

Fuzzy modell identifikáció

Botzheim János

Széchenyi István Egyetem

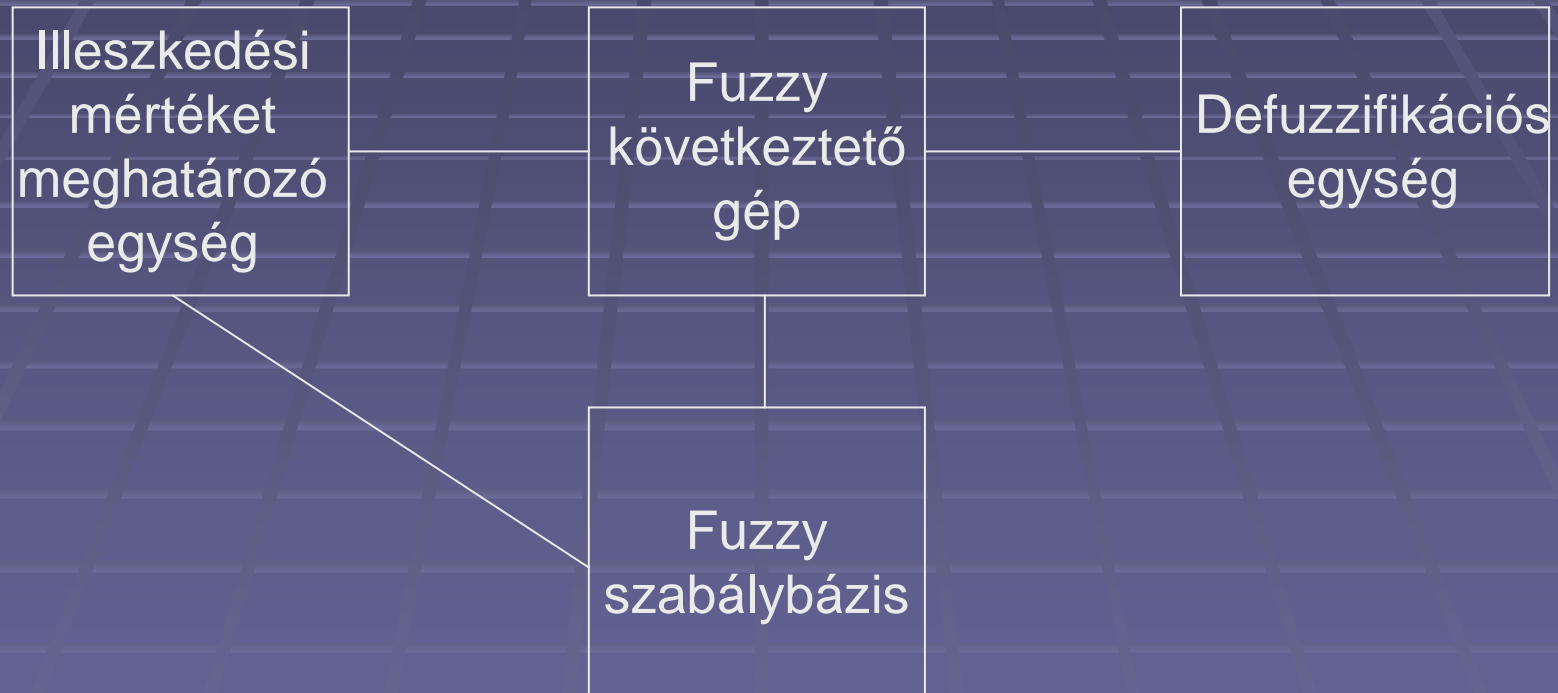
Automatizálási tanszék

Témák

- Bakteriális evolúciós algoritmusok
- Fuzzy szabálybázis optimalizálása
- Gradiens módszerek
- Bakteriális memetikus algoritmusok
- Változó kiválasztás
- Szabály kiválasztás
- B-spline neurális hálózatok
- Bakteriális programozás
- Utazó ügynök probléma

Fuzzy rendszerek

A fuzzy rendszer általános felépítése



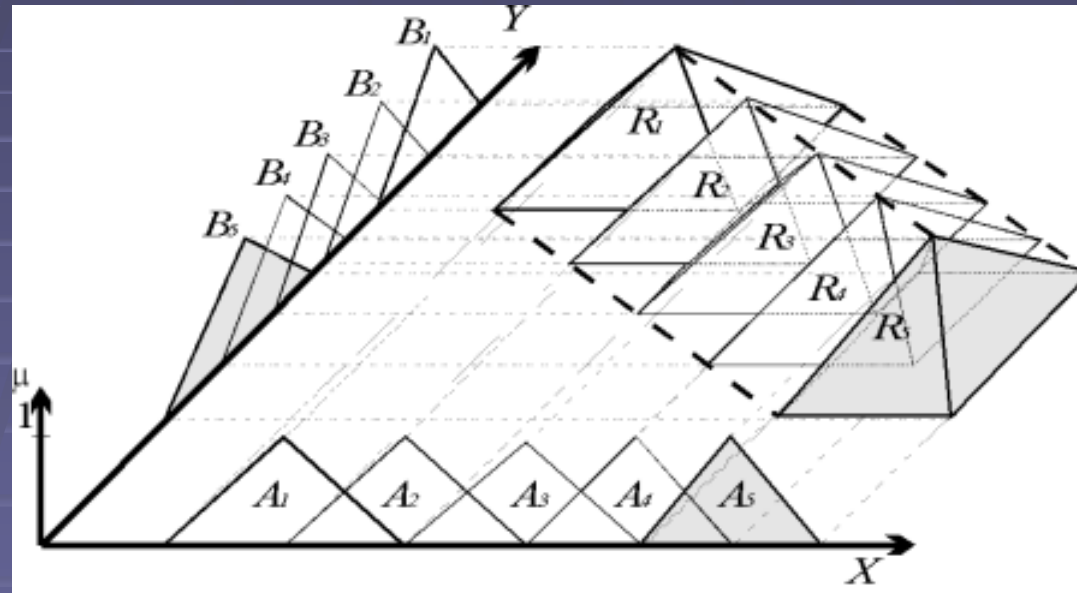
Nyelvi szabályok:

HA $x = A$ AKKOR $y = B$

A a szabály feltételrész, B a következményrész

Példa: „**HA** a forgalom ebből az irányból nagy **AKKOR** legyen hosszabb ideig zöld a lámpa”

Fuzzy szabálybázis reláció:

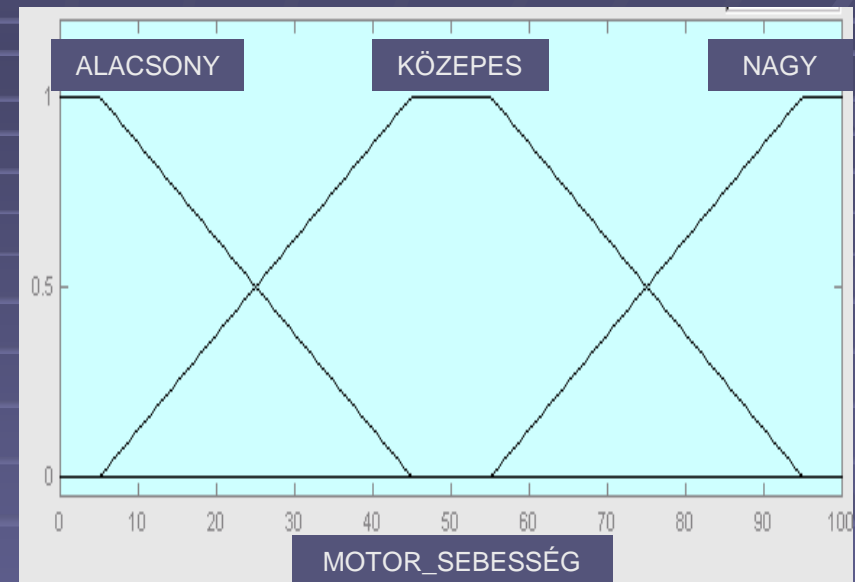
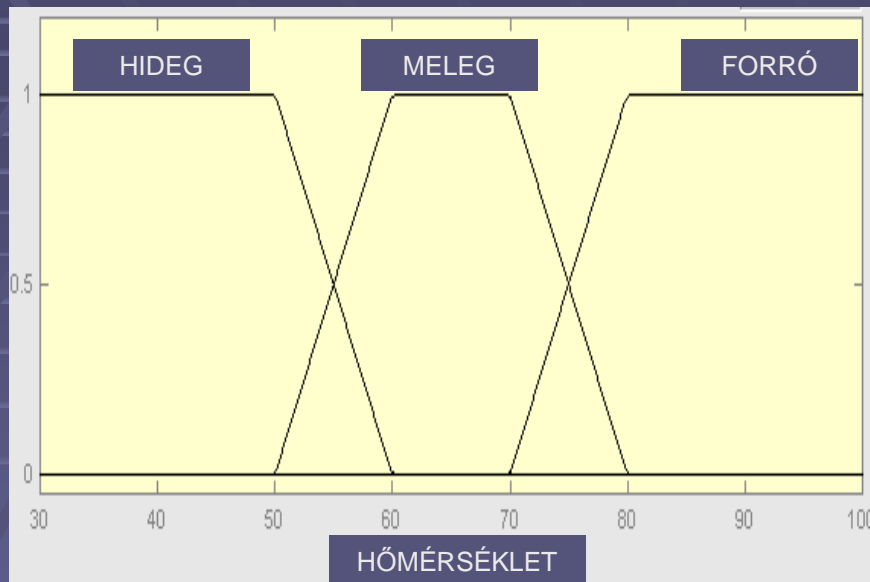


$$\mu_{R(x,y)}(x,y) = \max_{i=1}^r (\mu_{R_i(x,y)}(x,y)) \quad \mu_{R_i(x,y)}(x,y) = \min(\mu_{A_i(x)}(x), \mu_{B_i(y)}(y))$$

többdimenziós eset:

$$\mu_{R_i(x_1, x_2, \dots, x_n, y)}(x_1, x_2, \dots, x_n, y) = \min(\mu_{A_{1,i}(x_1)}(x_1), \dots, \mu_{A_{n,i}(x_n)}(x_n), \mu_{B_i(y)}(y))$$

Fuzzy rendszerek: egy példa



A fuzzy rendszerek fuzzy szabályokkal dolgoznak:

HA a hőmérséklet **HIDEG** **AKKOR** a motor_sebesség **ALACSONY**

HA a hőmérséklet **MELEG** **AKKOR** a motor_sebesség **KÖZEPES**

HA a hőmérséklet **FORRÓ** **AKKOR** a motor_sebesség **NAGY**

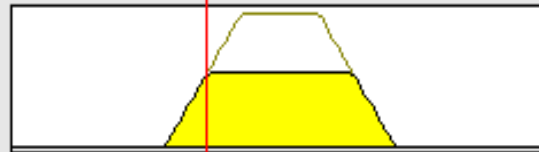
Következtetés (Mamdani)

1. szabály

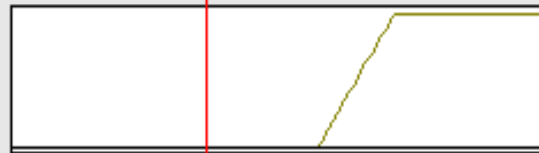
Hőmérséklet = 55



2. szabály



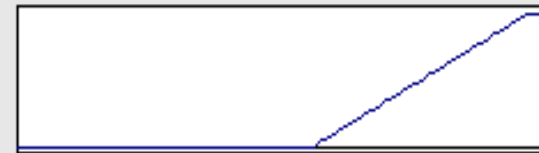
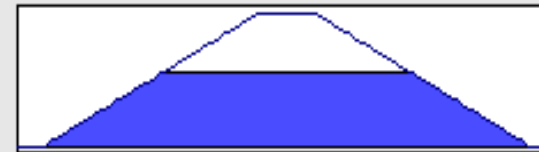
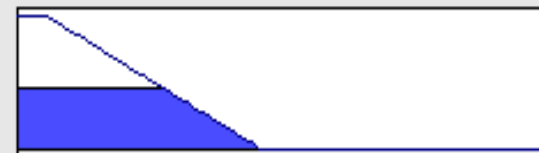
3. szabály



30

100

Motor sebesség



0

100

Motor sebesség =
43.6

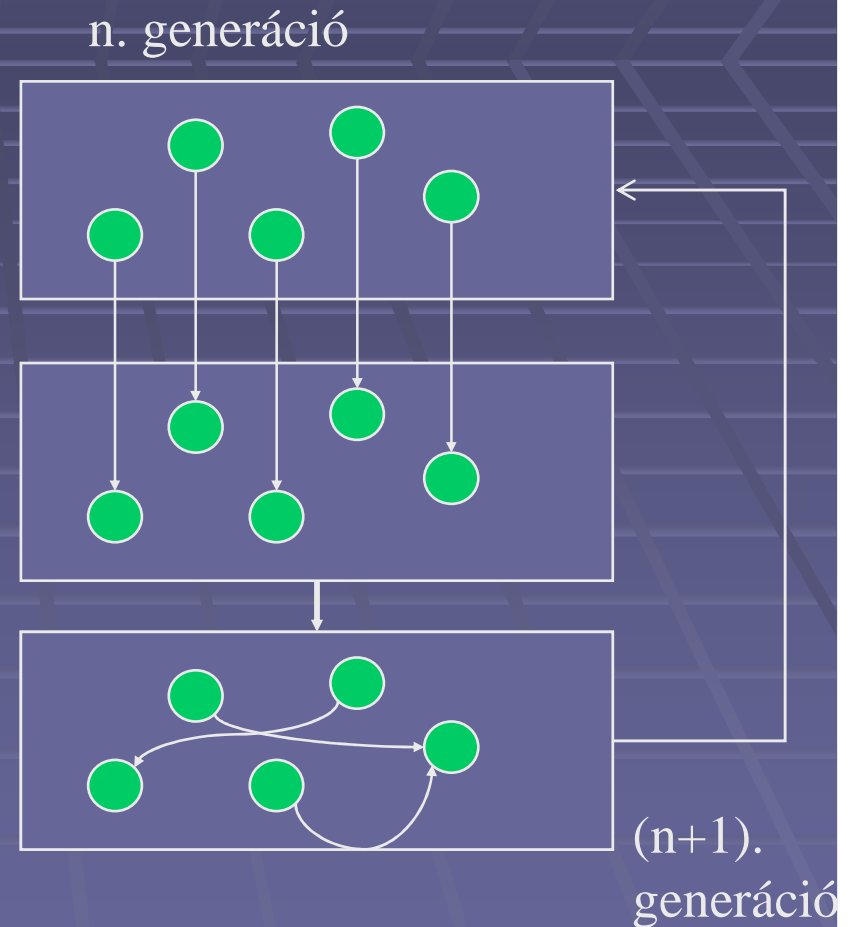
Bakteriális evolúciós algoritmusok

Bakteriális evolúciós algoritmusok

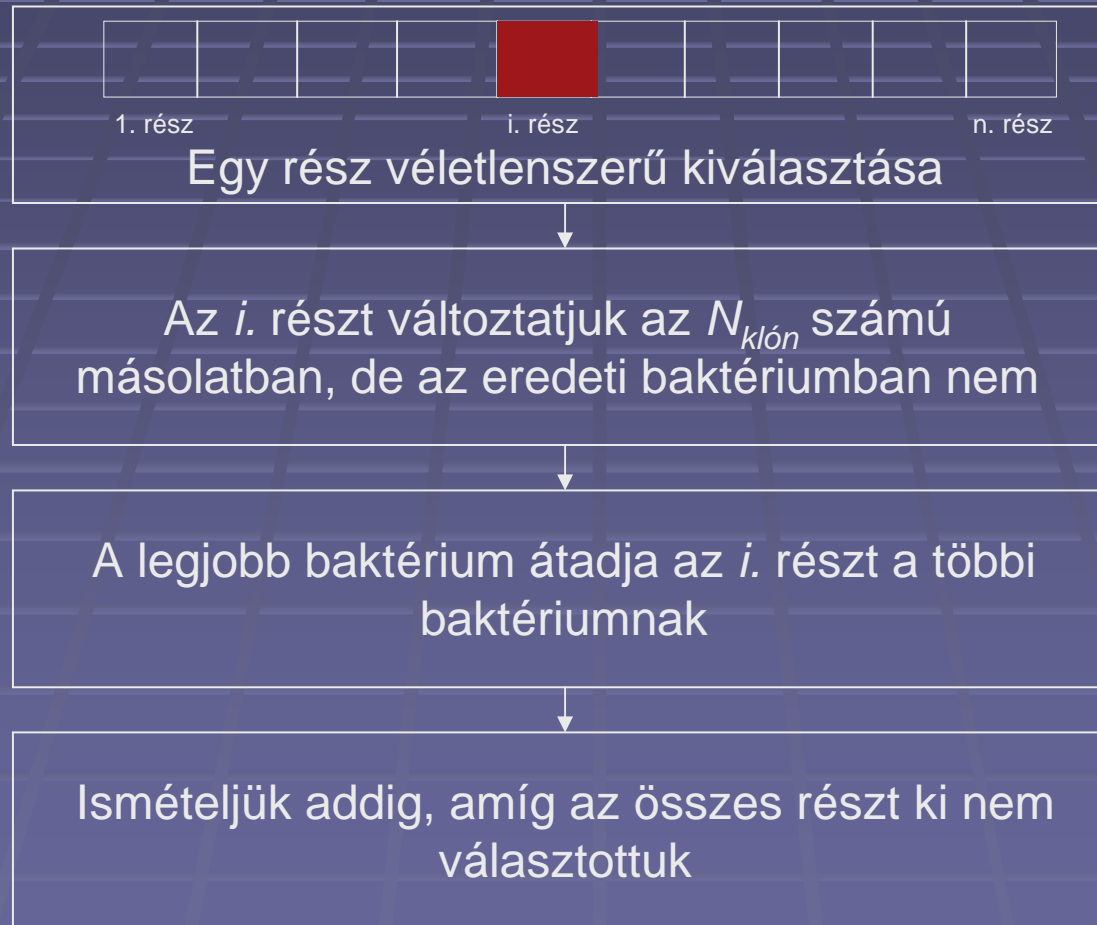
- Természetből ellesztett optimalizációs technika
- A baktériumok evolúciós folyamatán alapul
- Alkalmas bonyolult optimalizációs problémák megoldására
- Egyed: egy megoldás a problémára
- Intelligens keresési stratégia *elégé jó* megoldás keresésére (kvázi optimum)
- Gyors konvergencia (feltételesen)

Az algoritmus

- Kezdeti populáció véletlenszerű létrehozása
- Bakteriális mutáció végrehajtása minden egyeden
- Génátadás végrehajtása a populációban
- Ha megfelelő eredményt értünk el, akkor megállunk, különben folytatjuk a bakteriális mutációs lépéssel

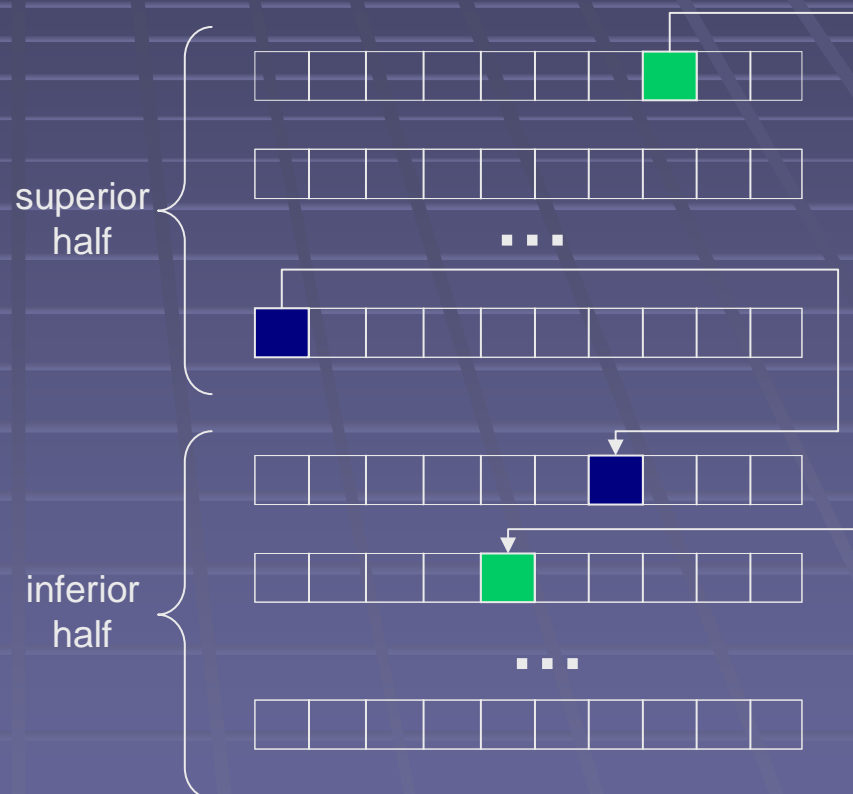


Bakteriális mutáció



Génátadás

1. A populációt 2 részre osztjuk, jó egyedekre (superior half), és rossz egyedekre (inferior half)
2. Egy baktériumot véletlenszerűen kiválasztunk a jobbik alpopulációból (forrás baktérium) egy másikat pedig a rossz egyedek közül (cél baktérium)
3. A forrás baktérium egy része felülírja a cél baktérium egy részét



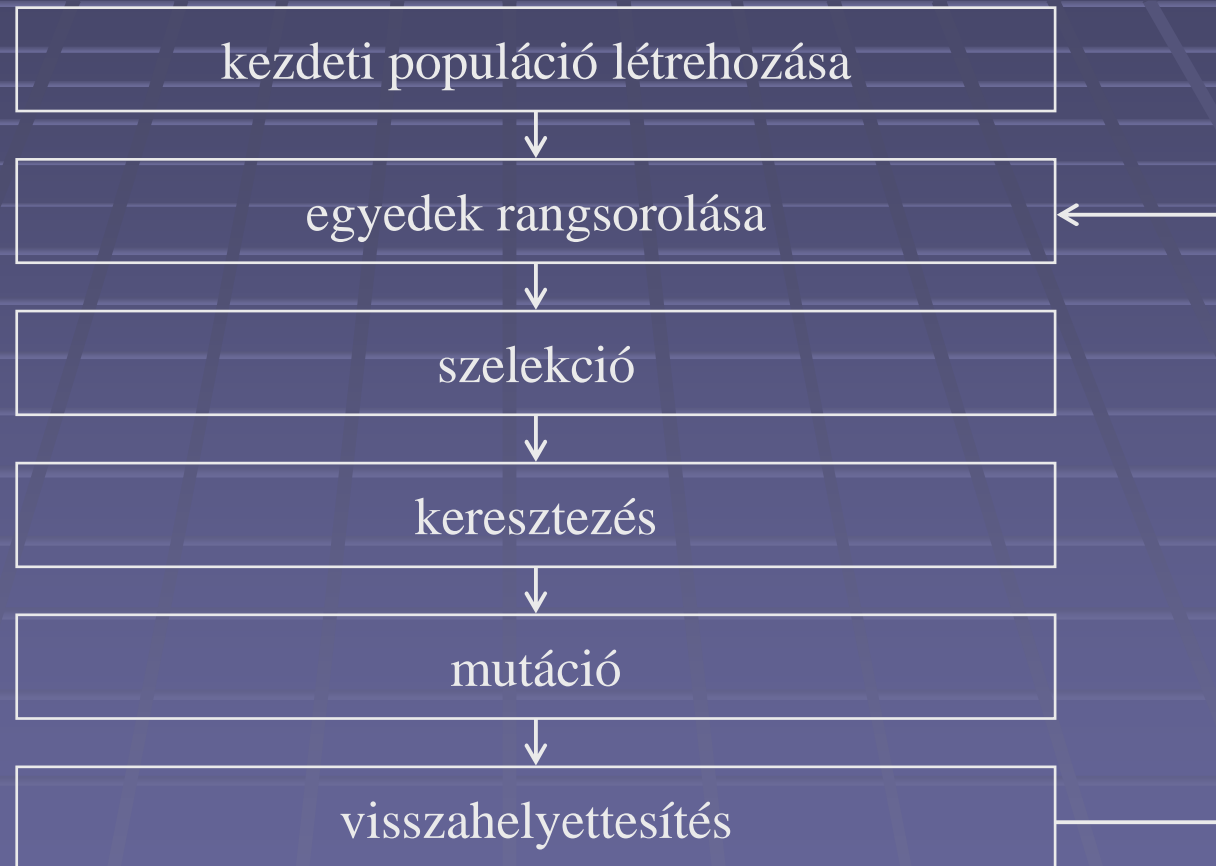
Ez a ciklus ismétlődik

N_{inf} -szer (“infekciók” száma)

Paraméterek

- N_{gen} : generációk száma
- N_{ind} : egyedek száma
- $N_{klón}$: másolatok (klónok) száma a bakteriális mutációban
- N_{inf} : génátadások (infekciók) száma a génátadásnál

