



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
TÁVKÖZLÉSI ÉS MÉDIAINFORMATIKAI TANSZÉK

Korszerű megközelítések az evolúciós algoritmusok alkalmazási módszereiben

Tézisfüzet

Balázs Krisztián
műszaki informatikus

Témavezető:
Dr. Kóczy T. László
egyetemi tanár, az MTA doktora

Budapest, 2013

1. Bevezetés és kutatási célok

Az úgynevezett *lágyszámítási* (más néven *számítási intelligencia*) módszerek a XX. század második felében jöttek létre. Hatékonyságban felülmúlják a korábbi eljárásokat nagybonyolultságú, de ugyanakkor szuboptimális megoldásokat elfogadó problématerületeken. Ennek oka az, hogy ezek a technikák viszonylag alacsony idő- és tárkomplexitással oldják meg a problémákat, illetve alkalmazhatók olyan esetekben is, amikor a probléma analitikus leírása nem, vagy csak részben ismert, illetve amikor a területről szerzhető tudás bizonytalan. Ezeknek az előnyös tulajdonságoknak az árát a megoldás pontatlanságában, szuboptimalitásában kell megfizetni. Alkalmazhatóságuk így olyan problémákra korlátozódik, melyekben a hatékonyság, a gyorsaság fontos szempont, ellenben elfogadható némi pontosságbeli hiányosság.

A lágyszámítási módszerek három fő nagy ágát az evolúciós számítási technikák, a fuzzy rendszerek, illetve a neurális hálózatok jelentik. Bár a fenti tulajdonságokkal mind rendelkeznek, lényeges különbség van közöttük.

Az evolúciós számítási technikák sztochasztikus optimalizáló eljárások, melyek a természetben megfigyelhető különböző evolúciós folyamatok mechanizmusával analóg módon törekednek az egyre kedvezőbb megoldások elérésére (egyre jobb megoldásokat reprezentáló „egyedek” létrehozására), mint ahogyan az az élővilágban is megfigyelhető fajok, egyedek közötti versengésben zajlik.

A fuzzy rendszerek, valamint a neurális hálózatok jó modellezőképességűek. Képesek olyan rendszereket modellezni, melyek szerkezetére nézve kezdetben semmilyen (*feketedoboz probléma*), vagy hiányos ismeretek állnak rendelkezésre (*szürkedomboz probléma*), viszont ismertek, vagy megismerhetők egyes bemenetekre adott válaszok.

Nagy különbség e két lágyszámítási megközelítés között az, hogy amíg a neurális hálózatok a problémáról nyert tudást az elemei közötti összeköttetésekben lévő súlyokban hordozzák nehezen kinyerhető és még nehezebben, vagy egyáltalán nem interpretálható formában, addig a (megfelelő körülményekkel létrehozott) fuzzy rendszerek eleve olyan szabálybázis építésével halmozzák fel a tudást, ahol minden információ explicit módon, interpretálható formában van jelen. Ez nagy előny a fuzzy rendszerek oldalán.

Mivel a szóban forgó lágyszámítási megközelítések (így ezen belül az evolúciós technikák) mindössze fél évszázados múltra tekintenek vissza, mind az elméletük, mind pedig a gyakorlati alkalmazásuk magától értetődő módon számos megoldatlan kérdést tartalmaz, még annak ellenére is, hogy ezek a módszerek kedvező tulajdonságaiknak köszönhetően intenzíven kutatottak, és egyre széleskörűbben alkalmazottak.

Az értekezés fő célja olyan új megközelítéseket javasolni evolúciós algoritmusok alkalmazásához, amelyek segítségével egyes optimalizálási problématerületeken magasabb minőségű megoldások elérésére nyílik lehetőség alacsonyabb számítási komplexitás mellett, mint az irodalomból ismert megközelítések esetében. Ahol értelmezhető az adott kontextusban, ott a másodlagos cél a megoldás minőségét meghatározó különböző minőségi mérőszámok közötti skálázás lehetőségének megteremtése.

A javasolt megközelítések újszerűsége a következőkben jut kifejezésre:

- egységes reprezentációk és operátorok,

- klasszikus és nem klasszikus optimalizáló és modellező technikákat felhasználó hibrid módszerek,
- az aktuálisan hatékonyan alkalmazható evolúciós algoritmusok adaptív kiválasztása.

Az alkalmazott technikák hatékonyságának alátámasztására mindenhol átfogó kísérleti, és ahol lehetséges, formális vizsgálatok szolgálnak.

Kutatási eredményeim röviden a következők szerint fogalmazhatók meg:

1. Sikeresen hibridizáltam evolúciós algoritmusokat más élvonalbeli (*state-of-the-art*) technikákkal permutáció alapú problématerületeken (1. téziscsoport).
2. Evolúciós algoritmusokat és az evolúciós módszerekben fuzzy modellezést alkalmaztam minőségi jellemzők feljavítása céljából (2. téziscsoport).
3. Evolúciós módszereket alkalmaztam különféle újszerű módokon különböző fuzzy szabályalapú tudáskinyerő architektúrákban (3. téziscsoport).
4. Javaslatot tettem interpretálható fuzzy rendszerek létrehozásához egy jelentést megőrző megközelítésre, ehhez kapcsolódóan pedig egy keresési tér szűkítő technikára (4. téziscsoport).
5. Ütemező eljárásokat vezettem be optimalizálási folyamatok során különböző optimalizáló algoritmusok közötti váltásokra (5. téziscsoport).

Az értekezésben szereplő új módszereket, amelyek magukba foglalják mind az általános megközelítéseket, mind pedig a konkrét algoritmusjavaslatokat, az 1. táblázat foglalja össze rendre a 3. szakasz egyes részeinek téziscsoportjai szerint. Az összefoglalás leírja az újonnan javasolt módszerek alkalmazási területeit és az általuk elérhető továbbfejlesztés lehetőségeket.

1. táblázat. Az újonnan javasolt módszerek összefoglalása téziscsoportonként.

	Alkalmazási terület	Konkrét problémaosztály	Optimalizált tulajdonság	Javulás típusa
1. téziscsoport	kombinatorikus	permutáció alapú optimalizálás	átfutási idő	teljesítmény
2. téziscsoport	kombinatorikus	ütemezés / hibalokalizálás	modellezőképesség / megbízhatóság	minőség
3. téziscsoport	folytonos	fuzzy tanulás	pontosság	teljesítmény
4. téziscsoport	folytonos	fuzzy tanulás	interpretálhatóság	minőség
5. téziscsoport	mindkettőre alkalmazható	mindegyikre alkalmazható	mindegyikre alkalmazható	mindkettőre alkalmazható

2. Módszertan

Annak érdekében, hogy elérjem a fent leírt célokat, kutatásaim során négy általam felállított gondolkodási mintát követtem. Ezek útmutatóként szolgáló általános ötletek voltak a meglévő algoritmusok és megközelítések továbbfejlesztéséhez. E minták konkrét esetekre történő adaptációja vezetett a különböző problémaosztályokra, és speciálisan konkrét feladatokra vonatkozó alkalmazási módszerek kifejlesztéséhez. Az általam használt gondolkodási minták a következők voltak:

- hatékony reprezentációk kifejlesztése (3.1., 3.2. és 3.3. alszakaszok),
- technikák és megközelítések hibridizálása (3.1., 3.2., 3.3. és 3.5. alszakaszok),
- ésszerű megszorítások bevezetése (3.3. és 3.4. alszakaszok),
- továbbfejlesztési lehetőségek intuitív megtalálása (3.4. alszakasz).

Az újonnan létrehozott technikák és megközelítések alkalmazhatósága és versenyképessége több lépcsőn keresztül került ellenőrzésre:

- heurisztikus érvelések (minden eredmény esetén),
- formális vizsgálat (ahol alkalmazható),
- kísérleti vizsgálat (minden eredmény esetén).

3. Új eredmények

Ebben a szakaszban az új kutatási eredményeim kerülnek röviden bemutatásra.

3.1. Egy- és többszálú hibrid megközelítések permutáció alapú ütemezési problémákra

1. téziscsoport. *Új egyedreprezentációs technikára tettem javaslatot, továbbá két operátorcsoportot javasoltam két különböző kódolási megközelítéshez, és különböző hibridizációs módszereket dolgoztam ki, melyeket hatékonyság szempontjából összehasonlítottam permutáció alapú ütemezési feladatok megoldására alkalmas egy- és többszálú algoritmusok esetén.*

Kapcsolódó publikációk: [J1], [C7], [C8].

3.1.1. Új kódolási módszerek és evolúciós operátorok permutáció alapú ütemezési problémákra

Két egyedreprezentáció típust (azaz kódolási módszert) alkalmaztam kromoszóma-alapú evolúciós technikákban permutáció alapú ütemezési problémák megoldásához. A permutáció alapú ütemezési problémák olyan optimalizálási feladatok, amelyek esetén a keresési tér adott méretű permutációk halmaza.

Az első kódolási módszer magukon a permutációkon alapszik, így a megfelelő evolúciós operátorok a permutációk elemeit módosítják közvetlen módon.

A második kódolási módszer egy általam javasolt, közvetett, valós érték alapú kódolási megközelítés, amely a numerikus optimalizálási feladatokra alkalmazott reprezentációknak egy nyilvánvaló kiterjesztése. Bár az operátorok ez esetben a valós értékű vektorok (tömbök) elemeit módosítják, – mivel a célfüggvény permutációk felett definiált, a kromoszómák valójában permutációkat reprezentálnak – valamilyen módon szükséges a valós értékű vektorokat permutációkká konvertálni. Erre az értékek sorrendezését javasoltam.

Az időkomplexitási költségek csökkentése érdekében a kromoszómák „tükrözhető” az egyedeken belül egy olyan módszerrel, amely az evolúciós operátorok által létrehozott módosításokat és az egyedek kiértékelését egyszerűbben elvégezhetővé teszi.

Mind a permutáció alapú, mind pedig a valós érték alapú kódolásokhoz három „atomi operátort” (mutáció, génátadás és lokális keresés) javasoltam, amelyekből evolúciós operátorok hozhatók létre különböző kromoszóma-alapú evolúciós algoritmusok számára.

1.1. tézis. *Új valós érték alapú kódolási módszerre tettem javaslatot permutáció alapú ütemezési feladatok számára, továbbá evolúciós operátorokat javasoltam permutáció alapú és az új valós érték alapú egyedreprezentációs technikákhoz.*

Kapcsolódó publikáció: [J1].

3.1.2. Az új technikák integrálása a PFSP problémára alkalmazott evolúciós módszerekbe

Az újonnan javasolt megközelítéseket Genetikus Algoritmus (Genetic Algorithm, GA) [9], Bakteriális Evolúciós Algoritmus (Bacterial Evolutionary Algorithm, BEA) [17] és Részecske-sereg Optimalizálás (Particle Swarm Optimization, PSO) [11], illetve memetikus (lokális keresést tartalmazó) változataik esetén alkalmaztam az úgynevezett Permutation Flow Shop Problem (PFSP) [20] feladat megoldására. A probléma során adottak munkadarabok és gépek. Minden munkadarabot minden gépen szükséges megmunkálni egymás után. A gépek egy sorban vannak elhelyezve, és egy gép egyszerre egyetlen munkadarabot tud megmunkálni, vagyis a munkadarabok megmunkálása „pipeline-szerű”. Szintén adott egy $n \times m$ -es megmunkálási időket leíró mátrix, amely azt definiálja minden munkadarab-gép párra, hogy az egyes munkadarabok mennyi időt töltenek a különböző gépeken. Egy adott munkadarab akkor munkálható meg egy gépen, ha a gép szabad (az előző munkadarab már lekerült róla), és a szóban forgó munkadarab már megmunkálásra került az előző gépen.

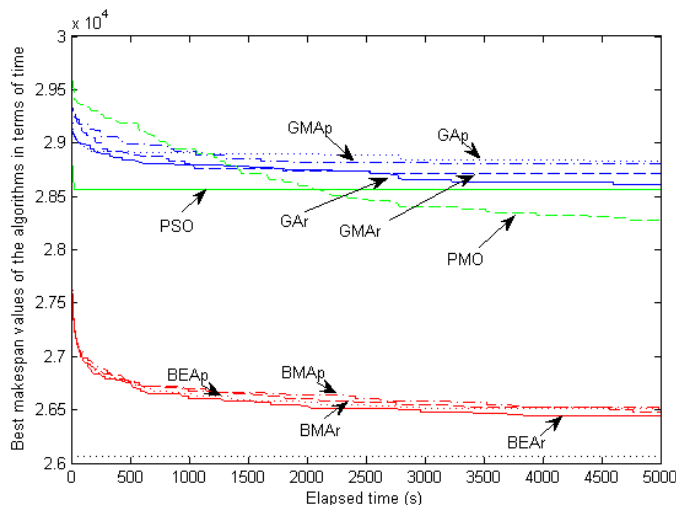
A feladat megtalálni a munkadarabok egy olyan permutációját (sorrendjét), amely esetén az összes munkadarab valamennyi gépen vett összes megmunkálási ideje (vagyis az úgynevezett *makespan* érték) minimális.

