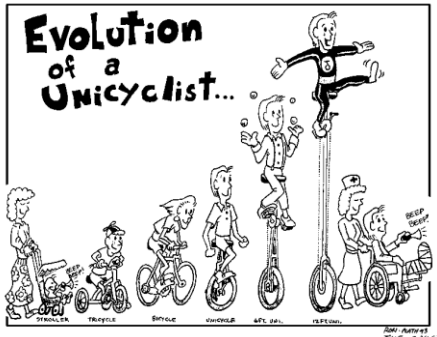
1.

Evolúciós algoritmusok

- Az evolúciós algoritmusok a biológiai evolúciót modellezik. Az ember számtalan probléma megoldásához merítette az alapötletet a természetből. Pl. madarak szárnya → repülőgépszárny, bambuszrúd → rúdugrók üvegszálas rúdja, halak léghólyagja → tengeralattjárók merülési berendezése, emberi szemlencse → fényképezőgép optikája, stb.

- Amit az ember gondolkodással próbál megoldani, azt a természet gyakran hatékonyabban oldotta meg a változatosság és a szelekció eszközével. A többnyire kétszülős biológiai szaporodás és a többnyire külső hatásoknak köszönhető véletlenszerű megváltozás, a mutáció **az egyedek sokszínűségét** eredményezi, míg a túlélésért folyó verseny a jobb, életképesebb egyedek **kiválogatódását**, tulajdonságaik továbböröklésének lehetőségét hordozza magában. E két működés a populáció környezeti elvárásoknak való jobb megfelelését, állandó alkalmazkodását jelenti.



1. www.ornl.gov/ORNLReview/ v34_1_01/p2b.jpg

www.tux.org/~bagleyd/unicycle_factory/ cartoons/evolution.gi

dr.Dudás László

Evolúciós algoritmusok és az optimumkeresés

- Az evolúciós algoritmusokkal végzett problémamegoldást **biológiai számításnak** is nevezik.

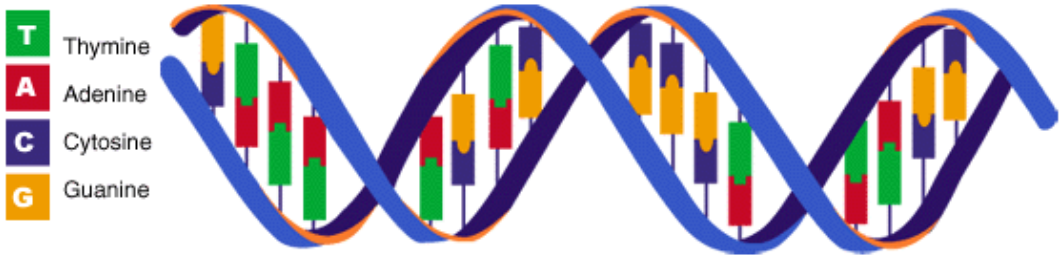
- Az evolúciós algoritmusok **populációjának** egyedei megfeleltethetők a keresés állapotterében található **állapotok** egy részhalmazának, az egyedek életképességét, jóságát mérő ún. **fittness függvény** megfeleltethető a keresés **kritériumfüggvényének**.

- Az evolúciós mechanizmus mintájára létrehozott programokkal akkor is megoldhatók a feladatok, ha a problémamegoldásra nem tudunk részletes algoritmust adni. Az irányított „próba-szerencse” módszer feladatérzéketlenségéért, **robusztusságáért számítási idővel kell fizetnünk**. Ebben is a biológiai modellre hasonlít: az evolúció nem gyorsan zajló folyamat.

- A biológiai evolúció és az evolúciós algoritmusok **eltérése**
  A biológiai evolúció célja a környezet elvárásaihoz való minél hatékonyabb **alkalmazkodás**, míg az evolúciós algoritmusokkal globális **optimumok keresését** végezzük. Ezen optimumkeresési feladatok általában nem biológiai beágyazó környezetben zajlanak.

Evolúciós algoritmusok osztályozása

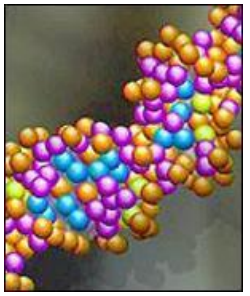
- **Evolúciós stratégiák**: a kezdet, (Rechenberg, 1973, repülőgépszárny optimalizálás.)
- **Evolúciós programozás**: programkód kifejlesztése a kódrészletek mutálódása és szelektálása által. Véges automaták automatikus kifejlesztésére (Fogel, Owens, Walsh, 1966.).
- **Genetikus algoritmus**, GA: keresztezés, mutáció és szelekció matematikai modellezése (Holland, 1975).
  - Osztályozó rendszerek
  - **Genetikus programozás**, GP: programok kitenyésztése adott feladatra. (Koza, 1992).



www.genecrc.org/site/ KO/images/helix.gif

Az Evolúciós algoritmusok általános lépései

1. Legyen P_0 a kezdeti populáció,
   k=0 a ciklusváltozó kezdőértéke.
2. Ha a megállási kritérium teljesül, add vissza a P_k populációt.
3. Egyébként bővítsd a P_k populációt új egyedekkel.
4. Szelekcióval állítsd elő a P_k populációból az új P_{k+1} populációt.
5. Inkrementáld a k ciklusváltozót és ismételd a 2. ponttól.



1.

A 3. lépésben a bővítés alkalmazhat szülőválasztást, keresztezést és mutációt.

A 4. lépés feladata a populáció létszámának eredeti értéken tartása a vesztes egyedek törlésével.

