiffer, T., Renz, J., Willems, C., & Meinel, C. (2015). Collaborative learning. In *ICERI2015 8th Annual International Conference of Education, Research and Innovation*, Seville, Spain (pp. 18–20).

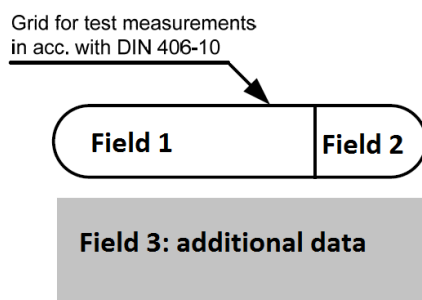
[5] Teague, D., & Roe, P. (2008). Collaborative learning: Towards a solution for novice programmers. *Proceedings of the Tenth Conference on Australasian Computing Education - Volume 78*.

[6] Vihavainen, A., Luukkainen, M., & Kurhila, J. (2012). Multi-faceted support for MOOC in programming. *Proceedings of the 13th Annual Conference on Information Technology Education*, ACM, New York, 171–176.

[7] Kuser Gábor, Havasi Gábor, Király Sándor, Kocsis-Baán Mária, Nehéz Károly, Hornyák Olivér, Mileff Péter, Introducing MeMOOC and Recent Results in e-Learning at University of Miskolc, In: Darco, Jansen; Lizzie, Konings (szerk.) MOOCs in Europe:

Overview of papers representing a collective European response on MOOCs as presented during the HOME conference in Rome November 2015 : Papers ‘WOW! Europe embraces MOOCs’, Heerlen, Hollandia: European Association of Distance Teaching Universities (EADTU) (2016) 221 p. pp. 75-78. , 4 p.

zsúfolt és bonyolult jellegéből adódóan jelentős figyelmet és precizitást is igényelt a mérnököktől.



3. ábra A mérési rács (Zeppelin) általános felépítése

A Zeppelin adattartalmának kinyeréséhez vizsgáltam a DIN 406-xx szabványcsoportot, amely három mezőt definiál. Az első mező egy jellemzőt határoz meg, például a méreteket és/vagy tűrőereket, egyéb követelményeket például a felületi adatokat vagy a hőkezelési adatokat, valamint a jellemző osztályozását. A második mező azt indexszámot határozza meg, amely például az eljárások megkülönböztetésére szolgál. Egy rajzon azonos indexet kapnak az azonos eljárások vagy műveletek, még akkor is, ha az első mező tartalma eltérő. A DIN 406-11 szabvány 5.6 szakasza szerinti tesztelés mértéke is megadható ebben a mezőben. A harmadik mező opcionális, és rövid megjegyzéseket tartalmazhat, például az alkatrészlista szerinti tételszámot, a végrehajtandó műveletet vagy eljárást, az indoklást, stb.

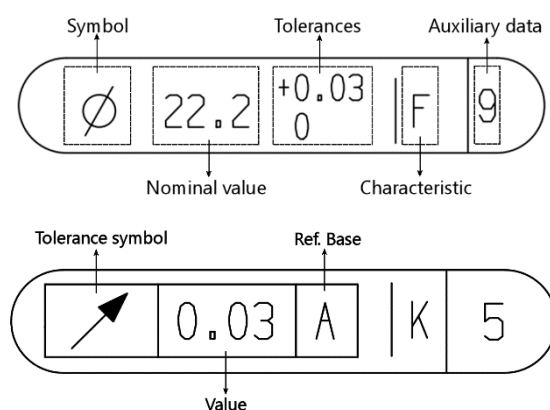
4.2 Az irodalomban ismert módszerek áttekintése

Az optikai karakterfelismerő (OCR) rendszerek széles körű alkalmazásuknak köszönhetően az informatika és a mérnöki tudományok fontos kutatási területévé váltak. Számos tudományos publikáció és szabadalom foglalkozik az OCR eljárások különböző aspektusaival, például szövegfelismeréssel, rajzi elemek elemzésével, valamint adatfeldolgozással. Azonban a témában végzett alapos irodalomkutatás és szabadalom-elemzés során azt tapasztaltuk, hogy a meglévő publikációk és szabadalmak nem fedték le azt a speciális alkalmazási területet, amely az általunk vizsgált rendszer fejlesztésére irányult. Különösen hiányzott olyan tudományos munka, amely az optikai karakterfelismerés és rajzi elemek azonosításának kombinált problémájával foglalkozik egyetlen, integrált alkalmazás keretében.

A releváns szabadalmak elemzése során több olyan eljárást azonosítottunk, amelyek részben megoldják az optikai karakterfelismerés bizonyos aspektusait. Ezek a szabadalmak nem adtak megoldást a kitűzött feladatra: vagy nem voltak képesek kezelni a rajzi elemek komplex orientációját és alakját, vagy nem voltak alkalmasak az összetett karakter-szimbólum kapcsolatok elemzésére. (Lásd még Az eredetiség kutatás eredménye c. fejezetet)

4.3 A kidolgozott, saját fejlesztésű, szabadalmaztatott eljárás

Az előzőekben feltárt és ismert eljárások hiányosságainak kiküszöbölésével kidolgoztunk egy szabadalmaztatott eljárást [P4.1], amely alkalmas kétdimenziós képi formában megjeleníthető rajzokon, térképeken, vagy kétdimenziós képet hordozó digitális adathordozón található szelektált rajzi alakzatok információtartalmának teljes kinyerésére megjelenítésére és azok adattárolására. Az eljárás a szelektíven válogatott és tárolt adatokat szövegesen vagy akár hanganyag formában is közli a felhasználóval, és így teljes információkinyerést hajt végre, még akkor is, ha az információt hordozó alakzat zárt vagy nyitott és tetszőleges szöghelyzetű tetszőleges rajzi elem, tartalma pedig tetszőleges, de két dimenzióban ábrázolható jelölés.



4. ábra A Zeppelin-ek általános felépítése

A működés alapja egy elektronikusan, tetszőleges adathordozón előálló műszaki, vagy más kétdimenziós rajz. Ez a rajz lehet szabadkézi vagy kinyomtatott műszaki rajz utólagosan digitalizált dokumentum vagy CAD rendszerekből kiemelt elektronikus dokumentum. A felismerési folyamat előtt alakzatfelismerés, majd alakzat szelekció történik. Kiválasztásra kerül a sokféle speciális alakzatú jelölés közül az, amelyik megjelenik a feldolgozandó dokumentumban vagy dokumentumokban (például a Zeppelin alakzat). Az eljárás következő lépéseként a dokumentum alakzatait vektorizált formára átalakítjuk. A vektorizált adatformátum egyeneseket és görbéket tartalmaz, amelyekből a következő lépésben virtuálisan matematikai gráfot építünk. Az átalakítás során figyelembe vesszük, hogy a vektor elemek végpontjai a digitalizálás/exportálás miatt pontatlanul illeszkedhetnek és az illesztési hibákat automatikusan kijavítjuk. Az előfeldolgozás célja a lehető legkevesebb csúcsponttal rendelkező, összefüggő gráf megkeresése. A keresés matematikai modelljét a következő alfejezetek írják le.

4.3.1 Címkezett izomorf algráf-keresés visszalépéses heurisztikával

A kidolgozott algoritmus a polimorfikus-algráfok keresésére épül, amely az alábbi módon formalizálható. Legyenek G és G' gráfok az alábbiak szerint definiálva:

$$G = (V, E), G' = (V', E') \quad (1)$$

ahol V és V' a csúcsok halmaza, E és E' pedig az élek halmaza. A két gráf csúcsainak halmaza között bijekció áll fenn:

$$f: V \rightarrow V', \text{ ha } \forall v_i \in V, f(v_i) \in V' \quad (2)$$

A G és G' gráfok izomorfok, ha:

$$\forall (v_i, v_j) \in E, \exists (f(v_i), f(v_j)) \in E' \quad (3)$$

Az izomorf gráfokat $G \cong G'$ ként jelöljük. Az (1) definícióban a G és G' gráfok irányítatlan, nem címkézett, nem súlyozott gráfokként voltak definiálva. Az algoritmusunkhoz azonban szükséges a görbesereg szerkezetének vizsgálata. Legyen a címkésokaság az alábbi módon definiálva:

$$\mathcal{L} (\{C, L\}) \quad (4)$$

ahol C a görbék, L pedig a szakaszokat jelöli, és a címkézési függvény:

$$\lambda: E \rightarrow \mathcal{L} \quad (5)$$

Legyen G és G' címkézett gráf:

$$G = (V, E, \lambda), G' = (V', E', \lambda') \quad (6)$$

G és G' címkézett gráfok izomorfok, ha a következő feltételek teljesülnek:

1. $G \cong G'$
2. $\lambda(e) = \lambda'(e'), \forall e \in E, e' \in E'$

A címkék halmaza görbék és szakaszokat tartalmaz. Az élek címkéi a következőképpen definiálhatók:

$$\lambda(e) = \begin{cases} C(t), & \text{ha az él (Bézier)görbe} \\ L, & \text{ha az él szakasz.} \end{cases} \quad (7)$$

ahol $C(t)$ egy Bézier-görbe négy kontrollponttal (P_0, P_1, P_2, P_3):

$$C(t) = (1-t)^3 \cdot P_0 + (1-t)^2 \cdot t \cdot P_1 + (1-t) \cdot t^2 \cdot P_2 + t^3 \cdot P_3, t \in [0,1] \quad (8)$$

Az algráf-keresési problémák NP-nehezek. A megoldási komplexitás csökkentése érdekében négy heurisztikát alkalmaztunk:

1. Csak a címkézett élek mintázatait keressük.
2. Csak zárt algráfokat keresünk.
3. A legnagyobb keresési mintázat hossza korlátozza a keresési fa mélységét, lehetővé téve a visszalépéses heurisztika alkalmazását.

4. Utófeldolgozást végzünk: kizárjuk a konkáv alakzatokat, a túl kicsi alakzatokat, és azokat, amelyek nem tartalmaznak további grafikus primitíveket.

A szabadalmaztatott eljárásban automatikus vagy kézi szűrési eljárással választjuk ki azokat az alakzatokat, amelyekből adatkinyerést végzünk. Az eljárás a dokumentum tetszőleges részében fellelhető, tetszőleges orientációjú és méretű, de azonos alakú alakzat megjelöléseket is kezel. Ezt követően a szelektált alakzataból karakterkinyeréssel rajzi elemeket nyerünk ki.