Vizsgálataim során a GA, BEA és PSO evolúciós algoritmusok által használt különböző evolúciós operátorokat a három újonnan javasolt atomi operátorra veztettem vissza.

Szimulációs futtatásokat hajtottam végre a javasolt megközelítések és a létrehozott algoritmusok kiértékelésének és összehasonlításának érdekében. Először az új módszerek egymással történő összehasonlítását végeztem el, majd a legjobbat összevettem más heurisztikákkal: a széles körben ismert Iterált Mohó (IG) heurisztikával, illetve memetikus változatával (IGLS) [20] és egy genetikus algoritmus alapú memetikus módszerrel (MA) [19], amely például az IG technikával kombináltan is (MA+MIG módszer) alkalmazásra került többprocesszoros rendszerekben.

E vizsgálatok érdekében egy tucat problémát használtam fel az irodalomból ismert Taillard-féle benchmark-halmazból [22], minden elérhető problémaméretből egy példányt.

Például a futási idő függvényében az új technikák legjobb egyedei célfüggvény-értékeinek a számtani közepét a benchmark-halmaz legnehezebb problémáján az 1. ábra jeleníti meg. A vízszintes tengely az eltelt számítási időt mutatja másodpercekben, amíg a függőleges tengely az adott időben a legjobb egyedek makespan értékeit.



1. ábra. Az 500x20-as méretű problémára (ta111) adódott eredmények.

Mivel a BMAr, a valós érték alapú Bakteriális Memetikus Algoritmus (Bacterial Memetic Algorithm, BMA) [4] adódott a legjobb hatékonyságú algoritmusnak, ez a technika szerepelt a további vizsgálatokban: ez a módszer került összehasonlításra az Iterált Mohó heurisztikával és a genetikus algoritmus alapú memetikus módszerrel.

Bár a legjobb általam létrehozott módszer hatékonyabbnak bizonyult, mint a többprocesszoros rendszerekben alkalmazott genetikus algoritmus alapú memetikus technika, az élvonalbeli Iterált Mohó módszer felülmúlta azt.

1.2. tézis. A két kódolási módszert és a javasolt evolúciós operátorokat Genetikus Algoritmusba (Genetic Algorithm, GA), Bakteriális Evolúciós Algoritmusba (Bacterial Evolutionary Algorithm, BEA) és Részecske-sereg Optimalizálásba (Particle Swarm Optimization, PSO), valamint ezek memetikus változataiba integráltam, továbbá az így létrehozott technikákat a Permutation Flow Shop Problem (PFSP) feladatra alkalmaztam. A létrehozott algoritmusokat hatékonyság szempontjából összehasonlítottam és kiértékeltem az úgynevezett Taillard-féle problémahalmazon. Szimulációs vizsgálatokkal igazoltam, hogy az alkalmazott technikák között a valós kódolású Bakteriális Memetikus Algoritmus (BMAr) rendelkezik a legkedvezőbb konvergencia-tulajdonságokkal. Ezeken felül hatékonyság szempontjából összehasonlítottam

a legjobb általam létrehozott algoritmust state-of-the-art technikákkal, és kísérletileg igazoltam, hogy a vizsgált módszerek közül az Iterált Mohó (Iterated Greedy, IG) heurisztika rendelkezik a legkedvezőbb konvergencia-tulajdonságokkal.

Kapcsolódó publikáció: [J1].

3.1.3. Új hibridizációs megközelítések és a Bakteriális Iterált Mohó hibrid algoritmus a PFSP problémára

Bár még a legjobb általam létrehozott kromoszóma-alapú evolúciós algoritmus, nevezetesen a Bakteriális Memetikus Algoritmus sem volt képes olyan jó teljesítményt nyújtani, mint a PFSP feladatra létező legjobb módszerek, például a jól ismert Iterált Mohó heurisztika, felmerült az ötlet, hogy ezen evolúciós algoritmus és az IG technikáknak valamilyen kombinációja gyümölcsöző algoritmusokat eredményezhet. Az ötlet mögött meghúzódó okok egyike az, hogy többszálú IG módszerek esetén az evolúciós algoritmusokkal történő kombináció képes volt feljavítani a teljesítőképességet, amint ezt a genetikusan alapú memetikus technikát (MA+MIG) alkalmazó párhuzamosított IG eljárást célzó vizsgálatok mutatják (például [19]).

Az említett hibrid heurisztikában szintén érdemes megpróbálkozni a genetikusan alapú memetikus módszer Bakteriális Memetikus Algoritmussal történő helyettesítésével, amely hatékonyabbnak tűnik a PFSP feladatra, és amely jobb tulajdonságokat mutat más optimalizálási területeken is.

Ezt az ötletet általánosítva javasoltam megközelítéseket tetszőleges populáció alapú evolúciós algoritmus (Population Based Evolutionary Algorithm, PBEA) és bármely más nem populáció alapú iteratív heurisztika (Non-Population Based Iterated Heuristic, NPBIH) hibridizálására.

Ha csupán egy szálát tekintünk, azaz nincs párhuzamosítás az algoritmusokban, akkor annak érdekében, hogy hibrid heurisztikát kapjunk, a PBEA és az NPBIH kombinálásának egyetlen útja az egyikük másikukba való beágyazása, ugyanis a két módszer egymás után történő végrehajtása nem igazán nevezhető hibrid módszernek.

Ha az alap-technika a PBEA és a beágyazott heurisztika az NPBIH, akkor az utóbbi egy új evolúciós operátornak tekinthető (csakúgy, mint egy lokális keresés operátor a memetikus algoritmusokban, melyet a tisztán evolúciós technikákhoz adunk), amely előre definiált iterációszámig fut le a PBEA minden egyes generációjában a megoldásjelöltek összes egyedére.

Az optimalizálási folyamat kezdetén, ha a kezdeti populáció véletlen eloszlás szerint definiált, a kezdeti egyedek rossz minőségűek lesznek. Ha a kezdeti megoldásjelöltek egy determinisztikus kezdeti heurisztika (Deterministic Initial Heuristic, DIH) által lennének generálva (például az úgynevezett NEH módszer [18] az IG esetében), akkor az egyedek sokkal jobbak volnának, azonban mindegyikük azonos lenne, így a populáció diverzitása zérussá válna, amely feltétlenül elkerülendő. Ezért a kezdeti populáció a DIH egy sztochasztikus változatával generálandó (például az általam javasolt Sztochasztikus NEH heurisztika) annak érdekében, hogy egy megfelelő populációdiverzitás adódjon ígéretes kezdeti megoldásjelöltek mellett.

Amennyiben az alap-módszer egy NPBIH és a beágyazott algoritmus egy PBEA, akkor a helyzet különbözik az előzőtől, mert az NPBIH-k csupán egy megoldásjelölttel foglalkoznak egyszerre, amíg a PBEA-k a megoldások egész populációját tartják karban. Tehát a beágyazási ponton egy sereg

egyedet szükséges származtatni egyetlen megoldásjelöltből. Ez azt jelenti, hogy a beágyazási pont az alap-algoritmus olyan pontja kell legyen, ahol az alap-heurisztika könnyen mellékutakra terelhető, azaz szétágaztatható enyhén különböző utakra, ezzel létrehozva a PBEA populációjának egyedeit.

Az alapvető beágyazási típusokon túl altípusok is létrehozhatók az algoritmusok egyes részinek (például operátorok) elhagyásával. Az elhagyás gyakran életképes lehet, mert bár az eredményül adódó technika egyszerűbb, és így kisebb javulást okoz iterációról iterációra, az algoritmus gyorsabbá válik, és így több iteráció kerülhet végrehajtásra azonos idő alatt. Tehát a lebutított módszer magasabb hatékonyságú lehet, mint az eredeti.

Az egyszálú esetben elmondottakkal ellentétben, ha több szálát tekintünk, azaz több algoritmus fut párhuzamosan, még ha az eredeti technikákat is hajtjuk végre minden szálon, hibrid módszert kaphatunk az egyes szálon különböző módszerek futtatásával. Egy példa ilyen hibrid többszálú heurisztikára a fent említett MA+MIG módszer, ahol az eredeti algoritmusok futnak az összes szálon, főként IG technikák párhuzamosan, de van egy kitüntetett szál, amelyen a genetikai algoritmus alapú memetikus módszer kerül végrehajtásra.

Hasonló hibrid módszereket javasoltam a szálon a heurisztikák lecserélésével. Tekintve az előzőleg tárgyalt kromoszóma-alapú evolúciós módszereket és egyszálú technikákat, két hibrid megközelítést javasoltam többszálú optimalizálásra. Ezeket a MA+MIG módszerből származtattam rendre a következő változtatásokkal:

1. A genetikai algoritmus alapú memetikus heurisztika és az iterált mohó szálakat cseréljük le rendre PBEA és NPBIH szálakra.
2. A genetikai algoritmus alapú memetikus heurisztika és az iterált mohó szálakat cseréljük le rendre PBEA és hibrid PBEA-NPBIH szálakra.

A fenti lépéseket követve a bakteriális technikákat az iterált mohó módszerekkel hibridizáltam.

A 2. táblázat az IG, a BMAr és a legjobb létrehozott egyszálú bakteriális iterált mohó technika (Single-threaded Bacterial Iterated Greedy, SBIG) által adott eredmények összehasonlítását mutatja 12 benchmark-probléma esetén, amelyek mindegyike különböző méretű (LIMÉ – Legjobb Ismert Makespan Érték¹, Rel. hiba – Relatív hiba).

Amint látható, a benchmark-problémák nehezebbik felén az SBIG adta minden esetben a legjobb makespan értékeket. Ezután nem meglepő, hogy az összes benchmark-problémát figyelembe véve, amíg a BMAr alacsonyabb hatékonyságú volt, mint az IG, kombinációjuk, az új hibrid módszer tisztán felülmúlta mindkét másik technikát eredeti várakozásaimmal összhangban.

1.3. tézis. *Új hibridizációs megközelítésekre tettem javaslatot, melyek alkalmasak populáció alapú evolúciós algoritmusok és egyéb nem populáció alapú iteratív heurisztikák kombinálására.*

A javasolt kombinációs megközelítéseket a Bakteriális Evolúciós Algoritmus (Bacterial Evolutionary Algorithm, BEA) és az egy-, illetve többszálú Iterált Mohó (Iterated Greedy, IG) heurisztika hibridizálására alkalmaztam. Hatékonyság szempontjából összehasonlítottam és

¹<http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>

2. táblázat. Az IG, a BMAr és az SBIG által adott eredmények.

Azonosító	Probléma			BMAr		IG		SBIG	
	Méret	LIMÉ	Időkorlát	Eredmény	Rel. hiba	Eredmény	Rel. hiba	Eredmény	Rel. hiba
ta001	20x5	1278	10	1278	0.00%	1278	0.00%	1278	0.00%
ta011	20x10	1582	10	1586.4	0.28%	1583.2	0.08%	1582.6	0.04%
ta021	20x20	2297	10	2303.8	0.30%	2301.6	0.20%	2304.2	0.31%
ta031	50x5	2724	50	2724	0.00%	2724	0.00%	2724	0.00%
ta041	50x10	2991	50	3045.6	1.83%	3035.2	1.48%	3035.4	1.48%
ta051	50x20	3847	50	3945.6	2.56%	3925	2.03%	3917.8	1.84%
ta061	100x5	5493	200	5493	0.00%	5493	0.00%	5493	0.00%
ta071	100x10	5770	200	5788.2	0.32%	5786.8	0.29%	5781.4	0.20%
ta081	100x20	6202	200	6392.4	3.07%	6350	2.39%	6319.2	1.89%
ta091	200x10	10862	1000	10882.2	0.19%	10888.6	0.24%	10874.6	0.12%
ta101	200x20	11181	1000	11432.4	2.25%	11392.4	1.89%	11373.2	1.72%
ta111	500x20	26059	5000	26476.4	1.60%	26395.6	1.89%	26298.2	0.92%

kiértékelttem a létrehozott hibrid Bakteriális Iterált Mohó algoritmusokat a Taillard-féle problémahalmazon. Hatékonyság szempontjából összehasonlítottam a legjobb létrehozott egy- és többszálú hibrid algoritmusokat az egy- és többszálú IG heurisztikákkal, és kísérletileg igazoltam, hogy az újonnan javasolt technikák rendelkeznek kedvezőbb konvergencia-tulajdonságokkal.