Amennyiben a populáció **egyelemű**, csak mutáció van, az algoritmus **helybeni**, és neve **sztochasztikus hegymászó**.

1. *"**news.bbc.co.uk/olmedia/1520000/ images/_1522808_new150dna.jp**"*

dr.Dudás László

A Genetikus algoritmus

- A genetikus algoritmus egy **globális kvázioptimum** megtalálására kifejlesztett kereső algoritmus, mely alapvetően **problémafüggetlen**. A lokális informáltságú, de globális célra törő algoritmusok, a többszörös indítású hegymászó, a szimulált hűtés és a tabu keresés rokona.

1.

- A problémára vonatkozóan rendelkezésre álló **információ bevihető** az algoritmusba a valós egyedeknek megfelelő **fenotípusról** a számításban alkalmazott modelljére,
  a **genotípusra** való áttéréskor, valamint
  a keresztezés és a mutáció módjának megválasztásakor.
  Az ily módon kialakított, problémára szabott genetikus algoritmusok **hatékonysága megnő**.

- A globális optimum megtalálásában az egzakt analitikus módszerekhez (differenciálás, gradiens módszer, szimplex módszer) viszonyítva lassú, pontatlan, de diszkrét állapottereken is működik, igénytelen a feladattal szemben.
  A hasonlóan robusztus és igénytelen kimerítő kereséstől hatékonyabb.

1. www.theatlantic.com/issues/ 98jan/climate.htm

A Genetikus algoritmus ..

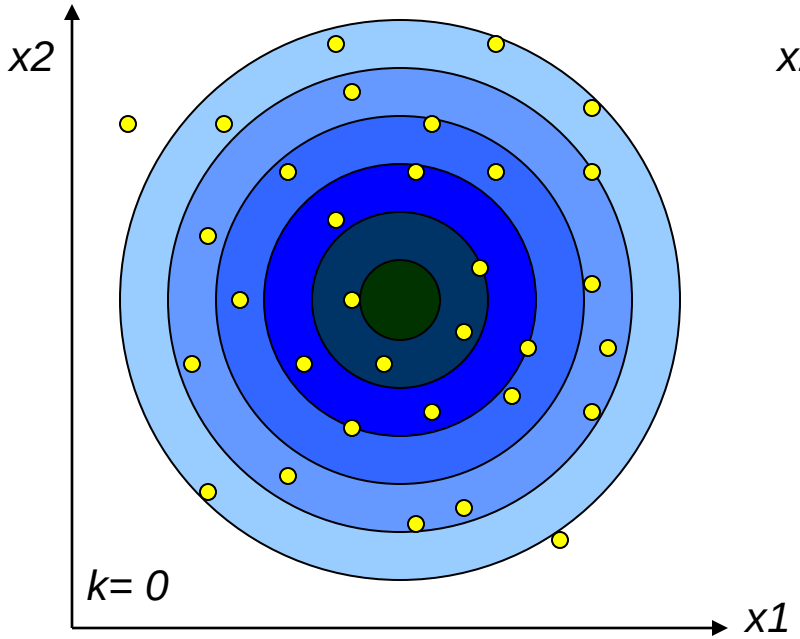
- A genetikus algoritmus **nevét** az élőlények utódaiban megjelenő kombinálódott szülői génekről, mint az utód tulajdonságainak hordozójáról kapta. A genetikus algoritmus alkalmazásának kulcskérdése a fenotípusról genotípusra való leképezés, azaz az egyed egyes tulajdonságait reprezentáló „gének”, azaz kromoszómarészletek kialakítása.

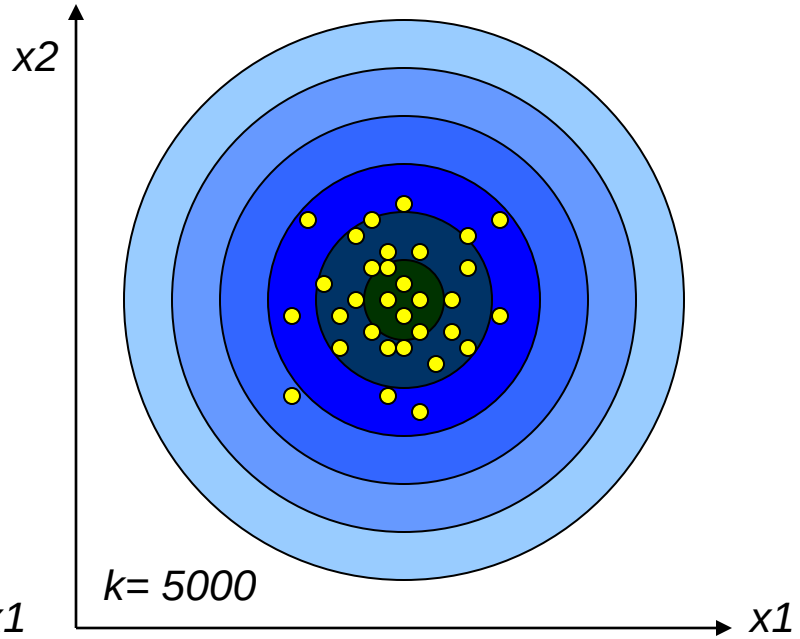
- Mivel egyszerre egy populációt, azaz egyedek egy csoportját kezeli, a globális optimum fellelésének megnő az esélye. Az állapottér feltárásában a következők játszanak szerepet:
  - A kezdeti populáció egyedeinek eloszlása az állapottéren
  - A keresztezések interpoláló hatása
  - A mutációk extrapoláló, a keresztezéssel kialakult belterjességből, lokalizáltságból kivezető hatása.