A karakter kinyerés (valójában szimbólumkinyerés) során olyan új, alakzat felismerő algoritmust használunk, amely tetszőleges méretű, elrendezésű és orientációjú, vektorizált formában eltárolt alakzatokat ismer fel. A felismerendő alakzatokat alakzatkezeléssel a már ismert alakzatokkal történő összevetés útján azonosítjuk. Az eljárás része az alakzattár elkészítésének és automatikus bővítésének folyamata. Az eljárás végül az alakzat tartalmát felismeri és a folyamat által kinyert információt feldolgozza, rendszerezi, és adatbázisba menti. Ennek során új optikai karakterfelismerő eljárás létrehozásával, olyan eljárási részfolyamatokból összekapcsolt olyan új eljárást hoztunk létre, amely elindulása után megtörténik a dokumentum bevitele, majd a dokumentumban az eljárás eredményeképpen automatikusan kijelölésre kerül az összes olvasandó alakzat. Ezt követően az alakzatokból karakterkinyerést végzünk, majd megjelenítjük és szükség szerint továbbítjuk a rajzi alakzatok tartalmát és megállítjuk az eljárást. Az eljárás működéséhez szükség van tudásbázis (alakzattár) bővítés végrehajtására, ami lehetővé teszi az alakzatok információ tartalmának teljes kinyerését és megjelenítését.

A szabadalmaztatott eljáráson alapuló szoftvert implementáltuk. Bemenete egy PDF formátumban elmentett műhelyrajz. A PDF formátumban tárolt műhelyrajzok egyik legnagyobb előnye a platformfüggetlenség, amely lehetővé teszi, hogy a fájlok bármilyen eszközön vagy operációs rendszeren megnyithatók legyenek, megőrizve azok eredeti formátumát. Ez különösen fontos a mérnöki és gyártási környezetekben, ahol a rajzok pontos részleteinek, méretarányainak és formázásának megőrzése elengedhetetlen. A PDF formátum további előnye, hogy használatához nincs szükség külön licencre, mivel ez egy nyílt szabvány. Ez azt jelenti, hogy a fájlok megnyitása, kezelése és feldolgozása ingyenes eszközökkel is megvalósítható, ami költséghatékony megoldást jelent a vállalatok számára. Emellett a PDF formátum stabilitása és széles körű támogatottsága biztosítja, hogy a fájlok hosszú távon is elérhetőek és kompatibilisek maradjanak a különböző szoftverekkel és rendszerekkel. A rajzokon az elemek száma elérheti a több milliót is, azonban a szoftver optimalizált működésének köszönhetően a 40-50 PDF fájl teljes feldolgozása csupán perceket vesz igénybe, így jelentősen megkönnyítve a mérnöki munkát.

A méréseink alapján a Zeppelinek felismerési aránya 98% körüli, a feliratok 80%-os kezdeti pontossággal felismerhetőek. A feliratok felismerési pontosságát alakzat betanítással tovább lehetett növelni.

4.4 Tézisek

T4.1 Kidolgoztam egy *eljárást szelektált rajzi alakzatok teljes tartalmának kinyerésére és megjelenítésére*, amely a műszaki rajzokon fellelhető rajzi alakzatokat szelektálja, majd azok teljes tartalmát felismeri és automatikusan olyan adatbázisba rendezi, amely közvetlenül a felhasználó személyek számára a rajz olvasása nélkül értelmezhető és/vagy amelyet számítógépes adatfeldolgozó szoftverek segítségével további műveletek elvégzésére lehet felhasználni. Az eljárás alkalmas elektronikus dokumentumokban szereplő tetszőleges helyzetű, géprajzi szabványokban vagy szabványon kívüli, de állandó formájú kétdimenziós rögzített rajzi alakzat és/vagy alakzatba foglalt rajzi elemek automatikus és teljes felismerésére, kinyerésére és megjelenítésére. A rajzi elemek lehetnek szimbólumok, karakterek, ismertetőjelek, kiegészítő jelzetek vagy ezektől eltérő, de információ tartalommal bíró jelölések. Az eljárás magában foglalja a jelölések felismerésének, az adatok kinyerésének és megjelenítésének folyamatát is. Ezt az eljárást elsősorban azok az ipari résztvevők használhatják a termelésük hatékonyságának emelésére, akik a tervezés és/vagy a gyártás során nagymennyiségű rajzi dokumentációt kezelnek, de felhasználhatják a térképészetben dolgozó szakemberek is egyedi szimbólumok felismerésére és szükség esetén azok tartalmának automatikus kinyerésére.

4.5 A témához kapcsolódó publikációk

[P4.1] Szabó M., Nehéz K., Mileff P., és Hornyák O., „Eljárás szelektált rajzi alakzatok teljes tartalmának kinyerésére és megjelenítésére” Benyújtás száma: P1800403, Magyar szabadalom, 2018.

4.6 Felhasznált irodalom

- [1] Almeida, R. A. S. G. de, Pinto, A. R., & Figueira, R. R. (2017). *Intelligent automatic license plate recognition for electronic tolling environment* (United States Patent Application No. US20170372161).
- [2] Dewey, M., Sims Jr., R., & Bains, B. (2013). *Systems, methods, and tools for proofing a computer-aided design object* (United States Patent No. US 8,370,117 B2).
- [3] Hideki, K. (1994). *Symbol recognizing system* (Japanese Patent No. JP3240303B2). Nippon Steel Corporation.
- [4] Kolton, A., & Bentov, A. (2017). *Location-based optical character recognition (OCR)* (United States Patent Application No. 20170337443).
- [5] Moriyama, M. (2000). *Method and device for extracting character information from CAD picture* (Japanese Patent Application No. JP2000176095A).
- [6] Suzuki, K., et al. (1992). *Drawing input device* (Japanese Patent No. JP19900140041). Tosgibe Corporation.

4.7 Az eredetiség kutatás eredménye

Az eredetiségkutatás szerepe egy szabadalmi eljárásban kulcsfontosságú, mivel az határozza meg, hogy az adott találmány valóban új, feltalálói tevékenységen alapuló és iparilag alkalmazható-e – ez a három fő kritérium a szabadalmazhatósághoz.

1. Újdonság vizsgálata

Az eredetiségkutatás során azt vizsgáltuk, hogy létezik-e már a találmánnyal azonos vagy nagyon hasonló megoldás (ún. „korábbi technika”). Ha létezik, akkor a találmány nem tekinthető újnak, és nem lehet rá szabadalmat kapni.

2. Feltalálói tevékenység meglétének ellenőrzése

Azt is vizsgáltuk, hogy a találmány egy szakember számára nyilvánvaló megoldásnak tekinthető-e a korábbi technika alapján. Ha igen, akkor nincs meg a feltalálói lépés, így a szabadalmi oltalom elutasítható.

3. A szabadalmi bejelentés megalapozása

Egy alapos eredetiségkutatás segít a feltalálónak vagy a bejelentőnek:

- pontosabban megfogalmazni az igénypontokat,
- erősebb szabadalmi védelem kialakításában,
- elkerülni a fölösleges idő- és pénzráfordítást egy nem szabadalmazható találmány bejelentésére.

4. A versenytársak figyelemmel kísérése

Az eredetiségkutatás során feltérképezhetők a hasonló területen tevékenykedő vállalatok, kutatók és azok szabadalmi – ez stratégiai előnyt adhat.

A következő táblázat a megvizsgált szabadalmakat mutatja meg.

Szabadalom száma	Kulcsjellezők	Korlátok	Megjegyzések
US0337443	Mobiltelefonokon használható optikai karakterfelismerés számlakezeléshez; a digitalizált képet központi szerver elemzi és csoportosított adatokat készít. Nem alkalmas zárt/nyitott alakzatokból történő felismerésre.	Nem képes alakzatszelektálásra és zárt/nyitott formák kiolvasására.	Adatokat küld vissza a beküldőnek, összehasonlító kliséket használ.
US8370117	Digitális rajzokon található rajzi jelek felismerése, méret- és tűrészadatok ellenőrzése. Csak	Nem tudja kezelni a tetszőleges irányú vagy formájú rajzi elemeket.	Vizsgálati eredményeket vizuálisan jelenít meg és csatolja a rajzhoz.

Szabadalom száma	Kulcsjellezők	Korlátok	Megjegyzések
	téglalap alakú vagy merőleges alakzatokat képes feldolgozni.		
JP2001318952	Számítógépen futtatható eljárás szövegmezők felismerésére, kizárólag keretezett szám- és betűkaraktereket ismer fel. Központi processzorral és megjelenítő egységgel rendelkezik.	Nem képes keretezetlen karakterek vagy tetszőleges formák felismerésére.	Kizárólag keretes szöveg felismerésére alkalmas; adatfeldolgozó egységgel működik.
US0372161	Jármű rendszámok felismerésére szolgáló eljárás; keretezett karakterekből dolgozik. Nem képes rajzi szimbólumok vagy alakzatok azonosítására.	Nem ismeri fel rajzi szimbólumokat vagy bonyolultabb alakzatokat.	Rendszámkaraktereket nagy pontossággal ismer fel, hibaüzenetet küld az olvashatatlan karakterekről.

1. táblázat A releváns szabadalmak

5 Termelésütemezési feladatok megoldása

5.1 Diszkrét gyártási folyamatok

A diszkrét gyártási folyamatok alatt olyan gyártási folyamatokat értünk, ahol az előállított termékek jól elkülöníthető, egyedi egységek, és a gyártás lépései is szakaszosak, időben és térben elhatárolhatók a tevékenységek. Ebben a környezetben a termelés egyértelműen elhatárolható lépésekből áll, például összeszerelésből, csomagolásból vagy tesztelésből, ahol minden egyes egység – legyen az alkatrész, modul vagy késztermék – külön kezelhető. Az ilyen típusú termelés jellemzően autóiipari, elektronikai, gépgyártási és más hasonló iparágakban gyakori, ahol a gyártás folyamatai szigorú sorrendet követnek, és az egyes műveletek egymástól függenek.

A termelésütemezés a gyártási és termelési rendszerek egyik központi folyamata, amely meghatározza, hogy az egyes termelési lépések mikor és milyen sorrendben kerülnek végrehajtásra, figyelembe véve a rendelkezésre álló erőforrásokat és a vállalat célkitűzéseit. A hatékony ütemezés kulcsfontosságú a termelés optimalizálása, a költségek csökkentése, az átfutási idők rövidítése és a vevői igények teljesítése szempontjából. A modern iparban a termelésütemezési problémák összetettsége folyamatosan nő az egyre nagyobb termelési volumenek, a testreszabott termékek iránti kereslet és a globális verseny miatt, ami a folyamatok folyamatos optimalizálását igényli.