Kapcsolódó publikációk: [C7], [C8].

3.2. Komplex rendszerek minőségi mérőszámainak javítása evolúciós algoritmusok és fuzzy modellezési eszközök segítségével végrehajtott kombinatorikus optimalizálás során

2. téziscsoport. Új evolúciós megközelítésekre tettem javaslatot komplex rendszerek minőségi mérőszámainak kombinatorikus optimalizálás során történő fejlesztése céljából, amelyek részben fuzzy modellezési technikákat alkalmaznak.

Kapcsolódó publikációk: [C6], [C17].

3.2.1. Új kódolási módszerek alkalmazása erőforrás-ütemezési feladatokra

Két reprezentációt alkalmaztam erőforrás-ütemezési feladatokra.

Az első problémamegfogalmazás egy új *naiv reprezentáció*. Ebben az esetben n -eseket használtam az ütemezések reprezentálására. E célból bevezettem két új fogalmat, az *igényegységet* és az *igényt*. Annak érdekében, hogy olyan egyszerű keretalgoritmust hozzak létre, amelyet csak lehet, az optimalizálás *igényekkel* számol erőforrások helyett. Ily módon a részeredmények könnyen összehíthetők, mert műszakok jönnek létre a probléma egyes részeihez tartozó igényekből, és ezek a műszakok rendelődnek hozzá az erőforrásokhoz.

Egy igényegység egy meghatározott munkafázist fejez ki (például vezetés, vagy szünet tartása személyzetütemezési feladatoknál), amelyet egy erőforrással szükséges teljesíteni (például sofőrrel). Egymást követő igényegységek listája az igény. Minden egyes igény egy rész-munkütemezésnek tekinthető. Mivel pontosan egy igény tartozik az ütemezés minden szakaszához, az alkalmazott n-esek $D^{|R|}$ halmazbeli elemek, ahol D az igények halmaza, $|R|$ pedig az ütemezésbeli szakaszok száma. A célfüggvény (költségfüggvény) az n-esek felett definiált, és a feladat a legjobb, azaz annak az n-esnek a megtalálása, amelyik a legkisebb költség mellett teljesíti az adott kényszereket (például üzleti kritériumokat).

Egy kifinomultabb problémamegfogalmazás a jól ismert Halmazfedési Problémán (SCP) alapszik. Ezen NP-nehéz feladat esetén adott egy U halmaz (az *alaphalmaz*), az alaphalmaz részhalmazainak egy C családja (*fedőhalmazok családja*) és egy $c: C \mapsto \mathbb{R}^+$ költségfüggvény, amely definiálja minden egyes fedőhalmaz költségét. A feladat az alaphalmaz valamennyi elemét lefedni fedőhalmazokkal úgy, hogy az alkalmazott fedőhalmazok összköltsége (*teljes költség*) minimális legyen.

A keresztezés, mutáció, bakteriális mutáció és génátadás operátorok megfelelő módosításaira szintén javaslatot tettem naiv reprezentáció esetén.

3.2.2. Üzleti követelmények kezelése evolúciós optimalizálás során fuzzy modellezés alkalmazásával

Az üzleti kritériumok megfogalmazhatók a keresési tér érvényes tartományát definiáló kényszerek formájában, ahol az érvényes tartományon kívüli megoldások üzleti megfontolások miatt nem elfogadhatók. Azonban gyakran az üzleti kritériumok nem definiálhatók egzakt módon, illetve bizonyos mértékű rugalmassággal rendelkeznek. Annak érdekében, hogy a bizonytalanság kezelhetővé, a rugalmasság pedig felhasználhatóvá váljon, az érvényességi tartomány fuzzy halmazként definiálható. Vagyis az üzleti kritériumok megfogalmazhatók fuzzy kényszerek formájában.

Bizonyos kritériumok szigorúbbak, amíg mások engedékenyebbek, így a különböző kényszerek fuzzy volta eltérő, és általánosan igaz, hogy minél fontosabb egy kritérium teljesülése, annál meredekebbek a hozzá tartozó fuzzy kényszerek tagsági függvényeinek oldalai.

Mivel az evolúciós algoritmusokban a fitness-értékek szerepe az egyedek jóságának a meghatározása az általuk az ütemezési feladatra adott megoldás szempontjából, a fitness-függvény irányítja az evolúciót. Ezért a legkézenfekvőbb választás a fitness-függvény alkalmassá tétele annyi cél kezelésére, amennyi csak lehetséges. A teljes költség és az üzleti kritériumokból származtatott fuzzy kényszerek mind belefoglalhatók a fitness-függvénybe. A teljes költség szolgálhat a függvény fő részeként, amíg a kényszersértések mértéke egy hozzáadott büntetőtagként jelenhet meg. A kényszerek fuzzy voltának növelése a kényszereket finomítja, és a fitness-függvényt „simábbá” teszi. Így az evolúciós folyamat érzékenyebbé válik, és jobb megoldást szolgáltatathat.

2.1. tézis. *Javaslatot tettem két új kódolási módszerre erőforrás-ütemezési feladatokhoz és egy fuzzy modellezést alkalmazó megközelítésre üzleti követelmények evolúciós optimalizálás során történő kezeléséhez.*

Kapcsolódó publikáció: [C6].

3.2.3. Az új technikák integrálása a CSP feladatra alkalmazott evolúciós algoritmusokba

A Személyzetütemezési Feladat (Crew Scheduling Problem, CSP) [6] jól ismert optimalizálási feladat, amely számos olyan üzleti folyamat közös absztrakciójaként szolgál, ahol személyzet, vagy más erőforrás hozzárendelését szükséges elvégezni adott feladatokhoz úgy, hogy bizonyos peremfeltételek és informális üzleti kritériumok teljesüljenek. Személyzetütemezési feladatokat elsősorban földi és légi közlekedési vállalatoknál, logisztikában és az ezekhez kapcsolódó területeken találhatunk.

A CSP feladatra mindkét javasolt reprezentációtípust alkalmaztam az ütemezés egyes szakaszait járatokkal az erőforrásokat pedig sofőrökkel azonosítva mindenhol.

A Bakteriális Memetikus Algoritmus alkalmas a naiv problémamegfogalmazás használatára. Ebben az esetben ütemezéseket leíró n -eseket reprezentálnak a kromoszómák (egyedek). A kromoszómákban minden egyes gén egy járhoz tartozik. A bakteriális operátorok felelősek a gének értékeinek cséréjéért, azaz ők társítanak új igényeket a járatokhoz az ütemezésekben annak érdekében, hogy minél jobb megoldást kapjunk.

Hangyakolónai Optimalizálás (Ant Colony Optimization, ACO) [7] esetén az SCP alapú problémamegfogalmazás alkalmazható. Ha az alaphalmaz teljes fedése, az egyes fedőhalmazok és az adott fedés összköltsége rendre az élelemforrással, az egyes részútvonalakkal és az adott útvonalak hosszával reprezentált, akkor a módszer képes létrehozni egy kvázi-minimális költségű fedőhalmaz-rendszert.

2.2. tézis. *A megtervezett kódolási módszereket, evolúciós operátorokat és fuzzy modellezési technikákat a Bakteriális Evolúciós (Bacterial Evolutionary Algorithm, BEA) és a Hangyakolónia Optimalizálás (Ant Colony Optimization, ACO) algoritmusokba integráltam, továbbá az így létrehozott technikákat a Személyzetütemezési Feladatra (Crew Scheduling Problem, CSP) alkalmaztam.*

Kapcsolódó publikáció: [C6].

3.2.4. Rendszermegbízhatóság növelése optikai hálózatokban evolúciós optimalizálást alkalmazó hibalokalizálás révén

Az elmúlt évtizedek során az optikai hálózatok fontos szerepet játszottak a kommunikációs rendszerekben főként nagyon hatékony adattovábbítási képességeiknek köszönhetően. Ennek eredményeként az optikai hálózatok megbízhatósága kritikussá vált mai információs társadalmunkban. Ezt jól tükrözik olyan statisztikák (például [10]), amelyek leírják néhány példát hibaköltségekre vonatkozóan, megmutatván, hogy akár egyetlen kiesett óra is dollármilliók elvesztését okozhatja a kommunikációs hiányból eredően.

Nemrégiben publikált hibalokalizálási megközelítések (például [1], [23] és [3]) úgynevezett *monitorozó utakat* (m -path), *monitorozó köröket* (m -cycle), vagy általánosságban *monitorozó részgráfokat* (m -trail) alkalmaznak. A hibalokalizálás során m -trail-ek halmazát hozzák létre úgy, hogy azzal megoldják az Egyértelmű Hibalokalizálási (Unambiguous Failure Localization, UFL) feladatot [3]. Az UFL feladat során az optikai hálózat gráfját összefüggő élhalmazokkal (m -trail-ekkel) szükséges

lefedni oly módon, hogy egy adott link hibája az m-trail-ek olyan kombinációjának hibáját okozza, amely egyértelműen azonosítja a hibás linket. Ez elérhető, ha az optikai hálózat gráfjában minden él egyedien fedett m-trail-ekkel, azaz nincsenek különböző élekhez tartozó azonos m-trail halmazok. Költséghatékonysági okokból az UFL feladatot a lehető legkevesebb m-trail-lel szükséges megoldani. Ez a követelmény a hibalokalizálási feladatot egy optimalizálási feladattá változtatja.

Az UFL feladatot megoldó legtöbb létező módszer (beleértve a [1], [23] és [3] publikációkban leírtakat) azzal a jelentős hátránnyal rendelkeznek, hogy szükségük van arra a szabadságra, miszerint tetszőlegesen összeállított m-trail-eket használjanak az optikai hálózat gráfjában. Azonban a gráfban az utaknak egy előre definiált halmazának, mint megengedett m-trail-eknek a feltételezése realiztikusabb és költséghatékonyabb, így ésszerűbb, mint megengedni tetszőleges m-trail-ek alkalmazását.

Javaslatot tettem ez utóbbi, megszorított UFL feladatnak egy speciális SCP feladat formájában történő megfogalmazására, és Hangyakolónia Optimalizálást (ACO) használó megoldására.

Szimulációs futtatásokat végeztem annak érdekében, hogy demonstráljam az újonnan javasolt evolúciós megközelítések alkalmazhatóságát a linkhiba-lokalizálási feladatra optikai hálózatokban, továbbá hogy összehasonlítsam azt néhány nemrég publikált módszerrel, nevezetesen a Véletlen Kódcserével (Random Code Swapping, RCS) [23] és a Link-kód Konstrukcióval (Link Code Construction, LCC) [3] technikákkal, amelyek által alkalmazott m-trail-ek halmaza nem előre definiálható.

A szimulációs futások során nyolc benchmark feladatot használtam véletlenszerűen generált összefüggő gráftopológiákon.

A szimulációs eredményekből látható, hogy figyelembe véve az újonnan javasolt módszer RCS és LCC technikákhoz képest megjelenő jelentős hátrányát, nevezetesen, hogy az ACO az egyetlen algoritmus, amelyik az előre definiált m-trail-ek alapján dolgozik, a javasolt új megközelítés ígéretes lehetőségekkel bír.

2.3. tézis. *Javaslatot tettem egy Hangyakolónia Optimalizálás (Ant Colony Optimization, ACO) alapú hibalokalizálási megközelítésre optikai hálózatok megbízhatóságának növelése céljából, amely monitorozó részgráfok előre definiált halmazát képes alkalmazni.*

Először megfogalmaztam a megszorított változatát az Egyértelmű Hibalokalizálási (Unambiguous Failure Localization, UFL) feladatnak Halmazfedési Feladatként (Set Covering Problem, SCP) Egyértelmű Halmazfedési Feladat (Unambiguous Set Covering Problem, USCP) néven, majd az ACO technikát erre a módosított SCP feladatra adaptáltam.

Kapcsolódó publikáció: [C17].

3.3. Új módszerek a fuzzy szabályalapú tudáskinyerésben

3. téziscsoport. *Új módszerek egy családjára tettem javaslatot fuzzy szabályalapú tudáskinyerő rendszerek számára, melyeket hatékonyság szempontjából összehasonlítottam különböző fuzzy architektúrákon.*

Kapcsolódó publikációk: [C16], [C1], [C3], [C2], [C5], [C4], [C18], [C19], [S1], [J6], [C10], [J3], [J4], [C11], [J5].

3.3.1. Evolúciós algoritmusok alkalmazása sűrű és ritka lapos szabálybázisokban

Számos evolúciós algoritmust alkalmaztam sűrű és ritka lapos fuzzy szabályalapú tudáskinyerő rendszerekben.

A lapos szabálybázisokban trapéz-alakú tagsági függvényeket használtam. Az optimalizált paraméterek a szabálybázis szabályaiban található tagsági függvények karakterisztikus pontjai (azaz a trapézok csúcsai) voltak.