- A kezdeti mutációk az állapottér hatékony feltárását eredményezik. A globális optimum köré gyűlt késői populációkból viszont már csak gyorsan elhaló mutációk származnak, mivel azok életképessége a populáció többi tagjához képest alacsony.

A Genetikus algoritmus ..

- A **populáció fejlődése** az egyedek fejlődésén keresztül realizálódik. A szelekciónak köszönhető **látszólag spontán** fejlődés nem csak a véletlen eredménye: a jobb célfüggvény (fitness) értéket reprezentáló egyedek aránya a populációban egyre nő, míg a kevésbé életképeseké csökken. Az egyedek a **módszer memóriájaként** is felfoghatók: a jó tulajdonságú egyedek megtartásával az algoritmus **megtanulja** a jó célfüggvényértéket adó tulajdonságokat, a gyengébbeket eredményező tulajdonságok pedig **elfelejtődnek** az azokat hordozó egyedek kihalásával.





A Genetikus algoritmus ..

- Leképezés fenotípusról genotípusra

  Feladata minden egyes állapothoz egy karaktersorozat, azaz bitminta hozzárendelése.

  A leképezés kihat az alkalmazható operátorokra, végeredményben a GA hatékonyságára.

- Példák egyszerű esetekre:
  - Fenotípus: 12 zöld, vagy piros téglalapból álló sorozat.
    Genotípus: 12 bites bináris szám, piros:1, zöld: 0.
    Optimum: minden téglalap piros.

  - Fenotípus: m*n méretű fekete-fehér bittérkép alakjában adott kép.
    Genotípus: m*n elemű bináris szám, fekete: 0, fehér: 1.
    Optimum: két fekete átló a bittérképen.

- Megjegyzés: Az egyetlen **tulajdonság különböző értékeinek** tárolására alkalmazott több bites, egységként, génként kezelt kromoszómarészlet által felvehető értékeket **allél**-eknek nevezzük.

A Genetikus algoritmus ..

- **A szelekciós operátor**

  Feladata a szülőegyedek kiválasztása a keresztezés számára. A különféle szelekciós módszerekben közös, hogy a rátermettebb egyedeket **jelentősebb arányban** választják ki tulajdonságaik továbböröklítésére, de általában a gyengébb egyedek is kapnak egy **kis esélyt**. Ha a régi populáció legrátermettebb egyede mindig átkerül az új populációba, a szelekció **elitista**.

- A szelekcióban alkalmazott technikák

  1. Rátermettség-arányos választás
  2. Párok versenyeztetése
  3. Rangsorolás

1.

1. cbs.sportsline.com/u/includes/ olympics/2000/kids/boxing.htm

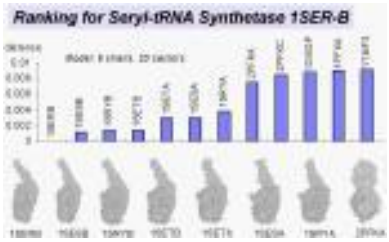
dr.Dudás László

A Genetikus algoritmus ..

1. **Rátermettség-arányos választás**: az egyed kiválasztásának valószínűsége arányos a populáció rátermettségi átlagához viszonyított rátermettségével. A rátermettség gyakran azonos az egyedhez tartozó függvényértékkel, ritkábban azonban csak a szelekcióhoz alkalmazott egyedjellemző értéke.

2. **Párok versenyeztetése**: véletlenszerűen kiválasztott két egyed közül a versenyt nyerő, azaz a nagyobb rátermettségi értékű lesz a kiválasztott. A módszer alkalmazható több, mint két versenyzős esetben is.

3. **Rangsorolás**: a keresztezés interpolációs hatásából eredő állapottérfeltárás hatásának lecsökkenését eredményezi, ha mindig ugyanazok az ősök szelektálódnak, örökítik tovább tulajdonságaikat. Ez a veszély fenyeget azoknál a módszereknél, amelyek a kiválasztást közvetlenül a rátermettség értékére alapozzák. Rangsorolás esetén a közvetlen **rátermettségi értékek helyett a rátermettségi sorrendre alapozott** szelekciót végzünk.



www.dbs.informatik.uni-muenchen.de/ Forschung/Bioinforma...

dr.Dudás László

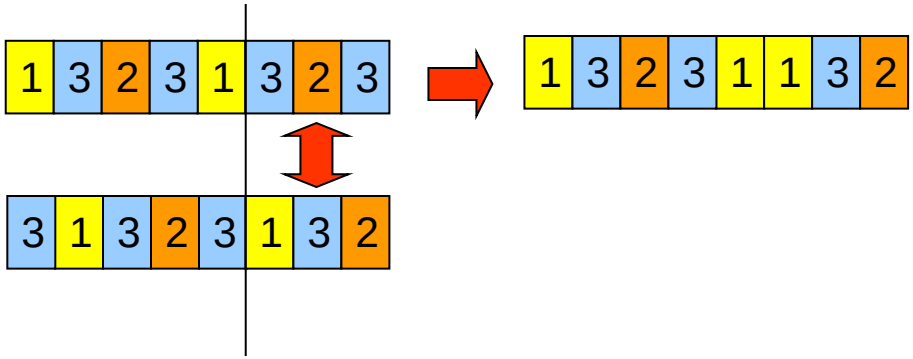
A Genetikus algoritmus ..

- **A keresztezés operátor**

  Szerepe utódok, azaz új egyedek előállítása.

1.