A termelésütemezés során számos kihívással kell szembenézni, például a munkaállomások, a gépek, az emberi erőforrások és az alapanyagok optimális elosztásával. Az ütemezési problémák bonyolultságát az is fokozza, hogy különböző típusú korlátokkal, például kapacitás-, idő- és költségkorlátokkal kell számolni. Ezen feladatok megoldására hagyományos matematikai módszerek és optimalizációs technikák alkalmazhatók, ám ezek nem mindig képesek hatékony megoldást nyújtani a nagy léptékű, dinamikus rendszerekben.

5.2 A flowshop ütemezési feladat

A flowshop ütemezési feladat egy olyan speciális termelésütemezési probléma, amelyben a gyártási folyamat minden egyes munkadarabra azonos sorrendben és fix munkafolyamatban zajlik le, több munkaállomáson vagy gépen keresztül. Ebben a modellben minden munkadarab ugyanazokon a műveleti lépéseken halad keresztül, általában az első munkaállomástól az utolsóig, és a termelési sorrend sem változtatható. A cél a gyártási folyamat optimalizálása, amely történhet a teljes átfutási idő minimalizálásával, a késések csökkentésével vagy az erőforrások maximális kihasználásával.

A flowshop problémát az jellemzi, hogy egy gépen egy időben csak egy munkadarab dolgozható fel, és minden munkadarab csak akkor léphet a következő munkaállomásra, ha az nem elfoglalt. Ez a szervezési struktúra gyakran előfordul olyan iparágakban, mint az

autógyártás, az elektronikai összeszerelés vagy az élelmiszeripar, ahol a termelési folyamat szigorú sorrendet köve.

A flowshop feladat a többgépes, több operációs ütemezési problémák egyik klasszikus feladattípusa, ahol a cél a feladatok minimális átfutási idejének elérése az optimális indítási sorrendjük változtatásával. Az optimális sorrend megtalálásához sokgépes esetben legtöbbször nincs egzakt algoritmus, ezért gyakran használnak különféle heurisztikákat és algoritmusokat egyaránt. 1993-ban a flowshop problémához benchmark dataset-et publikáltak [1]. Ezzel összehasonlíthatóvá váltak az egyes algoritmusok.

A flowshop ütemezési feladat általában NP-teljes problémának számít, különösen, ha az optimalizálási cél a teljes átfutási idő (makespan) minimalizálása több gépen. Az NP-teljesség azt jelenti, hogy nincs ismert polinomiális időben megoldható algoritmus a probléma általános esetére. Ezért a flowshop ütemezésnél, ahol az összes munkát meghatározott sorrendben kell végrehajtani különböző gépeken, a lehetséges sorrendek száma nagyon gyorsan növekszik a feladatok és gépek számával.

Mivel a probléma méretével exponenciálisan nő a lehetséges megoldások száma, a hagyományos módszerekkel a flowshop problémák megoldása nagyléptékű rendszerek esetén gyakorlatilag lehetetlen. Emiatt gyakran alkalmaznak heurisztikus és metaheurisztikus algoritmusokat (pl. genetikai algoritmusok, memetikus algoritmusok, szimulált lehülés) a flowshop ütemezési feladat közelítő megoldására.

A genetikai algoritmusok (GA) a flowshop ütemezési feladat megoldására az evolúció alapelveit alkalmazzák, hogy iteratív módon keressenek optimális vagy közel optimális ütemezési megoldásokat. A folyamat egy kezdeti populáció létrehozásával indul, ahol minden egyes egyed különböző ütemezési sorrendet, azaz lehetséges megoldást képvisel a problémára. Minden egyedhez egy fitness értéket rendelnek, amely a flowshop esetében gyakran a teljes átfutási idő, vagyis a makespan minimalizálását célozza; minél kisebb a makespan, annál jobb az ütemezés.

A genetikai algoritmus a természetes szelekciót utánozva választ ki egyedeket a populációból a következő generáció számára, előnyben részesítve a jobb fitness értékű megoldásokat, így növelve az optimalizálási hatékonyságot. A kiválasztott megoldásokon keresztül történik meg a keresztezés, amely két ütemezési sorrendből hoz létre újakat. A flowshop esetében speciális keresztezési módszereket használnak annak biztosítására, hogy az új megoldások megőrizzék a sorrendek egyediségét és teljességét.

A mutációs lépés szintén kulcsfontosságú az algoritmusban, mivel a feladatok sorrendjének kisebb módosításával növeli az új megoldások felfedezésének esélyét, így elkerülve a lokális optimumokba való beragadást. A genetikai algoritmus ezt a folyamatot ismételve generációról generációra javítja a populáció minőségét, közelítve az optimális ütemezéshez. Az algoritmus tehát a különböző ütemezési sorrendek folyamatos fejlődésén keresztül törekszik a teljes

átfutási idő minimalizálására, amely különösen hasznos a nagy léptékű, komplex flowshop problémák esetén.

A memetikus algoritmusok szerepe az, hogy kombinálják az evolúciós algoritmusok globális keresési képességeit a helyi keresési technikák finomítási erejével. Az ütemezési problémák gyakran rendkívül összetettek, sokféle megoldási lehetőséget kínálnak, és számos helyi optimumot tartalmaznak. A memetikus algoritmusok célja, hogy ebben a komplex térben hatékonyan navigáljanak, és elkerüljék a lokális optimumokba való beragadást.

A memetikus algoritmus globális keresés során alkalmazza az evolúciós műveleteket – például keresztezést és mutációt –, így a megoldások széles tartományát fedezi fel. Emellett minden generációban vagy bizonyos lépések után helyi keresési technikákat is bevet egy-egy megoldás finomítására, ami gyorsabb konvergenciát és pontosabb eredményeket eredményez. A helyi keresés lehet például gráfelméleti vagy heurisztikus módszer, amely a megoldások optimalizálását a problématerület specifikus jellemzőinek figyelembevételével végzi.

Az ütemezésben a memetikus algoritmusok előnyösnek bizonyultak, mivel az evolúciós algoritmusok globális keresési kapacitása mellett a helyi keresési módszerekkel jelentősen növelik a hatékonyságot, így a komplex, nagyméretű problémák esetén is jó ütemterveket hoznak létre.

5.3 Bakteriális memetikus algoritmus (DBMEA)

A bakteriális memetikus algoritmus egy számítógépes módszer, amelyet a biológiából ismert bakteriális szaporodási mechanizmus ihletett [8]. Ez a modell a természetben megfigyelhető génkeresztezési folyamatot utánozza, amely a baktériumok szaporodásakor játszódik le. A folyamat két alapvető műveletből áll: a bakteriális mutációból és a géntranszferből. Az algoritmus kezdetén egy véletlenszerű tulajdonságokkal rendelkező kezdeti populációt hozunk létre, majd ezen műveletek révén új egyedeket generálunk a következő generációban. Az így létrejövő második generáció egyedeit egy célfüggvény segítségével egyenként kiértékeljük: egyes egyedeket eltávolítunk, míg másokat a tulajdonságaik alapján előnyben részesítünk. Ezt a folyamatot ismételjük a leállási kritérium teljesüléséig.

Az n -opt operátorok kulcsfontosságú szerepet játszanak a bakteriális memetikus algoritmusokban (BMA) azért, hogy finomítsák a megoldásokat és javítsák az algoritmus konvergenciáját. A bakteriális memetikus algoritmusok az evolúciós és memetikus elveket ötvözik, és a globális és lokális keresés kombinációjára épülnek. Az n -opt operátorok a lokális keresés részét képezik, ahol a meglévő megoldásokat kismértékben módosítják annak érdekében, hogy jobb, lokálisan optimalizált megoldásokat találjanak. A legismertebb példák a 2-opt és 3-opt operátorok:

- 2-opt operátor: Két elem vagy kapcsolat cseréjével javítja a megoldást. Például a városokat összekötő útvonal-optimalizálási feladatban a 2-opt operátor két város közötti

utat megszünteti, majd a városokat egy másik sorrendben köti össze, így csökkentve a teljes útvonal hosszát. Ez az egyszerű, de hatékony módszer csökkenti a hurkokat és optimalizálja a részmegoldásokat.

- 3-opt operátor: Három elem vagy kapcsolat egyidejű cseréjét végzi. A 3-opt módszer több lehetséges változtatási kombinációt vizsgál meg az adott három elem között, ami nagyobb keresési tér átvizsgálását teszi lehetővé. A 3-opt operátor hatékonyabb megoldásokat találhat, de számításigényesebb, mint a 2-opt, mivel több kombinációt kell vizsgálni.

5.3.1 Az algoritmus bemutatása

Legyenek a baktériumok az ütemezési feladat lehetséges megoldásai. Az első lépésben a kezdeti populáció véletlenszerű generálása történik. A második lépésben a bakteriális mutációt alkalmazzuk. A harmadik lépésben a memetikus lépés, a lokális keresőt A lokális keresés megpróbálja tovább javítani (finomítani) az egyes megoldásokat (pl.: szomszéd megoldások generálásával), amíg nem talál egy jobb szomszédot.

A szomszédsági viszonyt az alábbiak szerint értelmezhetjük: Tekintsük a keresési teret egy X halmazként, amely a problémában lehetséges megoldásokat tartalmazza. Adott egy x megoldás. szomszédsági struktúra egy olyan $N(x) \subseteq X$ halmaz, amely tartalmazza azokat a megoldásokat, amelyek kis mértékben térnek el x -től. Ez a halmaz határozza meg a lokális keresés „lépésközét”.

Legyen $f: x \rightarrow \mathbb{R}$ egy célfüggvény, amely minden $x \in X$ megoldáshoz hozzárendeli annak az „értékét”.

A harmadik lépés (lokális keresés) akkor ér véget, ha a szomszédos megoldások között nem találunk jobbat. A negyedik lépés során a gén-transzfer mutációt hajtjuk végre. Az algoritmus a 2. - 4. lépéseket ismétli mindaddig, amíg el nem éri a leállási kritériumot. Végül a legjobb megoldással tér vissza.