Szimulációs futtatásokat hajtottam végre annak érdekében, hogy összehasonlítsam a következtetési technikák és a különböző numerikus optimalizáló módszerek hatékonyságát gépi tanulási problémákon, és hogy felfedjem a sűrű és ritka szabálybázisokat alkalmazó rendszerek tulajdonságai közötti különbségeket. Mamdani- [15], Stabilizált Kóczy-Hirota (Stabilized Kóczy-Hirota, SKH) [24] következtetési technikák és számos kromoszóma-alapú evolúciós optimalizáló algoritmusok minden ésszerű kombinációját alkalmaztam (beleértve a rögzített és a változó egyedhosszú optimalizálási technikákat, azaz a rögzített és a változó szabálybázisméretekkel is dolgozni képes módszereket) három gépi tanulási feladaton (az egydimenziós pH [4], a kétdimenziós ICT [4] és a hatdimenziós Nawa-Furuhashi feladatokon [17]).

Változó egyedhosszokkal dolgozó módszerek esetén új evolúciós operátorokat javasoltam a fuzzy szabályok kezelésére. Minden operátor végrehajtása során az adott egyed kromoszómája véletlenszerűen megnyúlik, összehúzódik, vagy megtartja a méretét. Ha a kromoszóma megnyúlik, akkor egy új szabályt illesztünk két véletlenszerűen választott szabály közé (esetleg az első szabály elé, vagy az utolsó szabály mögé). Ha a kromoszóma összehúzódik, akkor egy véletlenszerűen választott szabályt elhagyunk a meglévők közül.

Mivel a bakteriális alapú algoritmusok jobb teljesítményt nyújtottak, mint a genetikus és a részecske-sereg alapúak, a változó egyedhosszúságú szimulációk esetén a bakteriális alapú technikák módosításait (BEAv, BSDv, BMAv) alkalmaztam.

A szimulációs eredmények alapján a következő tendenciákat figyeltem meg a rögzített egyedhosszú módszereknél mind sűrű, mind pedig ritka szabálybázisokat alkalmazó következtetési technikák esetén:

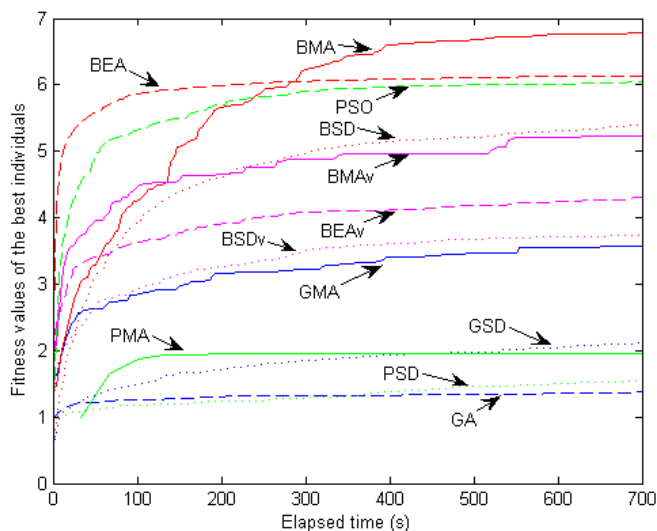
- A bakteriális technikák jobbnak bizonyultak, mint a megfelelő genetikus és részecske-sereg módszerek.
- A PSO mindig jobban teljesített, mint a GA, és olykor viszonylag közel olyan jól, mint a BEA.
- A részecske-sereg módszerek közül a PSO jobb eredményeket adott, mint a gradiens lépéseket használó algoritmusok.
- A PMA (LM lépéseket alkalmazó PSO) tűnt a legrosszabb Levenberg-Marquardt gradiens lépéseket használó technikának.
- A GMA (LM lépéseket alkalmazó GA) rendelkezett a legjobb konvergenciasebességgel a genetikus algoritmusok között.

- Általában elégséges mennyiségű idő elteltével a BMA (LM lépseket alkalmazó BEA) nem volt rosszabb bármely más algoritmusnál. Minél bonyolultabb egy feladat, annál inkább megmutatkozik e technika előnye.

Azt is mondhatnánk, hogy a BMA „lassan, de biztosan” közelít az optimum felé. „Lassan”, mert az esetek döntő többségében az optimalizálási folyamat elején nem rendelkezett a legmagasabb konvergenciasebességgel. „Biztosan”, mert a szimulációk során nem vesztett annyit a hatékonyságából, mint a többi technika.

Például a 2. ábra mutatja az algoritmusok fitnessz-fejlődését az idő függvényében a Nawa-Furuhashi probléma esetén Mamdani-következtetést alkalmazva.

Változó egyedhosszú módszerek esetén, ahogyan a tanulási probléma dimenziószáma nő, az interpolatív következtetési technikáknak a ritka szabálybázisok használatából származó előnye egyre inkább megjelenik. Mivel a szabálybázis mérete büntetve volt a fitnessz-függvényben egy szorzótényező révén, egyszerű problémák esetén előfordult, hogy a változó egyedhosszokat kezelő módszerek magasabb fitnessz-értékű megoldást eredményeztek, mint a rögzített egyedhosszú algoritmusok.



2. ábra. A hatdimenziós tanulási problémára adott eredmények Mamdani-következtetés alapú tanulást alkalmazva.

3.1. tézis. *Evolúciós operátorokra tettem javaslatot változó egyedhosszokat kezelő bakteriális algoritmusok számára a fuzzy szabályok számának fuzzy szabályalapú tudáskinyerő folyamatok közbeni változtatása céljából.*

Genetikus Algoritmust (Genetic Algorithm, GA), Bakteriális Evolúciós Algoritmust (Bacterial Evolutionary Algorithm, BEA), Részecske-sereg Optimalizálást (Particle Swarm Optimization, PSO) és ezen algoritmusok memetikus változatait (GSD, BSD, PSD, GMA, BMA

és PMA), valamint az újonnan létrehozott változó egyedhosszokat kezelő bakteriális algoritmusokat (BEAv, BSDv és BMAv) alkalmaztam sűrű és ritka lapos fuzzy szabályalapú tudáskinyerő rendszerek létrehozásához, ahol Legmeredekebb Lejtő (Steepest Descent, SD) és Levenberg-Marquardt (LM) technikákat használtam lokális keresési módszerek gyanánt. Ennek során az evolúciós algoritmusokban a gének a fuzzy szabályokban lévő tagsági függvények karakterisztikus pontjait reprezentálták. Hatékonyság szempontjából összehasonlítottam és kiértékeltem az így létrejött rendszereket referenciaproblémák széles skáláján, és kísérletileg igazoltam, hogy a vizsgált módszerek között a bakteriális technikák (BEA, BMA, BEAv és BMAv) rendelkeznek a legkedvezőbb konvergencia-tulajdonságokkal. Hatékonyság szempontjából összehasonlítottam a változó és a rögzített egyedhosszokat kezelő módszereket, és kísérletileg igazoltam, hogy bizonyos esetekben az újonnan javasolt megközelítések jobb eredményre vezettek a kisebb szabálybázis-komplexitás révén.

Kapcsolódó publikációk: [C16], [C1], [C3], [C2], [C18], [C19], [S1], [J6], [J4].

3.3.2. Új evolúciós megközelítések hierarchikus és hierarchikus-interpolatív fuzzy architektúrákhoz

Javaslatot tettem egy új sémára hierarchikus fuzzy szabálybázisok egyedekben történő reprezentálásának és módosításának céljából, azaz a szabálybázisok kódolására, és arra, hogy az optimalizálási algoritmusok miként módosítsák a megoldásjelölteket annak érdekében, hogy a tudásbázis minél pontosabb modellt reprezentáljon.

Az optimalizálási folyamatok során a szabályok trapéz-alakú fuzzy halmazai mindig a karakterisztikus pontjaik (azaz a csúcsaik elhelyezkedése) által reprezentáltak.

Az evolúciós programozási technikákban a megoldásjelöltek kifejezésfákkal reprezentálhatók [12], [13]. A kifejezésfák kétféle csúcst tartalmaznak: *nem-terminális (belső)* és *terminális (levél)* csúcsokat.

A belső csúcsok egy vagy több szabály antecedens (feltétel) részeit hordozzák. Minden egyes belső csúcs esetén a figyelembe vett bemeneti dimenziók (a *döntési változók*) száma egy véletlenszerű érték, amely maximuma paraméterezhető. Egy döntési változó csak olyan dimenzió lehet, amely még nem volt döntési változó az adott csúcs őseiben. (Ily módon a kifejezésfa mérete korlátossá válik.) A továbbiakban ezeket a döntési változókat (vagyis amelyek a csúcs egyik őseiben sem jelennek meg) *szabad döntési változóknak* nevezzük. Egy szabály konzekvens (következmény) része az adott csúcs egy gyereke, amely lehet belső csúcs, vagy levél. Így egy belső csúcsban szereplő szabályok számát meghatározza az adott csúcs gyerekeinek a száma. Egy csúcs gyerekeinek a száma egy véletlen érték, amely maximuma, ezzel együtt egy csúcshoz tartozó szabályok számának maximuma paraméterezhető.

Minden levél egyetlen fuzzy halmazt tartalmaz, egy konzekvenshalmazt.

Belső csúcsok és levelek is lehetnek azonos szinteken, ezért metaszabályok és szabályok együtt is megjelenhetnek.

Bevezettem a memetikus programozási technikákat, mint evolúciós programozási algoritmusokat (például [12], [5]), amelyek a következő lokális keresési operátort tartalmazzák.

A lokális keresési lépések előtt minden numerikus paraméter összegyűjtésre kerül egyetlen numerikus vektorba. Ezután gradiens módszereket alkalmazunk ezen a vektoron. A lokális keresési lépésekben a célfüggvény-kiértékelések során a vektor elemeit ideiglenesen beírjuk a kifejezésfa megfelelő helyeire. Ily módon az egyed kiértékelhetővé válik. A lokális keresési fázisok végén, ha a gradiens lépések továbbfejlesztették az egyedet, akkor az új vektor íródik vissza újra a megoldásjelöltbe, máskülönben az eredeti értékek maradnak a csúcsokban. Mivel a numerikus paraméterek bármelyike változhat a lokális keresés során, (a struktúrát leszámítva) az egész egyed optimalizálásra kerül a művelet végrehajtásakor.

3.2. tézis. *Új hierarchikus és hierarchikus-interpolatív szabálybázis reprezentációs technikára tettem javaslatot, amely segítségével hierarchikus és hierarchikus-interpolatív fuzzy rendszerek evolúciós programozási módszerekkel történő optimalizálására nyílik lehetőség. Bevezettem a memetikus programozási módszereket, melyek az evolúciós programozási algoritmusokat kombinálják lokális keresési technikákkal.*

Az új kódolási módszerek kifejezésfákon alapulnak. A belső csúcsok csupán antecedens részekkel rendelkező szabálybázisokat reprezentálnak, amíg a levél csúcsok tartalmazzák a fuzzy szabályok konzekvens halmazait.

Javaslatot tettem lokális keresési lépésként egy olyan technikára, amely összegyűjti a numerikus paramétereket a kifejezésfa csúcsaiból egy numerikus vektorba, és gradiens lépéseket hajt végre e vektor alapján. Ezt a módszert operátorként alkalmaztam evolúciós programozási technikákban, amely megközelítés memetikus programozási algoritmusokat eredményezett.

Kapcsolódó publikációk: [C5], [C4], [C10], [J3], [J4], [C11], [J5].

3.3.3. Az új megközelítések alkalmazása hierarchikus és hierarchikus-interpolatív fuzzy rendszerekben

Az előző pont alapján Genetikus Programozást (GProg) [12], Bakteriális Programozást (BProg) [5] és ezek memetikus változatait alkalmaztam hierarchikus és hierarchikus-interpolatív fuzzy szabálybázisok létrehozására a következő megfontolásokkal.

Mivel egy döntési változó nem lehet olyan dimenzió, amely már szerepelt döntési változóként az adott csúcs valamely ősében, a GProg technikában keresztezés esetén a döntési változókat érintően a következő inkonzisztencia jöhet létre. Tekintsünk két egyedet, amelyeket a keresztezés operátor kombinál. Tegyük fel, hogy egy adott x_i változó található az egyik egyedben a keresztezési pont alatt, és ugyanaz az x_i döntési változó szerepel a másik egyedben a keresztezési pont felett. Ebben az esetben az operátor létrehoz egy olyan kifejezésfát, ahol ugyanaz az x_i döntési változó jelenik meg különböző szinteken, ellentmondva a kódolási sémának (lásd előző pont). Annak érdekében, hogy ezt az inkonzisztenciát feloldjuk, amellet, hogy megtartjuk a részfa struktúráját, a keresztezés során átadott csúcsokban megváltoztatjuk a döntési változókat úgy, hogy az új döntési változók olyan dimenziók legyenek, amelyek még nem voltak döntési változók az ősökben. Ehhez egy új megköötésnek a bevezetése szükséges, nevezetesen, az átadott részfában található döntési változók száma nem lehet

több, mint a befogadó fában a keresztezési pontban jelenlévő szabad döntési változók száma. Így a keresztezési pont megválasztásakor ezt a megkötést figyelembe kell venni.