- Leggyakoribb módszerek
  - **Egypontos keresztezés**

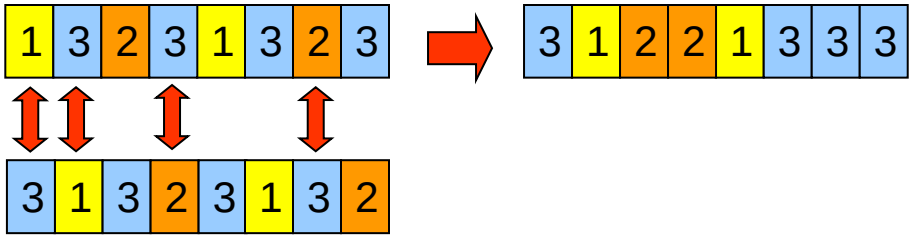


Véletlenszerű keresztezési pont választás.
(Többpontos keresztezés esetén több, mint két szülő is lehet.)

1. www.cnn.com/SPECIALS/ 2001/newamericans/

A Genetikus algoritmus ..

- **Egyenletes keresztezés**



1.

Ötven százalék eséllyel cserélődik minden egyes gén.

- Megjegyzések:
  - Nincsen általános módszer a keresztezés megválasztására, annak a feladathoz kell igazodnia.
  - Speciális keresztező operátort igényelhet az olyan eset, amikor összefüggés van a gének között (intelligens keresztezés).
  - A GA algoritmus érzékeny az operátorválasztásra.

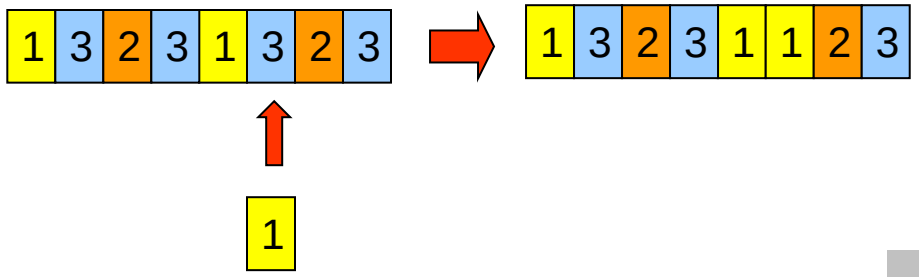
1. www.virginiasystems.com/

A Genetikus algoritmus ..

- **A mutáció operátor**

  Olyan új egyedet hoz létre, mely mentes a keresztezés belterjességétől, azaz az ősökétől merőben elütő tulajdonságokat hordozhat. A keresztezéstől kisebb gyakorisággal alkalmazott művelet.

- A módszer: az adott egyed kromoszómájának egy véletlenszerűen választott génjét egy másik génre cseréljük ki.



***Demo** található a*
http://www.taygete.demon.co.uk/java/ga/index.html
címen.

"http://www.grospixels.com/site/peve.html"

dr.Dudás László

A Genetikus algoritmus lépései

1. Add meg az algoritmus fő paramétereit
2. Add meg a kezdeti populációt
3. Rendezd sorba az egyedeket csökkenő rátermettség szerint
4. Ha a megállási kritérium teljesül, add vissza az aktuális populációt
5. Egyébként válaszd ki az egyedeket a keresztezéshez
6. Hajtsd végre a keresztezéseket
7. Válaszd ki az egyedeket a mutációhoz
8. Végezd el a mutációkat
9. Rendezd csökkenő rátermettség szerint az eredeti és a keresztezéssel és mutációval kapott egyedekből álló halmazt
10. Állítsd elő az új populációt a leggyengébb rátermettségű egyedek ejtésével.
11. Ismételd a 4. ponttól.

Megjegyzések:

A kezdeti populáció feltöltése történhet véletlen értékekkel, de történhet a feladatra vonatkozó információk felhasználásával célirányosan is. Kevés ismeret felhasználása is nagy hatékonyságnövekedést eredményezhet. A kezdeti populáció számossága 50 -100.

A Genetikus algoritmus paraméterei

1.

- A populáció számossága (pl. 16)
- Kromoszóma hossz (pl.12 bit)
- A szelekciós operátor
- A keresztezés operátor: a keresztezési ráta (új egyedek aránya, pl. 0.7), a keresztezési pontok száma, stb.
- A mutáció operátor: a mutációs ráta (pl. 0.001), a mutációs pontok száma, stb.

A genetikus algoritmus alkalmazhatósága

Bár a genetikus algoritmusok nagyon igénytelenek az állapottérrel, az állapottéren értelmezett kiértékelő függvénnyel szemben, van néhány követelmény, amelynek eleget kell tenniük a megoldandó feladatoknak:

- A feladat megoldási lépéseinek ismerete nem szükséges, de bizonyosnak kell lennünk a megoldhatóságában.
- A megoldások összetevői között lehet részleges függőség, de nem lehet az összes összetevő függőségi viszonyban.

1. www.usaid.gov/bd/ Population_Health.html

Feladatok reprezentálása genetikus algoritmussal történő megoldáshoz

A feladat megfelelő reprezentációja a genetikus algoritmus alkalmazhatóságának kulcskérdése.

A jó reprezentáció jellemzői

1. Az algoritmus legyen képes az állapottér lefedésére
2. A rátermettebb egyedek nagyobb arányban örökítsék tovább tulajdonságaikat, de ne alakuljon ki belterjes populáció
3. A közelálló szülők utódai hasonlítsanak a szülőkre, míg a nagyban eltérő szülők származtassanak változatos utódokat
4. A mutáció akadályozza meg a populáció korai belterjessé válását, de ne rontsa el a jó tulajdonságokat.
5. A mutációk származtassanak az állapottér feltáratlan területére eső életképes utódokat melyek közül a gyengék gyorsan szelektálódjanak
6. A kezdeti populáció használjon maximális informáltságot a kezdeti megoldások jó tulajdonságainak elérésére, de ugyanakkor fedje le kellő homogenitással az állapotteret, azaz legyen véletlen jellegű.