Az algoritmust az alábbi pszeudokód mutatja be. I_{trans} : a gén-transzfer szegmensek hossza.

1. Algorithm Diszkrét Bakteriális Memetikus Evolúciós algoritmus

procedure DBMEA

Step 1: generate initial population P

while termination criteria is not met **do**

Step 2: bacterial mutation (P, N_{clones}, I_{seg})

Step 3: local search

Step 4: $x_1 =$ gene transfer (P, N_{inf}, I_{trans})

if $f(x_1) < f(x_{best})$ **then**

Step 5: $x_{best} = x_1$

return x_{best}

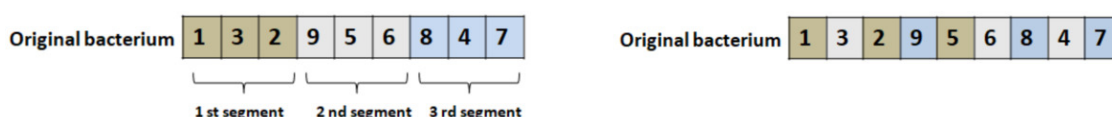
1. algoritmus DBMA áttekintő lépései

Az alkalmazott jelölések:

- P : az ütemtervek halmaza (populáció),
- x_I : a vizsgált ütemterv,
- x_{best} : eddigi legjobb ütemterv,
- f : jósági (fitness) függvény,
- N_{clones} : klónok száma,
- I_{seg} : mutációs szegmensek száma,
- N_{inf} : fertőzések száma a gén-transzferben,

5.3.2 A bakteriális mutációs eljárás

A P populációt alávétjük a bakteriális mutációnak. Adott N_{clones} számú klónt készítünk minden p baktériumból, ehhez a p baktériumot szegmensekre bontjuk. Két szegmentálási módot alkalmazunk véletlenszerű módon: koherens szegmensek, és laza szegmensek előállításával. A koherens szegmens mutáció esetén a szegmens elemei sorfolytonosan következnek, lásd 5. ábra a). Laza szegmens mutációnál nem feltétlenül egymás melletti elemek alkotják a szegmenst (5. ábra b)). Mindkét szegmens-variáns esetében a folyamat során sorra végig iterálunk az összes szegmensen, és kiválasztunk egyet, amelyen még nem hajtottunk végre mutációt. Az első lépésben a szegmens elemeit tükrözzük, ami az első klónt eredményezi. Ezután a szegmens elemeit véletlenszerűen átrendezzük, és ezzel hozzuk létre a további klónokat. Ezzel az eljárással összesen N_{clones} klónt generálunk, amelyek a szegmens különböző permutációit képviselik.



5. ábra Mutációk típusai: a) koherens b) laza

A mutációs algoritmus célja, hogy javítsa az eredeti megoldást, miközben feltárja a szegmens permutációs terét. A klónok értékelése egy adott célfüggvény alapján történik, amely lehet például a megoldás költsége vagy az átfutási idő minimalizálása. Az algoritmus végén a klónok közül a legjobb, azaz a célfüggvény szerint optimális klón kerül kiválasztásra, és átveszi az eredeti egyed helyét a populációban.

2. Algorithm Bakteriális mutációs eljárás

```

procedure BACTERIAL_MUTATION( $P, N_{clones}, I_{seg}$ )
  for all  $p$  in  $P$  do
    Step 1: create a random number between 0 and 1.
    Step 2: create  $N_{clones}$  clones of  $p$  and a random number  $r \in [0..1]$ 
    if  $r \leq COHERENT\_LOOSE\_RATE$  then
      Step 3: cut  $p$  into coherent segments with  $I_{seg}$  length
    else
      Step 4: cut  $p$  into loose segments with  $I_{seg}$  length
    Step 5: replace  $p$  with the best of the clones and  $p$ 
  return  $P$ 

```

2. algoritmus A bakteriális mutáció

5.3.3 Gén-traszfer művelet

A gén-traszfer operációt az egész populáción végrehajtjuk. Első lépésként a populáció egyedeit fitnessz értékeik alapján sorba rendezzük, hogy az egyedek relatív minősége alapján prioritizálhassuk őket. Ezt követően a populációt két részre osztjuk: egy felsőbbrendű csoportba, amely a rangsor élén álló, magasabb fitnessz értékű egyedeket tartalmazza, és egy alsóbbrendű csoportba, amely a gyengébb teljesítményű egyedeket foglalja magában.

Következő lépésként a gén-traszfer mutációt végrehajtjuk N_{inf} alkalommal mindig egy véletlenszerűen kiválasztott felsőbbrendű és alsóbbrendű egyed között. A gén-traszfer mutáció során egy véletlenszerűen kiválasztott I_{trans} hosszúságú szegmenst transzferálunk a felsőbbrendű baktériumból az alsóbbrendűbe. Ez a transzfer biztosítja, hogy a jobb teljesítményű egyedek értékes genetikai információi (tulajdonságai) bekerüljenek az alsóbbrendű egyedekbe, így potenciálisan javítva azok minőségét. A transzfer során ügyelünk arra, hogy az alsóbbrendű baktériumban ne keletkezzenek duplikált elemek, azaz a szegmens átvétele ne sértse a megoldás integritását.

A gén-traszfer művelet hatékonyan kombinálja a populáció diverzitásának növelését a legjobb tulajdonságok propagálásával. A felsőbbrendű egyedek legjobb szegmenseinek megosztása az alsóbbrendűekkel javítja a populáció általános jóságát (fitness értékét), miközben az alsóbbrendű egyedek továbbra is hozzájárulnak a genetikai variáció fenntartásához. Ez a mechanizmus kulcs a lokális optimumok elkerülésében és a globális optimalizáció elősegítésében.

3. Algorithm Gén-transzfer eljárás**procedure** GENE TRANSFER(P, N_{inf}, I_{trans})

Step 1: sort the population according to fitness values

Step 2: divide the population into superior and inferior parts

for i to N_{inf} **do**Step 3: select a random bacterium from the superior part (p_{src})Step 4: select a random bacterium from the inferior part (p_{dst})Step 5: select a random segment from p_{src} with I_{trans} lengthStep 6: copy the segment into p_{dst} in a random positionStep 7: eliminate duplicates in p_{dst} **return** P *3. algoritmus A gén transzfer eljárás lépései*

Ahol:

 p_{src} : a forrás baktérium, melynek felhasználásával transzformálunk, p_{dst} : a cél baktérium, melybe áthelyezzük a forrásból származó szegmenst.

5.3.4 A Monte-Carlo fakeresés algoritmus

A Monte-Carlo fakeresés (Monte Carlo Tree Search, MCTS) egy intelligens keresési és döntéshozatali algoritmus, amelyet elsősorban játékok és más szimulációs problémák megoldására alkalmaznak. Az MCTS célja egy fa struktúra bejárása és egy optimális döntés meghozatala a fa egy pontján, a lehetséges kimenetek szimulációjával és értékelésével.

Az MCTS az erősítő tanulás és a sztochasztikus megközelítések elemeit használja, hogy a fa ágainak vizsgálatát és bővítését dinamikusan optimalizálja. Az algoritmus azon az elven működik, hogy egy adott állapotból véletlenszerűen próbálja meg szimulálni a lehetséges jövőbeni lépéseket (Monte-Carlo szimulációk), majd ezeket az eredményeket visszacsatolja a keresési fa korábbi állapotaihoz. Ez lehetővé teszi, hogy az algoritmus egyensúlyt találjon a feltárás (új ágak vizsgálata) és a kihasználás (az ígéretesebb ágak további feltárása) között.

Az MCTS különösen hatékony az olyan problémákban, ahol a megoldási tér túl nagy a kimerítő kereséshez, és az optimális megoldás nem érhető el közvetlenül, de szimulációkon keresztül becsülhető. Például az MCTS-t gyakran használják stratégiai játékokban (sakk, Go) vagy más olyan területeken, ahol döntéshozatali faelemzés szükséges.

Az algoritmus az alábbi az alábbi négy lépés ciklikus végrehajtásán alapul:

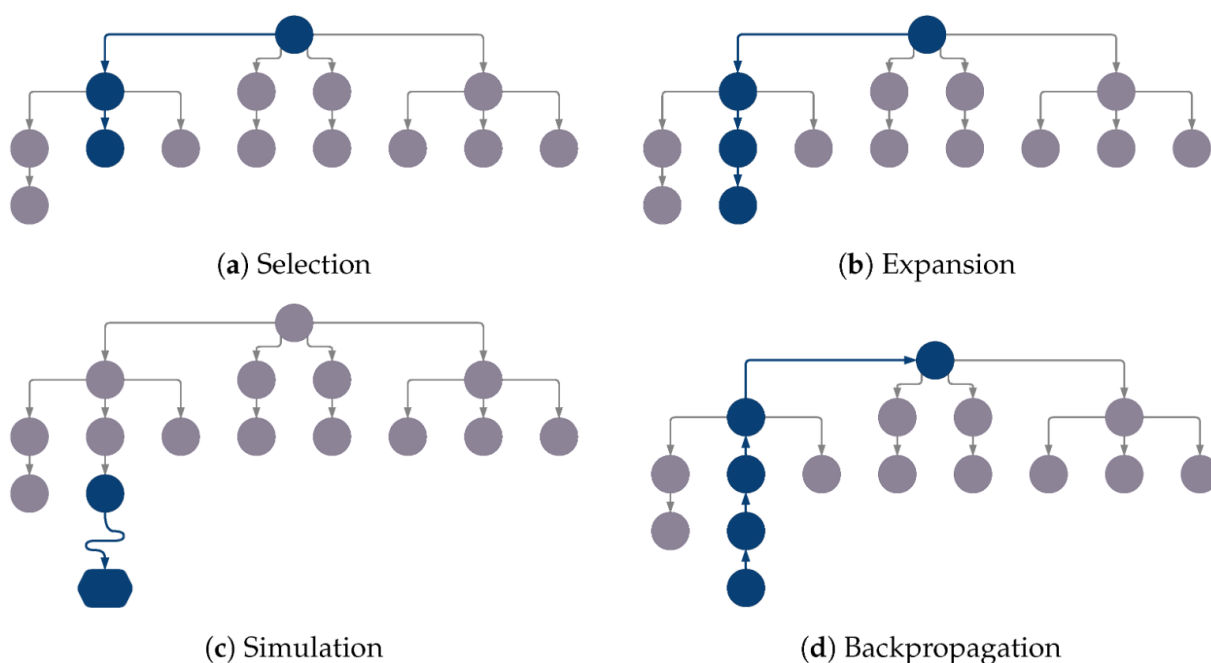
1. Kiválasztás: rekurzívan, fentről-lefelé kiválasztunk egy gyerek levelet a gyökérelemtől kezdve.
2. Kiterjesztés: véletlenszerűen létrehozunk egy új gyermek levelet.