Genetikus algoritmusok



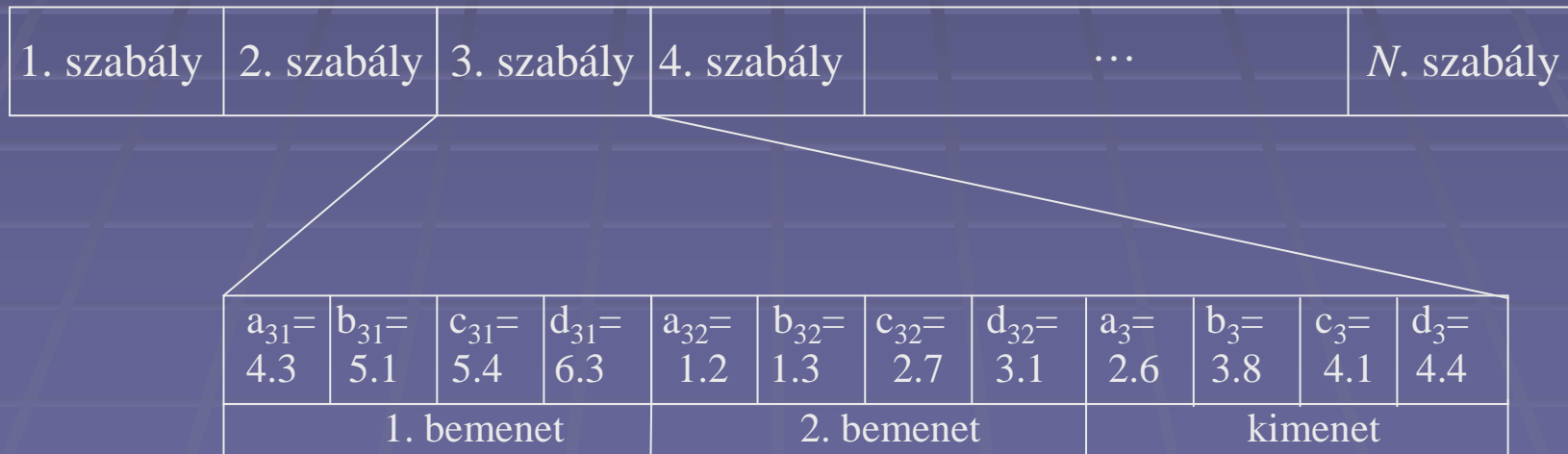
Különbségek a GA és BEA között

- A GA az emlősállatok evolúciós folyamatát utánozza, míg a BEA a baktériumok fejlődését
- Az információ terjesztésére a GA a keresztezés operátort használja, a BEA pedig a génátadást
- A bakteriális mutáció hatékonyabb a GA klasszikus mutációjánál
- A bakteriális evolúciós algoritmusban nincs szelekció, viszont van osztódás

Fuzzy szabálybázis optimalizálása

Fuzzy szabálybázis optimalizálása

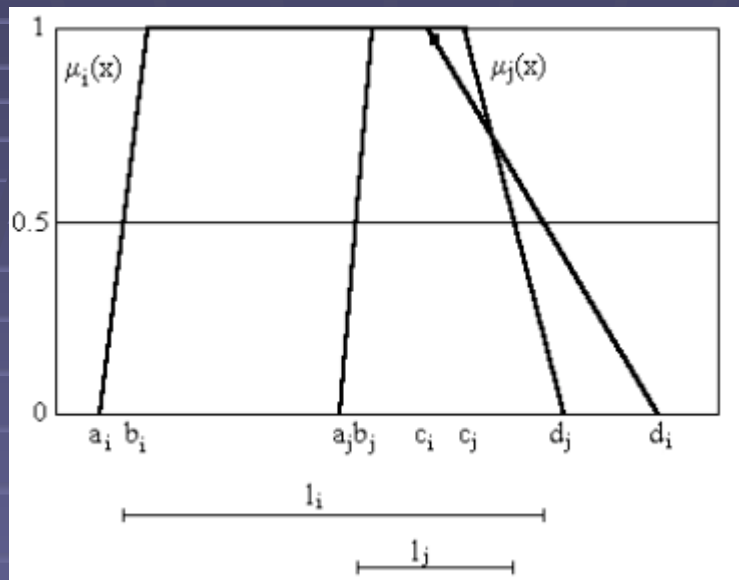
- Egy fuzzy rendszer egy baktérium
- Trapéz alakú tagsági függvények
 - 4 paraméterrel vannak leírva
 - $A_{kl}(a_{kl}, b_{kl}, c_{kl}, d_{kl})$: az i . bemenet tagsági függvénye a k . szabályban
- Pl. egy két bemenetű rendszer kódolása egy egyedbe:



Fuzzy szabálybázis optimalizálása

- További operátorokat alkalmazhatunk a szabálybázis méretének optimalizálására
- Ezeket a szabály redukáló operátorokat minden baktériumra alkalmazhatjuk a bakteriális operátorok közötti lépésben
- Redundáns szabályok fordulhatnak elő
- A hatástalan szabályokat törölhetjük és a hasonló szabályokat összevonhatjuk
- Kompromisszumot kell találni a közelítés pontossága és a bonyolultság között

Megsemmisítés



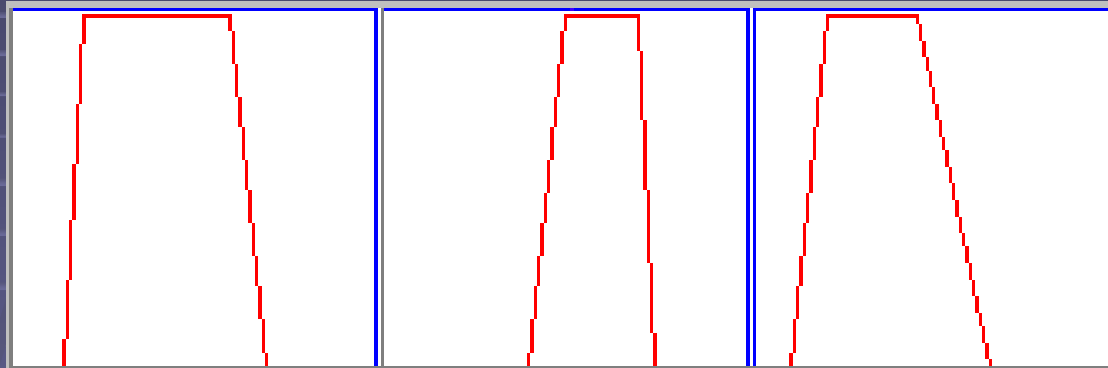
az alkalmazási kritérium:

$$l_i \mu_i \left(\frac{a_j + b_j + c_j + d_j}{4} \right) \geq \beta l_j$$

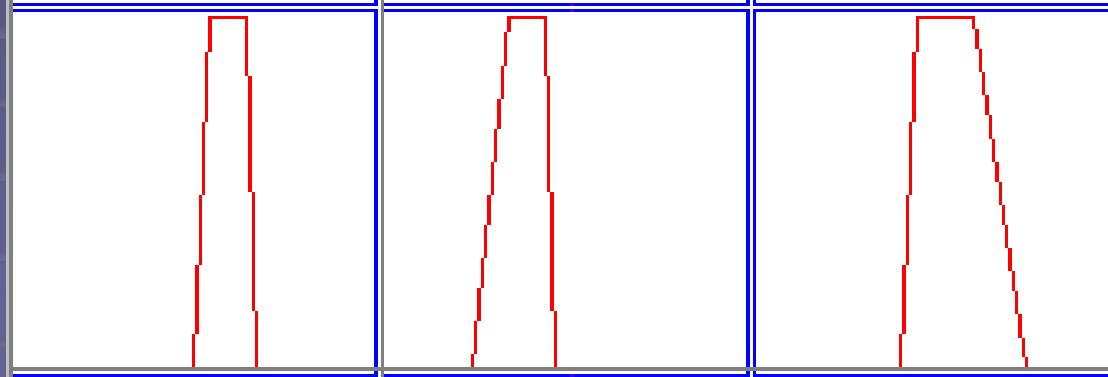
Megsemmisítés

1. bemenet 2. bemenet kimenet

i. szabály:

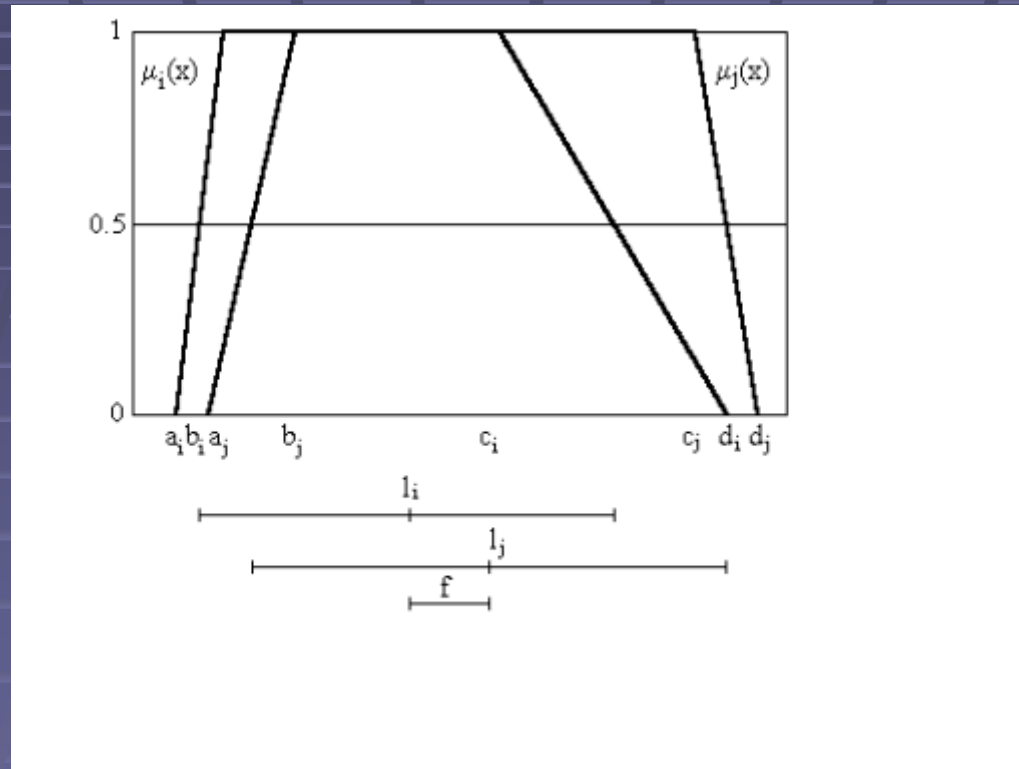


j. szabály:



j. szabály törlése

Egyesítés



kritériumok: $\left| \frac{l_i}{l_j} - 1 \right| < \gamma$ and $|f| < \gamma$

az egyesített tagsági
függvény paraméterei:

$$z_{fus} = \frac{z_i l_i + z_j l_j}{l_i + l_j}$$

Egyesítés

Egyesítés előtt:

i. szabály:

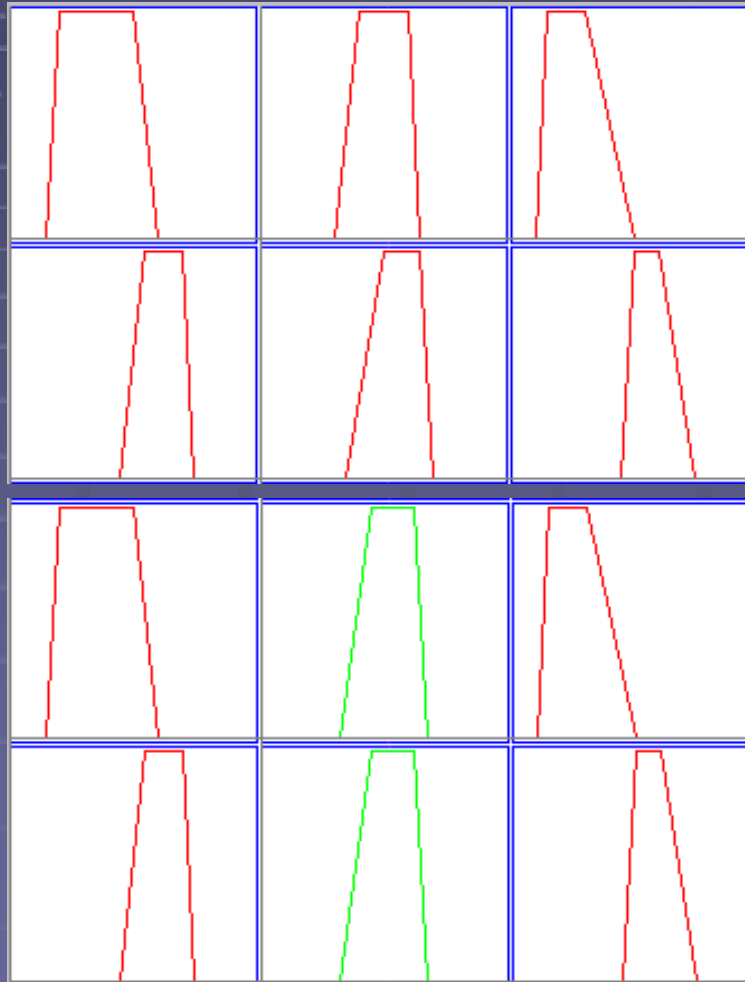
j. szabály:

Egyesítés után:

i. szabály:

j. szabály:

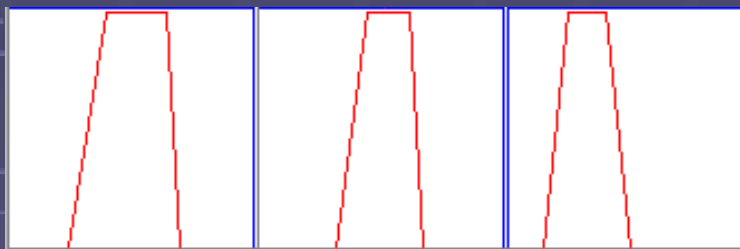
1. bemenet 2. bemenet kimenet



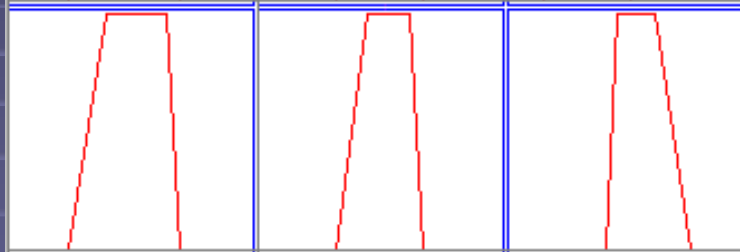
Szemantikus elemzés

1. bemenet 2. bemenet kimenet

i. szabály:

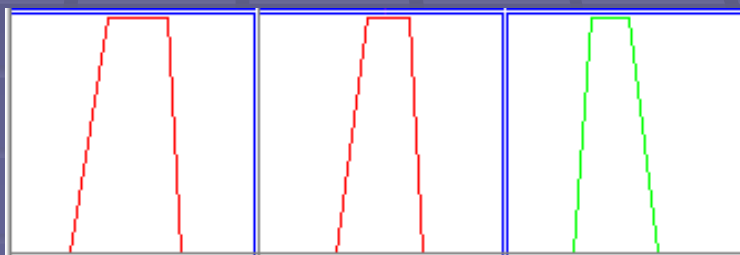


j. szabály:



Kimenet egyesítése
után:

i. szabály:



j. szabály törlése

Gradiens módszerek

A Levenberg-Marquardt módszer

- Másodrendű gradiensalapú tanítómódszer
- A 'megbízható-régió' vagy 'korlátozott-lépés' típusú módszerek csoportjába tartozik
 - Megkísérel egy olyan szomszédságot meghatározni ahol a kvadratikus függvény modell valamilyen értelemben megegyezik az aktuális függvénnyel. Az α paraméter irányítja a szomszédsági sugarat.
- Lokális optimalizáló

A Levenberg-Marquardt eljárás

A tanító algoritmus:

- A fuzzy szabálybázis struktúrája előre meghatározott

- Hiba kritérium

$$\Omega = \frac{\| \underline{t} - \underline{y} \|^2}{2} = \frac{\| \underline{e}[k] \|^2}{2} \quad \begin{array}{l} t = \text{kívánt kimenet} \\ y = \text{modell kimenet} \end{array}$$

- A Jacobi mátrix kiszámítása

$$\underline{J}[k] = \begin{bmatrix} \frac{\partial y(\underline{x}^{(p)})[k]}{\partial \underline{par}[k]} \end{bmatrix}$$

- par: a tagsági függvények (baktérium) paraméterei
- *k*: iterációs változó
- Az LM módosítást (*s*[*k*]) a következő egyenlet optimális megoldása adja:


$$(\underline{J}^T[k] \underline{J}[k] + \alpha \underline{I}) \underline{s}[k] = -\underline{J}^T[k] \underline{e}[k]$$

- Az új par-t *s*[*k*] segítségével kapjuk meg:

$$\underline{par}[k+1] = \underline{par}[k] + \underline{s}[k]$$

A Levenberg-Marquardt eljárás

Az alkalmazott algoritmus:

1. $J[k]$ és $e[k]$ számítása
2. $s[k]$ számítása 
3. A szomszédsági sugár ($r[k]$) kiértékelése
4. If $r[k] < 0.25$

$$\underline{s}[k] = - \begin{bmatrix} \underline{J}[k] \\ \sqrt{\alpha} \underline{I} \end{bmatrix}^+ \begin{bmatrix} \underline{e}[k] \\ \underline{0} \end{bmatrix}$$

$$\alpha[k+1] = 4 * \alpha[k]$$

Else if $r[k] > 0.75$

$$\alpha[k+1] = \alpha[k] / 2$$

Else

$$\alpha[k+1] = \alpha[k]$$

End

If $r[k] \leq 0$

$$\text{par}[k+1] = \text{par}[k]$$

Else

$$\text{par}[k+1] = \text{par}[k] + s[k]$$

if not ($a_{ij} \leq b_{ij} \leq c_{ij} \leq d_{ij}$)

$$\text{par}[k+1] = \text{par}[k] + g * s[k] \text{ és}$$

end

$$g = \frac{\text{par}_{i+1}[k] - \text{par}_i[k]}{2 * (s_{i+1}[k] - s_i[k])}$$

(a tagsági függvény
definíciója miatti
kényszer)

End

A Levenberg-Marquardt eljárás

Jacobi számítás:

- Minták alapján történő számítás:

$$\underline{J} = \begin{bmatrix} \frac{\partial y(\underline{x}^{(p)})}{\partial a_{11}} & \frac{\partial y(\underline{x}^{(p)})}{\partial b_{11}} & \dots & \frac{\partial y(\underline{x}^{(p)})}{\partial a_{12}} & \dots & \frac{\partial y(\underline{x}^{(p)})}{\partial d_1} & \dots & \frac{\partial y(\underline{x}^{(p)})}{\partial d_R} \end{bmatrix}$$

- A láncszabály alkalmazása a deriválásnál
- A bemeneti és kimeneti tagsági függvények deriváltjait és w_i (az i . szabály aktivizálódási foka) deriváltjait kell kiszámolni

A Levenberg-Marquardt módszer

- Használható az evolúciós algoritmusok feljavítására, amely ezáltal nagyobb pontossággal találja meg a globális optimumot
- Az evolúciós és a lokális kereső hibrid módszereket általában **memetikus algoritmusoknak** nevezik
- Egy új memetikus módszert javasolunk: **bakteriális memetikus algoritmus**

Memetikus algoritmusok

Bakteriális memetikus algoritmus

Evolúciós elméletek

- Jean-Baptiste Lamarck
 - A szerzett tulajdonságok öröklésének elmélete
 - Ha egy organizmus változik az élete során, hogy ezáltal jobban adaptálódjon a környezetéhez, ezeket a változásokat továbbadja az utódainak
- Charles Darwin
 - Az állatok vágyainak semmi köze a fejlődésükhöz
 - Az organizmus élete során bekövetkező változásai nincsenek hatással a faj fejlődésére
- James M. Baldwin
 - Új faktor az evolúcióban
 - A szerzett tulajdonságok indirekten öröklődhetnek

Memetikus algoritmus

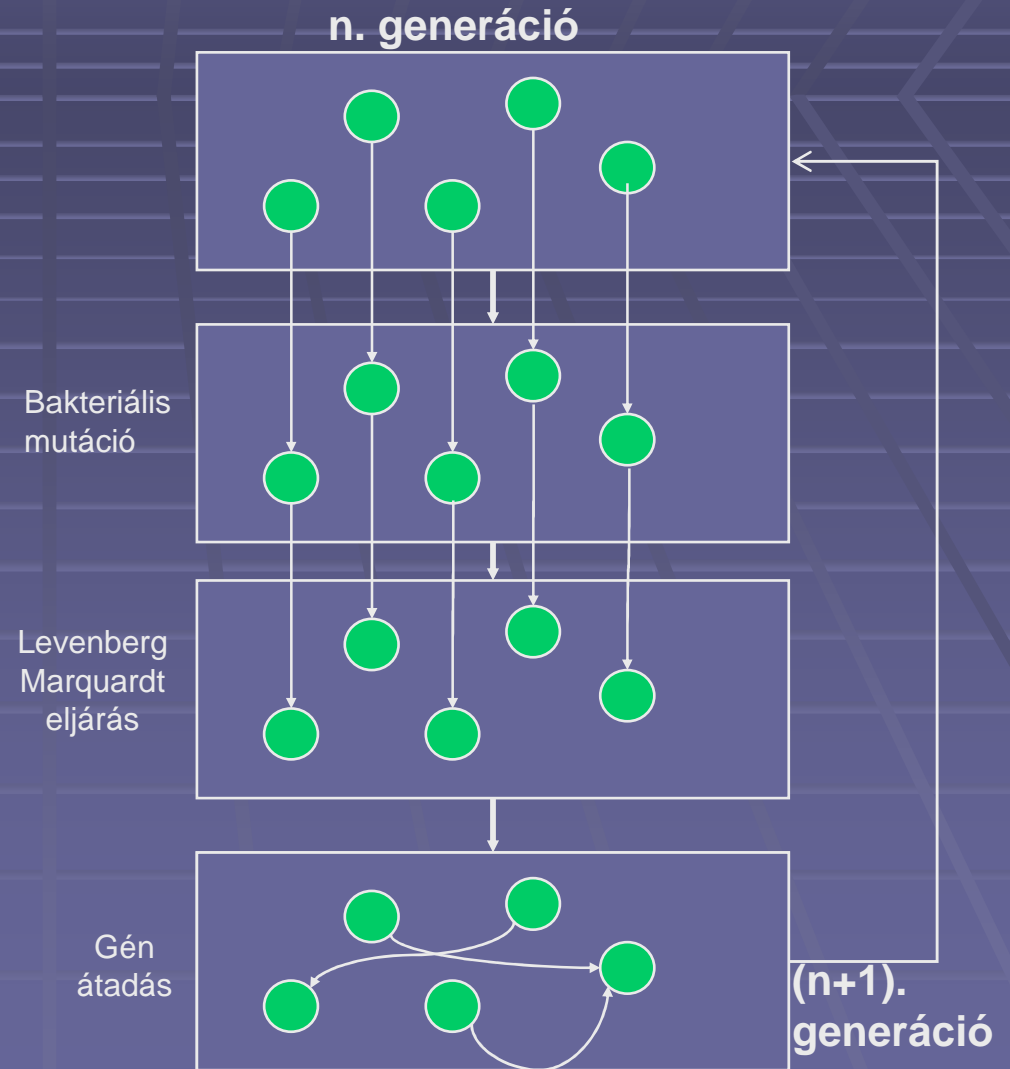
- Az evolúciós algoritmusoknak lokális kereső operátorokkal történő olyan kombinációja, ahol a lokális keresés az evolúciós cikluson belül zajlik
- A memetikus algoritmusokat Moscato javasolta (~1989)
- A lokális keresést élethosszig tartó tanulásnak lehet tekinteni
- Terminológia:
 - mém = a kulturális evolúció alapegysége („génje”) (Dawkins: Az önző gén, 1976)
 - ”mimema”: utánzás

Miért kombináljuk őket?