1.

1. droopyoh.multimania.com/

dr.Dudás László

A Genetikus algoritmus fö jellemzői

- Sokhegymászós sztochasztikus optimumkereső módszer, ebből eredően lassú
- Fő alkalmazási területe az összetett, más algoritmusokkal nehezen kezelhető feladatok területe
- Kvázioptimumok sokaságát szolgáltatja. Az optimum megtalálása további módszereket igényel
- Matematikailag nem alátámasztott, ennek ellenére összetett feladatokra is vannak sikeres alkalmazások
- A hatékonysága a megoldásokat felépítő elemi összetevők hatékony versenyeztetésében és szelektálásában rejlik
- A populáció egyedei és ezáltal a populáció is memóriaként működik: képes a jó tulajdonságok megtanulására és a szelekció révén a gyenge tulajdonságok felejtésére.
- Az állapottér csonkítása nélkül is megtalálja a globális optimumot a keresztezés interpoláló és a mutációk extrapoláló hatásának köszönhető állapottér-felderítő képességüknek köszönhetően.
- A mutáció az egyre belterjesebb populációban vérfrissítő hatással bír, és jó tulajdonsága esetén gyorsan elterjed, ami a keresés súlypontáthelyező képességét, lokális optimumból való kiszabadulását jelenti.

A Genetikus algoritmussal megoldható feladatok fö jellemzői

- Nemlineáris feladatok, melyek optimuma nem áll elő részoptimumok összegeként.
- Nagyméretű sokdimenziós állapottér
- Lokális extrémumok sokasága
- A feladat összetevőinek a hatása az eredményre nem becsülhető
- A különféle paraméterértékekkel adódó megengedett megoldások kiértékelése elégséges a legjobb megoldás kiválasztásához.

Példák az algoritmus működésének érzékeltetésére

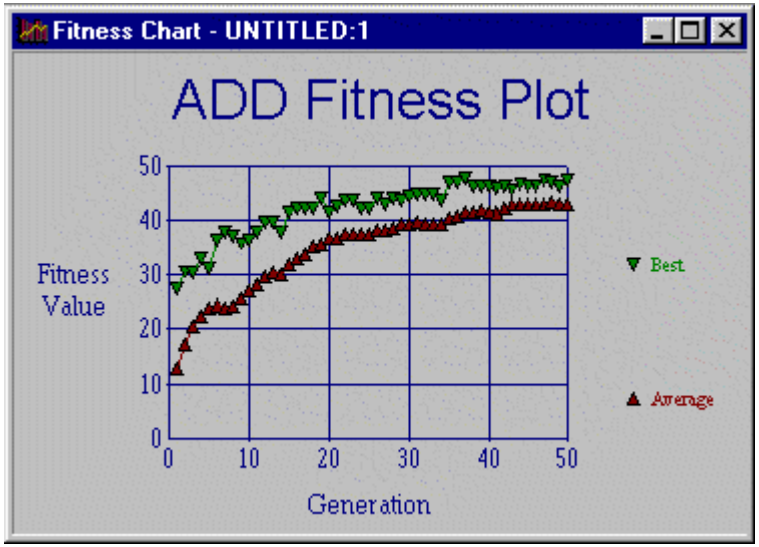
"ADD" példa

Az ” Add” példa bemutatja a GA algoritmus használatát egy öt tagból álló összeg maximumának megtalálására. Mindegyik tagot egy ötbites gén hordoz és a tagok értéke 0.0 – 10.0 közötti változhat.

A fitness függvény egyszerűen a gének által kódolt értékek összege:

Fitness = Tag1 + Tag2 + Tag3 + Tag4 + Tag5

A keresési folyamat előrehaladása a generációk számának függvényében:



dr.Dudás László

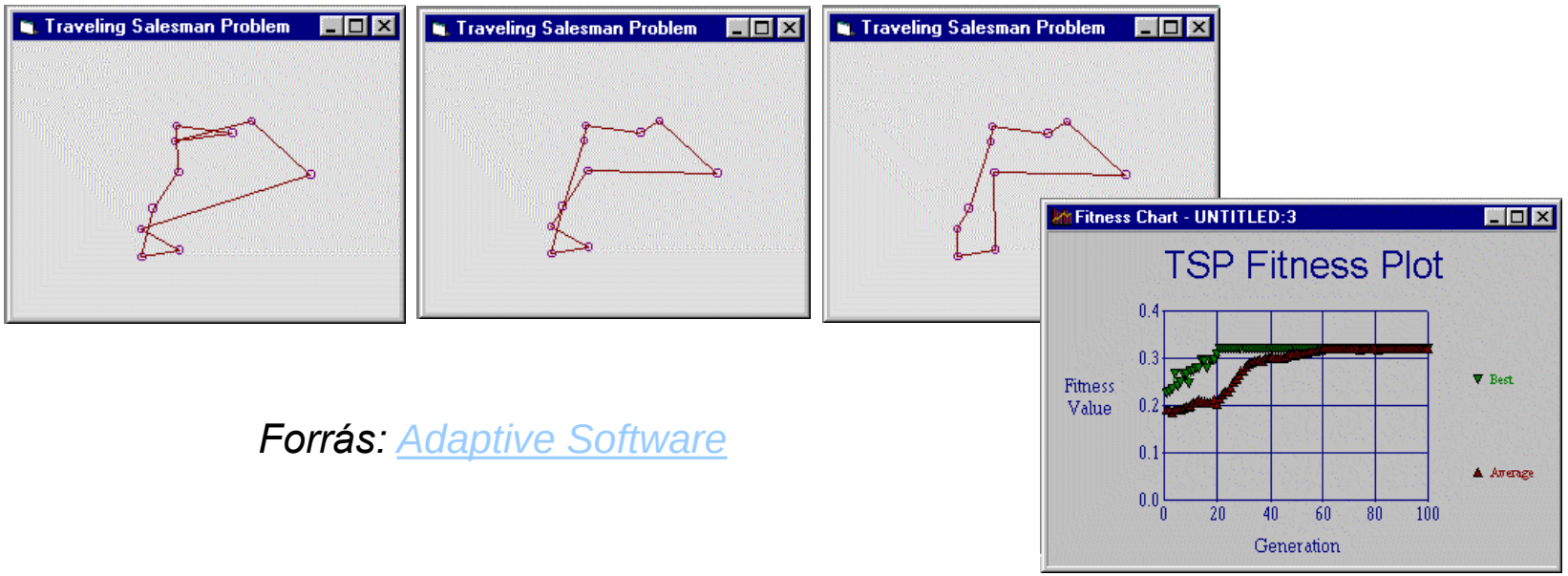
Példák az algoritmus működésének érzékeltetésére