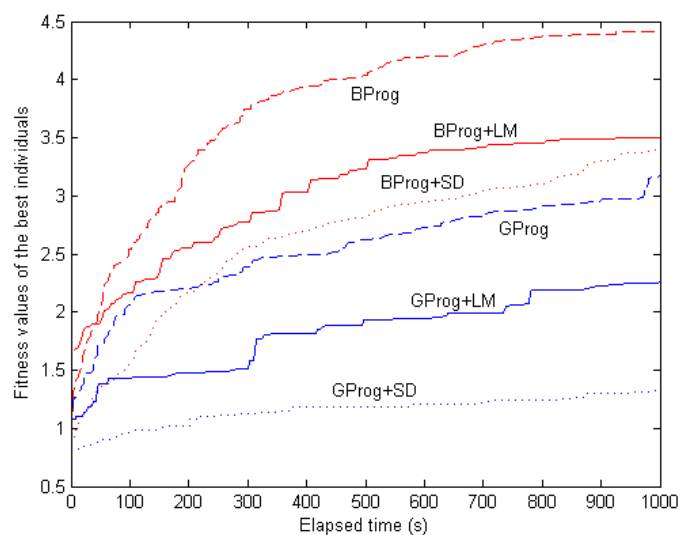
Ugyanez a probléma merül fel génátadás esetén a BProg algoritmusban, ahol az előzővel azonos megoldást alkalmaztam.

Mutáció esetén a GProg módszernél és bakteriális mutáció esetén a BProg technikánál nem adódnak technikai nehézségek; csupán új részfákat, vagy csúcsokat szükséges generálni az előző pontban leírt kódolási szabályok betartása mellett.

Következtető algoritmusok gyanánt a Mamdani-következtetés és a Stabilizált KH-interpoláció módszerek egyszerű kiterjesztéseit alkalmaztam. A kiterjesztések azért szükségesek, mert a szabályok konzekvens részeiben gyerekcsúcsok találhatóak. Mivel a részfák következtetései fuzzy halmazok (speciálisan a levél csúcsok tartalma szintén tekinthető következtetésnek), azok behelyettesíthetők a szülőcsúcsok szabályainak konzekvens részeibe. Mindez rekurzív módon elvégezhető.

Szimulációs futtatásokat végeztem öt különböző gépi tanulási benchmark problémán (Friedman, Daily electricity, Nawa-Furuhashi, Stock prices és Treasury adathalmazok) [2], [17] annak érdekében, hogy demonstráljam a kialakított fuzzy szabályalapú tanuló rendszerek alkalmazhatóságát, továbbá hogy összehasonlítsam őket egymással a tanulás sebességének és a létrehozott tudásbázis pontosságának szempontjából.

A kísérletekben a bakteriális programozási technikák felülmúlták a megfelelő genetikus programozási módszereket, és a BProg algoritmus mutatkozott a legalkalmasabb technikának a hierarchikus és hierarchikus-interpolatív rendszerekkel történő kombinációra (lásd például a Nawa-Furuhashi problémára adott eredményeket a 3. ábrán)



3. ábra. A Nawa-Furuhashi benchmark problémára adódott fitness-értékek.

3.3. tézis. A Genetikus Programozást (*Genetic Programming, GProg*) és Bakteriális Programozást (*Bacterial Programming, BProg*), valamint ezen algoritmusok memetikus változatait (*GProg+SD, BProg+SD, GProg+LM* és *BProg+LM*) az újonnan javasolt kódolási módszerekkel együtt hierarchikus és hierarchikus-interpolatív fuzzy szabályalapú tudáskinyerő rendszerek létrehozására alkalmaztam, ahol Legmeredekebb Lejtő (*Steepest Descent, SD*) és Levenberg-Marquardt (*LM*) technikákat használtam lokális keresési módszerek gyanánt. Hatékonyság szempontjából összehasonlítottam és kiértékeltem az így létrejött rendszereket referenciaproblémák széles skáláján, és kísérletileg igazoltam, hogy a vizsgált módszerek között a bakteriális technikák rendelkeznek a legkedvezőbb konvergencia-tulajdonságokkal.

Kapcsolódó publikációk: [C5], [C4], [C10], [J3], [J4], [C11], [J5].

3.4. Új jelentést megőrző megközelítés fuzzy rendszerek létrehozásához állítható interpretálhatóság-pontosság egyensúllyal

4. téziscsoport. Új jelentést megőrző és paraméterezzhető keresési tér szűkítési technikákra tettem javaslatot fuzzy tudásbázisokat optimalizáló evolúciós algoritmusok számára, lehetővé téve a pontosság és az interpretálhatóság közötti hangolást.

Kapcsolódó publikációk: [C12], [C15], [J2].

3.4.1. Új jelentést megőrző megközelítés interpretálható fuzzy szabálybázisok számára

Az interpretálhatóság és a pontosság közötti kompromisszum szükségessége miatt a konvencionális megközelítések interpretálható fuzzy szabálybázisok létrehozása esetén a következő jelentős hátránnyal rendelkeznek. A tanulási folyamat eredményeként adódó szabálybázistól függően merőben különböző halmazok címkéződhetnek azonos lingvisztikus kifejezésekkel, vagyis az alkalmazott szótár nem állandó a problémák széles skáláján. Még az is előfordulhat, hogy két különböző kontextusban a bemeneti tér partíciói olyan mértékben különböznek, hogy azonos halmazok különböző nyelvi kifejezésekkel címkéződnek.

Azonban még ha az egyes lingvisztikus kifejezésekhez tartozó halmazok egzaktul definiáltak is lennének, és így ha a címkék azonos halmazokat jelölnének minden egyes létrejövő szabálybázisban, azaz ha a szótár állandó lenne, az eredmény interpretációja jelentősen különböző lehetne két különböző ember számára, ugyanis a természetes nyelvek többértelműsége miatt egy természetes nyelvi kifejezés jelentése eltérhet különböző emberek esetén.

Ebből kiindulva tettem javaslatot egy új, személyre szabott megközelítésre interpretálható fuzzy rendszerek létrehozásához. A megközelítés fő gondolata az, hogy a lingvisztikus kifejezéseket azonos értelemben használjunk, mint ahogyan a felhasználó teszi, vagyis hozzunk létre egy *közös szótárat*. Például ha az interpretációt megelőzően a szabály az lenne, hogy „ha az időjárás 30°C, akkor kapcsolj be a légkondicionálást”, az interpretáció a hideg égövben élő számára az kellene legyen, hogy „ha az időjárás nagyon forró, akkor kapcsolj be a légkondicionálást”, amíg a forró égövben élő számára „ha az időjárás eléggé meleg, akkor kapcsolj be a légkondicionálást”.

Mivel az interpretálhatóság azt jelenti, hogy a tudás a felhasználó számára közvetlenül érthető módon kerül megfogalmazásra, az interpretálhatóság követelményei teljesítésének legegyszerűbb

módja az, ha az információt a felhasználó számára ismert reprezentációban tároljuk. Triviálisan ilyen reprezentációk a természetes nyelvek. Viszont ezek problémákat rejtenek magukban a pontatlanságuk miatt. Ha az emberek hallanak, vagy olvasnak valamit természetes nyelven megfogalmazva, jelentést társítanak a hallott, vagy olvasott szöveghez. Azonban amíg valaki egy bizonyos jelentést társít, más eltérő jelentéssel ruházhatja fel ugyanazt, mert a természetes nyelvekben a kifejezéseknek nem létezik egzakt meghatározása.

Annak érdekében, hogy a felhasználóval megegyező módon alkalmazzuk a kifejezéseket, a felhasználót szükséges *meginterjúvolni*. Egy egyszerű interjú lehet az, ha megkérjük a felhasználót, hogy definiálja a mellékneveket fuzzy halmazokként. Azonban feltételezve, hogy a felhasználó egyáltalán nem ismeri a fuzzy halmazokat (amely egy meglehetősen életszerű feltevés), az interjú lefolytatható úgynevezett fuzzy elicitációs technikákkal (lásd például [8]). Ekkor egy egyszerű interjú lehet egy kérdőív. Például hogy megtudjuk, mit ért a felhasználó „forró” alatt, feltehetünk egy kérdéssorozatot: „Mennyire érzi a 15, 20, 25, 30, 35, 40 fokot forrónak (de nem nagyon forrónak) egy 0 – 5 skálán (0 – egyáltalán nem, 5 – teljes mértékben)?” Ezek után a megfelelő fuzzy halmazok könnyedén létrehozhatók.

A lingvisztikus kifejezések nem csak mellékneveket tartalmazhatnak (például „forró”), hanem módosítószókat is (például „kicsit”, vagy „nagyon”). Ezek a módosítószók a módosított melléknevek halmazainak transzformációjaként foghatók fel. Tehát, ha a felhasználót meginterjúvöltük a „hideg”, a „meleg” jelentéseiről és arról, hogy ő hogyan módosítja egy melléknév jelentését, ha azt kombináljuk a „nagyon” módosítószóval, a „nagyon hideg” és a „nagyon meleg” jelentését nem szükséges meginterjúvolni, mert ezek kiszámíthatók a „nagyon” transzformációjának „hideg” és „meleg” halmazokon történő alkalmazásával. Ez komplexitáscsökkenéshez vezethet. (Természetesen a transzformációkat körültekintően kell definiálni egy jól megtervezett interjú alapján, ugyanis például a „nagyon” esetén az egyes halmazok pozitív értékkel való eltolása megfelelő lehet a „meleg” melléknévnél, de biztosan nem alkalmazható a „hideg” melléknévre.)

Az egész eljárás fordítva is elvégezhető. Interjú helyett a felhasználót *megtaníthatjuk*, vagyis a felhasználónak elmondhatjuk, hogy az egyes kifejezések (melléknevek és módosítószók) mit jelentenek.

A felhasználó által definiált lingvisztikus címkék alapján könnyedén létrehozhatunk fuzzy szabályokat és szabálybázisokat. Azonban nem minden így megalkotott szabály fogja teljesíteni az interpretálhatóság feltételeit, és így nem mindegyikük lesz interpretálható (például a konzisztencia hiánya miatt). Ezeket érvénytelen, amíg az interpretálhatósági feltételeket teljesítőket érvényes *interpretálható megoldásoknak* nevezhetjük.

4.1. tézis. *Jelentést megőrző megközelítésre tettem javaslatot interpretálható fuzzy szabálybázisok létrehozásához. A megközelítés fő gondolata az, hogy a lingvisztikus kifejezéseket azonos értelemben használjuk, mint ahogyan a felhasználó teszi, vagyis hozzunk létre egy „közös szótárat” a felhasználóval.*

Annak érdekében, hogy a felhasználóval megegyező módon alkalmazzuk a kifejezéseket, a felhasználót szükséges „meginterjúvolni”. Egy egyszerű interjú lehet az, ha megkérjük a felhasználót, hogy definiálja a mellékneveket fuzzy halmazokként, vagy az interjú lefolytatható úgynevezett fuzzy elicitációs technikákkal.

A lingvisztikus kifejezések nem csak mellékneveket tartalmazhatnak (például „forró”), hanem módosítószókat is (például „kicsit”, vagy „nagyon”). Ezek a módosítószók a módosított melléknevek halmazainak transzformációjaként foghatók fel.

Az egész eljárás fordítva is elvégezhető. Interjú helyett a felhasználót „megtaníthatjuk”, vagyis a felhasználónak elmondhatjuk, hogy az egyes kifejezések (melléknevek és módosítószók) mit jelentenek.

Kapcsolódó publikációk: [C12], [J2].

3.4.2. Új keresési tér szűkítő technika interpretálható fuzzy szabálybázisok létrehozására alkalmazott evolúciós algoritmusok számára

Az interpretált információ véges, de a gyakorlatban leginkább erősen korlátos mennyiségű jellemzővel írható le, ugyanis az ember csupán erősen korlátos számú információegységet képes kezelni. Továbbá az ember nem tud különbséget tenni jelentésben egymáshoz túlságosan közel lévő információegységek között, azaz a megkülönböztethető információ granularitása nem végtelenül kicsi, és ezért a lehetséges megoldások tere korlátos. Így a továbbiakban az interpretálható megoldások halmazát végesnek tekintjük és X_0 -val jelöljük.