- Evolúciós algoritmusok
 - Nagy, durva keresési teret járnak be
 - Nehézségek a finomhangolással
- Lokális kereső technikák
 - Gyorsan optimalizálnak, konvergálnak
 - Lokális optimumokba ragadnak
- Hátrányok:
 - A tanulás költséges
 - A tanulás nem mindig jó

Bakteriális memetikus algoritmus

- Kezdeti populáció véletlenszerű létrehozása
- **Bakteriális mutáció** végrehajtása minden egyeden
- **Levenberg-Marquardt** módszer végrehajtása minden baktériumon
- **Génátadás** végrehajtása a populációban
- Ha megfelelő eredményt értünk el, akkor megállunk, különben folytatjuk a bakteriális mutációs lépéssel



Az algoritmus paramétere

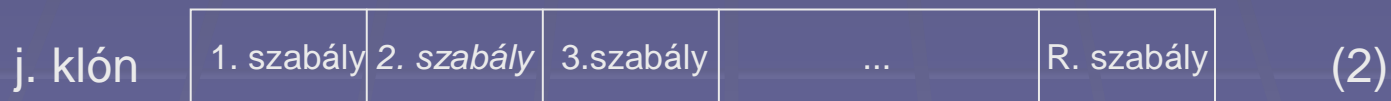
- N_{gen} : generációk száma
- N_{ind} : egyedek száma
- N_{clones} : másolatok (klónok) száma a bakteriális mutációban
- N_{inf} : génátadások (infekciók) száma a génátadásnál
- N_{iter} : iterációk száma az LM lépésben
- α : regularizációs paraméter az LM lépésben

Optimális szabálybázis méret

- Hogyan lehet a szabálybázis méretét is optimalizálni?
- Megengedünk különböző hosszúságú baktériumokat (különböző szabálysámú baktériumokat)
- A bakteriális operátorokat továbbfejlesztve kezelni tudjuk a különböző hosszúságú baktériumokat
- A kiértékelési kritériumnak tartalmaznia kell a baktérium hosszát is

Továbbfejlesztett bakteriális mutáció

- Az i^{th} szegmens mutációja közben a klónokban három különböző művelet történhet
 - szabály eltávolítás (1)
 - szabály helyettesítés (2)
 - szabály hozzáadás (3)

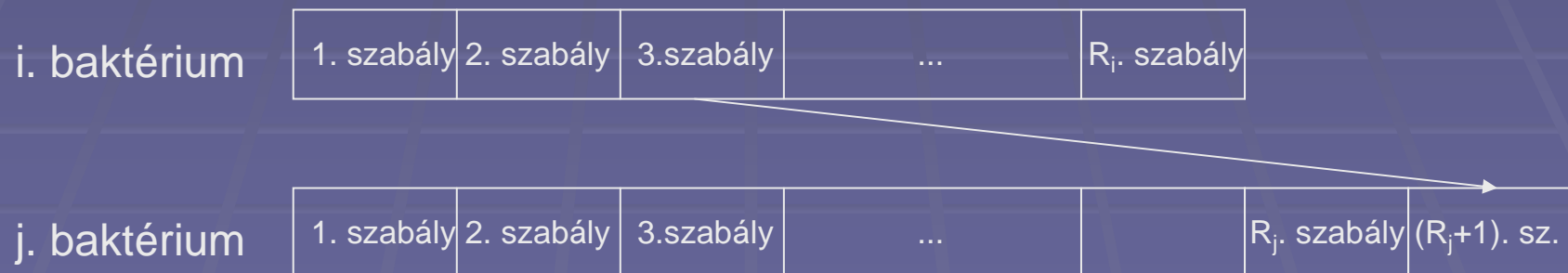


Továbbfejlesztett génátadás

Szabály helyettesítés művelete



Szabály hozzáadás művelete



Változó kiválasztás

Bevezetés

- Cél
 - A regressziós problémák nagyszámú változót foglalhatnak magukba
- Megoldási koncepció
 - Változó kiválasztás
- A legfontosabb változók kiválasztása
 - Interpretáció növelése
 - Túlilleszkedés elkerülése

Változó kiválasztás

- Különböző megközelítések vannak
 - Szűrő
 - A regresszió tanulás alkalmazása előtt kiválasztja a változókat
 - Hátrány: a kombinált függőségeket nem ismeri fel
 - Egyesült
 - Kombinálja a regressziós tanulást és a változó kiválasztást
 - Hátrány: csak egy tanulási módszerhez lehet alkalmazni
 - Becsomagolt
 - Magába foglalja a tanulási módszert
 - Előny: tetszőleges regressziós tanulási módszerhez alkalmazható
 - Hátrány: Nagy számítási teljesítményt igényel

Változó kiválasztás

- Motiváció a bakteriális optimalizáció használatára
 - Intelligens keresési stratégia *elég* jó megoldás keresésére
 - Alkalmazható komplex optimalizációs problémákra
 - Gyors konvergencia
- A bakteriális evolúciós algoritmust „csomagolásként” alkalmazzuk a regressziós tanuló köré

A kódolási módszer

- Egy baktérium a probléma egy megoldásának felel meg ($\xi_i, i \in I$)
- A probléma jelen esetben m változó kiválasztása egy n változót tartalmazó készletből ($m \leq n$)
- A baktérium egész számok vektorából épül fel $\xi_i = \{\xi_i^1, \dots, \xi_i^m\}$, $1 \leq \xi_i^k \leq n$, ahol $\xi_i^k \neq \xi_i^l$ ha $k \neq l$
- Például egy ξ_i baktérium, ha $n = 50$ és $m = 5$:

44	17	36	2	7
ξ_i^1	ξ_i^2	ξ_i^3	ξ_i^4	ξ_i^5

A kiértékelési függvény

- A baktérium jósági értéke határozza meg, hogy az információját szétterjeszti-e vagy sem
- Először a baktérium és a tanító adatkészlet alapján egy regressziós modellt számítunk:

$$f_i(\vec{x}) : X_{\xi_i^1} \times \dots \times X_{\xi_i^m} \mapsto Y$$

- Ennek a modellnek egy tesz adatkészleten számított átlagos négyzetes hibája alapján definiáljuk a kiértékelési függvényt:

$$\phi(\xi_i) = E(f_i, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} (f_i(\vec{x}) - y)^2$$

Változó hosszúságú eset

- Nem határozzuk meg előre, hogy mennyi feature-t szeretnénk kiválasztani
- A cél ilyenkor a legfontosabb változók kiválasztása egy n változót tartalmazó készletből
- A baktériumok hossza is különböző lehet, és folyamatosan változhat
- Az egyedek kiértékelése ezért a következő kritérium szerint történik:

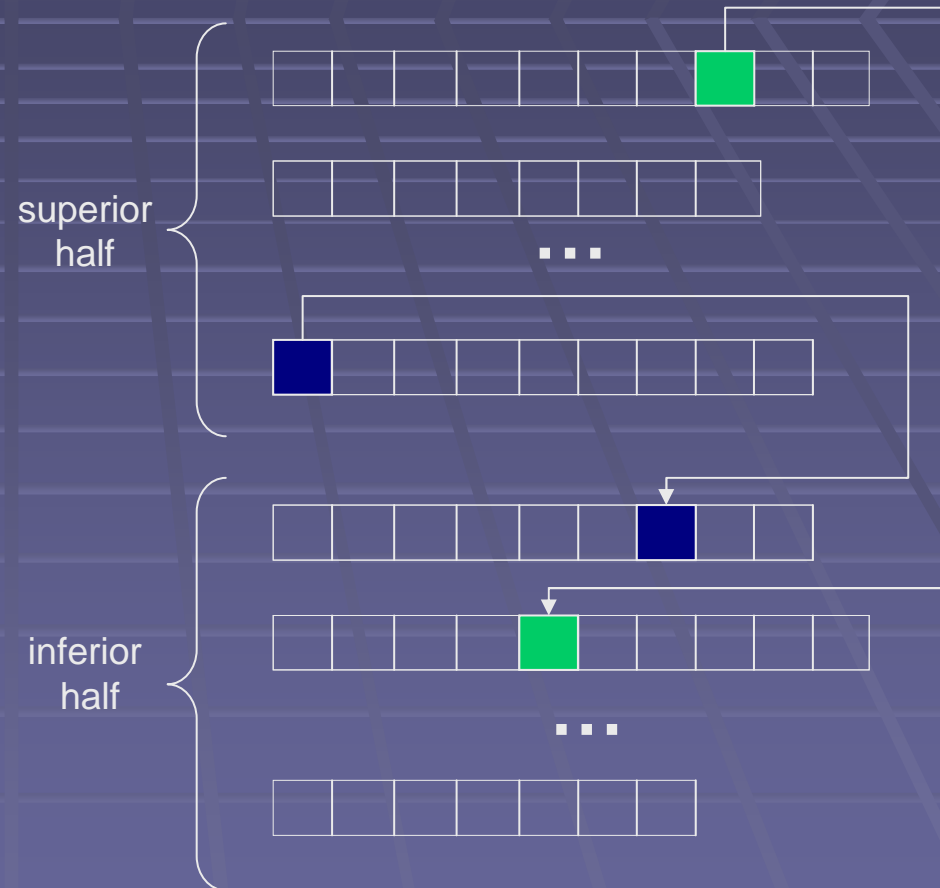
$$\phi(\xi_i) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} (f_i(\vec{x}) - y)^2 + \beta \frac{l(\xi_i)}{MAXLEN}$$

Továbbfejlesztett bakteriális mutáció

- A bakteriális mutáció paramétereit a klónok száma ($N_{klón}$) és a szegmens hossza (l_{BM})
- A mutáció nemcsak a tartalmat változtathatja meg, hanem a hosszt is
- Az új elemek hosszát véletlenszerűen választjuk a $l_{BM} \pm l_{BM}^*$ tartományból, ahol az l_{BM}^* paraméter határozza meg, hogy egy mutációnál mennyi lehet a hossz maximális megváltozása
- Amikor a baktérium egy szegmensét változtatjuk, figyelni kell arra, hogy az új szegmens elemei egyediek legyenek a baktériumban

Továbbfejlesztett génátadás

1. A populációt két részre osztjuk az egyedek alkalmassági értéke alapján
2. Egy baktériumot véletlenszerűen kiválasztunk a jobbik alpopulációból (forrás baktérium) egy másikat pedig a rossz egyedek közül (cél baktérium)
3. A forrás baktérium egy része felülírja a cél baktérium egy részét ha a forrásszegmens még nincs a célbaktériumban
Ez a ciklus ismétlődik N_{inf} -szer (“infekciók” száma)



Továbbfejlesztett génátadás

- A génátadás paramétereit az infekciók száma (N_{inf}) és a forrás szegmens hossza (l_{GT})
- A génátadás nemcsak a tartalmat változtathatja meg, hanem a célbaktérium hosszát is
- Az új elemek hosszát véletlenszerűen választjuk a célbaktériumban a $l_{GT} \pm l_{GT}^*$ tartományból, ahol az l_{GT}^* paraméter határozza meg, hogy mennyi lehet a hossz maximális megváltozása a célbaktériumban

Paraméterek

- N_{gen} : generációk száma
- N_{ind} : egyedek száma
- N_{clones} : másolatok (klónok) száma a bakteriális mutációban
- N_{inf} : génátadások (infekciók) száma a génátadásnál
- *MutationLength*: a mutációs szegmens hossza (I_{BM})
- *ModifiedMutationLength*: a maximális megengedett különbség egy szegmens mutáció előtti és utáni hossza között (I_{BM}^*)
- *GeneTransferLength*: a forrásszegmens hossza (I_{GT})
- *ModifiedGeneTransferLength*: a változás hossza a célbaktériumban (I_{GT}^*)
- *MAXLEN*: baktérium maximális megengedett hossza ($\cdot n$)
- β : trade-off paraméter a pontosság és a komplexitás között

Szabály kiválasztás

Motivációk

- Döntési problémákban a létrehozott modellnek teljesítenie kell domén specifikus minőségi kritériumokat
 - a specificitás és szenzitivitás fontos minőségi kritériumok
- A fuzzy szabálybázisok képesek az értelmezhetőségre és pontosságra vonatkozó igényeket megvalósítani
 - olyan alkalmazásokban használhatók jól, ahol a szakértői tudás nem explicit módon adott, de az eredményként kapott modell tudása elengedhetetlen
- A fuzzy szabálybázisokra alkalmazott induktív tanuló módszerek általában nem alkalmasak ilyen komplex és problémafüggetlen célfüggvények optimalizálására
 - a szabályok hatnak egymásra a szabálybázisban
 - a célfüggvényt szigorúan a probléma határozza meg

Motivációk

- Az evolúciós megközelítések képesek különböző optimalizációs kritériumokat kezelni
 - Még ezek a technikák is általában nagyon komplexek és időigényesek amikor a keresési tér nagyon nagyá válik
- A cél a komplexitási akadály leközdése, ennek érdekében elkülönítjük a szabálylétrehozási és a végső szabálybázis létrehozási feladatokat két külön lépésbe
 - Az első lépésben egy nagy szabályhalmazt hozunk létre ami csak minimális igényeket teljesít
 - A második lépésben ezen szabályoknak egy sokkal kisebb részhalmazát választjuk ki bakteriális evolúciós algoritmussal

A probléma megfogalmazása

- A predikciós mátrix jó képet ad az osztályozó teljes minőségéről
 - minden egyes esetre az aktuális feltételt és az előrejelzett diagnózist hasonlítjuk össze
 - a null predikciókat külön kell tekinteni

		<i>Feltétel</i>		
		pos	neg	
<i>Teszt eredmény</i>	pos	truePos	falsePos	posVals
	neg	falseNeg	trueNeg	negVals
	<i>Null</i>	nullPos	nullNeg	
		szenzitivitás	specificitás	

A probléma megfogalmazása

f az osztályozó (a fuzzy szabálybázis) S^* pedig az adathalmaz

$$\begin{aligned} \text{sens}(f, S^*) &= \frac{\text{truePos}}{\text{truePos} + \text{falseNeg} + \text{nullPos}} \\ \text{spec}(f, S^*) &= \frac{\text{trueNeg}}{\text{trueNeg} + \text{falsePos} + \text{nullNeg}} \\ \text{posVals}(f, S^*) &= \frac{\text{truePos}}{\text{truePos} + \text{falsePos}} \\ \text{negVals}(f, S^*) &= \frac{\text{trueNeg}}{\text{trueNeg} + \text{falseNeg}} \end{aligned}$$

- Ezen paraméterek segítségével lehet a döntési modell minimális követelményeit megfogalmazni
- Egy átfogó minőségi kritériummal lehet az optimális modellt identifikálni

A probléma megfogalmazása

- Az f osztályozó hibája egy adott S^* adathalmazra $F(f, S^*)$, amit a szenzitivitás és specificitás harmonikus átlagából számítunk

$$F(f, S^*) = \frac{1}{LE(MS(f), \bar{m})} \left(1 - 2 \frac{\text{sens}(f, S^*) + \text{spec}(f, S^*)}{\text{sens}(f, S^*) \cdot \text{spec}(f, S^*)} \right)$$

- Definiálunk egy $LE(s, m)$ fuzzy predikátumot, melynek jelentése "s nem nagyobb, mint m", szerepe pedig, hogy elkerüljük a végtelen nagy szabályhalmaz kialakulását:

$$LE(s, \bar{m}) = \begin{cases} 1 & s \leq \bar{m} \\ e^{-\frac{1}{2} \left(\frac{\bar{m}-s}{\bar{m}/2} \right)^2} & \text{egyébként} \end{cases}$$

MS az aktuális modell méret (szabályok száma vagy tagsági függvények száma)

s jelöli a modell méretet, \bar{m} a maximális szabálysám

A probléma megfogalmazása

- Amikor egy szabály létrejön, annak hatását nem ismerjük a teljes szabálybázis viselkedésére
 - gyakran helyi minőségi mértékeket, mint például tartó és konfidencia értékeket, használnak annak eldöntésére, hogy melyik szabályokat hozzanak létre a lokális optimum elérése érdekében

- Tartó:
$$\text{supp}(A \rightarrow C) = \frac{1}{|X|} \sum_{\vec{x} \in X} T(A(\vec{x}), C(\vec{x}))$$

ahol X a minták halmaza, T pedig egy t-norma

- Konfidencia:

$$\text{conf}(A \rightarrow C) = \frac{\text{supp}(A \rightarrow C)}{\text{supp}(A)}$$

Javasolt megközelítés

- Kifejező predikátumok létrehozása a vizsgált domén (adatok) számára
=> **Comp-FS** módszer javasolt erre a feladatra
- Összes szabály generálása melyek minimális követelményeknek tesznek eleget ezen predikátumok konfidencia és tartó értékeire vonatkozóan
=> nagyszámú (átlapolódó) szabály
FS-Miner módszer javasolt erre a feladatra
- Ezen szabályok olyan részalmazának kiválasztása, amely optimalizálja a teljes minőségi kritériumot
=> nagy keresési tér (komplexitás!)
Bakteriális Evolúciós Algoritmus alkalmazása erre a feladatra

A kódolási módszer

- Egy baktérium a probléma egy megoldásának felel meg ($\xi_i, i \in I$)
- A probléma jelen esetben m_i szabály kiválasztása egy n szabályt tartalmazó készletből ($m_i \leq n$), a baktérium szabály indexek vektorából épül fel $\xi_i = \{\xi_i^1, \dots, \xi_i^{m_i}\}$, $1 \leq \xi_i^k \leq n$, ahol ξ_i^k a k . szabály indexe, és $\xi_i^k \neq \xi_i^l$ ha $k \neq l$
- Például egy ξ_i baktérium $n = 50$ és $m_i = 5$ esetén:

44	17	36	2	7
ξ_i^1	ξ_i^2	ξ_i^3	ξ_i^4	ξ_i^5

A kiértékelési függvény

- A baktérium jósági értéke határozza meg, hogy az információját szétterjeszti-e vagy sem
- A ξ_i baktérium jósági értékét egy $\Phi(\xi_i)$ kiértékelési függvény segítségével számítjuk
- Egy ξ_i baktérium (szabály halmaz) hibája egy adott S^* adathalmazra $F(f, S^*)$, amit a szenzitivitás és specificitás harmonikus átlagából számítunk

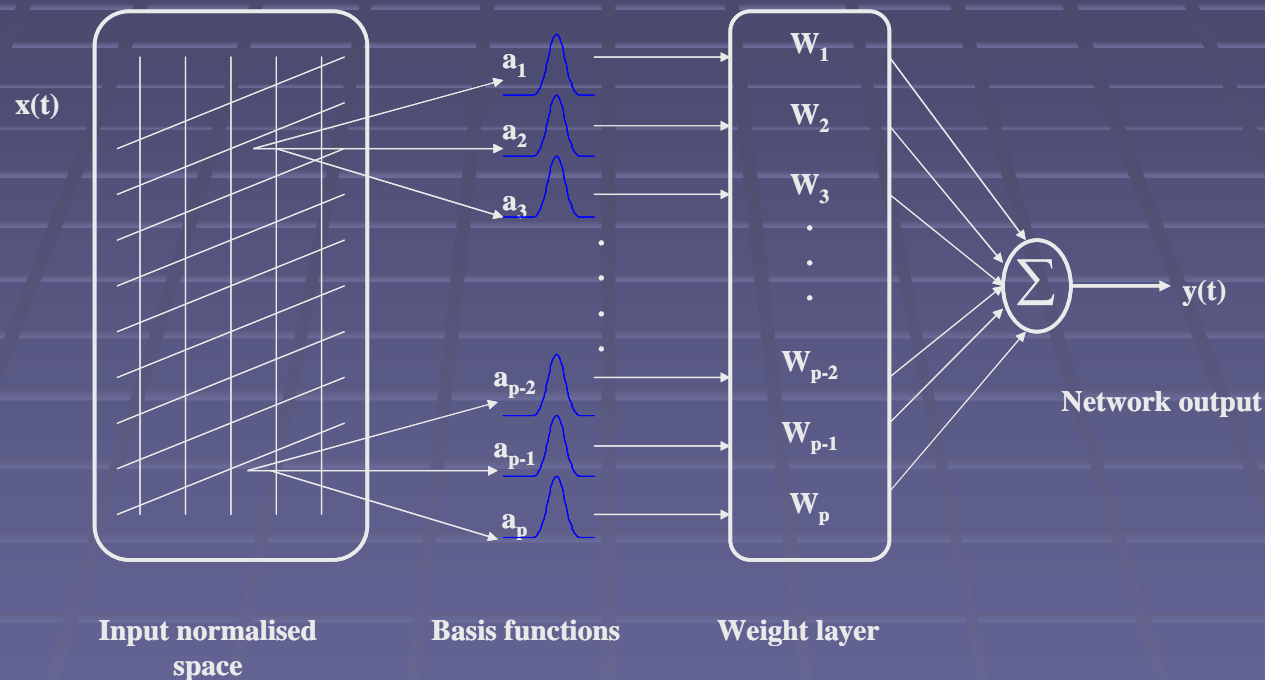
B-spline neurális hálózatok

B-spline neurális hálózatok

- A B-spline neurális hálózatok olyan asszociatív memória hálózatok, amelyek B-spline-okat használnak bázisfüggvényként
- Információ lokális tárolása
- Alkalmas online adaptív modellezésre és szabályozási alkalmazásokra
- Transzparens a rácsszerkezetű struktúrája miatt
- Általában kis (<5) bemeneti dimenziós problémákra alkalmazzák

B-spline neurális hálózatok

- Alap struktúra



- Normalizált bemeneti réteg

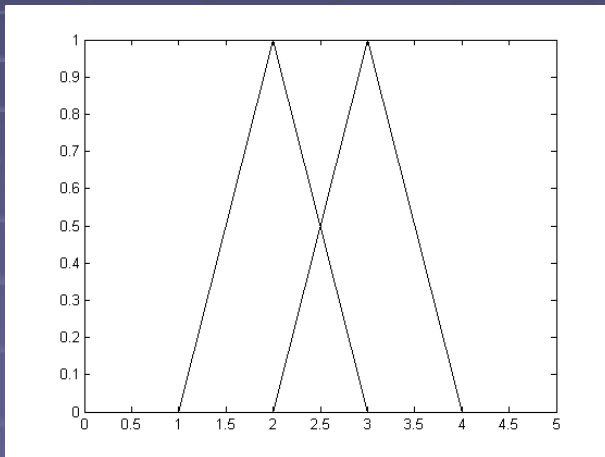
- Vektorokkal definiált rácsozat
- Cellák száma a rácsozatban

$$x_i^{\min} < \lambda_{i,1} \leq \lambda_{i,2} \leq \dots \leq \lambda_{i,r_i} < x_i^{\max}$$

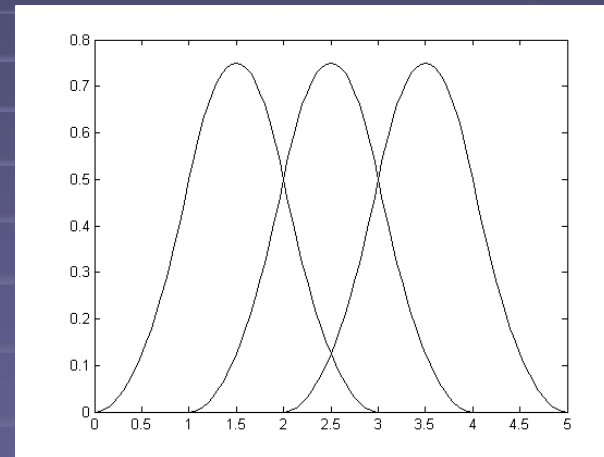
$$p' = \prod_{i=1}^n (r_i + 1)$$

B-spline neurális hálózatok

- Bázisfüggvények rétege
 - a spline-ok rendje meghatározza a bázisfüggvények alakját és tartóját



$k=2$



$k=3$

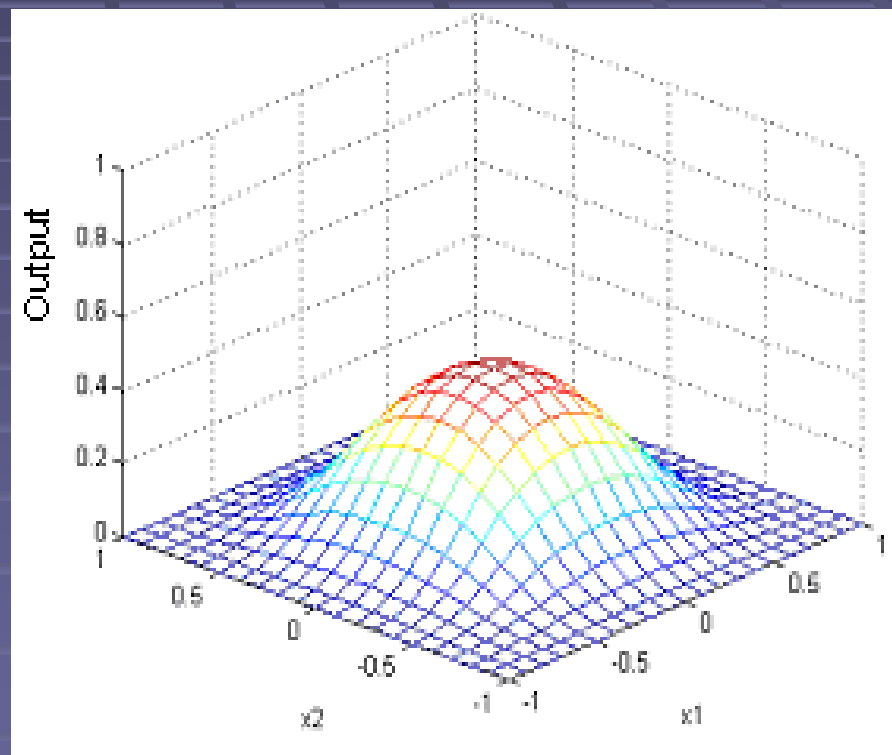
$$N_k^j(x) = \left(\frac{x - \lambda_{j-k}}{\lambda_{j-1} - \lambda_{j-k}} \right) N_{k-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-k+1}} \right) N_{k-1}^j(x)$$

$$N_1^j(x) = \begin{cases} 1 & \text{ha } x \in I_j \\ 0 & \text{egyébként} \end{cases}$$

B-spline neurális hálózatok

- Többváltozós bázisfüggvények képzése: az egyváltozós bázisfüggvények tenzorszorzatával

$$N_{\mathbf{k}}^j(\mathbf{x}) = \prod_{i=1}^n N_{\mathbf{k}_i, i}^j(\mathbf{x}_i)$$



B-spline neurális hálózatok

- Súlyréteg

- A bázisfüggvények kimeneteinek a lineáris kombinációja

$$y = \sum a_i w_i = a^T w, \quad a_i = N_k^i(x), \quad i = 1, \dots, p$$

- Egyszerre $p'' = \prod_{i=1}^n k_i$ aktív, ezért $y = \sum_{i=1}^{p''} a_{act(i)}(x) w_{act(i)}$