„Utazó ügynök" példa
A példa bemutatja a GA algoritmus használatát egy 10 városon áthaladó legrövidebb körút megtalálására. Az intelligens keresztező operátor biztosítja, hogy minden város csak egyszer szerepeljen a körútban.

A fitness függvény az úthossz reciproka:

Fitness = 1/ ” Teljes úthossz”

A keresési folyamat előrehaladása a generációk számának függvényében:



Forrás: Adaptive Software

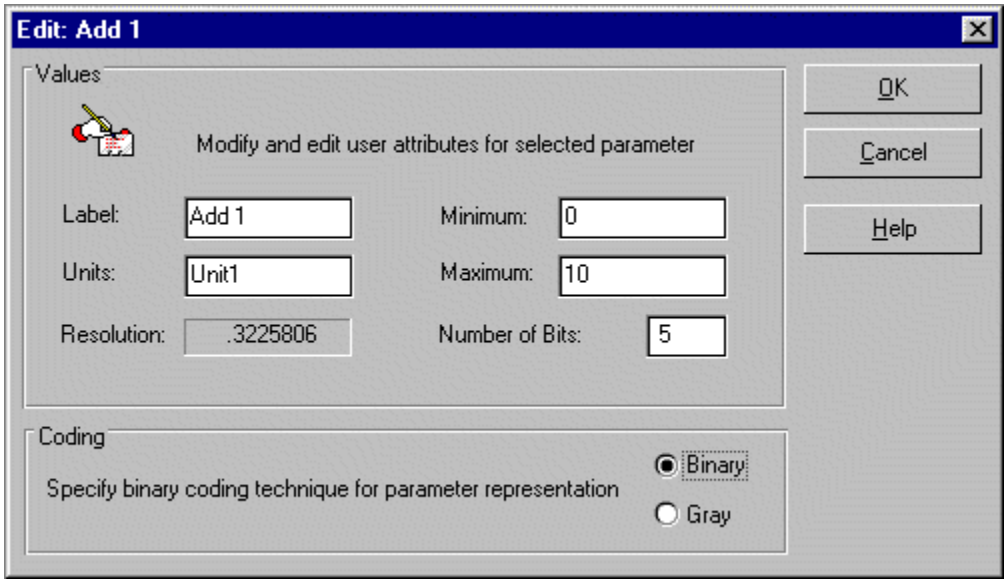
dr.Dudás László

Az Adaptive Software cég GaUI programja

Forrás: Adaptive Software

- A program fő funkciói:
  1. Felhasználói paraméterek megadása
  2. A GA paramétereinek beállítása
  3. Fitness értékek alakulásának futás közbeni kijelzése
  4. A genetikus kódok kijelzése
  5. A genetikus kódok rendezése és rangsorolása
  6. A GaUI szolgáltatásainak automatizálása script-ek segítségével

1. Felhasználói paraméterek megadása az egyes gének jellemzésére, génenként (Add1 = Tag1)

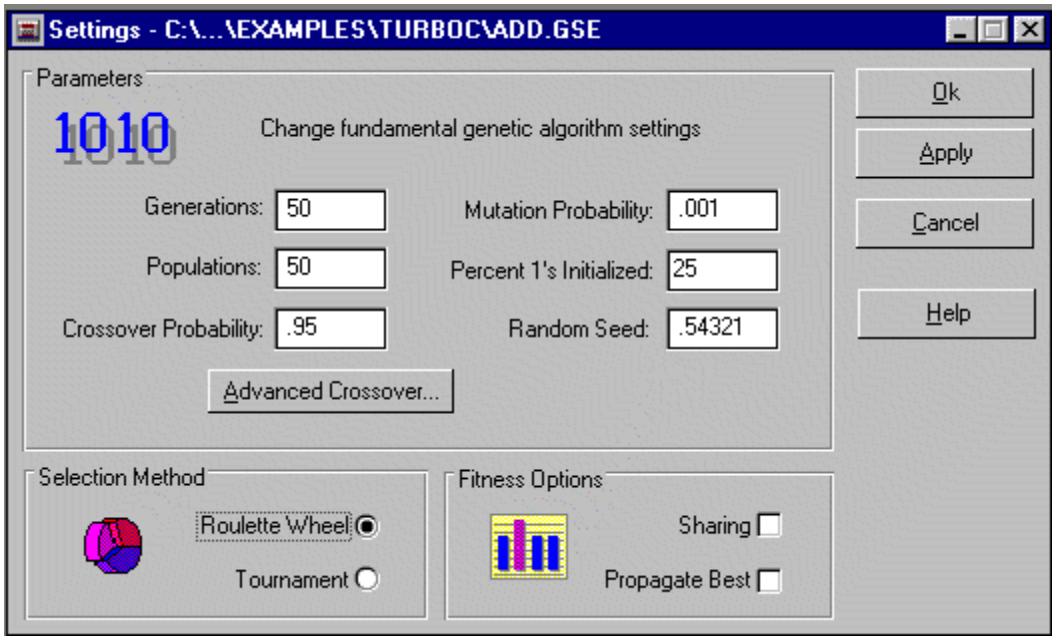


dr.Dudás László

Az Adaptive Software cég GaUI programja ..