Ha a fuzzy rendszer mintákból került létrehozásra felügyelt gépi tanulási technikák alkalmazásával, és a folyamat fő célja az interpretálhatóság biztosítása, akkor a tanulás feladata egy $x_0^* \in X_0$ meghatározása úgy, hogy $\forall x \in X_0 : A(x_0^*) \geq A(x)$, ahol $A(\cdot)$ a (relatív) pontosság mértéke, a hibának egy szigorúan monoton csökkenő függvénye, amely hiba például a kívánt kimenetek és a rendszer kimenetei közötti különbségek alapján számolható. Ez a x_0^* globális keresést végző numerikus optimalizáló algoritmusokkal érhető el megfelelő idő alatt. A tanulási folyamat eredménye a legpontosabb tudásbázis az interpretálható megoldások között. Nyilvánvalóan a hangsúly ez esetben az interpretálhatóságon van.

Jelölje X_∞ az adott szabálybázis paramétervektorainak a legnagyobb figyelembe vehető halmazát. Ha a fuzzy rendszer mintákból került létrehozásra felügyelt gépi tanulási technikák alkalmazásával, és a folyamat fő célja a pontosság biztosítása, akkor a tanulás feladata egy $x_\infty^* \in X_\infty$ meghatározása úgy, hogy $\forall x \in X_\infty : A(x_\infty^*) \geq A(x)$, ahol $A(\cdot)$ a fent leírt (relatív) pontossági függvény. Ez az x_∞^* tetszőleges pontossággal közelíthető globális keresést végző numerikus optimalizáló algoritmusokkal (mint ismeretes, a globális keresési módszerek sztochasztikusan konvergálnak a globális optimumhoz). A tanulási folyamat eredménye a legpontosabb tudásbázis az interpretálhatóság teljes figyelmen kívül hagyása mellett.

Nyilvánvaló az a tény, hogy ha a keresési terek egy egymásba ágyazott $X_0 \subset X_{r_1} \subset X_{r_2} \subset \dots \subset X_{r_n} \subset X_\infty$ sorozatát definiáljuk (ahol az r_i indexek sorozata egy szigorúan monoton növekvő sorozat), akkor pozitív annak a valószínűsége, hogy egy optimális megoldás egy tágabb térben nagyobb pontossággal rendelkezik, mint egy szűkebb tér valamennyi eleme.

Azonban ha $r_i > 0$ és $x_{r_i}^* \notin X_0$ (ahol $x_{r_i}^*$ az optimális megoldás az X_{r_i} halmazban), akkor is adható interpretáció a megoldásra, amennyiben definiálunk egy $\mathcal{J} : X_\infty \mapsto X_0$ interpreter függvényt úgy, hogy $\mathcal{J}(x_{r_i}^*)$ valamilyen módon a „legközelebbi” eleme legyen az X_0 halmaznak az $x_{r_i}^*$ ponthoz, azaz $\forall x \in X_0 : d(x_{r_i}^*, \mathcal{J}(x_{r_i}^*)) \leq d(x_{r_i}^*, x)$, ahol $d : X_\infty \times X_\infty \mapsto \mathbb{R}^+ \cup \{0\}$ egy eltolás invari-

áns metrika. Vagyis egy $x_{r_i}^* \notin X_0$ megoldás interpretációja az $x_{r_i}^*$ ponthoz legközelebbi $x_0 \in X_0$ interpretálható megoldás egy tetszőlegesen rögzített eltolás invariáns távolságfüggvény szerint.

Természetesen definíció szerint $\mathcal{J}(x_\infty^*)$ sosem lehet pontosabb, mint x_0^* , csakúgy, mint ahogyan x_0^* sosem lehet pontosabb, mint x_∞^* .

Ez azt mutatja (intuitív elvárásainknak megfelelően), hogy az interpretálhatóság és a pontosság egymásnak ellentmondó követelmények: ha egy interpretálható tudásbázist hozunk létre, az kevésbé lesz pontos, és ha egy pontosabbat készítünk, akkor várhatóan az interpretációt követően kevésbé pontos megoldást kapunk, mintha az interpretálhatóságot tartottuk volna szem előtt, és a pontosság csupán mellékes szempont lett volna.

Ezen ellentétes megközelítések különböző súlyozásokkal köztes megközelítéseké kombinálhatók, ha mind a nem interpretált tudásbázis pontossága, mind pedig az interpretálté fontos. Ilyen kombinációk létrehozhatók a lehetséges tudásbázisok keresési terének szűkítésével és egymásba ágyazott keresési terek $X_0 \subset X_{r_1} \subset X_{r_2} \subset \dots \subset X_{r_n} \subset X_\infty$ sorozatának definiálásával. Például ha a keresési terek az X_0 elemei körül létrehozott tömör hipergömbök (azaz hipergolyók) uniói, amely hipergolyók azonos sugárral rendelkeznek, továbbá a tágabb és a szűkebb keresési terek csupán a sugár értékében különböznek egymástól.

Meglehetősen kedvező lenne, ha fennállna egy olyan tendencia, miszerint az interpretált megoldás várhatóan pontosabb lenne egy szűkebb keresési tér esetén, mert ekkor a sorozatból egy szűkebb keresési teret választva, bár a nem interpretált tudásbázis pontossága kisebb lenne, az interpretált tudásbázis nagyobb pontosságot mutatna. Tehát, pongyolán fogalmazva, lehetőség nyílna az interpretálhatóság és a pontosság közti egyensúlyozásra a megfelelő keresési tér megválasztásával.

4.2. tézis. *Új keresési tér szűkítési technikára tettem javaslatot evolúciós algoritmusok számára, amellyel interpretálható fuzzy rendszerek hozhatók létre úgy, hogy fennálljon a lehetősége a pontosság és az interpretálhatóság közötti hangolásnak.*

Legyen $X_r := \{x \in X_\infty \mid \exists x_0 \in X_0 : d(x, x_0) < r\} \subseteq \mathbb{R}^n$ a „szűkített keresési tér” egy adott $r \in \mathbb{R}_0^+ \cup \{\infty\}$ sugárra és egy adott d eltolás invariáns metrikára vonatkozóan, ahol X_0 jelöli az interpretálható szabálybázisok halmazát és X_∞ jelöli a lehetséges szabálybázisok halmazát.

Ebben az esetben az egymásba ágyazott keresési terek $X_0 \subset X_{r_1} \subset X_{r_2} \subset \dots \subset X_{r_n} \subset X_\infty$ sorozatának alkalmazásával lehetőség nyílik a pontosság és az interpretálhatóság közötti hangolásra.

Kapcsolódó publikációk: [C12], [C15], [J2].

3.4.3. A szűkítési technika formális és kísérleti vizsgálata

Egy sztochasztikus modell alapján sikeresen igazoltam formális eszközökkel az említett kedvező tulajdonságot.

1. tétel. *Tegyük fel, hogy a keresési terek a fenti módon definiáltak. Ekkor a sorozatból egy szűkebb keresési tér választásával, bár a nem interpretált tudásbázis várható pontossága kisebb, az interpretált tudásbázis várható pontossága nagyobb.*

A következő tétel azt világítja meg, hogy miért javasolt a legközelebbi interpretálható megoldást választani az interpretáció folyamán.

2. tétel. *Tegyük fel, hogy a keresési terek a fenti módon definiáltak. Ekkor tetszőleges x_r^* esetén a legközelebbi interpretálható megoldás adja a legnagyobb várható pontosságot.*

Annak érdekében, hogy demonstráljam a javasolt megközelítések alkalmazhatóságát, és hogy kísérletileg is alátámasszam a tárgyalt kedvező tulajdonság fennállását, szimulációs futtatásokat végeztem.

Ehhez az úgynevezett pH-problémát alkalmaztam. A Mamdani-következtetést és a Stabilizált KH-interpolációt alkalmazó tanulási folyamatok eredményeit rendre a 3. és a 4. táblázat mutatja.

A táblázatokban a *-gal jelölt értékek elméletileg helytelenek, ugyanis az optimális interpretálható megoldás nem lehet rosszabb bármely más interpretálható megoldásnál (vö. például 3. táblázat, Maximum-metrika, 0 sugárérték), továbbá egy keresési tér optimális pontja nem lehet kevésbé pontos, mint bármely részalmazának az optima (vö. például 4. táblázat, Manhattan-metrika, 0.1 sugárérték). Ezek az eredmények az optimalizálási folyamat során elért nem megfelelő kvázi-optimumból adódnak. Ezért ezeket érvénytelen eredményeknek tekintettem, és így figyelmen kívül hagytam a vizsgálat során levont következtetések megfogalmazásakor.

Tisztán látható, hogy \dagger -tel jelölt eredményeket kivéve, azaz 31 alkalommal a 33 érvényes esetből, $MSE(x_{r_i}^*) \geq MSE(x_{r_j}^*)$ és $MSE(\mathcal{J}(x_{r_i}^*)) \leq MSE(\mathcal{J}(x_{r_j}^*))$ fennáll, ha $r_i \leq r_j$.

Mivel a pontosság szigorúan csökkenő függvénye a hibának, ha $r_i \leq r_j$, akkor 31 alkalommal a 33 érvényes esetből $A(x_{r_i}^*) \leq A(x_{r_j}^*)$ és $A(\mathcal{J}(x_{r_i}^*)) \geq A(\mathcal{J}(x_{r_j}^*))$ szintén fennáll.

A kísérleti eredmények alapján kijelenthető, hogy a szimulációs eredmények meggyőzőek, és alátámasztják az interpretálható fuzzy rendszerek létrehozására javasolt megközelítések elméletileg várt tulajdonságait.

3. táblázat. A pH problémára adott eredmények Mamdani-következtetés esetén.

sugár(r)	Manhattan-metrika		Euklideszi-metrika		Maximum-metrika	
	$MSE(x_r^*)$	$MSE(\mathcal{J}(x_r^*))$	$MSE(x_r^*)$	$MSE(\mathcal{J}(x_r^*))$	$MSE(x_r^*)$	$MSE(\mathcal{J}(x_r^*))$
0	$4.71 \cdot 10^{-3}$	$4.71 \cdot 10^{-3}$	$4.71 \cdot 10^{-3}$	$4.71 \cdot 10^{-3}$	* $4.98 \cdot 10^{-3}$	* $4.98 \cdot 10^{-3}$
0.01	$4.06 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$	$3.46 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$	$2.74 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$
0.02	$3.64 \cdot 10^{-3}$	† $2.26 \cdot 10^{-2}$	$2.73 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$	$1.64 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$
0.05	$3.26 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$	$1.82 \cdot 10^{-3}$	$5.31 \cdot 10^{-3}$	$2.81 \cdot 10^{-4}$	† $4.80 \cdot 10^{-2}$
0.1	$1.96 \cdot 10^{-3}$	$1.13 \cdot 10^{-2}$	$5.60 \cdot 10^{-4}$	$1.58 \cdot 10^{-2}$	$6.90 \cdot 10^{-5}$	$2.16 \cdot 10^{-2}$
0.2	$7.40 \cdot 10^{-4}$	$1.58 \cdot 10^{-2}$	$2.17 \cdot 10^{-4}$	$1.82 \cdot 10^{-2}$	* $1.48 \cdot 10^{-4}$	* $2.73 \cdot 10^{-2}$

4.3. tézis. *Formálisan bebizonyítottam, hogy az újonnan javasolt keresési tér szűkítési technika rendelkezik a kívánt hangolási tulajdonságokkal bizonyos feltételek fennállása esetén, vagyis:*

Ha $0 \leq r_1 < r_2 \leq R$ (ahol R a „kritikus sugár”), akkor

$$\mathbb{E}A(\mathcal{J}(x_{r_2}^*)) < \mathbb{E}A(\mathcal{J}(x_{r_1}^*)) \leq \mathbb{E}A(x_{r_1}^*) < \mathbb{E}A(x_{r_2}^*).$$

4. táblázat. A pH problémára adott eredmények Stabilizált KH-interpoláció esetén.

sugár(r)	Manhattan-metrika		Euklideszi-metrika		Maximum-metrika	
	$MSE(x_r^*)$	$MSE(\mathcal{J}(x_r^*))$	$MSE(x_r^*)$	$MSE(\mathcal{J}(x_r^*))$	$MSE(x_r^*)$	$MSE(\mathcal{J}(x_r^*))$
0	$1.42 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$
0.01	$1.40 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.42 \cdot 10^{-2}$	$1.32 \cdot 10^{-2}$	$1.46 \cdot 10^{-2}$
0.02	$1.40 \cdot 10^{-2}$	$1.47 \cdot 10^{-2}$	$1.36 \cdot 10^{-2}$	$1.46 \cdot 10^{-2}$	$1.20 \cdot 10^{-2}$	$1.46 \cdot 10^{-2}$
0.05	$1.12 \cdot 10^{-2}$	$1.55 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$	$1.46 \cdot 10^{-2}$	$8.46 \cdot 10^{-3}$	$1.55 \cdot 10^{-2}$
0.1	* $1.18 \cdot 10^{-2}$	* $1.47 \cdot 10^{-2}$	$9.24 \cdot 10^{-3}$	$1.55 \cdot 10^{-2}$	$7.19 \cdot 10^{-3}$	$1.55 \cdot 10^{-2}$
0.2	$1.02 \cdot 10^{-2}$	$1.69 \cdot 10^{-2}$	$7.90 \cdot 10^{-3}$	$1.55 \cdot 10^{-2}$	$6.23 \cdot 10^{-3}$	$3.12 \cdot 10^{-2}$

A középen elhelyezkedő egyenlőség pontosan akkor áll fenn, ha $r_1 = 0$.