- Almodulok kompozíciója

- a számítási bonyolultság csökkentésére az n -dimenziós modellt általában kisebb dimenziós almodulokra dekomponáljuk

$$y(x) = \sum_{u=1}^{n_u} S_u(x_{\underline{u}})$$

B-Spline neurális hálózatok tervezése

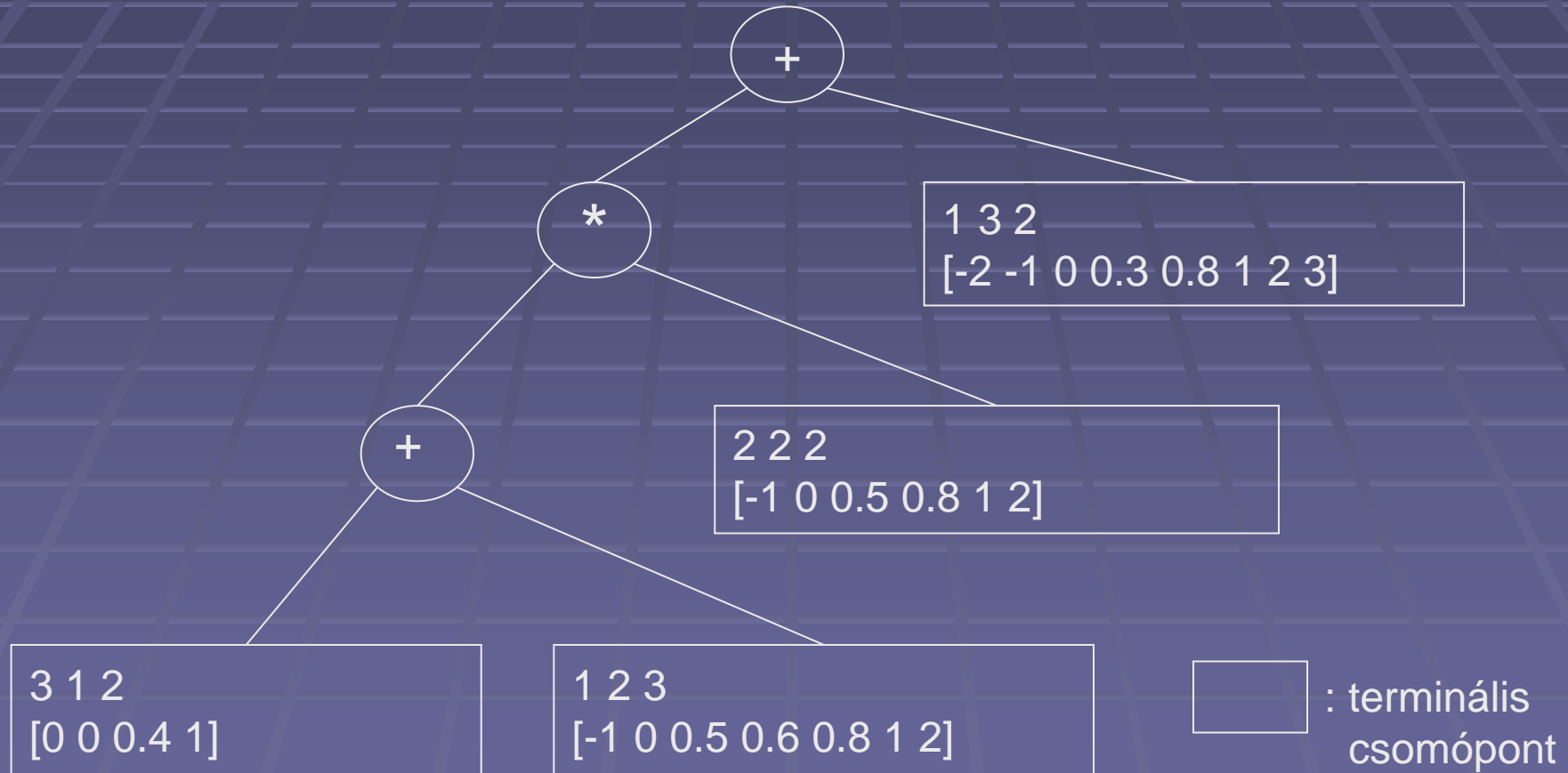
- Almodulok számának a meghatározása
- Mindegyik almodulra a bemeneti változók megadása
- A spline-ok rendjének meghatározása mindegyik bemeneti változóra
- Belső csomópontok száma mindegyik bemenetre
- Belső csomópontok helyzete mindegyik bemenetre
- A lineáris kimeneti súlyok értékének meghatározása

Bakteriális programozás

Bakteriális programozás

- A bakteriális evolúciós algoritmusok és a genetikus programozás ötvözése
- Ugyanazokat az operátorokat használja, mint a bakteriális evolúciós algoritmus
 - Bakteriális mutáció
 - Génátadás
- Az egyedek kifejezésfával vannak reprezentálva, mint a genetikus programozásban
 - Terminális csomópontok
 - Függvény csomópontok

Egy példa B-spline hálózatok kifejezésekéjára



$$y(\mathbf{X}) = f(\mathbf{X}_3 \times \mathbf{X}_2) + f(\mathbf{X}_1 \times \mathbf{X}_2) + f(\mathbf{X}_1)$$

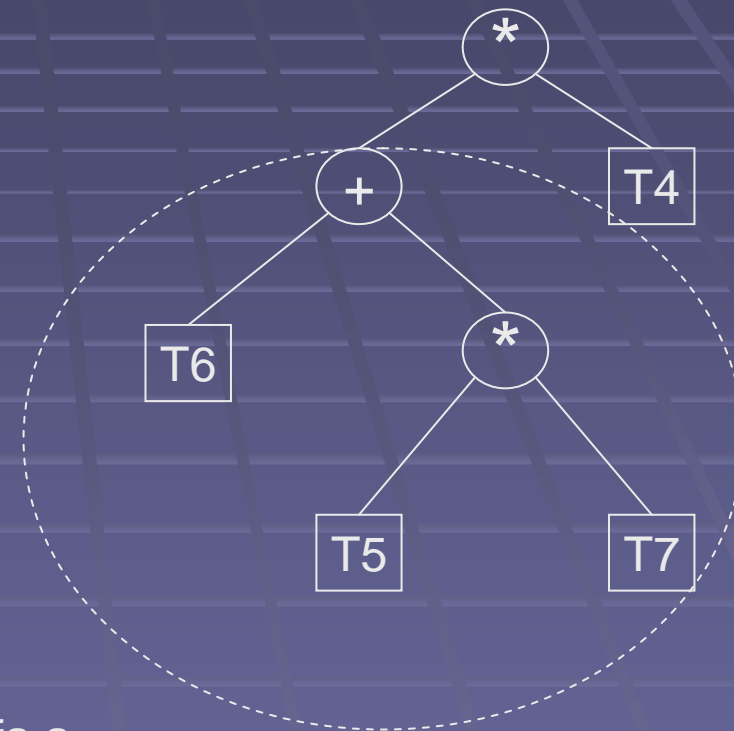
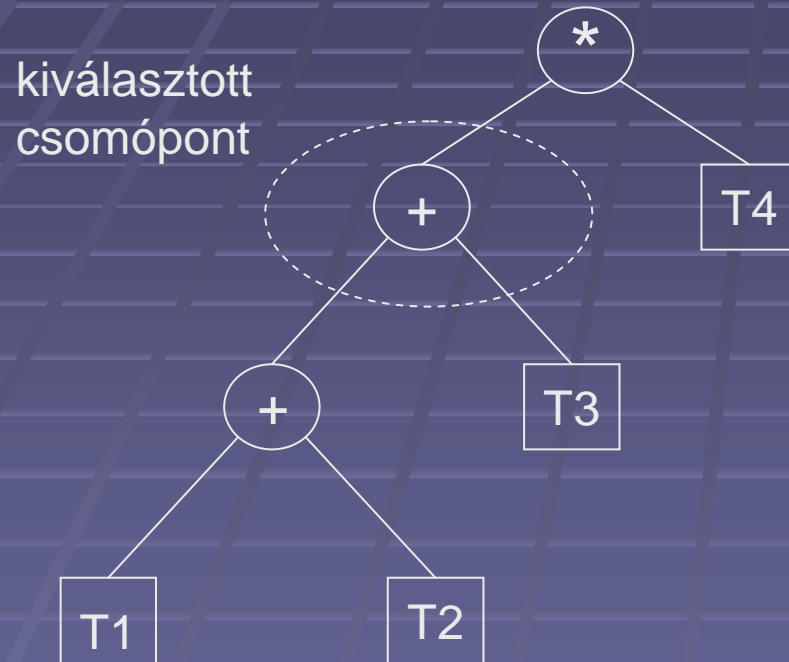
Csomópontok

- Függvény csomópontok:
 - „+”: almodulok összeadása
 - „*”: almodulok létrehozása kisebb dimenziós almodulokból
 - „/”: almodulok szétvágása kisebb dimenziós almodulokra
- Terminális csomópont:
 - bemeneti változó
 - spline rendje
 - belső pontok száma
 - pontok elhelyezkedése

Evolúciós folyamat

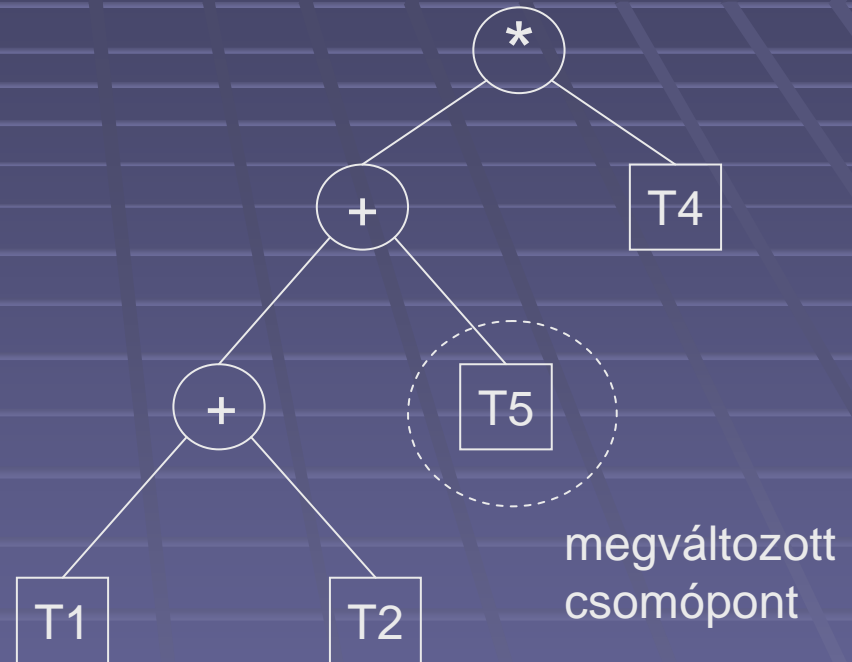
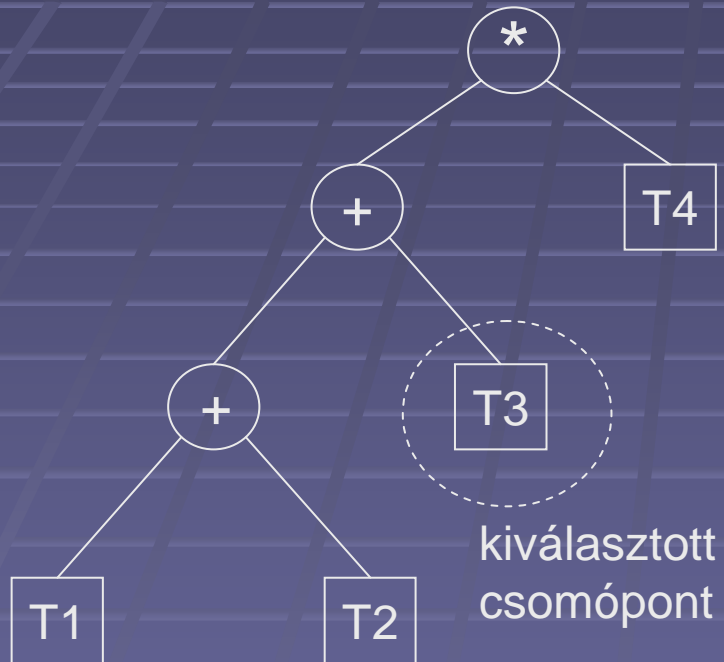
1. Kezdeti populáció létrehozása
2. Bakteriális mutáció alkalmazása minden baktériumra
3. Génátadás alkalmazása a populációra
4. Ha teljesül a megállási kritérium, akkor a módszer véget ér, különben folytatódik a 2. lépéssel
 - A megállási kritérium általában a maximális generációszám elérését jelenti

Függvény mutáció



új részfa a
függvény mutáció után

Terminális mutáció

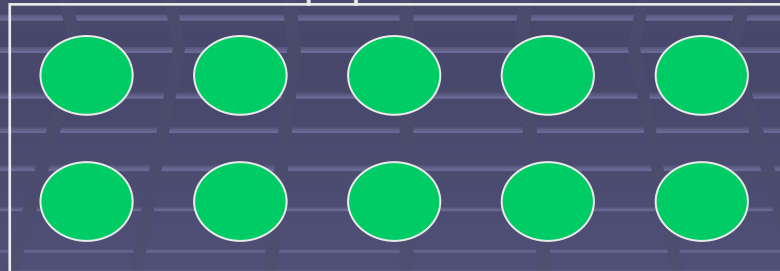


Terminális mutáció

- Többféle lehetőség van a terminális csomópont megváltoztatására
 1. A terminális teljes lecserélése
 2. A változó lecserélése
 3. A spline rendjének lecserélése
 4. Egy belső pont véletlenszerű áthelyezése
 5. N db újabb belső pont véletlenszerű felvétele
 6. N db belső pont törlése
- A terminális mutációs ráta egy vektort jelent
 - $p_{\text{term_mut}} = [\%1 \ \%2 \ \%3 \ \%4 \ \%5 \ \%6]$, ahol $\%i$ jelenti az i . típusú mutációs rátát

Génátadás

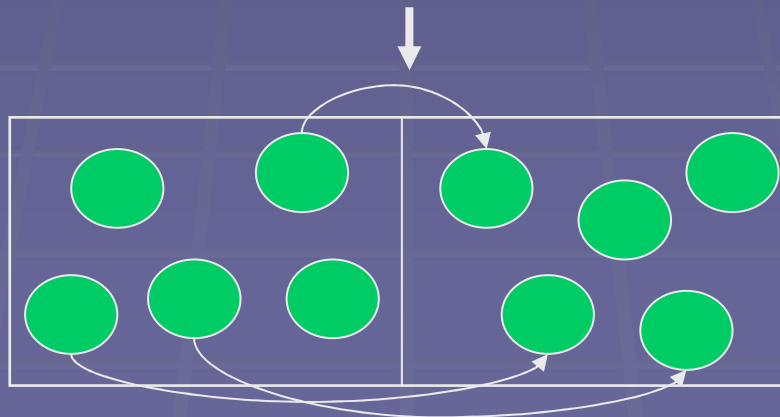
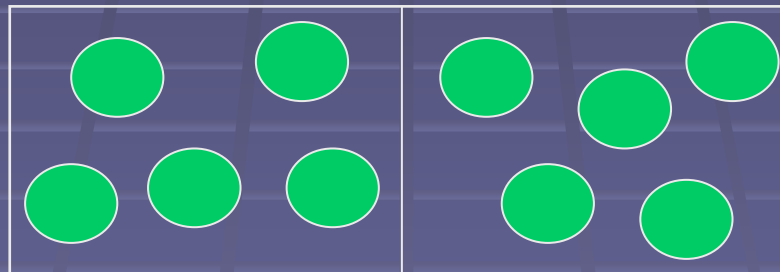
a populáció



egyedek rendezése

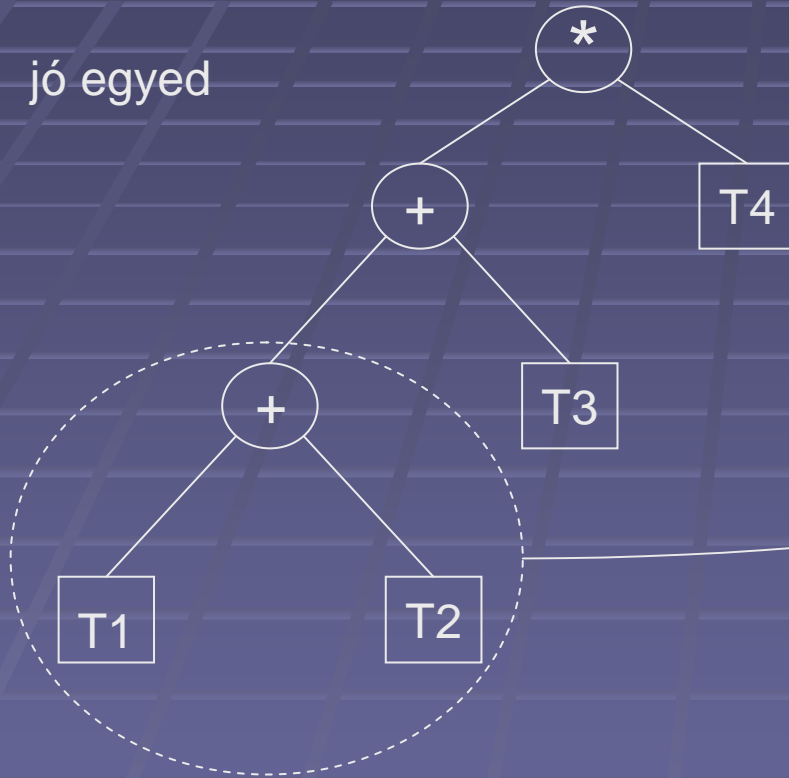
jó egyedek

rossz egyedek

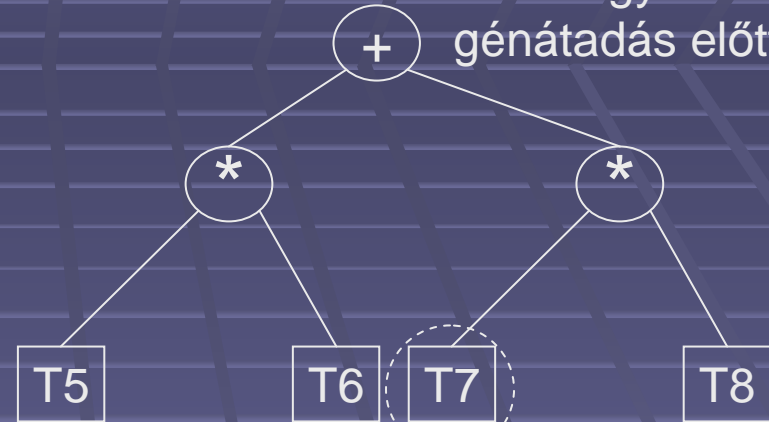


Génátadás

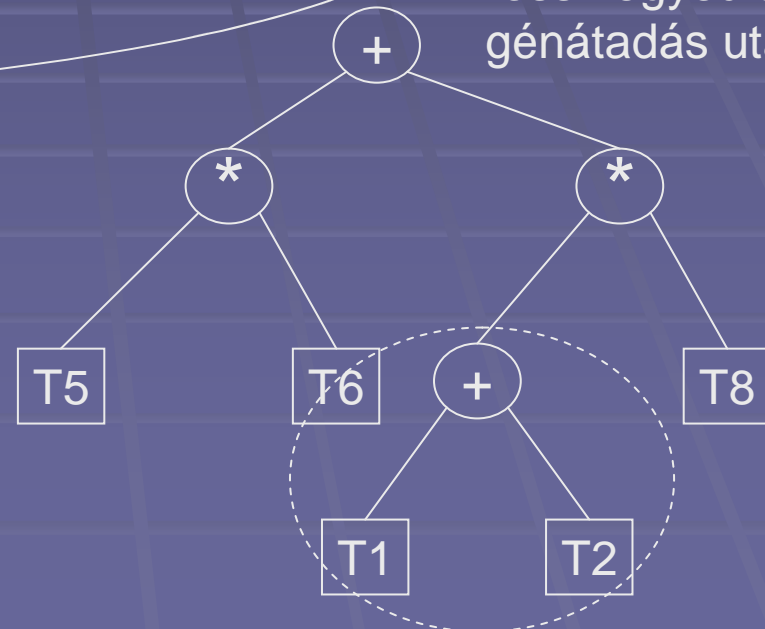
jó egyed



rossz egyed az
génátadás előtt



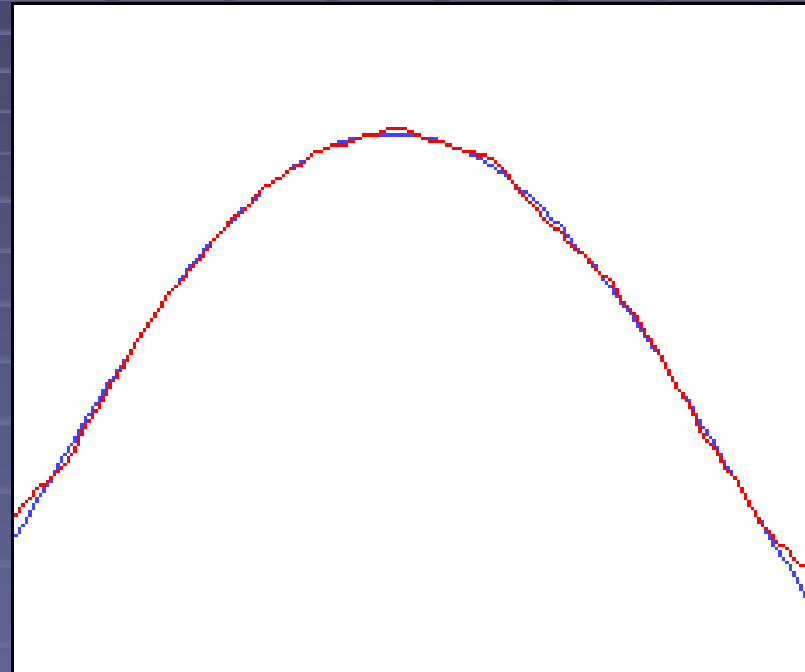
rossz egyed a
génátadás után



Szimulációs eredmények

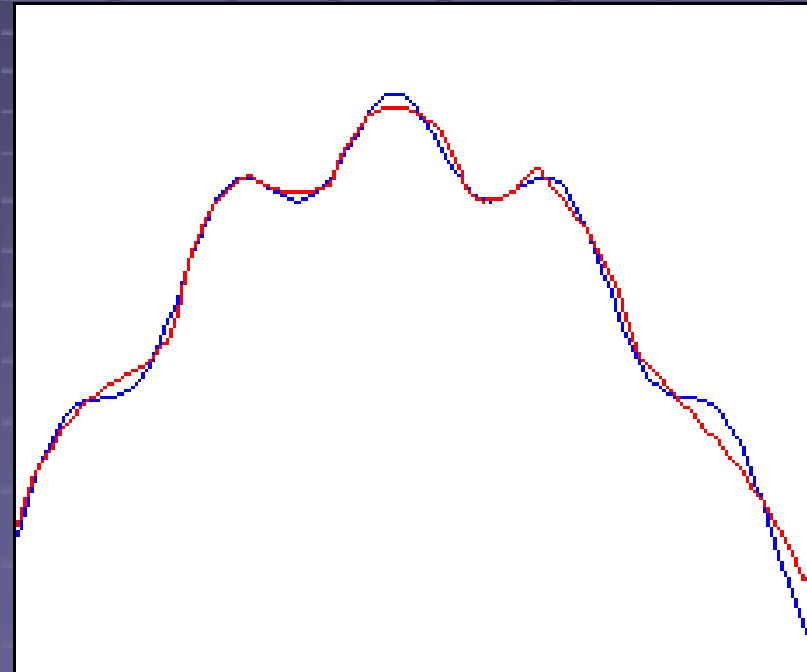
BEA fuzzy szabálybázis optimalizációjára