3. Szimuláció: kiértékelünk, új levélből kiindulva elvégezzük a jósági függvény értékének kiszámítását. Kiértékeljük a „játzmát”: nyer, veszít vagy döntetlen (+1, -1, 0) értékalmazra leképezve az eredményt.
4. Visszacsatolás: a játszma értékelése alapján rekurzívan felfelé lépve frissítjük a levelek nyilvántartásait, növeljük a próbálkozások számát, frissítjük a nyert játszmák számát.

A lépéseket a 6. ábra mutatja.

A feltárás (exploration) és a kihasználás (exploitation) közötti egyensúlyt az UCB (Upper Confidence Bound) számítása határozza meg. A két feladattípus definíciója:

- Hasznosítás (Exploitation): Az algoritmus előnyben részesíti azokat a csomópontokat (node-okat), amelyeket már jól feltérképezett, és amelyek eddig ígéretes eredményeket adtak.
- Feltárás (Exploration): Az algoritmus olyan csomópontokat is vizsgál, amelyeket eddig kevésbé vizsgált, hátha ezek jobb eredményeket hoznak.



6. ábra A MCTS algoritmus 4 alaplépése

A számítás az alábbi képlet alapján történik:

$$UCB = \frac{\omega_i}{n_i} + c \sqrt{\frac{\ln(N)}{n_i}}, \quad (9)$$

ahol

- ω_i az az i -edik csomópontnál elért összesített nyereség (reward).
- n_i az i -edik csomópontot eddig hányszor választotta az algoritmus.
- N az i -edik szülő levélen keresztül játszott összes játszma darabszáma.

- c a feltárás mértékét szabályzó konstans.

Az algoritmus futása során, a fa fokozatosan és aszimmetrikusan növekszik a feltárási és hasznosítási stratégiák által meghatározott módon. A c konstans növelése/csökkentése befolyásolja a súlyt a két stratégia között. *Feltárás* alatt azt értjük, hogy az algoritmusnak véletlenszerű választásra van szüksége arra, hogy bejárja a keresési teret. A *hasznosítás* alatt az ismert legjobb opció újrahasznosítását értjük.

Az eredetileg kétjátékos táblás játékokra tervezett algoritmust itt ütemezési problémák megoldására alkalmazzuk. Ebben a modellben a fa levelei egy-egy permutációt, vagyis egy lehetséges ütemezési sorrendet képviselnek. A levelek gyermekeit a szülő permutációjából módosító műveletek segítségével hozzuk létre. A fa minden csomópontja egy potenciális megoldást jelent, amelyek között szomszédsági reláció áll fenn. A fában ugyanaz a megoldás többször is megjelenhet, akárcsak a táblás játékok algoritmusában.

5.3.5 Hibrid DBMEA algoritmus

Az eredeti DBMEA algoritmus a *2-opt* és *3-opt* módszereket alkalmazza. Vizsgálataink alapján kiderült, hogy a flow shop feladatnál ezek nem hatékonyak. A lokális keresés hatékonyságának növelésére Monte Carlo fa-keresést (MCTS) alkalmazunk, ahol az egyes szimulációkat szimulált hűtéssel (SA) értékeljük ki, ezzel kialakítva a DBMEA + MCTS + SA megközelítést. Az MCTS kiterjesztési lépésében meghatározzuk a szimulált hűtés kezdeti permutációját, amelyből a játszmák során jobb, rosszabb vagy azonos permutációkat találunk. A szimulált hűtés alkalmazása lehetővé teszi, hogy a módszer rosszabb megoldásokat is elfogadjon, így elkerülve a lokális optimumokba való beragadást. A lokális minimumokból való kijutás elősegítésére, egy ún. egy halandósági (N_{mort}) rátát alkalmaztunk. a populáció bizonyos része eldobásra kerül (meghal), véletlenszerűen generált új egyedek lépnek ezek helyébe.

5.3.6 Futási eredmények

Eric Taillard [10] publikációjában olyan benchmarkokat vezetett be, amelyek célja az ütemezési algoritmusok teljesítményének objektív és összehasonlítható értékelése. A benchmarkok különféle ütemezési problémákra, például flow shop, job shop és open shop problémákra lettek kifejlesztve, amelyek a diszkrét ütemezési problémák alapvető típusai. Az egyes példák úgy lettek kialakítva, hogy valóság-hű feldolgozási időket tartalmazzanak, és különböző méretű, komplexitású problémákat modellezzenek. A méretek a kisebb, könnyebben kezelhető problémáktól (például öt gép és tíz munka) a közepes és nagyobb méretű problémákig terjednek, amelyek közelebb állnak a valós ipari környezetek kihívásaihoz.

Ezek a benchmarkok lehetőséget adnak arra, hogy a kutatók az új algoritmusokat egységes és standardizált környezetben teszteljék. Mivel minden benchmark példához tartozik egy meghatározott célfüggvény, például a makespan minimalizálása, az algoritmusok által elért eredmények könnyen összehasonlíthatók. A benchmarkokhoz tartozó példákban meghatározott

feldolgozási idők és struktúrák egyértelmű formátumban érhetők el, amely lehetővé teszi azok egyszerű implementációját.

Ezt a klasszikus benchmarkot vettük alapul az algoritmusok kiértékelésére és összehasonlítására. Különböző algoritmusok átlagos teljesítményének az összehasonlításához az eddigi legjobb találatához viszonyítjuk az eredményeket, a következőképpen.

$$q = \frac{C_{BS} - C_{uB}}{C_{uB}} \quad (10)$$

ahol C_{BS} jelöli az algoritmus által talált legjobb átfutási idő értéket, C_{uB} pedig az eddigi ismert legjobb eredményt.

Eredményeink azt mutatják, hogy az alkalmazott Hibrid Bakteriális Memetikus algoritmus a benchmark feladatok esetében átlagosan kevesebb mint 1%-os eltéréssel közelíti meg az ismert legjobb megoldásokat. Kilenc olyan esetet is bemutatunk, ahol sikerült túlszárnyalni az eddig ismert legjobb eredményeket; ezekben az esetekben megadtuk az új átfutási idő értéket és a hozzájuk tartozó permutációkat, így az elért eredmények egyszerűen ellenőrizhetők és reprodukálhatók.

Kutatásunk során tehát azt vizsgáltuk, hogy mely lokális kereső alkalmazásával növelhető tovább a hatékonyság. A [P5.1]-ben publikált megközelítésünkben a szimulált hűtéssel kombináltuk a diszkrét bakteriális memetikus algoritmust (DBMA) és kimutattuk, hogy jobb eredményeket kapunk, mint a kombinálatlan alapváltozatokban. A [P5.2]-ben tovább finomítottuk az eljárást.

5.4 Kiterjesztett flowshop ütemezési feladat

Az Extended Flexible Flow Shop (EFFS) modell kiterjesztett folyamatrendszerű ütemezési problémákat kezel, amelyek komplexebb feltételeket és rugalmasságot biztosítanak a hagyományos flow shop modellekhez képest. Az „kiterjesztett” megközelítés lényege, hogy számos további jellemzőt és korlátozást is figyelembe vesz, amelyeket a valós ipari gyártási folyamatok igényelnek.

A kiterjesztés lehetővé teszi, hogy a valós ipari gyártási problémákat pontosabban modellezzék. Az EFFS modellben figyelembe vehetőek az alternatív technológiai útvonalak, amelyek különböző gépeket és sorrendeket tartalmaznak, valamint nem kapcsolódó párhuzamos gépek, amelyek eltérő termelékenységgel működnek a különböző termékek gyártása során.

A sorfüggő átállási idők, ahol az egyik termékről a másikra való váltás időigénye változó, szintén a modell részét képezik. Ezen túlmenően a gépek elérhetősége is korlátozott lehet, hiszen bizonyos időszakokban nem állnak rendelkezésre. Az ütemezés során olyan speciális gyártási korlátokat is kezelni kell, mint a termékek indulási ideje vagy határideje. Az EFFS továbbá lehetőséget nyújt több cél egyidejű optimalizálására, például a késések, az átállási idők vagy a gépkihasználat együttes minimalizálására.

A modellben a munkák csoportosítása is fontos szerepet játszik, hiszen ez elősegíti az ütemezési folyamat hatékonyságát. Az EFFS tehát a klasszikus flow shop alapvető struktúráját ötvözi a valós gyártási folyamatok bonyolultságával, ezáltal pontosabb és alkalmazhatóbb megoldásokat kínál.

5.5 Többcélú heurisztikus ütemezés

A kidolgozott modell olyan jellemzőket integrál, mint az alternatív technológiai útvonalak, párhuzamos, de nem azonos teljesítményű gépek, valamint a sorfüggő beállítási idők és az erőforrások időbeli korlátozott elérhetősége. Az újdonságot az jelenti, hogy ezek a tényezők egyetlen keretrendszeren belül kerülnek kezelésre, miközben a modell egyidejűleg több célkitűzést optimalizál.

A modell egy új, relatív összehasonlításon alapuló metrikát mutat be, amely a megoldások minőségét nem abszolút, hanem viszonylagos szempontok alapján értékeli. Ez a megközelítés hatékonyabbá teszi a célfüggvények közötti kompromisszumok kezelését. Emellett a kifejlesztettünk egy integrált megoldási stratégiát, amely nem bontja alproblémákra az ütemezési feladatot, hanem egyszerre kezeli a csoportosítást, hozzárendelést, sorrendbe állítást és időzítést.

A modell tabu keresési algoritmust használ. A tabu keresést speciális szomszédsági operátorokkal és egyedi tabu lista struktúrával, amely a megoldások kódolt formáját tárolja. A többcélú optimalizációhoz igazított kiértékelési függvény használtunk, amely a különböző célkitűzések közötti egyensúlyt támogatja.

Végül, a módszerek validációja valós ipari esetre épül, amely egy multinacionális világítástechnikai cég gyártási folyamatait modellezi. Ez igazolja a megközelítés gyakorlati alkalmazhatóságát és hatékonyságát a valós gyártási környezetekben.