Kísérletileg megerősíttem a formálisan bizonyított hangolási tulajdonságokat referencia-problémák széles skáláján.

Kapcsolódó publikációk: [C15], [J2].

3.5. Optimalizáló algoritmusok adaptív ütemezése

5. téziscsoport. *Új adaptív ütemező sémára tettem javaslatot optimalizálási folyamatokban az optimalizáló algoritmusok közötti adaptív váltások céljából.*

Kapcsolódó publikációk: [C9], [C14], [C13].

3.5.1. Az adaptív ütemezés problémája optimalizálási folyamatokban

Az irodalomból nagyszámú numerikus optimalizálási algoritmus ismert. Jelentős részüket az úgynevezett „orákulum-alapú” (más néven „feketedoboz”) technikák teszik ki, amelyek a célfüggvényt minden egyes iterációban kiértékelik új állapotok előállításához. Ezek a technikák alkalmasak komplex optimalizálási problémák megoldására, ugyanis csupán néhány feltételezéssel élnek az adott problémákkal kapcsolatban, így meglehetősen általános algoritmusok. Egy részüket intuitív módon fejlesztették ki (például a Legmeredekebb Lejtő [21] és a Levenberg-Marquardt [14], [16] módszerek), másik részüket a természetben lejátszódó folyamatok inspirálták (például Genetikus Algoritmus [9] és Bakteriális Evolúciós Algoritmus [17]). Azonban az általános alkalmazhatóságuknak az az ára, hogy nem ismertek egzakt eredmények arra vonatkozóan, hogy melyik technika mennyire hatékony általánosságban, vagy akár csak egy adott problématerületen belül.

Ezért csupán heurisztikák léteznek annak eldöntésére, hogy melyikeket és milyen paraméterezésekkel célszerű alkalmazni. Ezek intuíción és szimulációs eredményeken alapulnak (például [17]).

Gyakran nagy a különbség az algoritmusok bonyolultsága között, amely különböző típusú karakterisztikákat eredményez. Az egyszerűbb technikák gyorsabbak lehetnek, de kevésbé hatékonyak, amíg a bonyolultabbak lassabbak, de iterációról iterációra sokkal hatékonyabbak.

Sok esetben az optimalizálás korai szakaszaiban könnyű egyre jobb állapotokat elérni a problémában, de sok iterációt követően már nehezzé válik a továbbfejlődés. Így az egyszerű algoritmusok hasznosak lehetnek kezdetben (globálisabb keresésként) a nagy iterációs sebességüknek köszönhetően, amíg a bonyolultabb algoritmusok jobb választásnak bizonyulhatnak később (lokálisabb keresésként) a nagyobb hatékonyságuk miatt.

Ezért gyakran hasznos lenne egyszerű algoritmusokat alkalmazni a numerikus optimalizálási folyamatok elején és bonyolultabbakat a végén. Általánosabban fogalmazva, kívánatos lenne annak meghatározása, hogy melyik optimalizáló algoritmust használjuk egy probléma egyes részein. (Ez a feladat általánosabb, mint az a kérdés, hogy „Melyik algoritmust alkalmazzuk egy adott optimalizálási problémára?”). Más szavakkal: az optimalizáló algoritmusok hatékony ütemezését lenne kívánatos megtalálni adaptív módon az egyes numerikus optimalizálási problémákra. (Természetesen nem csupán optimalizáló algoritmusok ütemezhetőek, hanem teljes optimalizáló architektúrák, vagy éppen egyszerű paraméterezések is.)

Ez a probléma egy döntési fát definiál, ahol a szintek, az élek és a csúcsok rendre iterációs szinteket, optimalizáló algoritmusok végrehajtásait és állapotokat reprezentálnak. A döntési fa bejárható teljesen, vagy részben. Az előbbi hatalmas (exponenciális) számítási igényű, az utóbbi alacsonyabb (akár lineáris) a bejárt rész méretétől függően.

5.1. tézis. *Bevezettem és formálisan definiáltam az optimalizálási algoritmusok adaptív ütemezésének problémáját.*

Gyakran hasznos egyszerű algoritmusokat alkalmazni a numerikus optimalizálási folyamatok elején és bonyolultabbakat a végén. Ezt általánosítva, kívánatos annak meghatározása, hogy melyik optimalizáló algoritmust használjuk egy probléma egyes részein. Vagyis az optimalizáló algoritmusok hatékony ütemezését célszerű adaptív módon az egyes numerikus optimalizálási problémákra megtalálni.

Kapcsoló publikáció: [C9].

3.5.2. Ütemező algoritmusok

A fent leírt problémára a következő három ütemező módszert javasoltam.

Mohó Ütemező. A *Mohó Ütemező (Greedy Scheduler, GS)* valamennyi algoritmust egyidejűleg hajtja végre, és minden egyes iterációt (vagy időrést) követően az aktuálisan legjobb megoldásjelöltet választja ki a célfüggvényértékének megfelelően. Populációalapú technikák esetén (mint a BEA, vagy a BMA) a legjobb populációt választjuk ki valamilyen minőségi mérték szerint, például a legjobb egyed fitness-értéke alapján. A következő iterációban (vagy időrésben) minden algoritmust felinitializálunk a kiválasztott megoldásjelölttel, vagy populációval, és egyidejűleg futtatjuk. Ezek után újabb összehasonlítást végzünk, és így tovább.

Ezen egyszerű ütemező reménytelennek tűnő hátránya az a hatalmas számítási többletköltség, amely az összes algoritmus egész optimalizálási folyamat alatti párhuzamos végrehajtásából ered.

Gyors Mohó Ütemező. Olykor az optimalizálási feladat kedvező tulajdonságokkal rendelkezik, és egyes optimalizáló algoritmusok hosszú ideig a legjobb teljesítmény tudják nyújtani. Ebben az esetben az algoritmusok közti váltások száma alacsony lehet az optimális ütemezésben.

Annak érdekében, hogy ez az előny kiaknázható legyen, a fent tárgyalt Mohó Ütemezőnek egy új változatára tettem javaslatot. Ha az ütemezési probléma az említett kedvező tulajdonsággal rendelkezik, akkor ez az ütemezési módszer gyorsabb, mint az előző.

A *Gyors Mohó Ütemező (Fast Greedy Scheduler, FGS)* nem végez összehasonlítást minden lépésben, azaz nem hajtja végre az összes algoritmust egyidejűleg, csupán egy előre definiált *vakon futási idő* után, amely alatt csak a lokálisan legjobb algoritmust alkalmazza.

Ily módon az előző ütemező esetén említett hátrány enyhíthető.

Monoton Gyors Mohó Ütemező. Ha feltehető, hogy a karakterisztikák, azaz a konvergenciasebességek a fitness-szintek függvényében monoton csökkenő függvények, akkor a következő kézenfekvő továbbfejlesztés alkalmazható a Gyors Mohó Ütemezőn, amelynek eredményeként kapjuk a *Monoton Gyors Mohó Ütemezőt (Monotonic Fast Greedy Scheduler, MFGS)*.

Minden vakon futási fázist követően az aktív algoritmus konvergenciasebességét összehasonlítjuk az inaktív módszerek utoljára számolt konvergenciasebességeivel, amely értékek az utolsóként elvégzett összehasonlítási fázisok végén számolt konvergenciasebességek. Ha az aktív algoritmus konvergenciasebessége még mindig nagyobb, vagy egyenlő, mint az összes többi érték, akkor az aktuális fitness-szinten biztos, hogy még mindig az aktív algoritmus a leghatékonyabb a karakterisztikák feltételezett monotonitása miatt. Ekkor a következő összehasonlítási fázist késleltetjük, és a vakon futást folytatjuk. E meghosszabbított fázis további részében folyamatosan összevetjük a konvergenciasebességet az inaktív módszerek utolsóként számított értékeivel. Amikor az aktív algoritmus hatékonysága az inaktív technikák legmagasabb utoljára mért hatékonysági szintje alá süllyed, a késleltetett összehasonlítási fázist elvégezzük, melynek során az algoritmusokat egy rövid ideig egyidejűleg futtatjuk.

Ezzel a továbbfejlesztéssel az MFGS nyilvánvalóan tovább csökkenti az ütemezők összehasonlításokból eredő számítási többletköltségét.

5.2. tézis. *Három új ütemező algoritmusra tettem javaslatot, amelyek az optimalizálási folyamatokban az optimalizáló módszerek között konvergencia-tulajdonságoktól függő adaptív váltásokat végeznek. Rávilágítottam az újonnan javasolt ütemezők különböző körülmények közötti hatékonyságára, valamint az ütemezők közötti kapcsolatra.*

Kapcsolódó publikációk: [C9], [C14], [C13].

3.5.3. A javasolt ütemezőket alkalmazó optimalizálási folyamatok kísérleti vizsgálata

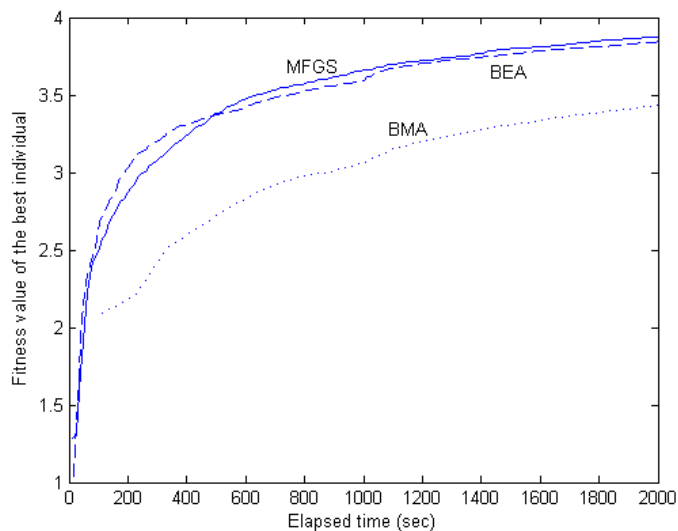
Szimulációs futtatásokat végeztem benchmark problémákon annak érdekében, hogy demonstráljam az imént tárgyalt elmélet alkalmazhatóságát. A szimulációk az MFGS hatékonyságát mutatják a Friedman függvényen, a Stock prices és a Treasury adathalmazokon [2] végzett fuzzy szabályalapú tanulások során.

A tanuló architektúrák Mamdani- és SKH-következtetési módszereket alkalmaztak, továbbá Bakteriális Evolúciós Algoritmust (BEA) és Bakteriális Memetikus Algoritmust (BMA), mint optimalizálási algoritmusokat a fuzzy szabályok megfelelő paramétereinek megtalálásához.

A különböző fuzzy rendszerek által adott eredményeket megfigyelve, talán a legnyilvánvalóbb tény az, hogy az MFGS alulmaradt az öt és a kilencdimenziós esetekben (rendre a Friedman függvény és a Stock adathalmaz), azonban a tizenöt dimenziós probléma esetén (Treasury adathalmaz) az MFGS volt a leghatékonyabb (mind Mamdani-, mind pedig SKH-következtetés esetén). Ezen eredmények magyarázata a következő.

Az egyszerű problémáknál nem érdemes az ütemező technikát használni, mert a számítási többletköltsége lerontja az aktuálisan legjobban teljesítő optimalizáló algoritmusra történő adaptív átkapcsolás lehetőségéből származó előnyét.

Azonban a bonyolultabb probléma esetén az adaptív váltás előnye nagyobb, mint az összehasonlításokból származó számítási többletköltség hátránya, és így az ütemező megközelítést alkalmazó fuzzy rendszerek felülmúlják a többi (vö. 4. ábra, ahol az SKH-következtetést használó rendszer eredményei láthatók az Treasury adathalmazon).



4. ábra. A Treasury adathalmazra adott fitnessz-értékek SKH-interpoláció esetén.