- Egyváltozós függvény
 - Egy szinuszos komponens
- 10 szabály
- 200 generáció



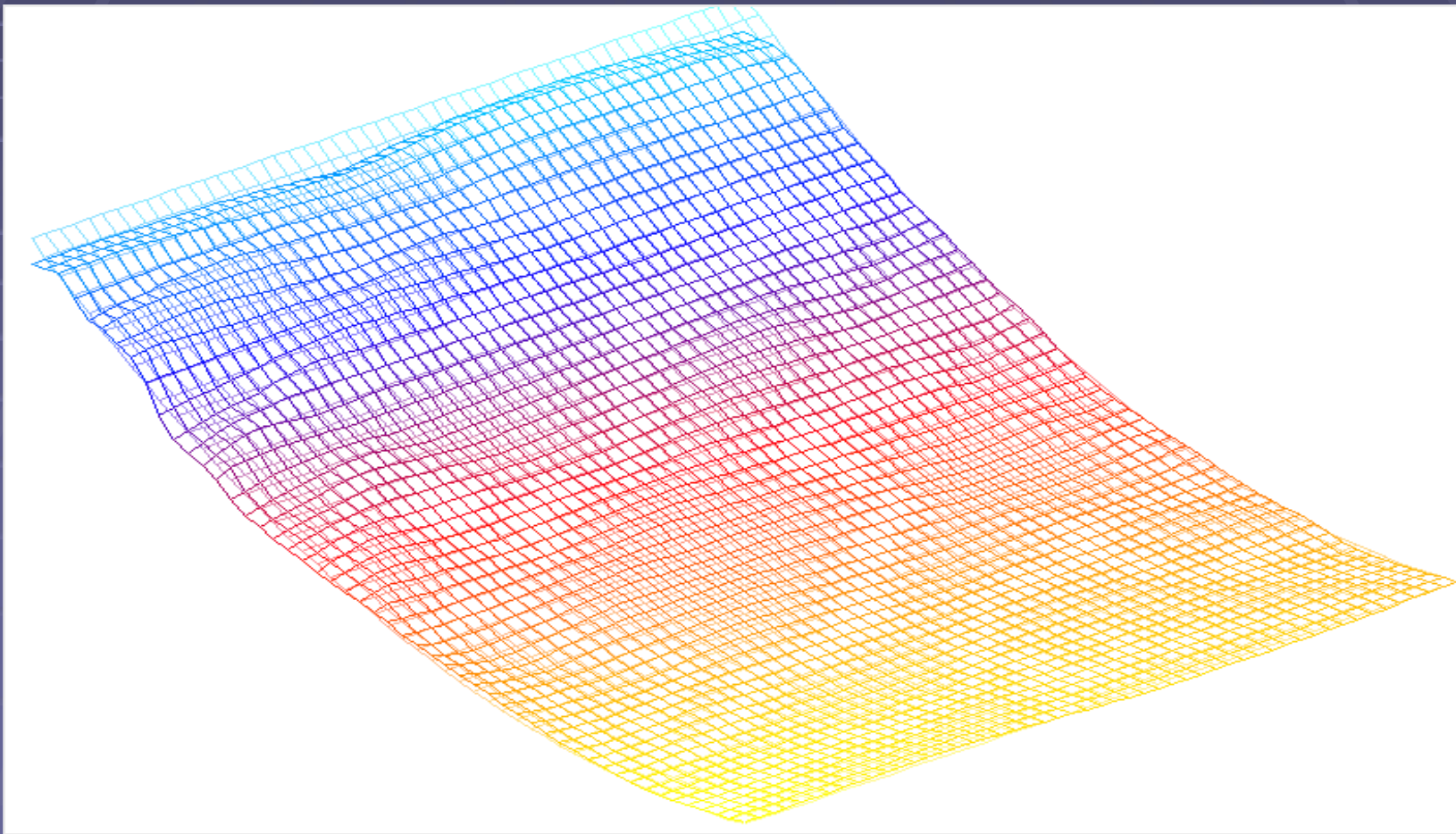
BEA fuzzy szabálybázis optimalizációjára

- Egyváltozós függvény
 - Két szinuszos komponens
- 10 szabály
- 200 generáció



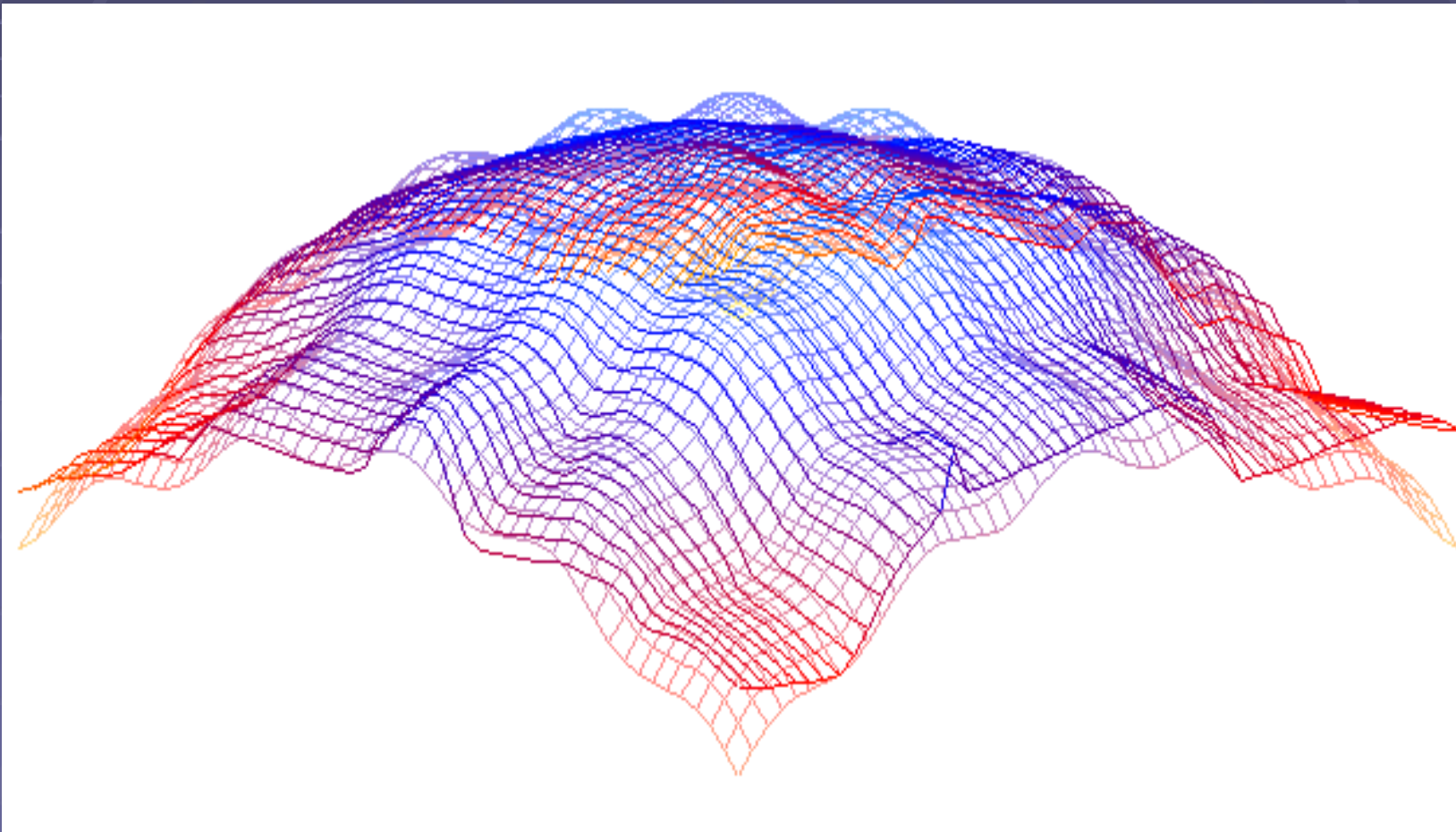
BEA fuzzy szabálybázis optimalizációjára

- Kétváltozós függvény ($f(\mathbf{x})=x_1+x_2^2$)



BEA fuzzy szabálybázis optimalizációjára

- Kétváltozós függvény két szinuszos komponenssel



BEA fuzzy szabálybázis optimalizációjára

6 változós függvény: $y = x_1 + x_2^{0.5} + x_3x_4 + 2e^{2(x_5-x_6)}$

A *megsemmisítés* szabály redukciós operátor β paraméterének hatása:

kezdeti
szabálysám: 10

$\gamma = 5\%$

β	Átlagos szabály szám	Tanítási hiba	Teszt hiba
5	4.7	5.23	8.94
6	5.4	5.02	8.47
8	7.6	4.52	8.32
10	7.8	4.13	8.17

BP a B-spline neurális hálózatokra

- 5 változós probléma
 - PID hangolási feladat
 - 2 rendszer identifikációs paraméter, 3 PID paraméter (k_p, T_i, T_d)
- 145 tanítóminta, 64 tesztminta
- Hibakritériumok:

$$MSRE = \frac{1}{N_{pat}} \sum_{i=1}^{N_{pat}} \frac{(t_i - y_i)^2}{y_i^2}$$

$$\%MRE = \frac{100}{N_{pat}} \sum_{i=1}^{N_{pat}} \left| \frac{t_i - y_i}{y_i} \right|$$

$$BIC = m \ln(RMS) + n \ln(m)$$

t_i : kívánt kimenet az i . mintára

y_i : modell kimenet az i . mintára

m : tanítóminták száma

n : modell komplexitás
(bázisfüggvények száma)

RMS : Root-Mean-Square

BP a B-spline neurális hálózatokra

- A módszerek paraméterbeállításai

Paraméterek	GP	BP
N_inf	-	4
N_clones	-	8
N_ind	80	10
N_gen	10	10
Keresztezés	populáció 50%-a	-
Mutációs arány	0.8	-

- A terminális mutációs ráta
[5%, 10%, 5%, 10%, 60%, 10%]

BP a B-spline neurális hálózatokra

- Átlagértékek az összes egyedre nézve

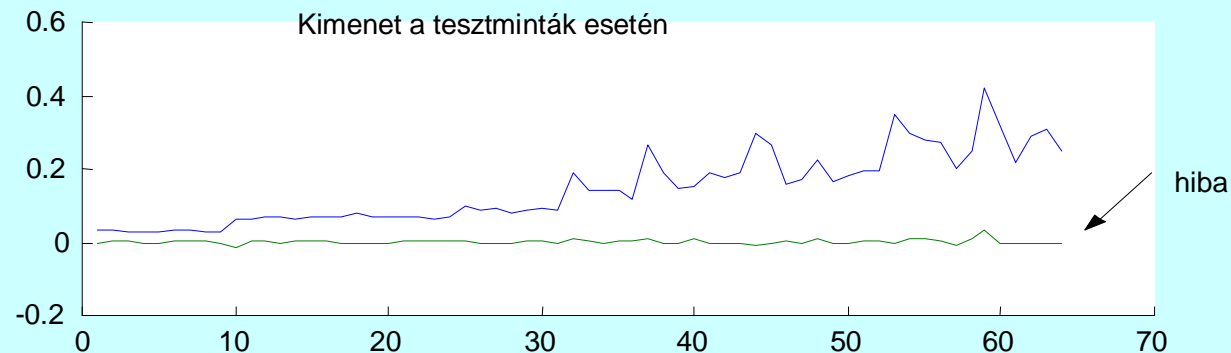
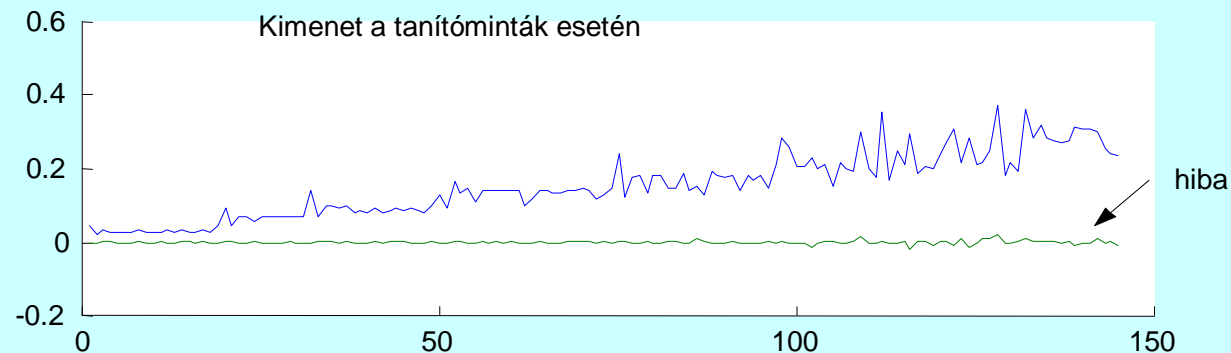
módszer	BIC	MSE	MSRE	%MRE	MSE_v	MSRE_v	%MRE_v	Kompl.
<i>GP</i>	-1361	5.2×10^{-5}	1.8×10^{-3}	2.0	1.1×10^{-2}	10.1	17	29.7
<i>BP</i>	-1376	5.5×10^{-5}	1.6×10^{-3}	1.9	2.4×10^{-4}	1.2×10^{-2}	4	32.6

- Legjobb egyed értékei

módszer	BIC	MSE	MSRE	%MRE	MSE_v	MSRE_v	%MRE_v	Kompl.
<i>GP</i>	-1414.6	1.2×10^{-5}	4.7×10^{-4}	1.5	4.4×10^{-5}	1.1×10^{-3}	2.5	34
<i>BP</i>	-1467.7	4.3×10^{-6}	2.1×10^{-4}	0.97	1.3×10^{-3}	1.1×10^{-2}	5.8	65

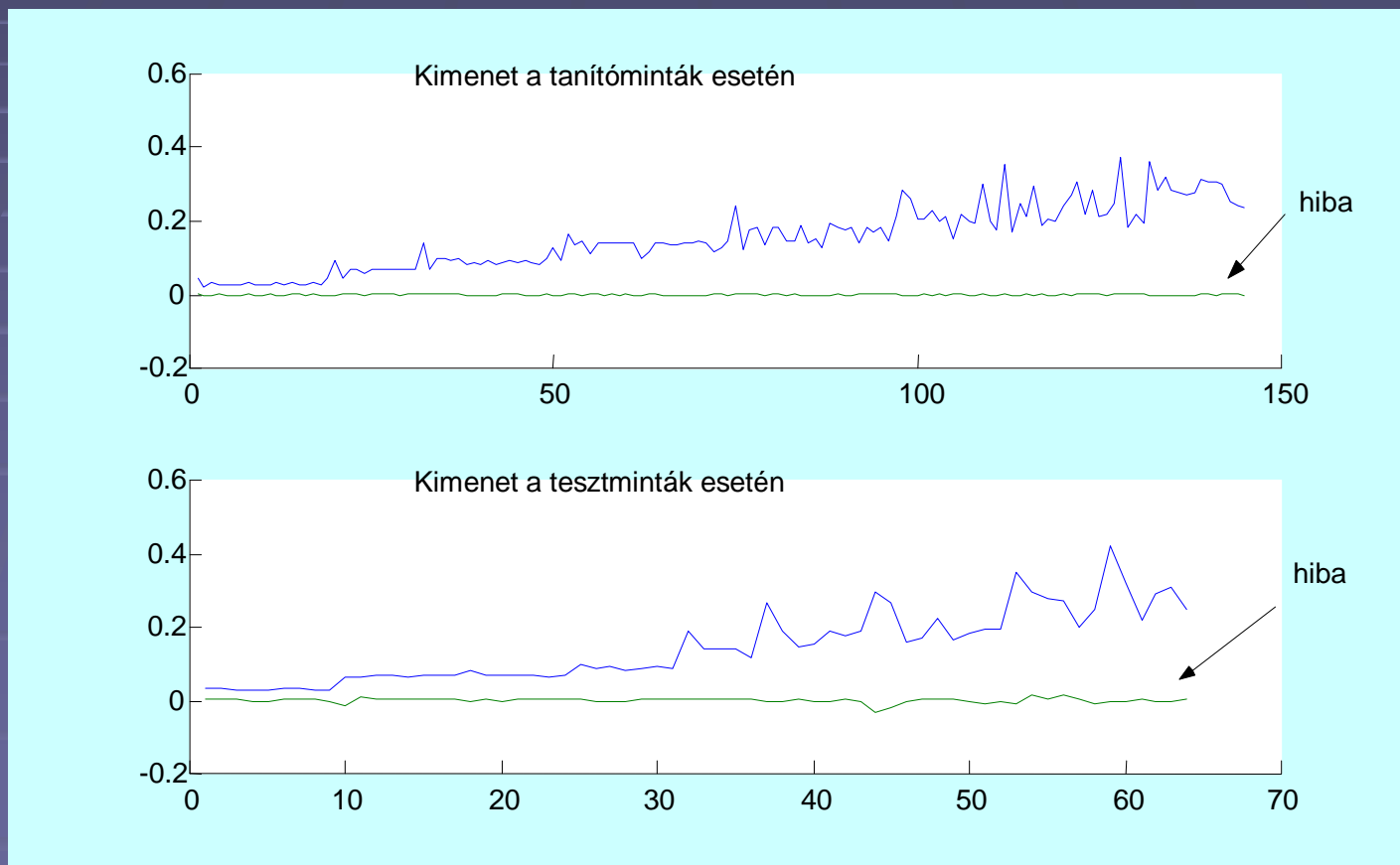
BP a B-spline neurális hálózatokra

- Tanító és tesztelő minták illeszkedése GP módszerrel a legkisebb BIC-jű modell esetén (felül), és a legkisebb %MRE_v-jű modell esetén (alul)



BP a B-spline neurális hálózatokra

- Tanító és tesztelő minták illeszkedése BP módszerrel a legkisebb BIC-jű modell esetén (felül), és a legkisebb %MRE_v-jű modell esetén (alul)



BEA változó kiválasztásra

- Három különböző regressziós módszert használtunk az algoritmus viselkedésének az elemzésére
 - Legkisebb négyzetes (LS) optimalizáció
 - Fuzzy szabály indukciós módszer regularizált optimalizációval (RENO)
 - Induktív fuzzy döntési fa tanulási módszer (FS-ID3)

BEA változó kiválasztásra

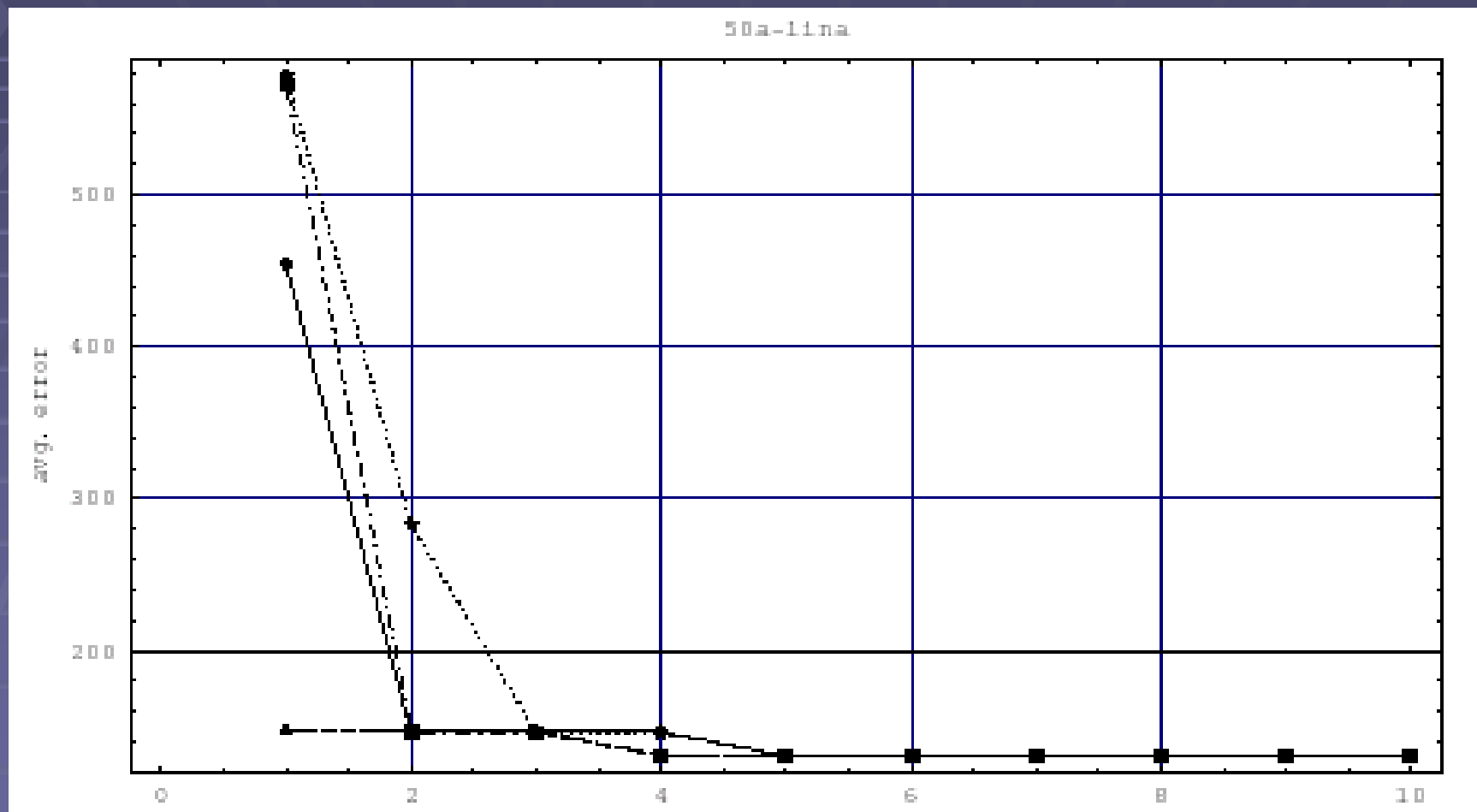
- A bakteriális optimalizációt mindegyik módszerre alkalmaztuk, hogy megoldást adjon 3 nagy-dimenziós probléma változó kiválasztására

$$f_{20}(\vec{x}) = x_1 x_2^2 x_{13}^3 - x_{20} + 5 \sin(x_{16}) - 25 \cos(x_5 x_{18}) + e^{x_3 x_5} + x_4 x_{19} + x_{10}^2 + x_{11}^5$$

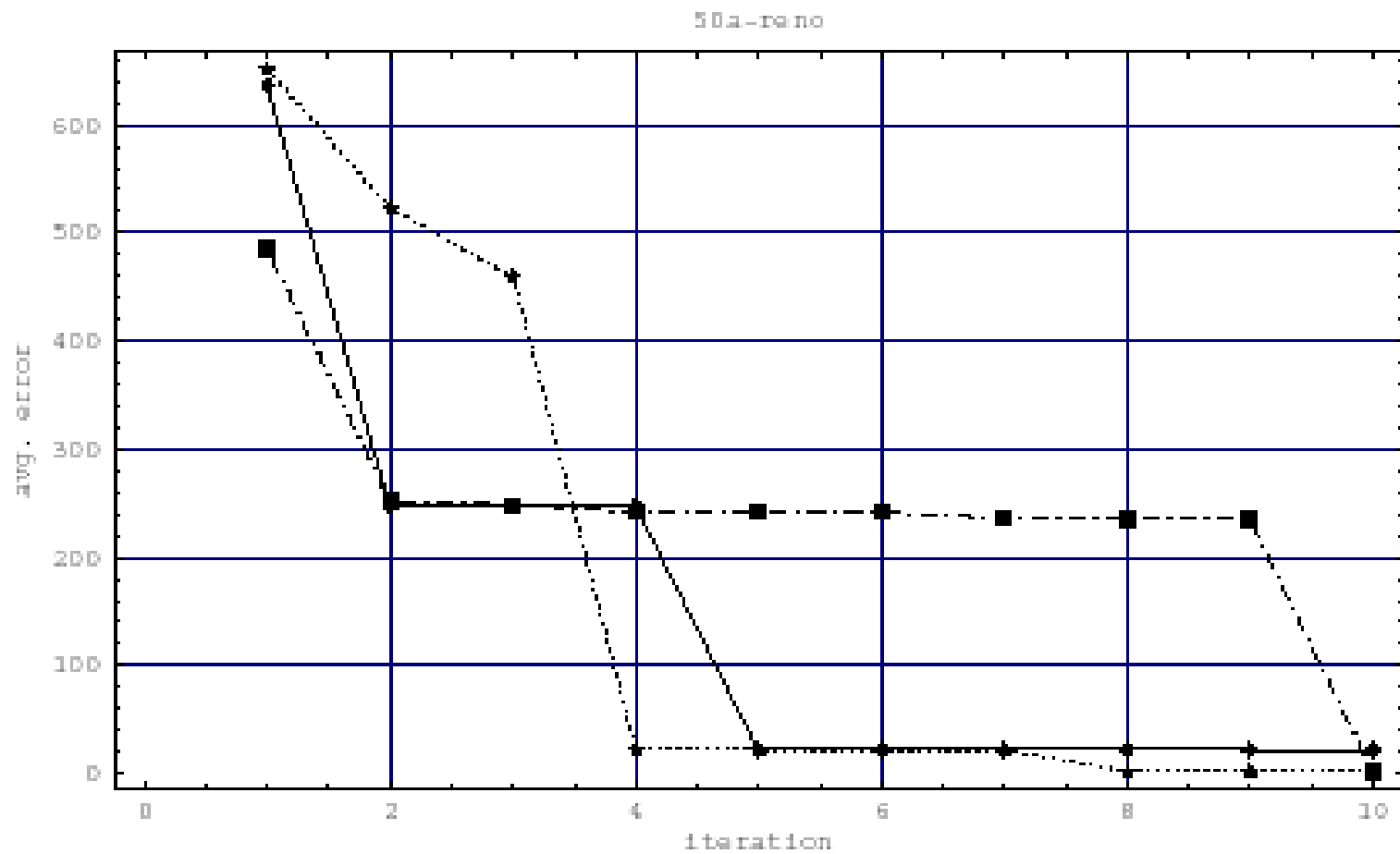
$$f_{50a}(\vec{x}) = x_1 x_{40}^2 + 0.01 x_{12} + 0.01 x_{15} + 3.1 x_{49}$$

$$f_{50b}(\vec{x}) = x_1 + \sqrt{x_2} + x_3 x_4 + 2e^{2(x_5 - x_6)} + x_7 x_8 x_9 - \frac{x_{10}}{x_{11} x_{12}} + 0.5 x_{13}^3 + x_{14}^2 x_{15} - x_{16} - x_{17} + 50 \frac{x_{18} + x_{19}}{x_{20}} + x_{46}$$