Az újraütemezési folyamat a már meglévő ütemezések módosítását célozza a gyártási folyamatot érintő váratlan események – például géphibák, anyaghiány, vagy prioritásváltozások – hatékony kezelése érdekében. A kidolgozott modell újdonsága abban rejlik, hogy a módosítások során figyelembe veszi a korábbi ütemezési döntéseket, és törekszik a rendszer stabilitásának megőrzésére azáltal, hogy minimalizálja az eredeti ütemezéstől való eltéréseket. Ehhez olyan minőségi mutatókat vezet be, amelyek az új és az eredeti ütemezések közötti változásokat mérik, például a módosított munkák, gépek vagy sorrendek számát.

Az újraütemezéshez alkalmazott algoritmus többcélú optimalizációra épül, és egyedi „fagyasztási technikákat” alkalmaz. Ezek a technikák biztosítják, hogy a már elkészült vagy éppen futó munkák állapota változatlan maradjon, és csak azokat a munkákat érintse a módosítás, amelyek szükségesek az ütemezési probléma megoldásához. Az algoritmus a részleges, teljes és időeltolódásos újraütemezési módszerek rugalmas kombinációját használja, amely lehetővé teszi a különböző helyzetekhez való adaptációt.

Az újraütemezési módszerek integrálják a tabu keresési algoritmusokat, amelyek szomszédsági operátorai az új ütemezések gyors és hatékony előállítását biztosítják, miközben figyelembe veszik az új körülményeket és az eredeti ütemezéshez való hasonlóság fenntartását. A módszert úgy tervezték, hogy a felhasználók valós időben beavatkozhatnak, módosíthatják az algoritmus paramétereit, és manuálisan is szerkeszthessék az ütemezéseket, biztosítva a gyakorlati alkalmazhatóságot és rugalmasságot.

Ez a kombinált megközelítés jelentős előrelépést képvisel az újraütemezés területén, mivel nemcsak a hatékonyságot növeli, hanem a rendszer stabilitását is fenntartja, miközben lehetőséget ad a felhasználói kontrollra.

5.6 Tézisek

T5.1 A diszkrét gyártási folyamatok ütemezésére egy hibrid Bakteriális Memetikus Evolúciós Algoritmust (DBMEA) fejlesztettünk ki, és kimutattuk, hogy ez képes hatékonyan kezelni a flowshop típusú termelésütemezési problémákat. Az algoritmus a hagyományos evolúciós technikák mellett helyi keresési módszereket is alkalmaz, növelve az optimalizáció hatékonyságát.

T5.2 Kimutattuk, hogy a DBMEA algoritmusba integrált Monte Carlo fa-keresés (MCTS) jelentősen növeli a lokális keresés hatékonyságát a flowshop problémák esetén. Az MCTS során az egyes állapotokat szimulált hűtéssel értékeljük ki, amely lehetővé teszi a lokális optimumokból való kilépést, ezáltal javítva a globális megoldásminőséget.

T5.3 Kimutattuk, hogy a szimulált hűtés (SA) alkalmazása az MCTS folyamat során elősegíti a rosszabb megoldások elfogadását, amely hozzájárul a keresési tér hatékonyabb bejárásához és az optimálisabb eredmények eléréséhez. Ezzel a megközelítéssel az algoritmus képes elkerülni a lokális optimumokba való beragadást.

T5.4 A bakteriális mutációs technikák finomhangolására az algoritmusban alkalmazott bakteriális mutáció során véletlenszerűen generált koherens és laza szegmensek átrendezésével új megoldásokat hoztunk létre. Az így generált klónok közül a legjobb megoldásokat választva sikerült javítani a populáció általános minőségét. A gén-transzfer művelet során a populáció legjobb megoldásaiból származó szegmenseket integráltuk gyengébb megoldásokba, ezzel növelve az új megoldások diverzitását és a globális optimum elérésének esélyét. Az algoritmusba integrált „halandósági ráta” segítségével véletlenszerűen generált új egyedekkel helyettesítjük a populáció bizonyos részeit, amely tovább javítja az algoritmus képességét a keresési tér felfedezésére és a globális optimum elérésére.

T5.5 A DBMEA + MCTS + SA hibrid megközelítés hozzájárul egy új algoritmikus standard kidolgozásához a komplex ütemezési problémák, különösen a flowshop feladatok megoldásában, amely a gyakorlati alkalmazásokban is kiváló eredményeket mutatott. Kimutattuk, hogy az algoritmus a benchmark feladatok esetében átlagosan kevesebb mint 1%-os

eltéréssel közelíti meg az ismert legjobb megoldásokat. A 120 benchmark feladat közül 9 esetben az eddig ismert legjobb megoldásnál jobb megoldást találtunk.

5.7 A téziszhez kapcsolódó publikációk

[P5.1] A. Agárdi, K. Nehéz, O. Hornyák, and L. T. Kóczy, “A Hybrid Discrete Bacterial Memetic Algorithm with Simulated Annealing for Optimization of the Flow Shop Scheduling Problem,” SYMMETRY (BASEL), vol. 13, no. 7, paper 1131, 12 p, 2021., WoS: Q2 SJR: Q2

[P5.2] Fazekas, B. Tüü-Szabó, L. T. Kóczy, O. Hornyák, and K. Nehéz, “A Hybrid Discrete Memetic Algorithm for Solving Flow-Shop Scheduling Problems”, ALGORITHMS, vol. 16, no. 9, p. 406, 2023., WoS: Q2 SJR: Q2

5.8 Felhasznált irodalom

- [1] Botzheim, J., Cabrita, C., Kóczy, L. T., & Ruano, A. E. (2009). Fuzzy rule extraction by bacterial memetic algorithms. *International Journal of Intelligent Systems*, 24(3), 312–339.
- [2] Chen, S.-H., Chang, P.-C., Cheng, T. C. E., & Zhang, Q. (2012). A self-guided genetic algorithm for permutation flowshop scheduling problems. *Computers & Operations Research*, 39(7).
- [3] Das, S., Chowdhury, A., & Abraham, A. (2009). A bacterial evolutionary algorithm for automatic data clustering. In *2009 IEEE Congress on Evolutionary Computation* (pp. 2403–2410). IEEE.
- [4] Gao, K., Yang, F., Zhou, M., Pan, Q., & Suganthan, P. N. (2018). Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm. *IEEE Transactions on Cybernetics*, 49, 1944–1955.
- [5] Inoue, T., Furuhashi, T., Maeda, H., & Takaba, M. (2002). A study on interactive nurse scheduling support system using bacterial evolutionary algorithm engine. *IEEE Transactions on Electronics, Information and Systems*, 122(10), 1803–1811.
- [6] Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Technical Report, Caltech Concurrent Computation Program Report 826*. California Institute of Technology: Pasadena, CA, USA.
- [7] Muyldermans, L., Beullens, P., Cattrysse, D., & Van Oudheusden, D. (2005). Exploring variants of 2-opt and 3-opt for the general routing problem. *Operations Research*, 53(6), 982–995.
- [8] Nawa, N. E., & Furuhashi, T. (1999). Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Transactions on Fuzzy Systems*, 7(5), 608–616.
- [9] Nawaz, M., Enscore, E. E., Jr., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11, 91–95.
- [10] Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2), 278–285.

- [11] Tseng, L.-Y., & Lin, Y.-T. (2010). A hybrid genetic algorithm for no-wait flowshop scheduling problem. *International Journal of Production Economics*, 128(1), 144–152.
- [12] Van Laarhoven, P. J., & Aarts, E. H. (1987). Simulated annealing. In *Simulated Annealing: Theory and Applications* (pp. 7–15). Springer: Berlin/Heidelberg, Germany.

6 Az állapotfelügyelet AdaBoost algoritmussal és az alkalmazási korlátok feltárása

6.1.1 Osztályozási feladatok

A felügyelt gépi tanulás egyik leggyakrabban vizsgált problémátípusa az osztályozás, amelynek során egy adott bemenethez tartozó címkét kell meghatározni. Az osztályozási feladatok során a cél egy olyan függvény betanítása, amely egy jellemzővektor (feature vector) alapján helyesen határozza meg a jelenséghez tartozó osztály kategóriát. A gyakorlati alkalmazások széles skáláját fednek le az orvosi diagnosztikától kezdve az ipari állapotfelügyeleten át az képfelismerésig vagy a természetes nyelvfeldolgozásig.

Az osztályozási algoritmusok működhetnek lineáris- (pl.: regressziós), nemlineáris- (pl. döntési fák, k-legközelebbi szomszéd algoritmus, k-nearest neighbor (kNN)) vagy statisztikai alapú modelleken (pl. Naiv Bayes). Az utóbbi évtizedben jelentős figyelmet kaptak az ensemble, azaz többszörös modell-alapú tanulási módszerek, amelyek célja, hogy több – külön-külön gyenge teljesítményű – osztályozó kombinálásával egy erősebb, megbízhatóbb modellt állítsanak elő.

Ezek közé tartozik az AdaBoost (Adaptive Boosting) algoritmus is, amely a gyenge tanulók egymás után fűzésével próbálja meg csökkenteni az osztályozási hibát. Az AdaBoost működésének lényege, hogy az egyes iterációk során növeli azoknak a mintáknak a súlyát, amelyeket az előző gyenge modell hibásan osztályozott. Így a későbbi modellek „tanulnak” az előzőek hibáiból. A végeredmény egy súlyozott többségi szavazással működő „erős tanuló”, amely a gyenge tanulók együttes teljesítményét egyesíti. Az AdaBoost algoritmus tehát egy meta-algoritmus, amely különösen jól működik kis hibával rendelkező gyenge tanulókra építve, például egyszerű döntési fákra.

Az AdaBoost egyik előnye, hogy nem hajlamos a túltanulásra, amennyiben nem túl bonyolult gyenge tanulóra épül. Ugyanakkor érzékeny lehet az adathalmazok kiegyensúlyozatlanságára és a zajos adatokra, mivel a hibásan osztályozott minták súlya folyamatosan növekszik, ami túlzott fókusz eredményezhet a marginális példákra.

A tézisfüzet további részében részletesen bemutatom, hogyan alkalmaztam az AdaBoost algoritmust ipari állapotfelügyeleti adatokra, és milyen kihívások merültek fel az algoritmus prediktív teljesítményével és alkalmazási korlátaival kapcsolatban. A vizsgálatok során külön figyelmet fordítottam az osztályok közötti aránytalanság kezelésére, valamint a jellemzők kiválasztásának és súlyozásának hatására az előrejelzési pontosságra.