Az eredményeket látva felmerülhetne a gyanú, hogy a BMA technika messze alulmarad ellentétben a 3.3.3 részben megfogalmazott következtetéssel. Azonban gondos tanulmányozást követően észrevehető, hogy az optimalizálási folyamat végén a BMA fitnessz-görbéje a legmeredekebb, előrevetítve a BEA görbéjével történő metszést. Valóban, a 3.3.3 részben a BMA „lassan, de biztosan” viselkedést mutatott, és amikor a BMA elég futási idővel rendelkezett, akkor felülmúlt minden más technikát.

5.3. tézis. *Hatékonyság szempontjából összehasonlítottam és kiértékeltem az újonnan javasolt ütemezőket alkalmazó és nem alkalmazó optimalizálási folyamatokat, valamint kísérletileg igazoltam az újonnan javasolt ütemezőket alkalmazó optimalizálási folyamatok megnövekedett hatékonyságát.*

Kapcsolódó publikációk: [C14], [C13].

4. Az eredmények alkalmazása

Az új eredmények mind elméleti operációkutatási, diszkrét és folytonos optimalizálási, valamint gépi tanulási alapkutatásokban mind pedig olyan ipari műszaki és informatikai területeken használhatók, mint például a termelésoptimalizálás, útvonalválasztás, erőforrás-hozzárendelés, vagy az adatokból történő tudáskinyerés.

Az 1. téziscsoporthoz kapcsolódó eredményeket gyártósorokra alkalmazva a termelési folyamatok hatékonysága az új módszerek fejlett optimalizáló tulajdonságai révén növelhető.

A 2. téziscsoporthoz kapcsolódóan javasolt megközelítések előnyei közé sorolható a bizonytalan üzleti típusú kényszerek jobb modellezése és olyan gyakorlati megkötések kezelése, amellyel más létező módszerek nem képesek megbirkózni.

A 3. és a 4. téziscsoportokhoz kapcsolódó eredményekkel fuzzy szabályalapú gépi tanuló komponensek mennyiségi (pontosság) és minőségi (interpretálhatóság) tulajdonságai javíthatók mesterséges intelligencia rendszerekben.

Végül, általános természetükből fakadóan, az 5. téziscsoporthoz kapcsolódóan javasolt megközelítések teljesítményt és minőséget képesek növelni minden említett területen.

Eredményeimet ez idáig számos tudományos alapkutatás és ipari kutatás-fejlesztési projekt keretében hasznosítottam, melyek során sikeresen oldottam meg gyártósorok folyamatoptimalizálási problémáit, erőforrás hozzárendelési és gépi tanulási feladatokat.

Néhány publikált alkalmazási példát említve: gyártósorok termelési hatékonyságának növelése a megmunkálendő munkadarabok megfelelő sorrendezésével (1. téziscsoport) [J1], sofőrök ütemezése tömegközlekedési járművekre (2. téziscsoport) [C6], hibalokalizálás optikai hálózatokban (2. téziscsoport) [C17], interpretálható szabályok kinyerése járművek fogyasztására vonatkozóan fogyasztási, tömeg, teljesítmény és egyéb tulajdonságokra vonatkozó adatokat tartalmazó adatbázisból (3. és 4. téziscsoportok) [J2].

Kapcsolódó publikációk

Folyóiratcikkek

[J1] K. Balázs, Z. Horváth, and L. T. Kóczy. Different chromosome based evolutionary approaches for the permutation flow shop problem. *Acta Polytechnica Hungarica*, 2(2):115–138, 2012. Estimated impact factor: **0.385**.

- [J2] K. Balázs and L. T. Kóczy. New meaning preserving approaches for constructing fuzzy systems with adjustable interpretability-accuracy trade-off. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. Submitted for publication.
- [J3] K. Balázs and L. T. Kóczy. Fuzzy szabályalapú modellek és rendszerek felépítése evolúciós technikák segítségével. *Híradástechnika*, 66:44–50, 2011. In Hungarian.
- [J4] K. Balázs and L. T. Kóczy. Constructing dense, sparse and hierarchical fuzzy systems by applying evolutionary optimization techniques. *Applied and Computational Mathematics*, 11(1):81–101, 2012. Estimated impact factor: **0.551**.
- [J5] K. Balázs and L. T. Kóczy. Hierarchical-interpolative fuzzy system construction by genetic and bacterial memetic programming approaches. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(supp02):105–131, 2012. Estimated impact factor: **1.781**.
- [J6] Zs. Dányádi, K. Balázs, and L. T. Kóczy. A comparative study of various evolutionary algorithms and their combinations for optimizing fuzzy rule-based inference systems. *Scientific Bulletin of „Politechnica” University of Timisoara, Romania*, 55(69):247–254, 2010.

Cikk szerkesztett könyvben

- [S1] K. Balázs, L. T. Kóczy, and J. Botzheim. Comparative investigation of various evolutionary and memetic algorithms. In I. J. Rudas, J. Fodor, and J. Kacprzyk, editors, *Computational Intelligence in Engineering*, volume 313 of *Studies in Computational Intelligence*, pages 129–140. Springer, 2010.

Konferenciacikkek

- [C1] K. Balázs, J. Botzheim, and L. T. Kóczy. Comparative analysis of various evolutionary and memetic algorithms. In *Proceedings of the 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, CINTI 2009*, pages 193–205, Budapest, Hungary, 2009.
- [C2] K. Balázs, J. Botzheim, and L. T. Kóczy. Comparative analysis of interpolative and non-interpolative fuzzy rule based machine learning systems applying various numerical optimization methods. In *World Congress on Computational Intelligence, WCCI 2010*, pages 875–982, Barcelona, Spain, 2010.
- [C3] K. Balázs, J. Botzheim, and L. T. Kóczy. Comparison of various evolutionary and memetic algorithms. In *Proceedings of the International Symposium on Integrated Uncertainty Management and Applications, IUM 2010*, pages 431–442, Ishikawa, Japan, 2010.

- [C4] K. Balázs, J. Botzheim, and L. T. Kóczy. Hierarchical fuzzy system construction applying genetic and bacterial programming algorithms with expression tree building restrictions. In *World Automation Congress, WAC 2010*, pages 1–6, Kobe, Japan, 2010.
- [C5] K. Balázs, J. Botzheim, and L. T. Kóczy. Hierarchical fuzzy system modeling by genetic and bacterial programming approaches. In *World Congress on Computational Intelligence, WCCI 2010*, pages 1866–1871, Barcelona, Spain, 2010.
- [C6] K. Balázs, M. Farkas, and L. T. Kóczy. Proposal on formalizing uncertain business criteria as fuzzy constraints in multi-objective csp optimization. In *International Fuzzy Systems Association World Congress and Asia Fuzzy Systems Society International Conference, IFSA 2011*, pages 1–5, Surabaya, Indonesia, 2011.
- [C7] K. Balázs, Z. Horváth, and L. T. Kóczy. Hybrid bacterial iterated greedy heuristics for the permutation flow shop problem. In *World Congress on Computational Intelligence, WCCI 2012*, pages 1–8, Brisbane, Australia, 2012.
- [C8] K. Balázs, Z. Horváth, and L. T. Kóczy. Multi-threaded bacterial iterated greedy heuristics for the permutation flow shop problem. In *13th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2012*, pages 63–66, Budapest, Hungary, 2012.
- [C9] K. Balázs and L. T. Kóczy. A remark on adaptive scheduling of optimization algorithms. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2010*, pages 719–728, Dortmund, Germany, 2010.
- [C10] K. Balázs and L. T. Kóczy. Hierarchical-interpolative fuzzy system construction by genetic and bacterial programming algorithms. In *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2011*, pages 2116–2122, Taipei, Taiwan, 2011.
- [C11] K. Balázs and L. T. Kóczy. Genetic and bacterial memetic programming approaches in hierarchical-interpolative fuzzy system construction. In *World Congress on Computational Intelligence, WCCI 2012*, pages 1–8, Brisbane, Australia, 2012.
- [C12] K. Balázs and L. T. Kóczy. New parameterizable search space narrowing technique for adjusting between accuracy and interpretability in fuzzy systems. In *13th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2012*, pages 323–328, Budapest, Hungary, 2012.
- [C13] K. Balázs and L. T. Kóczy. Adaptive scheduling of optimization algorithms in the construction of interpolative fuzzy systems. In *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2013*, Hyderabad, India, 2013. To appear.
- [C14] K. Balázs and L. T. Kóczy. Constructing dense fuzzy systems by adaptive scheduling of optimization algorithms. In *International Fuzzy Systems Association World Congress, IFSA 2013*, Edmonton, Canada, 2013. To appear.

- [C15] K. Balázs and L. T. Kóczy. A stochastic model for analyzing the interpretability-accuracy trade-off in interpretable fuzzy systems using nested hyperball structures. In *8th conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013*, Milano, Italy, 2013. Accepted for publication.
- [C16] K. Balázs, L. T. Kóczy, and J. Botzheim. Comparison of fuzzy rule-based learning and inference systems. In *Proceedings of the 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, CINTI 2008*, pages 61–75, Budapest, Hungary, 2008.
- [C17] K. Balázs, P. B. Soproni, and L. T. Kóczy. Improving system reliability in optical networks by failure localization using evolutionary optimization. In *IEEE International Systems Conference, SysCon 2013*, Orlando, USA, 2013.
- [C18] Zs. Dányádi, K. Balázs, and L. T. Kóczy. A comparative study of various evolutionary algorithms used for fuzzy rule-based inference and learning systems. In *IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics, ICC-CONTI 2010*, pages 49–54, Timisoara, Romania, 2010.
- [C19] Zs. Dányádi, K. Balázs, and L. T. Kóczy. Using multiple populations of memetic algorithms for fuzzy rule-base optimization. In *Proceedings of the 11th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2010*, pages 113–118, Budapest, Hungary, 2010.

Hivatkozások

- [1] S. Ahuja, S. Ramasubramanian, and M. Krunz. Single-link failure detection in all-optical networks using monitoring cycles and paths. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1080–1093, 2009.
- [2] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2–3):255–287, 2011.
- [3] P. Babarcsi, J. Tapolcai, and P.-H. Ho. Srg failure localization with monitoring trails in all-optical mesh networks. In *International Conference on Design of Reliable Communication Networks, DRCN 2011*, pages 188–195, Krakow, Poland, 2011.
- [4] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. E. Ruano. Fuzzy rule extraction by bacterial memetic algorithms. In *Proceedings of the 11th World Congress of International Fuzzy Systems Association, IFSA 2005*, pages 1563–1568, Beijing, China, July 2005.

- [5] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. E. Ruano. Genetic and bacterial programming for B-spline neural networks design. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 11(2):220–231, February 2007.
- [6] B. Crawford, C. Castro, and E. Monfroy. A hybrid ant algorithm for the airline crew pairing problem. In *MICAI 2006: Advances in Artificial Intelligence*, volume 4293 of *Lecture Notes in Computer Science*, pages 381–391. Springer, 2006.
- [7] M. Dorigo, V. Maniezzo, and A. Coloni. The ant system: Optimization by a colony of co-operating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26(1):29–41, 1996.
- [8] D. Dubois and H. M. Prade. *Fundamentals of fuzzy sets*. Springer, New York, 2000.
- [9] J. H. Holland. *Adaption in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts, 1992.
- [10] R. W. Kember. *Fibre Channel: A Comprehensive Introduction*. Northwest Learning Associates, 1998.
- [11] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, Perth, Australia, 1995.
- [12] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [13] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge MA, USA, May 1994.
- [14] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2(2):164–168, 1944.
- [15] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.*, 7:1–13, 1975.
- [16] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11(2):431–441, Jun. 1963.
- [17] N. E. Nawa and T. Furuhashi. Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Transactions on Fuzzy Systems*, 7(5):608–616, Oct. 1999.
- [18] M. Nawaz, E. E. Ensore Jr., and I. Ham. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *OMEGA, The International Journal of Management Science*, 11(1):91–95, 1983.
- [19] M. G. Ravetti, C. Riveros, A. Mendes, M. G. C. Resende, and P. M. Pardalos. Parallel hybrid heuristics for the permutation flow shop problem. Research technical report, AT&T Labs, 2010.

- [20] R. Ruiz and T. Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177:2033–2049, 2007.
- [21] J. A. Snyman. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer, New York, 2005.
- [22] E. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285, 1993.
- [23] J. Tapolcai, B. Wu, and P.-H. Ho. On monitoring and failure localization in mesh all-optical networks. In *IEEE INFOCOM*, pages 1008–1016, Rio de Janeiro, Brasil, 2009.
- [24] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon. Stability of interpolative fuzzy kh-controllers. *Fuzzy Sets and Systems*, 125(1):105–119, 2002.