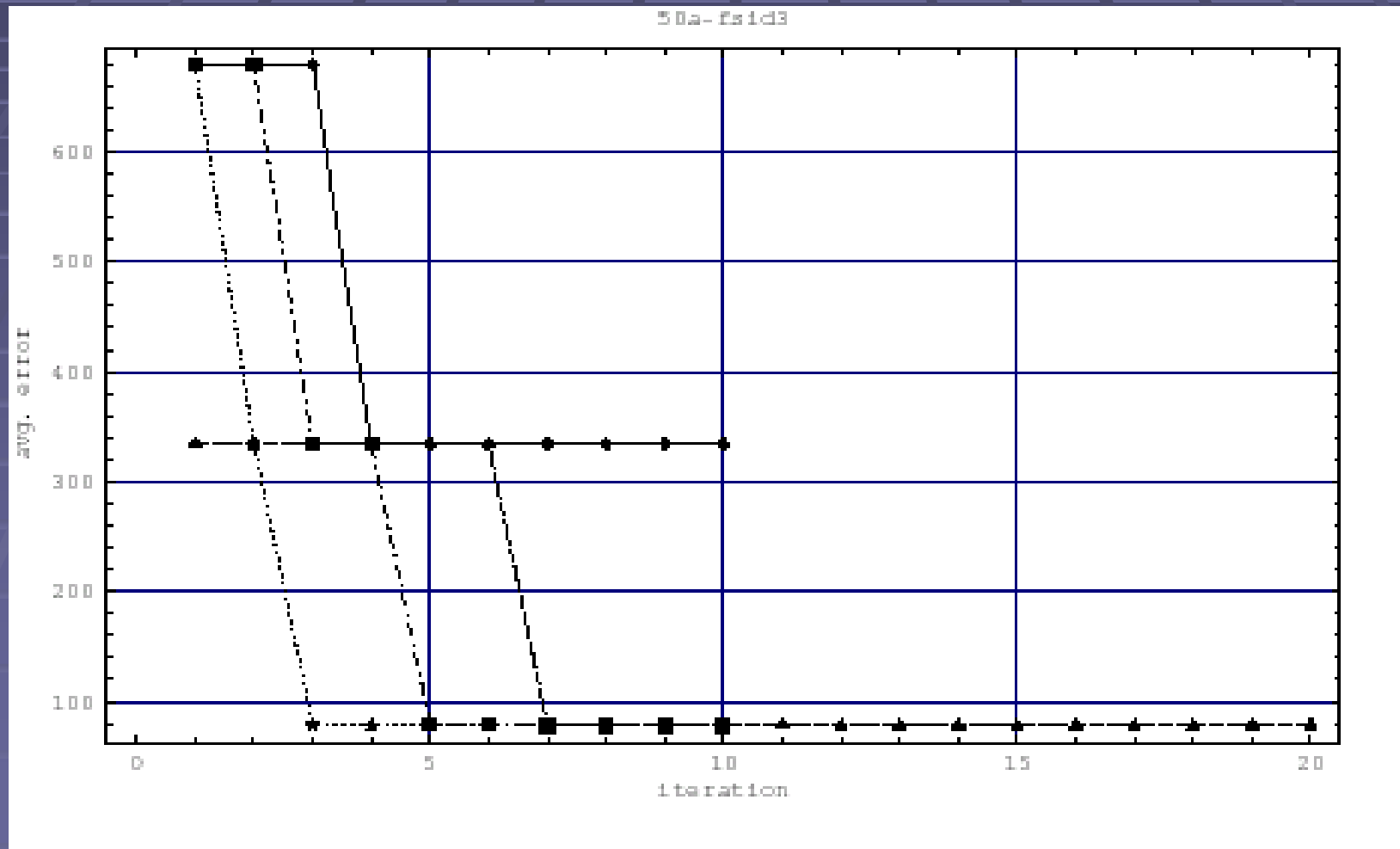
BEA változó kiválasztásra



BEA változó kiválasztásra



BEA változó kiválasztásra



BEA változó kiválasztásra

Változó hosszúságú eset

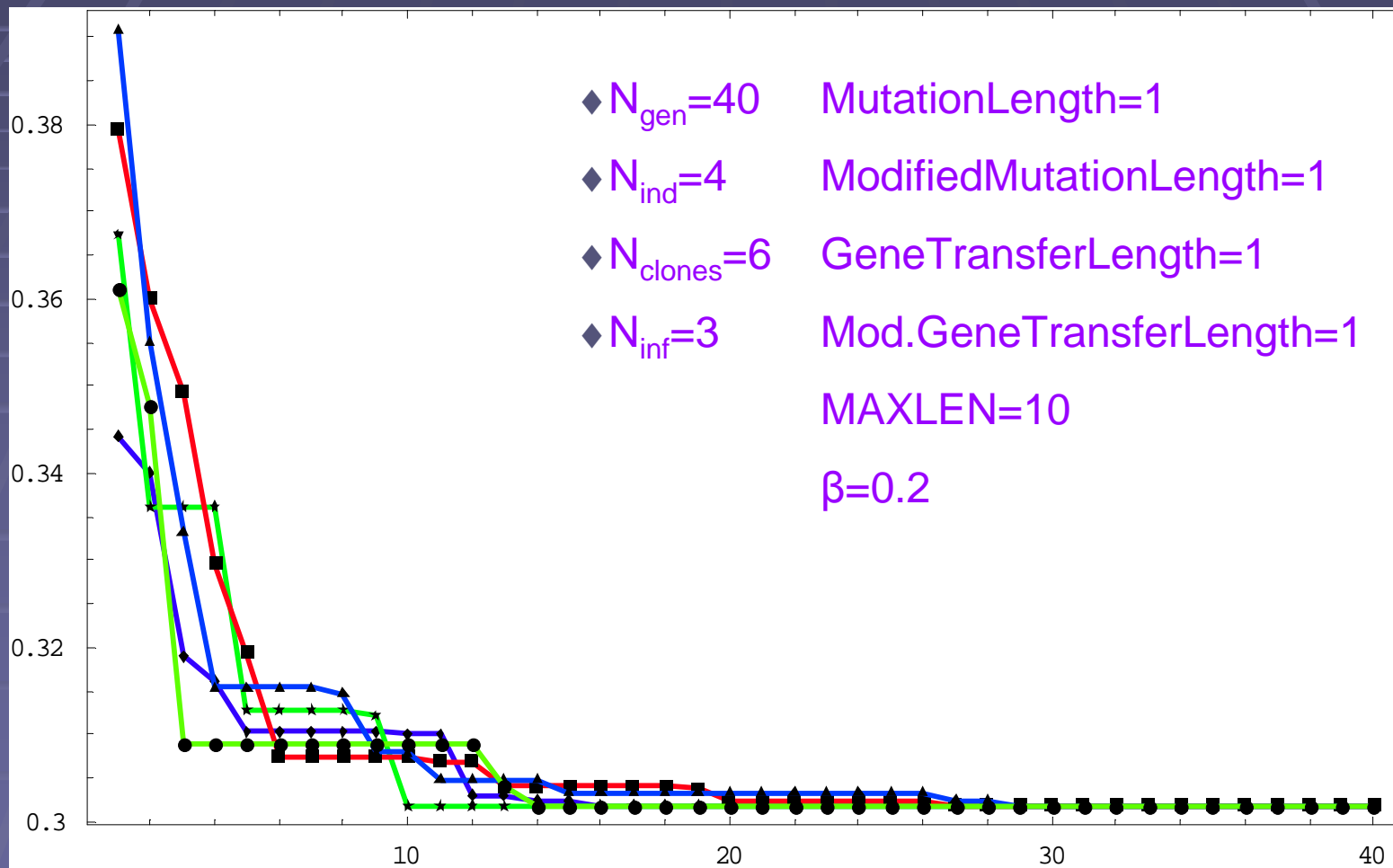
- Az algoritmus továbbfejlesztett változatának teszteléséhez a következő nagy dimenziós számú problémát definiáltuk:

$$f_{20}(\vec{x}) = x_1 x_2^2 x_{13}^3 - x_{20} + 5 \sin(x_{16}) - 25 \cos(x_5 x_{18}) + e^{x_3 x_5} + x_4 x_{19} + x_{10}^2 + x_{11}^5$$

Ennek bemeneti változóin egy lineáris, exponenciális, négyzetes és köbös transzformációt hajtottunk végre → 80 lehetséges bemeneti feature-t kaptunk

- Predikciós modell készült az adatokra történő, a bemeneti változók lineáris kombinációjával való legkisebb-négyzetes illeszkedés számításával

BEA változó kiválasztásra Változó hosszúságú eset



BEA szabály kiválasztásra

Orvosi alkalmazás

- Hepatitis analízis

- 562 felnőtt hepatitises beteg eset rekordja
- 231 nem fertőzött beteg eset rekordja
- Bécsi Általános Kórház (AKH-Wien) 1976 és 1986 között
- A diagnózisokat orvosok ellenőrizték (=> gold standard)

- A következő paramétereket vizsgálták:

Kor

Albumin

Alpha 1 Globulin

Beta Globulin

Gamma-Glutamyltranspeptidase

Alanin-Aminotransferase (GPT)

Bilirubin

Nem

Alkalin Phosphatase

Alpha 2 Globulin

Gamma Globulin

Aspartat-Aminotransferase (GOT)

Lactate Dehydrogenase

- Bináris döntési probléma

- fertőzött (pozitív) és nem fertőzött (negatív) eseteket kell megkülönböztetni

BEA szabály kiválasztásra

- Az *FS-Miner* 466 szabályt hozott létre
 - minimum tartó 0.01
 - minimum konfidencia 0.06
 - az átlapolódó szabályokat nem távolítottuk el
- Ezután a BEA került alkalmazásra a következő paraméterekkel:

N_{gen}	5
N_{ind}	4
N_{clones}	5
$I_{BM} \pm I_{BM}^*$	3 ± 2
N_{inf}	20
$I_{GT} \pm I_{GT}^*$	2 ± 2
MAXLEN	20

BEA szabály kiválasztásra

- Az eredményeket három tanuló algoritmus által kapott eredményekkel hasonlítottuk össze ugyanazon predikátumhalmazok használata mellett, hogy biztosítsuk a kompatibilitást
 - *FS-ID3*: egy fuzzy döntési fa tanuló módszer
 - *FS-FOIL*, *FS-Miner*: szabály tanuló algoritmusok

BEA szabály kiválasztásra

5x Cross Validáció	BEA	FS-ID3	FS-FOIL	FS-Miner
$F(f,S^*)$	0.234	0.28	0.473	1
Szenzitivitás	0.759	0.791	0.755	0.861
Specifititás	0.781	0.695	0.424	0
PosVals	0.927	0.87	0.915	0.891
NegVals	0.606	0.59	0.651	0
FractionCorrect	0.798	0.766	0.844	0.891
ChiSquared	49.305	34.349	39.533	0
PLevel	0	0	0	0
Kölcsönös információ	0.259	0.167	0.237	0
Normalizált kölcsönös Információ	0.297	0.189	0.304	0
Modell méret	18.2	19.8	7.2	36
Null predikciók arány	0.038	0	0.219	0.32

- Az FS-Miner nem sikeres, mert csak pozitív eseteket jelez előre
 - magas szenzitivitás
 - nincs specifititás
- FS-ID3
 - egy kicsit jobb, mint BEA a szenzitivitás szempontjából
 - de jelentősen rosszabb a specifitásra vonatkozóan
- FS-FOIL
 - összehasonlítható a BEA-val a szenzitivitás alapján
 - sikertelen specifititás
- A BEA jelentősen jobb a χ^2 statisztika vonatkozásában és a kölcsönös információs mérték alapján
- A BEA jobb eredményt ad az F mértéket nézve, ami az optimalizációs feladat aktuális célfüggvénye

BEA szabály kiválasztásra

- Az eredményként kapott szabálybázis

Class	Condition
HEP2_Is_NEG	← ALKPHOS_IsAtLeast_M && GPT_IsAtMost_L && GAMMA_GT_IsAtMost_L && GAMMA_IsAtMost_L && LDH_IsAtLeast_L SERBILI_Is_VL && ALKPHOS_Is_VL && BETA_Is_M ALKPHOS_IsAtMost_L && LDH_Is_VL && ALBUMIN_IsAtLeast_M ALKPHOS_IsAtMost_L && LDH_Is_VL && SEX_Is_W ALKPHOS_IsAtLeast_M && GAMMA_GT_Is_VL && GPT_Is_VL && SEX_Is_W && GAMMA_IsAtMost_M GPT_Is_VL && ALKPHOS_Is_VL && SERBILI_IsAtMost_L
HEP2_Is_POS	← ALKPHOS_IsAtLeast_M LDH_IsAtLeast_M GOT_IsAtLeast_M ALKPHOS_IsAtLeast_H ALKPHOS_Is_VL && AGE_IsAtLeast_L && LDH_IsAtLeast_L && ALPHA2_IsAtMost_L ALKPHOS_Is_VL && GAMMA_GT_IsAtMost_H && SEX_Is_M && AGE_IsAtLeast_L && ALPHA1_IsAtMost_H GAMMA_GT_IsAtLeast_L && ALBUMIN_IsAtMost_H ALKPHOS_Is_VL && GAMMA_GT_IsAtMost_H && AGE_IsAtMost_L && ALPHA2_IsAtMost_L && ALPHA1_IsAtMost_H SERBILI_IsAtLeast_L && LDH_IsAtLeast_L ALKPHOS_Is_VL && GAMMA_GT_IsAtMost_H && SEX_Is_M && ALPHA2_IsAtMost_L && ALBUMIN_IsAtMost_H ALKPHOS_Is_VL && GAMMA_GT_IsAtMost_H && SEX_Is_M && AGE_IsAtLeast_L && ALPHA2_IsAtMost_M && GPT_IsAtMost_H ALKPHOS_Is_VL && GAMMA_GT_IsAtMost_H && SEX_Is_M && AGE_IsAtLeast_L && GPT_IsAtMost_H

BMA fuzzy szabálybázis optimalizációjára

- Öt példát használtunk az algoritmus teljesítményének bemutatására

- pH probléma
$$pH = -\frac{\log\left(\sqrt{\frac{y^2}{4} + 10^{-14}} - \frac{y}{2}\right) + 6}{26}, y = 2 \times 10^{-13} x - 10^{-3}$$

- ICT probléma
$$\theta_2 = \pm \operatorname{tg}^{-1}\left(\frac{s}{c}\right) \quad s = \sqrt{1 - c^2} = \sin(\theta_2) \quad c = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} = \cos(\theta_2)$$

- Hatdimenziós függvény
$$y = x_1 + x_2^{0.5} + x_3x_4 + 2e^{2(x_5 - x_6)}$$

- Mezőgazdasági probléma: kapcsolatot ír le a talajjellemzők és a kukorica hozam között (hat változós)

- Kémiai probléma: egy kémiai berendezés vezérlésének operátorát valósítja meg (öt változós)

- Megállási feltétel

$$\Omega[k-1] - \Omega[k] < \theta[k] \quad \theta[k] = \tau_f (1 + \Omega[k]) \quad \tau_f = 10^{-4}$$

$$\|\mathbf{par}[k-1] - \mathbf{par}[k]\| < \sqrt{\tau_f} (1 + \|\mathbf{par}[k]\|)$$

$$\|\mathbf{g}[k]\| \leq \sqrt[3]{\tau_f} (1 + |\Omega[k]|)$$

BMA fuzzy szabálybázis optimalizációjára

- Paraméter értékek:
 - $N_{gen}=20$, $N_{ind}=10$, $N_{clones}=8$, $N_{inf}=4$, $N_{iter}=10$
 - A szabályok száma 3
 - Tanító/ellenőrző minták száma:
 - pH:101, ICT: 110, 6-dim.: 200

- Hibakritériumok:

$$MSRE = \frac{1}{N_{pat}} \sum_{i=1}^{N_{pat}} \frac{(t_i - y_i)^2}{y_i^2}$$

t_i : az i . minta kívánt kimenete
 y_i : modell kimenet az i . mintára

$$MREP = \frac{100}{N_{pat}} \sum_{i=1}^{N_{pat}} \left| \frac{t_i - y_i}{y_i} \right|$$

BMA fuzzy szabálybázis optimalizációjára

Alg.	MSE	MSRE	MREP	MSEv	MSREv	MREPV
BEA	4.5e-003	2.1e-001	2.8e+001	7.6e-003	4.2e-001	5.1e+001
BMA	1.5e-005	3e+002	1.7e+002	3e-005	1.2e+003	3.5e+002

A legjobb MSE egyed teljesítménye (pH)

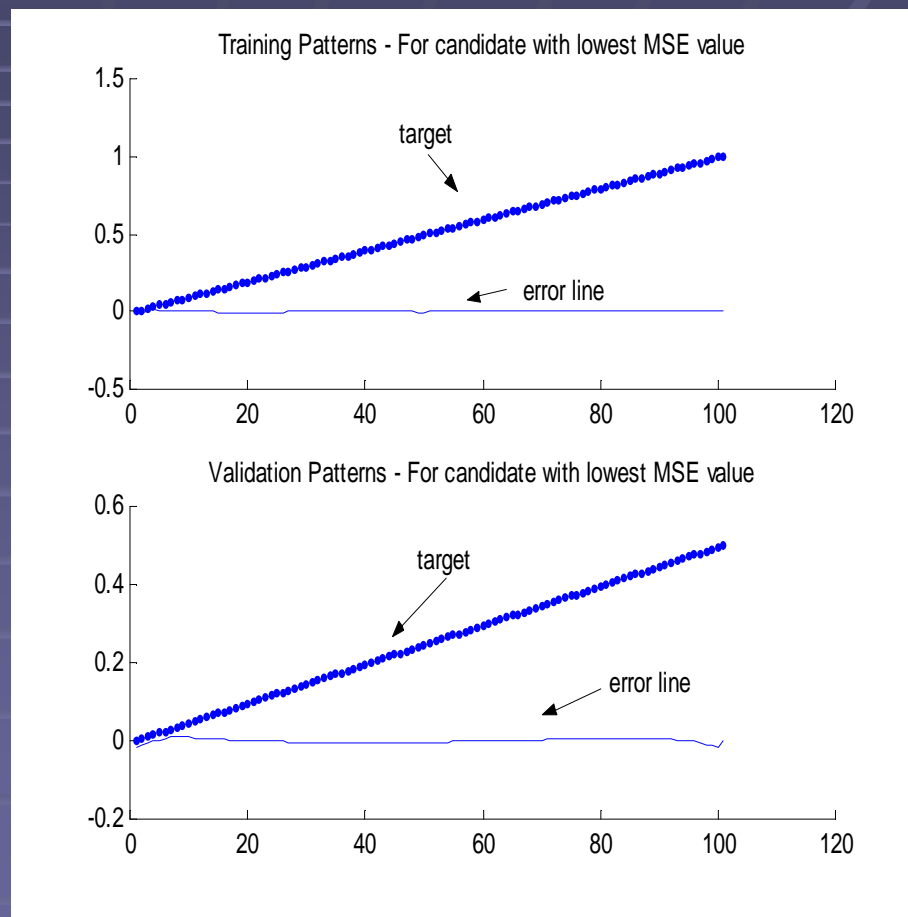
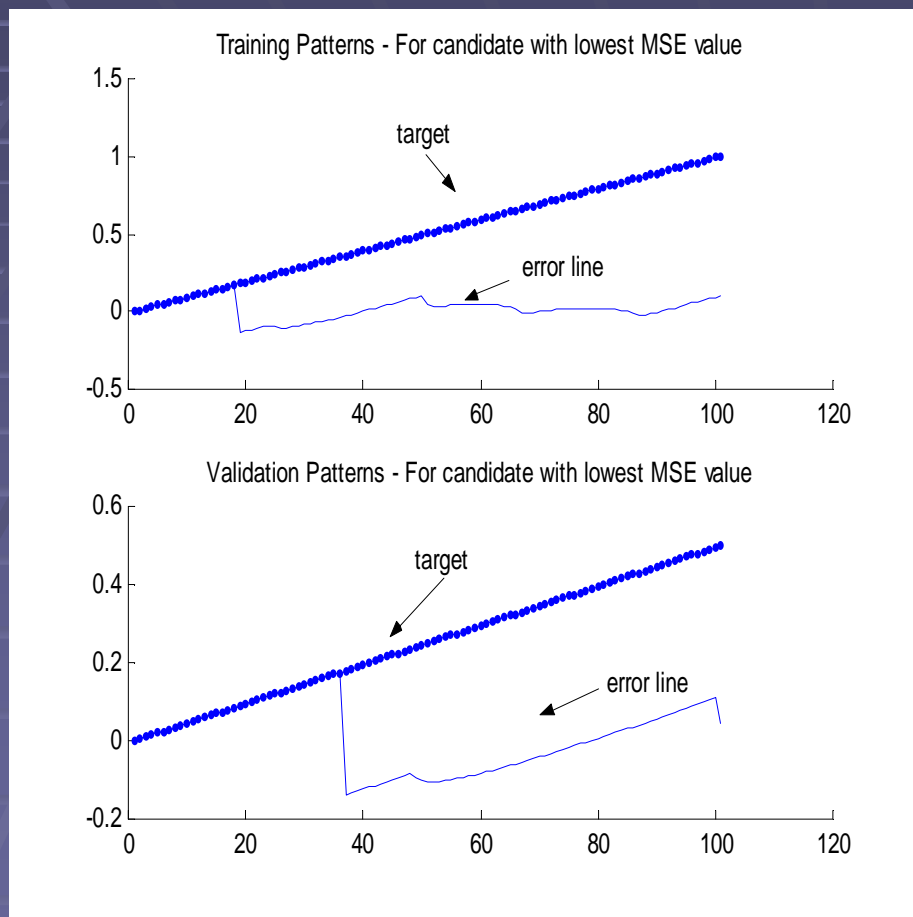
Alg.	MSE	MSRE	MREP	MSEv	MSREv	MREPV
BEA	1.2	6.3e+013	2.1e+008	2.5	3.1e-001	4.7e+001
BMA	1.3e-001	2.9e+013	1.1e+008	3.4e-002	5.2e-003	5.50

A legjobb MSE egyed teljesítménye (ICT)

Alg.	MSE	MSRE	MREP	MSEv	MSREv	MREPV
BEA	4.5e+001	7.9e-001	8.3e+001	5.4e+001	8.1e-001	8.5e+001
BMA	1.5e+001	3.3e-001	4.5e+001	1.4e+001	2.8e-001	3.9e+001

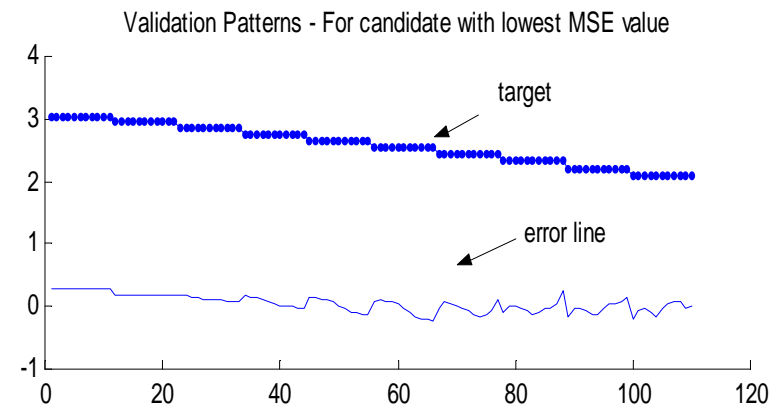
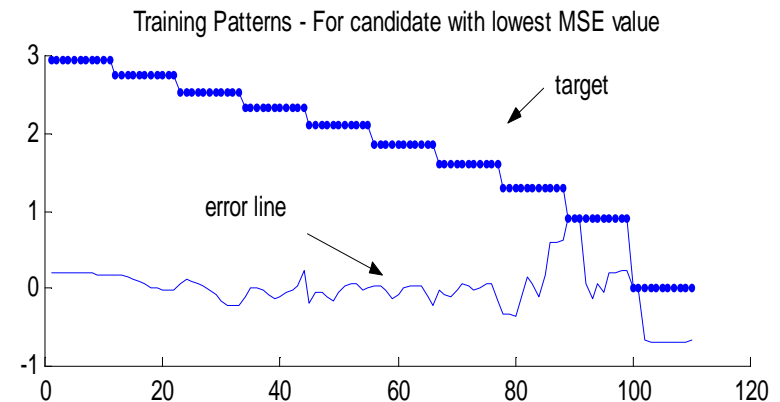
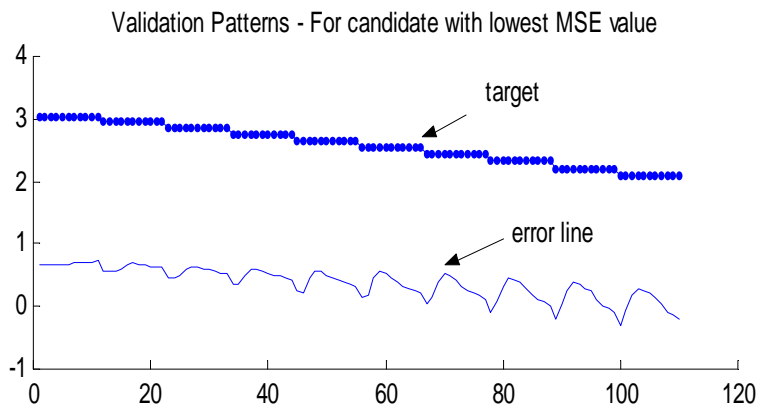
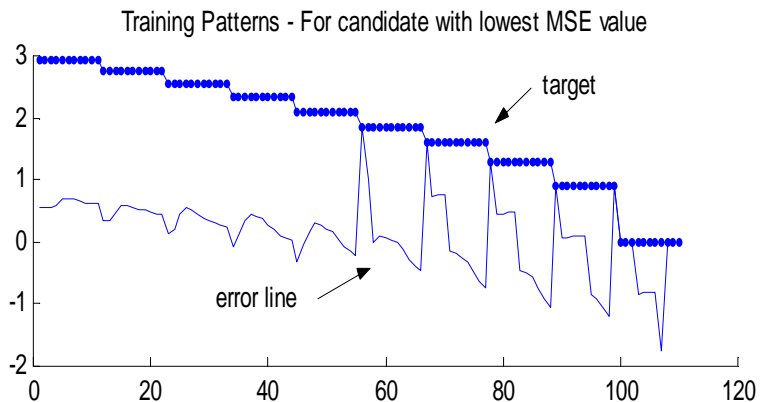
A legjobb MSE egyed teljesítménye (6 dim.)