6.1.2 Állapotfelügyelet és karbantartás

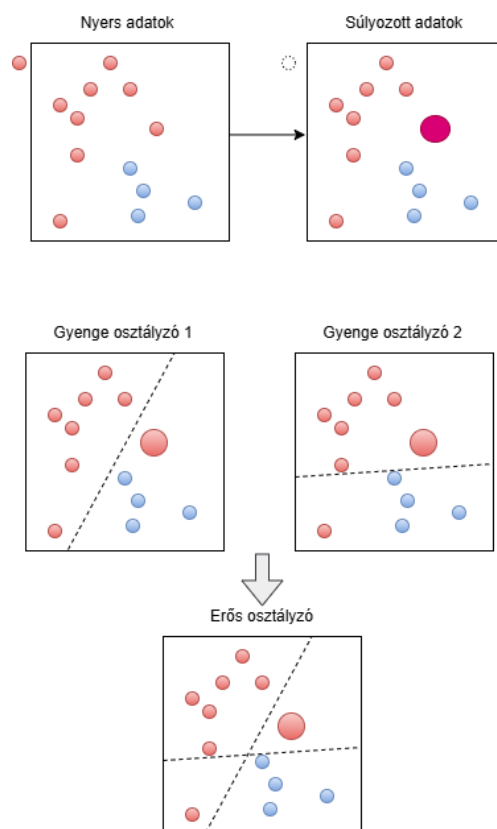
Az adatvezérelt technológiák és a gépi tanulás algoritmusai egyre jelentősebb szerepet töltenek be az ipari rendszerek állapotfigyelésében és karbantartási folyamataiban. Az intelligens alkalmazások fejlődése lehetővé tette, hogy hatalmas mennyiségű érzékelőadat gyűljön össze valós időben, melyek elemzése révén előrejelezhetőek a gépek meghibásodásai és

optimalizálhatóak a karbantartási tevékenységek. Az AdaBoost algoritmus, mint egy népszerű boosting-alapú gépi tanulási technika, ígéretes eszköz az ilyen típusú problémák megoldásában. A tézisfűzetben közölt vizsgálatok a hibák és meghibásodások osztályozásában foglalják össze az elért eredményeket.

Az adatvezérelt az algoritmusok teljesítménye jelentősen függ az adatok jellemzőitől és a predikciós problémák természetétől. Ezért a vizsgálatok kiterjedtek az AdaBoost algoritmus hatékonyságának és korlátaira valós ipari szenzoradatok feldolgozása mellett, különösen az előrejelzések pontosságára és az algoritmus érzékenységére a bemeneti adatok jellemzőire.

6.2 Az AdaBoost algoritmus

Az AdaBoost algoritmus tehát egy olyan gépi tanulási technika, amely a gyenge osztályozók kombinációjával hoz létre egy erős osztályozót. az algoritmus iteratív: minden lépésben egy úgynevezett gyenge osztályozót generál.



7. ábra Az AdaBoost osztályozás lépései [16]

Az első iterációban az összes adatpont azonos súlyozással szerepel. Az osztályozók teljesítményét a hibaarányuk alapján értékeli, amely a helytelenül osztályozott adatpontok súlyozott összegéből adódik. Az iteráció során a hibásan osztályozott adatpontok súlyát megnöveli, hogy a következő lépésben ezekre nagyobb figyelmet fordítson. Eközben a jól osztályozott pontok súlya csökken, így kevésbé befolyásolják a további osztályozók tanítását.

A végső osztályozót a gyenge osztályozók eredményeinek súlyozott „szavazása” alkotja meg. Ez az erős osztályozó képes pontosabban kezelni azokat a mintákat is, amelyek gyenge osztályozóval nehezen különíthetők el.

Az algoritmus pszeudo kódja a következő:

Input:

X: tanító adatpontok $\{x_1, x_2, \dots, x_n\}$

Y: címkék $\{y_1, y_2, \dots, y_n\}$, ahol $y_i \in \{-1, 1\}$

T: iterációk száma (gyenge osztályozók száma)

Output:

H(x): a végső erős osztályozó

1. Inicializáld a súlyeloszlást D_0 minden adatpontra:

$$D_0(i) = 1 / n, \text{ ahol } i = 1, 2, \dots, n$$

2. Iterációk (t = 1-től T-ig):

2.1. Taníts egy gyenge osztályozót $h_t(x)$ a D_t súlyozott adathalmazon.

2.2. Számítsd ki a $h_t(x)$ hibáját:

$$\epsilon_t = \sum D_t(i) * I(h_t(x_i) \neq y_i),$$

ahol I a hibaindikátor (1, ha hiba, különben 0).

2.3 Számítsd ki az α_t frissítési súlyt:

$$\alpha_t = 0.5 * \ln((1 - \epsilon_t) / \epsilon_t)$$

2.4 Frissítsd a súlyeloszlást:

$$D_{t+1}(i) = D_t(i) * \exp(-\alpha_t * y_i * h_t(x_i)) \text{ úgy, hogy a normalizálás után, } \sum D_{t+1}(i) = 1.$$

3. Állítsd össze a végső osztályozót:

$$H(x) = \text{sign}(\sum (\alpha_t * h_t(x)))$$

4. algoritmus Az AdaBoost algoritmikus lépései

6.3 Élettartam tervező és meghibásodás előrejelző komplex döntéstámogató rendszer, facility management szolgáltatás kialakításához

A 2020-1.1.2-PIACI-KFI-2020-00147 számú „OmegaSys – Élettartam tervező és meghibásodás előrejelző komplex döntéstámogató rendszer, facility management szolgáltatás kialakításához” című projektben kidolgoztam egy adatvezérelt predikciós modellt, amely az AdaBoost algoritmus alkalmazásán alapul. A modell célja az volt, hogy az érzékelők által gyűjtött adatok elemzése alapján előre jelezzem a gépek meghibásodásait, ezzel támogatva a feltételes alapú karbantartási döntéseket. Az érzékelő adatokból kiválasztottam azokat a jellemzőket, amelyek kapcsolatban állhatnak a hibajelenségekkel, majd egy múltbeli és jövőbeli adatokra osztott adatbázison teszteltem a predikciós képességeket. A modell teljesítményét különböző mutatókkal értékeltem, és az eredmények megmutatták, hogy az algoritmus

bizonyos típusú adatokkal, például egyensúlyhiányos adatkészletekkel, gyengén teljesít. Ez rámutatott arra, hogy az algoritmus hatékonysága nagymértékben függ az adatok jellemzőitől.

A kutatáshoz kapcsolódóan hajtóművek meghibásodásait is vizsgáltam, különös tekintettel a telemetriai adatok és a meghibásodások közötti kapcsolatra, hogy az adatvezérelt modellek segítségével pontos(abb) előrejelzéseket lehessen készíteni a várható hibák időpontjáról és valószínűségéről. A célom az volt, hogy a prediktív algoritmusok, mint az AdaBoost, alkalmazásával támogassam a feltételes alapú karbantartási stratégiákat, és csökkentsem a nem tervezett leállások és karbantartási költségek gyakoriságát.

A projekt eredményeképpen kifejlesztésre került egy komplex döntéstámogató rendszer.

6.4 Tézisek

T6.1 A gépek működési paramétereinek elemzése és a gépi tanulási algoritmusok közé tartozó AdaBoost alkalmazása lehetővé teszi a gépek meghibásodásainak előrejelzését, ezzel támogatva az optimális ütemezésű karbantartási stratégiákat. Az AdaBoost algoritmus képes a hibás és normál működés osztályozására, alkalmazása kimutathatóan javítja a potenciális hibák előrejelzését, azonban teljesítménye az adathalmazok egyensúlytalanságától és a minták jellemzőitől jelentősen függ. A modell lehetővé teszi az üzemeltetési kockázatok pontosabb elemzését és jelentős csökkentését, miközben az éves karbantartási költségek várható értéke is mérséklődik az adaptív, valós idejű predikciós algoritmus révén. Az OmegaSys rendszer által kínált új szolgáltatáscsomag jelentős versenyelőnyt biztosít a létesítmény-menedzsment szolgáltatások piacán, különösen az irodaházak, sportlétesítmények, rendészeti és katonai létesítmények üzemeltetésében.

T6.2 A telemetriai adatok és a meghibásodások közötti kapcsolat mélyreható elemzése lehetőséget teremt a komplex gépek, gépegységek állapotának pontosabb monitorozására. Az adatvezérelt megközelítés alkalmazása révén nemcsak a hibák előfordulási valószínűsége, hanem azok várható időpontja is előre jelezhető. A prediktív karbantartási stratégia sikeres alkalmazása ezzel nemcsak a működési biztonságot növeli, hanem hozzájárul a berendezések élettartamának meghosszabbításához és az erőforrás-felhasználás optimalizálásához is.

6.5 A tézishoz kapcsolódó publikációk

[P6.1] Hornyák, O., Iantovics, L. B., "AdaBoost Algorithm Could Lead to Weak Results for Data with Certain Characteristics," MATHEMATICS, vol. 11, no. 8, paper 1801, pp. 1–24, 2023. DOI: 10.3390/math11081801. Available: <https://www.mdpi.com/2227-7390/11/8/1801>. [Indexed in: WoS, Scopus, SJR Q2].

[P6.2] Hornyák, O., "Data-Driven Engine Health Monitoring with AI," ENGINEERING PROCEEDINGS, vol. 79, no. 1, paper 39, 2024. SJR Q3

[P6.3] Hornyák, O., "From Data to Decisions: AdaBoost-based Approach for Remaining Useful Life Estimation," in Proceedings of the 2024 25th International Carpathian Control Conference (ICCC), IEEE, 2024, pp. 1–7. ISBN: 9798350350708; 9798350350692. DOI: 10.1109/ICCC62069.2024.10569647.

[P6.4] Hornyák, O., "An Overview on Evaluation Methods of Sequence Prediction Problems" LECTURE NOTES IN NETWORKS AND SYSTEMS, vol. 929, pp. 427–440, 2024. DOI: 10.1007/978-3-031-54674-7_32. SJR: Q4

[P6.5] Hornyák, O., "AdaBoost algoritmus alkalmazhatóságának vizsgálata," XXIV. Energetika-Elektrotechnika – ENELKO és XXXIII. Számítástechnika és Oktatás – SzámOkt Multi-konferencia, 2023, pp. 134–139.