Forrás: Adaptive Software

2. A GA paramétereinek beállítása
   - *Generációk száma*
   - *Populáció létszáma*
   - *Keresztezési ráta*
   - *Mutációs ráta*
   - *A kezdeti populáció inicializálási módszere*
   - *Kiválasztási módszer*
   - *Fitness opciók*

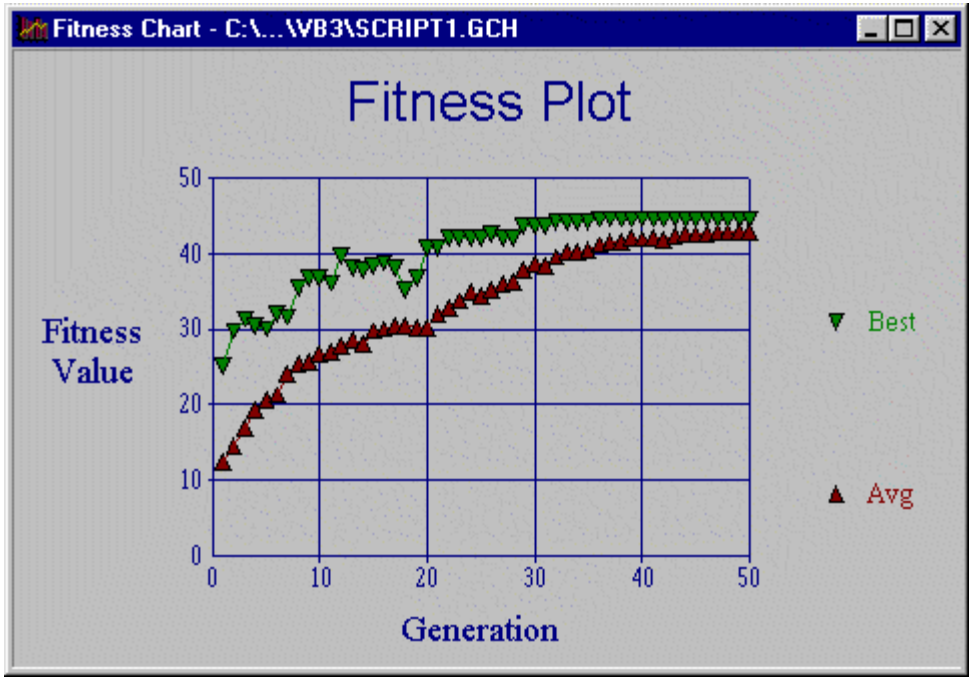


dr.Dudás László

Az Adaptive Software cég GaUI programja ..

3. Fitness értékek alakulásának futás közbeni kijelzése

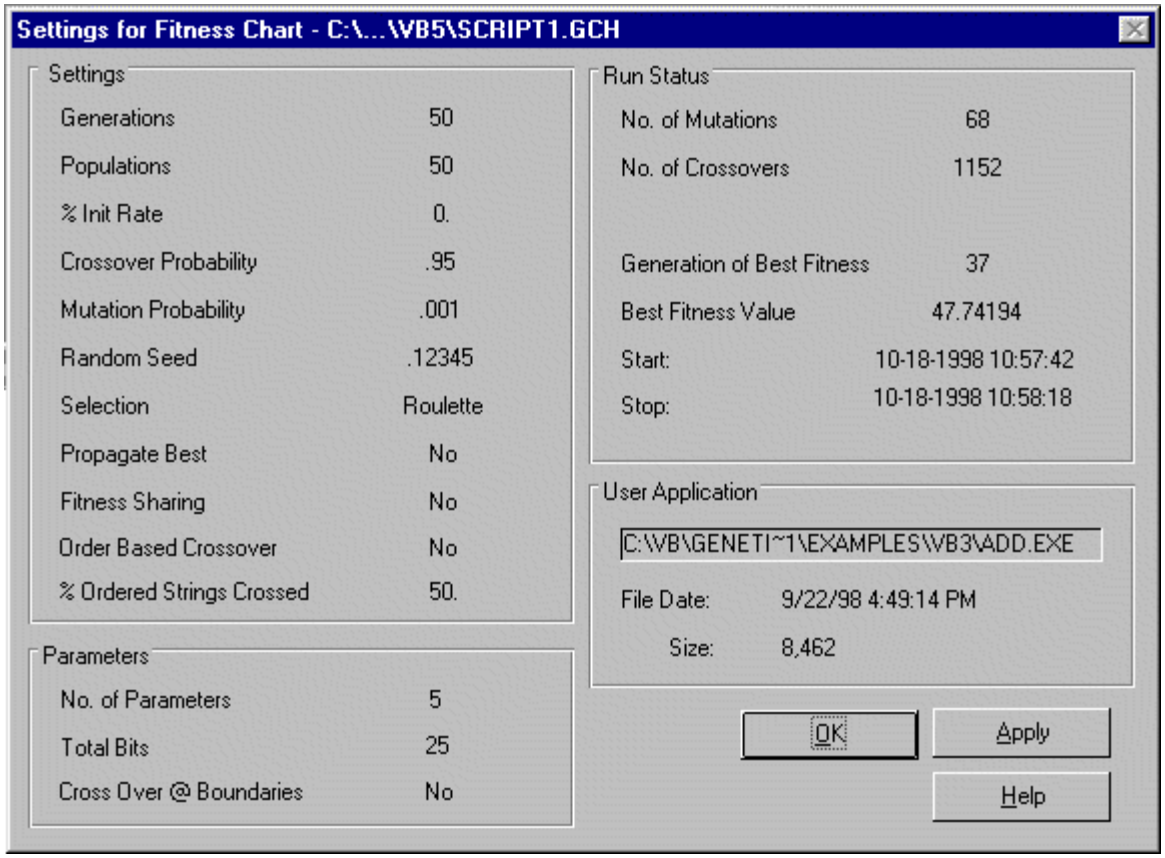
Kijelzi a legjobb egyed fittnessének és a populáció átlagos fittnessének alakulását.