BMA fuzzy szabálybázis optimalizációjára



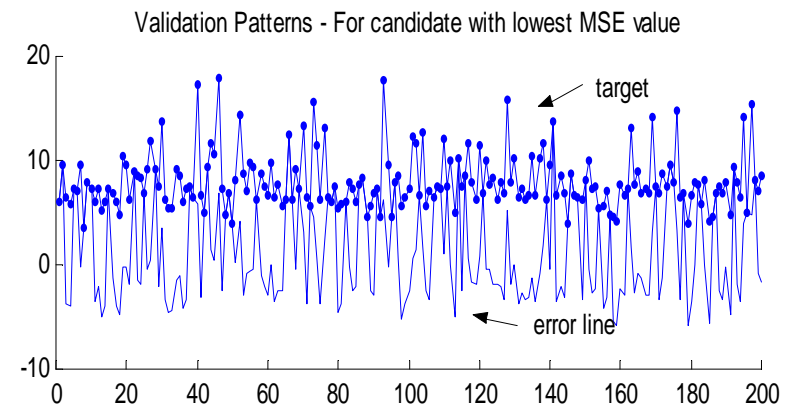
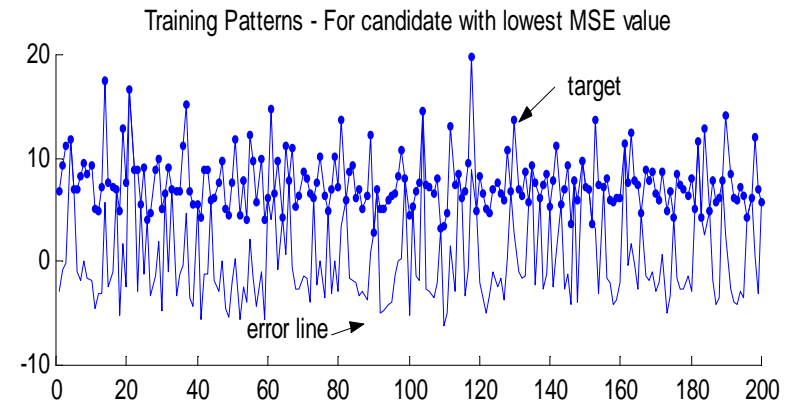
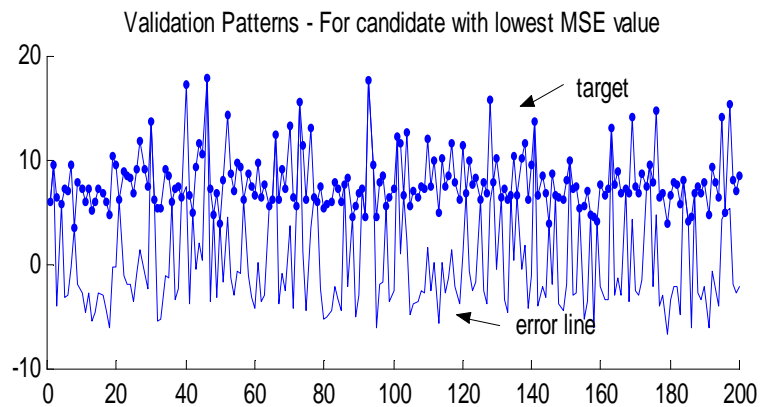
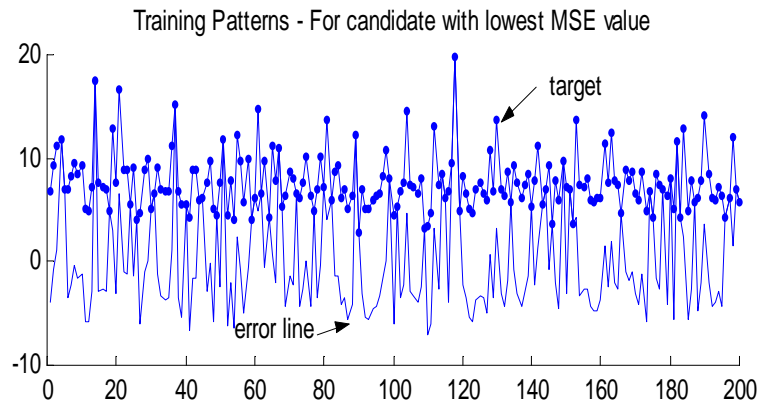
A kívánt kimenet és a hiba ábrázolása a legjobb egyedre a pH probléma esetén BEA (baloldal) és BMA (jobboldal) algoritmusokat alkalmazva

BMA fuzzy szabálybázis optimalizációjára



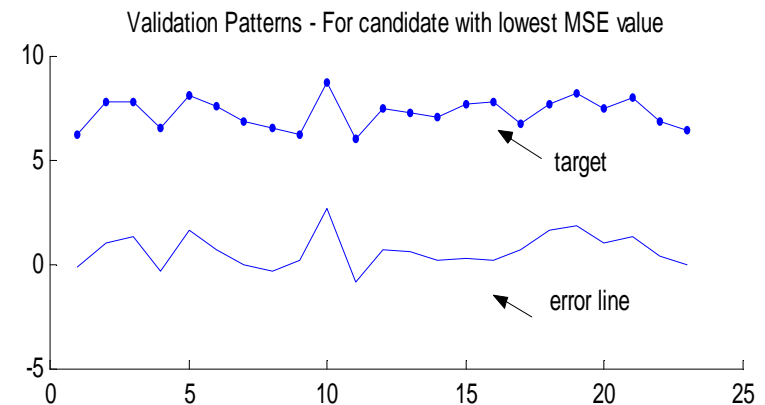
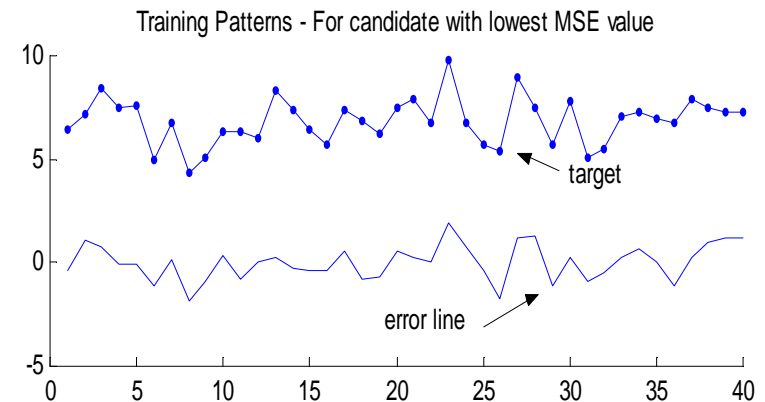
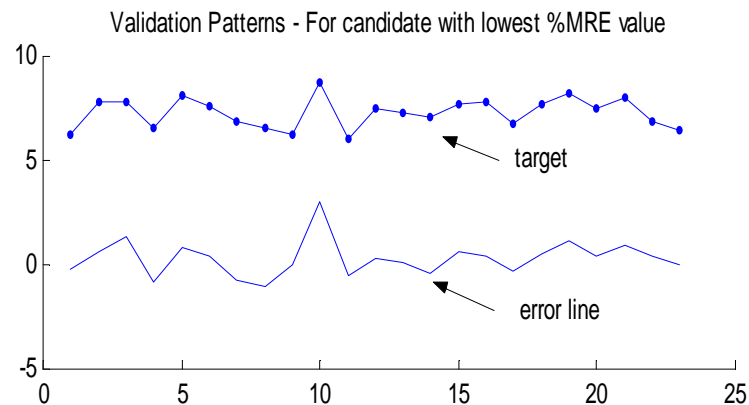
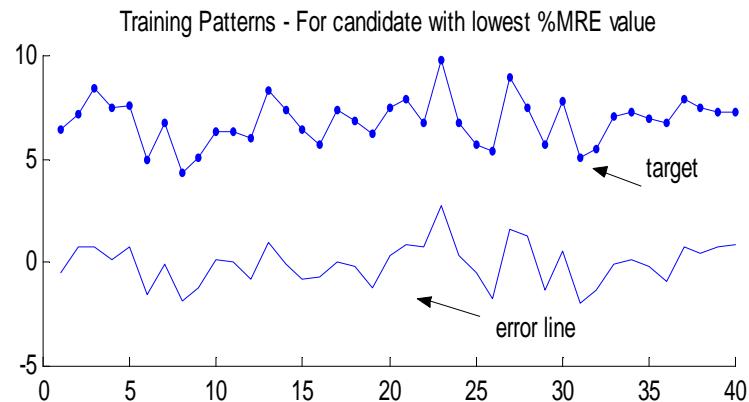
A kívánt kimenet és a hiba ábrázolása a legjobb egyedre a ICT probléma esetén BEA (baloldal) és BMA (jobboldal) algoritmusokat alkalmazva

BMA fuzzy szabálybázis optimalizációjára



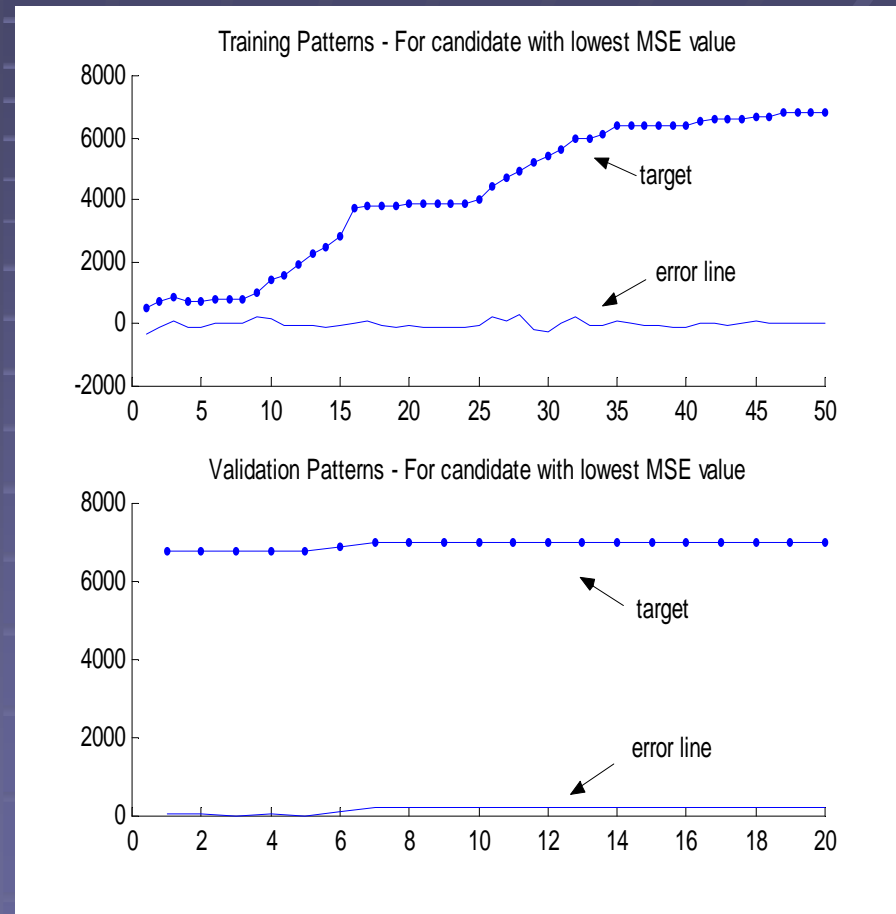
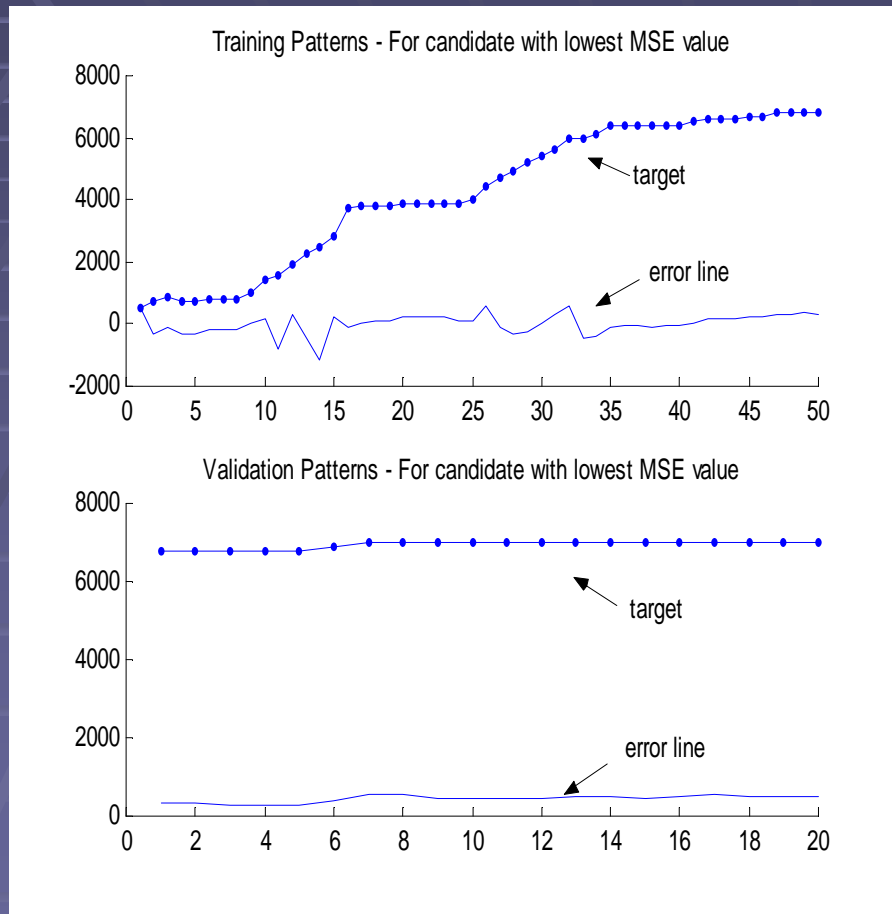
A kívánt kimenet és a hiba ábrázolása a legjobb egyedre a hatváltozós probléma esetén BEA (baloldal) és BMA (jobboldal) algoritmusokat alkalmazva

BMA fuzzy szabálybázis optimalizációjára



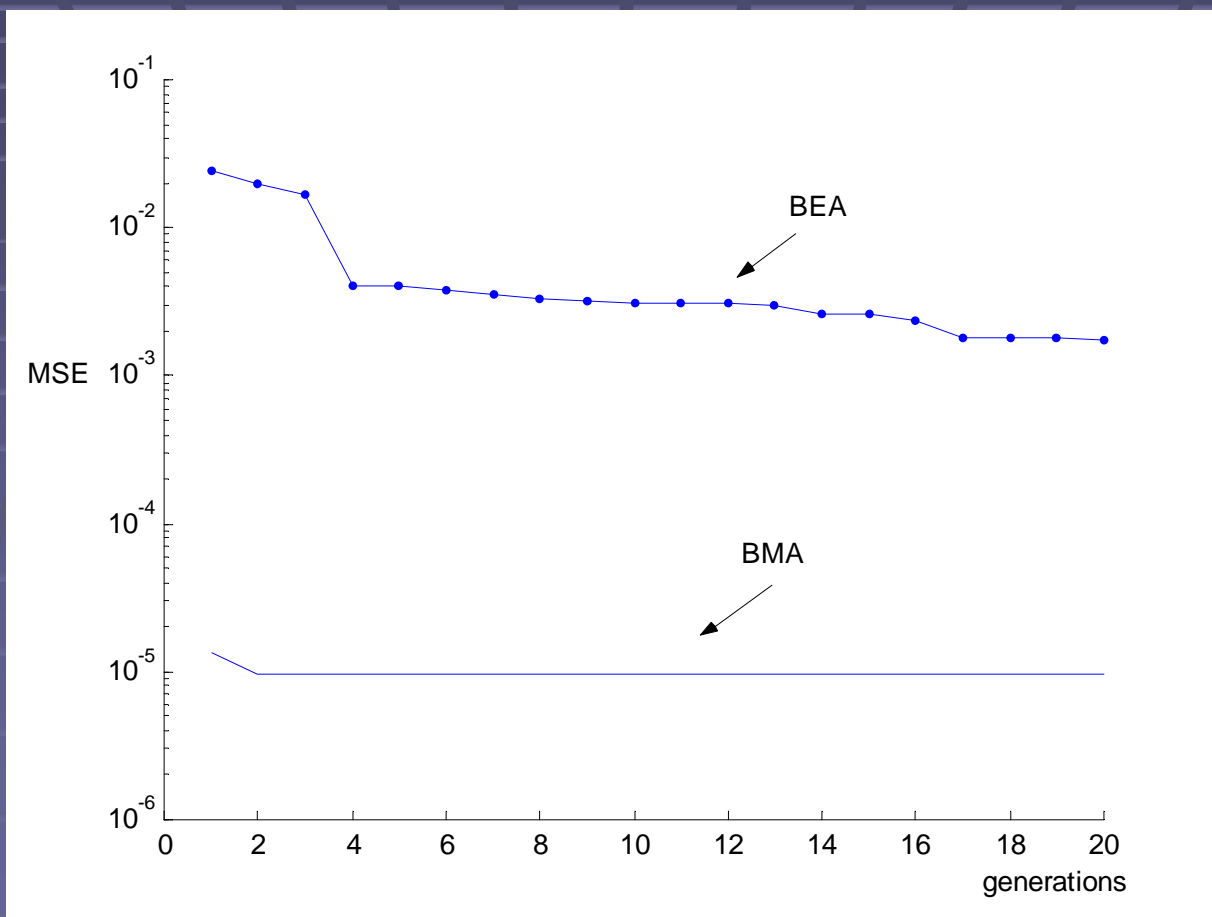
A kívánt kimenet és a hiba ábrázolása a legjobb egyedre a mezőgazdasági probléma esetén BEA (baloldal) és BMA (jobboldal) algoritmusokat alkalmazva

BMA fuzzy szabálybázis optimalizációjára



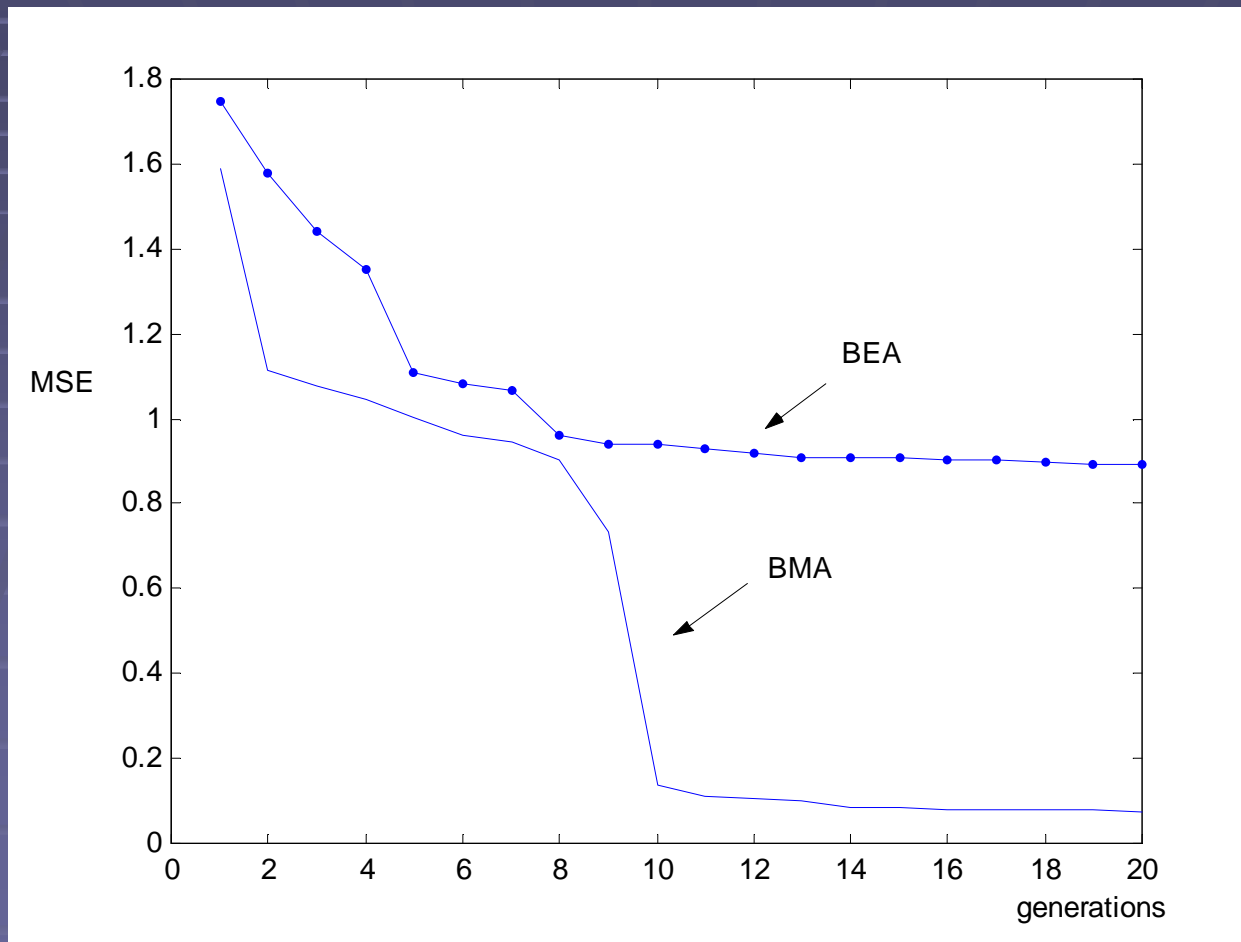
A kívánt kimenet és a hiba ábrázolása a legjobb egyedre a kémiai probléma esetén BEA (baloldal) és BMA (jobboldal) algoritmusokat alkalmazva