6.6 Felhasznált irodalom

- [1] Ahmad, R., & Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, 63, 135–149.
- [2] Alanne, K., & Sierla, S. (2022). An overview of machine learning applications for smart buildings. *Sustainable Cities and Society*, 76, 103445.
- [3] BS EN 13306. (2017). *Maintenance-Maintenance Terminology*. British Standards Institution: London, UK.
- [4] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861–874.
- [5] Ferreira, J. M., Pires, I. M., Marques, G., Garcia, N. M., Zdravevski, E., Lameski, P., Flórez-Revuelta, F., & Spinsante, S. (2020). Identification of daily activities and environments based on the AdaBoost method using mobile device data: A systematic review. *Electronics*, 9, 192.
- [6] Fumeo, E., Oneto, L., & Anguita, D. (2015). Condition-based maintenance in railway transportation systems based on big data streaming analysis. *Procedia Computer Science*, 53, 437–446.
- [7] Gentile, C., & Littlestone, N. (1999). The robustness of the p-norm algorithms. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, Santa Cruz, CA, USA, 6–9 July 1999 (pp. 1–11).
- [8] Gouriveau, R., Medjaher, K., & Zerhouni, N. (2016). From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics. John Wiley & Sons: Hoboken, NJ, USA.
- [9] Haq, I. U., Ullah, A., Khan, S. U., Khan, N., Lee, M. Y., Rho, S., & Baik, S. W. (2021). Sequential learning-based energy consumption prediction model for residential and commercial sectors. *Mathematics*, 9, 605.
- [10] Hiruta, T., Uchida, T., Yuda, S., & Umeda, Y. (2019). A design method of data analytics process for condition-based maintenance. *CIRP Annals*, 68, 145–148.

-
- [11] Idé, T. (2018). Collaborative anomaly detection on blockchain from noisy sensor data. In *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, 17–20 November 2018 (pp. 120–127).
- [12] Iantovics, L. B. (2021). Black-box-based mathematical modelling of machine intelligence measuring. *Mathematics*, 9, 681.
- [13] Jo, O., Kim, Y. K., & Kim, J. (2017). Internet of Things for smart railway: Feasibility and applications. *IEEE Internet of Things Journal*, 5, 482–490.
- [14] Kumar, A., Shankar, R., & Thakur, L. S. (2018). A big data-driven sustainable manufacturing framework for condition-based maintenance prediction. *Journal of Computer Science*, 27, 428–439.
- [15] Leon, F., & Gavrilescu, M. (2021). A review of tracking and trajectory prediction methods for autonomous driving. *Mathematics*, 9, 660.
- [16] Misra, S., & Li, H. (2020). Chapter 9 – Noninvasive fracture characterization based on the classification of sonic wave travel times. In S. Misra, H. Li, & J. He (Eds.), *Machine learning for subsurface characterization* (pp. 243–287). Gulf Professional Publishing. <https://doi.org/10.1016/B978-0-12-817736-5.00009-0>
- [17] Muhonen, T. (2015). Standardization of Industrial Internet and IoT (Internet of Things) – Perspective on Condition-Based Maintenance. University of Oulu: Oulu, Finland.
- [18] Oliveira, M., Torgo, L., & Costa, V. S. (2021). Evaluation procedures for forecasting with spatiotemporal data. *Mathematics*, 9, 691.
- [19] Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: A review. *International Journal of Advanced Manufacturing Technology*, 50, 297–313.
- [20] Prajapati, A., Bechtel, J., & Ganesan, S. (2012). Condition-based maintenance: A survey. *Journal of Quality in Maintenance Engineering*, 18, 384–400.
- [21] Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation – A review on statistical data-driven approaches. *European Journal of Operational Research*, 213, 1–14.
- [22] Stefansky, W. (1972). Rejecting outliers in factorial designs. *Technometrics*, 14, 469–479.
- [23] Vijayan, D. S., Rose, A. L., Arvindan, S., Revathy, J., & Amuthadevi, C. (2020). Automation systems in smart buildings: A review. *Journal of Ambient Intelligence and Humanized Computing*, 1–13.

7 Summary

This habilitation thesis presents my main scientific contributions in the field of applied informatics. The research spans several domains, including blockchain-based innovations, automated evaluation of programming tasks in online education, pattern recognition from engineering drawings, advanced scheduling algorithms in production systems, and data-driven predictive maintenance using the AdaBoost algorithm.

The blockchain-related research focuses on open innovation models for smart cities and the application of smart contracts in the automotive industry. It highlights how distributed ledger technologies can support transparency, traceability, and automation in urban management and industrial supply chains.

The second major topic addresses the challenge of automatic grading in Massive Open Online Courses (MOOCs). A four-aspect evaluation model was developed for programming tasks, along with a dual-level asynchronous queuing system, enabling scalable and secure assessment of thousands of submissions in real time.

A significant contribution of the thesis is a patented method for the automatic recognition of selected graphical elements in engineering drawings. The novelty of the solution lies in its dual capability:

- (1) a labelled isomorphic subgraph search algorithm supported by a backtracking-based heuristic enables efficient and precise selection of relevant geometric elements within complex technical diagrams, and
- (2) a custom-developed character recognition module designed specifically for engineering environments allows robust interpretation of annotation texts in arbitrary orientation and scale, including rotated or mirrored symbols frequently used in mechanical drawings.

This hybrid method has been implemented as a multi-user, interactive industrial application, capable of extracting structured data from heterogeneous 2D CAD drawings. It enables the automation of engineering documentation processes, supports digital inventory creation, and facilitates downstream integration with enterprise resource planning (ERP) and product lifecycle management (PLM) systems. Its industrial use demonstrates the commercial relevance of the scientific innovation and its potential to reduce human error and improve operational efficiency in technical data processing.

Research in production scheduling led to the design and application of a discrete bacterial memetic evolutionary algorithm (DBMEA). The algorithm outperformed traditional methods in benchmark scheduling problems, offering practical solutions for optimizing flowshop operations.

Finally, a data-driven predictive model was developed using the AdaBoost algorithm to support condition-based maintenance decisions in facility management. Although the algorithm showed

limitations with imbalanced datasets, the results demonstrated that predictive accuracy and risk reduction can be significantly improved by proper feature selection and adaptive modeling strategies. The work culminated in a decision-support system prototype for monitoring the remaining useful life of machinery.

The scientific results have been published in international journals and conferences and have practical applications in smart city infrastructures, digital education platforms, industrial engineering systems, and maintenance optimization. Together, these contributions demonstrate the potential of intelligent informatics solutions to address real-world challenges through innovation and interdisciplinary research.

8 Köszönetnyilvánítás

Szeretnék köszönetet mondani mindazoknak, akik bármilyen módon hozzájárultak szakmai és személyes fejlődésemhez – legyen szó inspiráló beszélgetésekről, tanácsokról, támogatásról vagy akár apró gesztusokról.

Mély hálával gondolok Erdélyi Ferenc professzor úrra, PhD témavezetőmre, aki szakmai útmutatásával, inspirációjával meghatározó szerepet játszott tudományos pályám elindításában. Kiemelkedő szakértelme és támogatása nemcsak a doktori kutatásaim előrehaladását segítette, hanem hozzájárult ahhoz is, hogy kritikusan szemléljem a tudományos problémákat. Köszönöm az értékes tanácsokat, a közös munkát és az ösztönzést, amelyet a doktoranduszi évek során kaptam, és amely hosszú távon is meghatározó marad számomra.

Szeretném kifejezni őszinte hálámat volt Intézetigazgatónak, Tóth Tibor professzor úrnak, aki nemcsak szakmai fejlődésemet segítette, hanem jelentős mértékben hozzájárult tudományos munkásságom elindulásához. Külön köszönettel tartozom az angol nyelvű lektorálásban nyújtott segítségéért, amely nagyban hozzájárult publikációim minőségéhez és érthetőségéhez. Emellett értékes kapcsolatrendszerének és elkötelezett munkájának köszönhetően számos lehetőség nyílt meg számomra a tudományos közösségekben. Nagyra értékelem támogatását és irányítását, amely meghatározó szerepet játszott kutatói pályafutásom során.

Köszönetet mondok jelenlegi intézetigazgatónak Nehéz Károlynak, aki nemcsak az egyik legsokoldalúbb informatikus, akit ismerek, hanem hosszú évek óta jó barátom is. Mindig számíthattam a támogatására, bátorítására és értékes tanácsaira. Nagyra értékelem a támogatását, motiváltsága mindig ösztönzött a fejlődésre, és hosszú beszélgetéseink segítettek eligazodni a tudományos élet kihívásai között. Barátsága és szakmai útmutatása nélkül sokkal nehezebb lett volna elérni azokat az eredményeket, amelyeket ma magaménak tudhatok.

Hálás köszönetemet szeretném kifejezni kollégáimnak és szerzőtársaimnak a közös munkáért, az inspiráló ötleteikért. Külön köszönöm az együttműködéseink során tanúsított elhivatottságukat, türelmüket és konstruktív visszajelzéseiket, amelyek mindig előre mozdították munkánkat. Öröm és megtiszteltetés egy ilyen kiváló közösség tagjának lenni, és remélem, hogy a jövőben is számos értékes tudományos eredményt érhetünk el együtt.

Végül, de nem utolsósorban szeretném kifejezni legmélyebb hálámat családomnak, szeretteimnek, akik mellettem álltak, és feltétel nélküli támogatásukkal segítettek ezen az úton. Köszönöm türelmüket, megértésüket és azt a rengeteg szeretetet, amely erőt adott a kihívásokkal teli időszakokban. Nélkülük ez a munka sokkal nehezebb lett volna, és az elért eredményeim az ő biztatásuk és áldozatkészségük nélkül nem születhettek volna meg. Köszönöm, hogy hittek bennem, és hogy megoszthattam velük ennek az útnak az örömeit és nehézségeit egyaránt. Külön hálás vagyok a gyerekeimnek, akik sokszor alkalmazkodtak a munkával járó elfoglaltságaimhoz, és türelmükkel, szeretetükkel segítettek átvészelni a sűrűbb

időszakokat. Jó érzés volt tudni, hogy ott vannak mellettem, és még ha nem is mindig értették pontosan, min dolgozom, az érdeklődésük és jelenlétük nagyon sokat jelentett.

Miskolc, 2025. október 08.

Dr. Hornyák Olivér