BMA fuzzy szabálybázis optimalizációjára



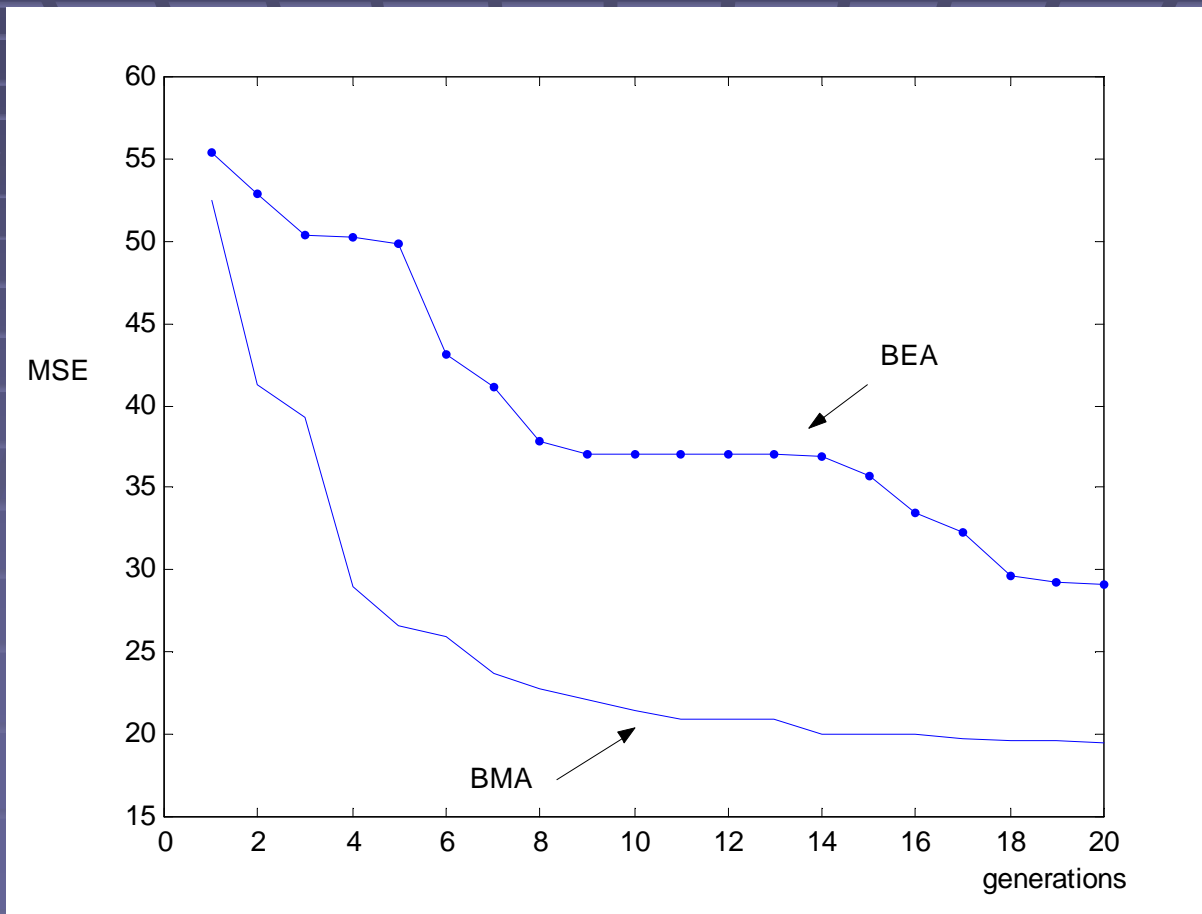
MSE alakulása a pH probléma esetén

BMA fuzzy szabálybázis optimalizációjára



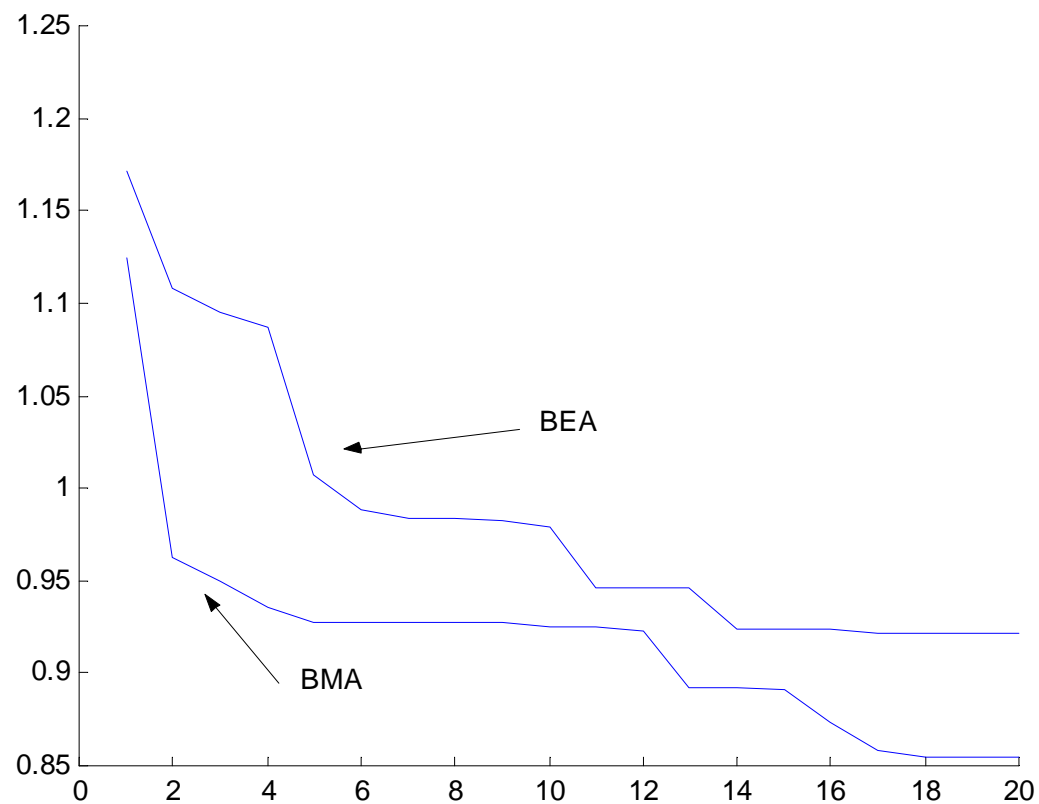
MSE alakulása az ICT probléma esetén

BMA fuzzy szabálybázis optimalizációjára



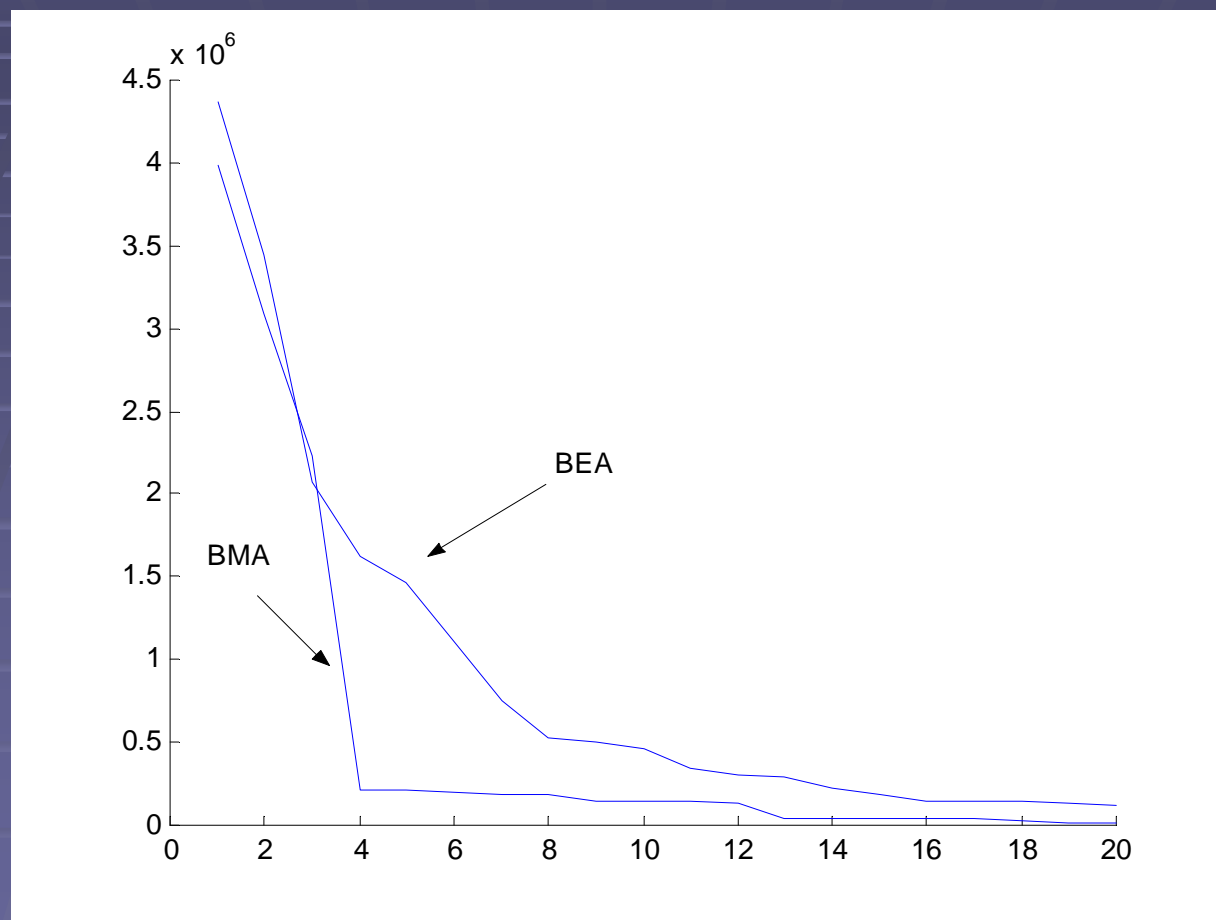
MSE alakulása a hat változós probléma esetén

BMA fuzzy szabálybázis optimalizációjára



MSE alakulása a mezőgazdasági probléma esetén

BMA fuzzy szabálybázis optimalizációjára



MSE alakulása a kémiai probléma esetén

BMA fuzzy szabálybázis optimalizációjára

A BEA által kapott optimális szabályok a pH problémára:

R_1 : Ha $x_1 = [0.059341, 0.28597, 0.40514, 0.7313]$ akkor $y = [0.18589, 0.27486, 0.52066, 0.70602]$

R_2 : Ha $x_1 = [0.38301, 0.39652, 0.41925, 0.76578]$ akkor $y = [0.75291, 0.81982, 1.0024, 1.0372]$

R_3 : Ha $x_1 = [0.051951, 0.36896, 0.56908, 0.70039]$ akkor $y = [0.086398, 0.24551, 0.37881, 0.55217]$

A BMA által kapott optimális szabályok a pH problémára:

R_1 : Ha $x_1 = [0.14845, 0.4463, 0.64292, 0.67123]$ akkor $y = [0.34233, 0.51977, 0.58192, 0.80283]$

R_2 : Ha $x_1 = [-0.10635, 0.39081, 0.57236, 0.74319]$ akkor $y = [-0.4978, 0.21469, 0.31095, 0.45583]$

R_3 : Ha $x_1 = [0.037302, 0.23532, 0.7653, 0.84955]$ akkor $y = [0.6916, 0.87295, 1.0073, 1.3837]$

BMA fuzzy szabálybázis optimalizációjára

A BEA által kapott optimális szabályok az ICT problémára:

R_1 : Ha $x_1 = [-0.027427, 0.013643, 0.027407, 0.27058]$ és $x_2 = [-0.072697, 0.1739, 0.43876, 0.88877]$ akkor $y = [1.9286, 1.9644, 2.2663, 2.8419]$

R_2 : Ha $x_1 = [0.023605, 0.82814, 0.83875, 0.91526]$ és $x_2 = [0.039025, 0.37465, 0.76291, 0.94203]$ akkor $y = [0.083984, 0.599, 0.83059, 1.6556]$

R_3 : Ha $x_1 = [0.0055574, 0.14652, 0.57419, 0.57709]$ és $x_2 = [-0.090017, 0.16984, 0.88549, 0.89452]$ akkor $y = [1.5032, 2.4553, 2.6697, 3.2334]$

A BMA által kapott optimális szabályok az ICT problémára:

R_1 : Ha $x_1 = [-0.05557, 0.096831, 0.096831, 0.61764]$ és $x_2 = [-0.38741, 0.15828, 0.42377, 0.75708]$ akkor $y = [1.9286, 2.5199, 2.841, 3.6312]$

R_2 : Ha $x_1 = [-0.13014, 0.49711, 0.79235, 0.99016]$ és $x_2 = [0.30314, 0.72457, 0.76876, 1.114]$ akkor $y = [0.01816, 0.38369, 1.0447, 1.3301]$

R_3 : Ha $x_1 = [0.11687, 0.62462, 0.69643, 0.80899]$ és $x_2 = [-0.083077, 0.22048, 0.24413, 0.70602]$ akkor $y = [1.0631, 1.6966, 1.9513, 2.1227]$

BMA fuzzy szabálybázis optimalizációjára

A BEA által kapott optimális szabályok a hat változós problémára:

R_1 : Ha $x_1 = [3.2691, 3.7803, 4.5245, 4.9707]$ és $x_2 = [0.45688, 1.6764, 1.8366, 5.2796]$ és $x_3 = [-0.19167, 3.0767, 3.5534, 4.1727]$ és $x_4 = [0.098077, 0.32278, 0.55876, 0.5754]$ és $x_5 = [-0.028333, 0.40267, 0.99707, 1.0857]$ és $x_6 = [-0.0039945, 0.36039, 0.96744, 1.2172]$ akkor $y = [3.77353, 8.49603, 10.0623, 16.1108]$

R_2 : Ha $x_1 = [0.69229, 1.4028, 4.0655, 4.5688]$ és $x_2 = [1.3221, 2.2397, 3.2113, 3.9014]$ és $x_3 = [0.81474, 2.5275, 3.817, 4.7368]$ és $x_4 = [0.019088, 0.43315, 1.8155, 2.9786]$ és $x_5 = [0.80924, 0.90894, 1.2484, 2.6031]$ és $x_6 = [0.13657, 1.8188, 2.5877, 3.1743]$ akkor $y = [3.22641, 11.4497, 12.7276, 15.884]$

R_3 : Ha $x_1 = [0.1308, 1.6514, 2.2276, 4.9155]$ és $x_2 = [-0.46576, 0.13091, 3.1385, 4.4794]$ és $x_3 = [0.37358, 1.4722, 2.3596, 4.1944]$ és $x_4 = [0.0015921, 0.40295, 0.43801, 0.55955]$ és $x_5 = [0.067956, 0.10459, 0.44455, 0.85976]$ és $x_6 = [-0.014537, 0.76469, 0.93777, 1.1354]$ akkor $y = [4.48621, 8.30233, 15.3879, 16.4223]$

BMA fuzzy szabálybázis optimalizációjára

A BMA által kapott optimális szabályok a hat változós problémára:

R_1 : Ha $x_1 = [1.0195, 1.0823, 2.2247, 4.4724]$ és $x_2 = [0.45709, 1.3472, 3.4677, 4.4184]$ és $x_3 = [0.12743, 2.4164, 3.8943, 3.8966]$ és $x_4 = [-0.0092932, 2.9735, 3.6027, 4.597]$ és $x_5 = [0.35931, 0.81271, 0.93897, 3.6913]$ és $x_6 = [0.46638, 1.53, 4.2455, 4.4091]$ akkor $y = [4.14193, 9.83104, 11.5653, 11.6497]$

R_2 : Ha $x_1 = [0.29934, 0.42357, 4.3054, 5.1422]$ és $x_2 = [-0.45532, 2.787, 4.5064, 4.9697]$ és $x_3 = [0.4231, 0.77611, 1.9622, 3.4471]$ és $x_4 = [0.014251, 0.36256, 0.4216, 0.72609]$ és $x_5 = [0.27691, 0.556, 0.89273, 1.035]$ és $x_6 = [-0.027394, 0.061566, 0.20996, 0.41618]$ akkor $y = [9.16778, 10.0454, 11.5467, 14.1264]$

R_3 : Ha $x_1 = [-0.23359, 1.7672, 2.9245, 4.608]$ és $x_2 = [0.029085, 0.14005, 3.1426, 4.6243]$ és $x_3 = [0.33395, 0.62798, 2.8702, 4.0896]$ és $x_4 = [-0.013502, 0.042693, 0.51082, 0.77926]$ és $x_5 = [-0.040665, 0.041077, 0.92361, 1.0095]$ és $x_6 = [0.017653, 0.5352, 0.86467, 0.87896]$ akkor $y = [3.52437, 7.38148, 10.069, 17.0449]$

BMA fuzzy szabálybázis optimalizációjára

A BEA által kapott optimális szabályok a mezőgazdasági problémára:

R_1 : Ha $x_1 = [1.5142, 1.7934, 2.9077, 5.7027]$ és $x_2 = [0.455225, 0.938563, 8.55417, 10.8499]$ és $x_3 = [11.3405, 20.4192, 35.1192, 56.4783]$ és $x_4 = [6.147601, 61.01089, 139.3491, 235.2254]$ és $x_5 = [22.62425, 47.09296, 49.69601, 105.8604]$ és $x_6 = [0.017295, 2.6764, 3.0311, 6.3414]$ akkor $y = [4.48543, 7.08479, 10.5221, 12.1178]$

R_2 : Ha $x_1 = [0.30569, 0.39484, 1.9894, 3.9017]$ és $x_2 = [0.73, 0.93597, 5.9333, 7.5257]$ és $x_3 = [0.121577, 16.6376, 32.5122, 52.2857]$ és $x_4 = [97.24715, 106.2515, 219.0978, 409.6491]$ és $x_5 = [6.921508, 35.79438, 64.01843, 136.3694]$ és $x_6 = [0.0287838, 0.315806, 8.8421, 10.5541]$ akkor $y = [3.4876, 3.9486, 7.6912, 9.4939]$

R_3 : Ha $x_1 = [0.68888, 1.1326, 1.4349, 2.8142]$ és $x_2 = [1.7471, 1.7911, 7.1462, 9.0641]$ és $x_3 = [3.35896, 15.2122, 30.9944, 49.8447]$ és $x_4 = [65.98891, 138.6865, 150.962, 254.418]$ és $x_5 = [27.99438, 89.80611, 91.47118, 194.8481]$ és $x_6 = [0.018529, 0.75039, 3.9289, 6.7939]$ akkor $y = [1.8734, 3.4901, 4.781, 5.0093]$

BMA fuzzy szabálybázis optimalizációjára

A BMA által kapott optimális szabályok a mezőgazdasági problémára:

R_1 : Ha $x_1 = [0.47796, 1.6484, 1.9865, 3.6762]$ és $x_2 = [0.94773, 2.79118, 12.5175, 15.567]$ és $x_3 = [14.0982, 17.7734, 39.3691, 63.3048]$ és $x_4 = [18.12948, 135.9387, 224.7332, 524.1069]$ és $x_5 = [46.75971, 50.43507, 79.21961, 168.7417]$ és $x_6 = [-0.243827, 3.81297, 10.9592, 11.0626]$ akkor $y = [2.5372, 3.8271, 8.3034, 9.576]$

R_2 : Ha $x_1 = [0.12006, 1.7442, 2.8891, 3.1227]$ és $x_2 = [0.261424, 1.18546, 8.71874, 11.1638]$ és $x_3 = [4.25414, 7.04729, 36.4863, 58.7372]$ és $x_4 = [103.0778, 104.6215, 195.9549, 403.3658]$ és $x_5 = [20.37879, 33.63517, 50.7738, 108.1339]$ és $x_6 = [-0.12446, 3.8411, 8.8812, 9.2889]$ akkor $y = [3.67955, 8.87042, 8.87042, 11.4879]$

R_3 : Ha $x_1 = [2.5302, 2.7748, 3.6654, 5.0787]$ és $x_2 = [1.49618, 1.57757, 8.70348, 11.0376]$ és $x_3 = [11.6207, 16.1909, 35.5542, 57.0146]$ és $x_4 = [56.59098, 69.52069, 146.8576, 268.0168]$ és $x_5 = [5.144961, 10.33069, 60.76468, 129.2059]$ és $x_6 = [0.32495, 0.9483, 2.0883, 7.1438]$ akkor $y = [4.43149, 9.07721, 11.6889, 13.0019]$

BMA fuzzy szabálybázis optimalizációjára

A BEA által kapott optimális szabályok a kémiai problémára:

R_1 : Ha $x_1 = [0.17002, 0.40194, 3.3061, 6.0523]$ és $x_2 = [-0.29799, 0.2135, 0.30264, 0.39925]$ és $x_3 = [270.80893, 3170.9721, 4898.488, 5649.0938]$ és $x_4 = [-0.47485, 0.044146, 0.29019, 0.68514]$ és $x_5 = [-0.10644, 0.30708, 0.35014, 0.39027]$ akkor $y = [408.357, 6297.509, 6454.9829, 6870.354]$

R_2 : Ha $x_1 = [0.12287, 0.62496, 2.8252, 5.172]$ és $x_2 = [-0.25833, 0.081853, 0.18509, 0.25415]$ és $x_3 = [418.74117, 1776.8845, 5600.8669, 8734.971]$ és $x_4 = [-0.37049, 0.038079, 0.067936, 0.22641]$ és $x_5 = [-0.1374, 0.13502, 0.38861, 0.50378]$ akkor $y = [720.16852, 5761.8159, 9433.87108, 11990.8662]$

R_3 : Ha $x_1 = [1.31078, 5.6382, 7.49383, 13.7185]$ és $x_2 = [-0.37862, 0.022461, 0.27127, 0.5022]$ és $x_3 = [154.20948, 1963.9383, 2431.6353, 3216.8208]$ és $x_4 = [-0.3917, 0.16706, 0.23937, 0.29018]$ és $x_5 = [-0.065146, 0.038172, 0.15266, 0.23887]$ akkor $y = [119.59077, 413.20153, 1467.155, 1900.3928]$

BMA fuzzy szabálybázis optimalizációjára

A BMA által kapott optimális szabályok a kémiai problémára:

R_1 : Ha $x_1 = [0.64389, 3.5881, 7.8709, 8.5822]$ és $x_2 = [-0.33968, -0.28496, 0.29334, 0.69235]$ és $x_3 = [267.21254, 428.59773, 1321.6508, 6830.8565]$ és $x_4 = [-0.38738, -0.31509, 0.30362, 0.58666]$ és $x_5 = [-0.20359, -0.053113, 0.35114, 0.40443]$ akkor $y = [-153.16266, 157.7469, 1159.5847, 2627.3857]$

R_2 : Ha $x_1 = [0.533977, 2.97557, 6.69248, 10.3774]$ és $x_2 = [-0.44198, -0.26061, 0.24069, 0.56808]$ és $x_3 = [204.99338, 297.67489, 1135.1634, 5479.9971]$ és $x_4 = [-0.59209, -0.22992, 0.3832, 0.74042]$ és $x_5 = [-0.42551, -0.23569, 0.31779, 0.36602]$ akkor $y = [-883.87131, -168.00231, 1013.4558, 2409.1277]$

R_3 : Ha $x_1 = [1.2743, 3.3716, 6.4247, 8.3991]$ és $x_2 = [-0.38969, -0.16209, 0.20013, 0.22057]$ és $x_3 = [824.239661, 9134.08527, 11066.4981, 12272.1434]$ és $x_4 = [-0.41796, -0.31585, 0.30276, 0.47112]$ és $x_5 = [-0.26288, -0.096488, 0.34021, 0.53332]$ akkor $y = [1572.58381, 6280.96997, 10414.0883, 10414.0883]$

BMA fuzzy szabálybázis optimalizációjára

Változó hosszúságú eset

- Kiértékelési kritérium:

$$BIC = m \cdot \ln(MSE) + n \cdot \ln(m)$$

m : minták száma

n : szabályok száma

MSE : átlagos négyzetes hiba

$$MSE = \frac{1}{m} \sum_{i=1}^m (t_i - y_i)^2$$

t_i : az i . minta kívánt kimenete

y_i : modell kimenet az i . mintára

- 10 futtatás, 20 generációval
- populáció méret: 10, klónok száma: 8, infekciók száma: 4. A Levenberg-Marquardt lépés minden generációban 10 iterációig fut és a maximális megengedett baktérium hossz 10

BMA fuzzy szabálybázis optimalizációjára

Változó hosszúságú eset

A BEA és a BMA által kapott átlagértékek

Specifikáció	pH		ICT		6 változós probléma	
	BEA	BMA	BEA	BMA	BEA	BMA
MSE	6.28×10^{-3}	2.3×10^{-6}	8.69×10^{-1}	3.67×10^{-2}	3.21	1.29
MSRE	4.06×10^4	1.02×10^1	5.63×10^{13}	1.80×10^{13}	6.20×10^{-2}	3.07×10^{-2}
MREP	2.05×10^3	2.69×10^1	1.77×10^8	1.08×10^8	1.86×10^1	1.21×10^1
MSE _v	7.76×10^{-3}	2.08×10^{-6}	1.30	1.12×10^{-2}	4.29	2.22
MSRE _v	1.63×10^5	4.18×10^1	1.92×10^{-1}	1.62×10^{-3}	6.50×10^{-2}	3.37×10^{-2}
MREP _v	4.12×10^3	5.46×10^1	2.56×10^1	3.04	1.91×10^1	1.32×10^1
#rules	4.90	7.40	2.20	5.00	6.50	7.30

BMA fuzzy szabálybázis optimalizációjára

Változó hosszúságú eset

A BEA és a BMA által kapott legjobb értékek

Specifikáció	pH		ICT		6 változós probléma	
	BEA	BMA	BEA	BMA	BEA	BMA
MSE	2.73×10^{-3}	4.7×10^{-7}	8.42×10^{-1}	1.04×10^{-2}	2.07	4.10×10^{-1}
MSRE	3.71×10^4	3.63	5.32×10^{13}	7.09×10^{12}	4.78×10^{-2}	9.56×10^{-3}
MREP	1.97×10^3	1.92×10^1	2.03×10^8	6.39×10^7	1.59×10^1	7.06
MSE_v	5.12×10^{-3}	6.07×10^{-7}	1.26	2.6×10^{-3}	2.66	9.9×10^{-1}
MSRE_v	1.48×10^5	1.53×10^1	1.87×10^{-1}	3.74×10^{-4}	4.28×10^{-2}	1.5×10^{-2}
MREP_v	3.96×10^3	3.93×10^1	2.34×10^1	1.48	1.47×10^1	9.28
#rules	9	8	2	5	7	10

Konklúziók

Konklúziók

- A generációszám és az egyedszám növelésével a módszer teljesítménye is nő (de ez több számítási teljesítményt igényel)
- Optimális klón szám?! lokális minimumok problémája
- Génátadás: kapcsolatot teremt a baktériumok között
- Gyorsabb konvergencia
- A génátadás műveletet egyszerűbben lehet megvalósítani, mint a keresztezést a genetikai algoritmusokban

Konklúziók

- A **változó kiválasztási feladat** esetén a módszert számos létező, regressziós vagy osztályozási feladatot megoldó technikával lehet kombinálva használni
- Az algoritmus továbbfejlesztett változata az optimális feature halmaz méretét is képes meghatározni
- Továbbfejlesztett bakteriális operátorok teszik lehetővé a hosszváltozást
- Sub-sampling stratégia alkalmazásával a fitness kiértékelés számítási komplexitása csökken

Konklúziók

- A **bakteriális memetikus algoritmus** jobb eredményt ad mint a bakteriális evolúciós algoritmus
 - Az optimalizációs kritérium alapján
 - Más általánosított kritériumok értelmében
- A bakteriális operátorokkal a lokális optimumok elkerülhetőek
- A Levenberg-Marquardt módszert használva a lokálisan legjobb megoldás található meg
- Ezért a globális optimum nagyobb pontossággal található meg
- A továbbfejlesztett bakteriális operátorokkal a szabályok számát nem szükséges előre definiálni