Az Adaptive Software cég GaUI programja ..

3. Fitness értékek alakulásának futás közbeni kijelzése ..

További folyamat-állapot jelzők megjelenítése:



Az Adaptive Software cég GaUI programja ..

4. A genetikus kódok kijelzése

- Az összes, vagy csak a legjobb kromoszómák kijelzése
- A bináris kódalak és a valós értékek kijelzése
- A populáció tagjaihoz a szülők kijelzése

Best String View - C:\VB\GENETI~1\EXAMPLES\VB3\ADD.GBS

Generation: 1 - 50

Generation	Add 5 Unit1	Add 4 Unit1	Add 3 Unit1	Add 2 Unit1	Add 1 Unit1	Fitness	Average
1	01100	10011	11000	00010	11100	27.41935	12.8
2	10011	11000	01000	11010	10010	30.64516	17.2258
3	01100	11001	01100	11101	10000	30.32258	20.39354
4	11110	10010	01100	11000	10010	32.90322	22.44515
5	01100	10000	11100	11001	10000	31.29032	23.89032
6	11110	10000	11000	11101	01110	36.45161	24.39355
7	11100	11001	11000	10000	11000	37.74194	23.78064
8	11100	11000	11100	11101	00110	37.09677	24.28387
9	11100	10011	11000	10000	11000	35.80645	25.57419
10	11100	10011	11000	11000	10010	36.45161	27.18064
11	11110	10010	11100	11001	10000	37.74193	28.41935
12	11110	10011	11000	11000	11010	39.67742	29.65806
13	11110	10011	11000	11000	11010	39.67742	30.45161
14	10010	11010	11111	10010	11000	37.74193	30.00645
15	11110	11101	11100	11000	10010	41.6129	31.88387
16	11110	11101	11100	11010	10010	42.25806	33.05806
17	11110	11101	11100	11010	10010	42.25806	33.87096
18	11110	11101	11100	11010	10010	42.25806	35.23225
19	11110	11101	11100	11010	11000	44.19355	35.65161
20	11110	11101	11100	11000	10010	41.6129	36.85161

dr.Dudás László

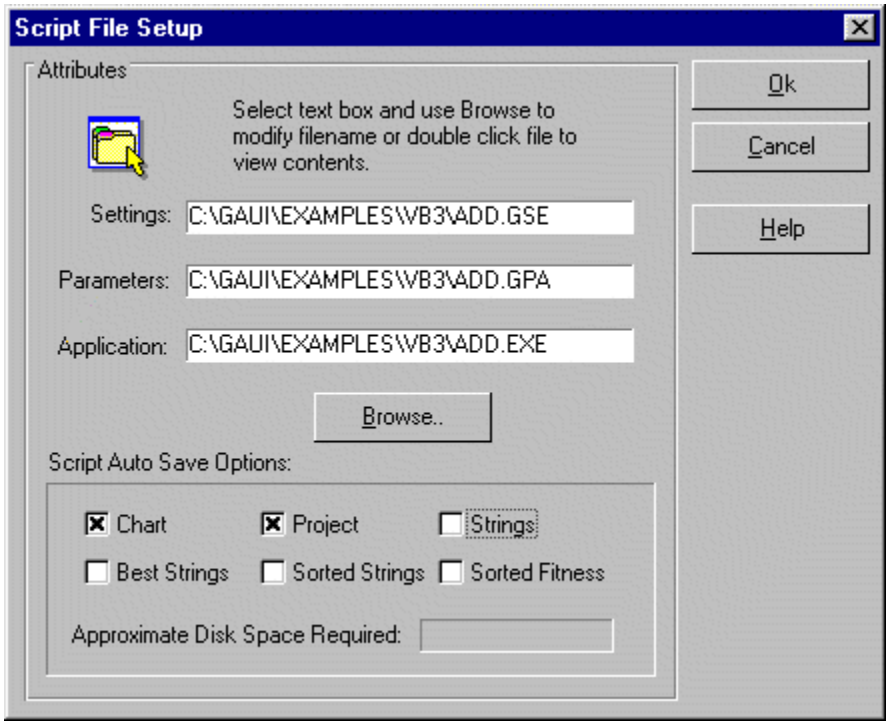
Az Adaptive Software cég GaUI programja ..

5. A genetikus kódok rendezése és rangsorolása

   - Csökkenő fittnessérték szerinti kijelzés
   - Rangsorolás szerinti kijelzés

6. A GaUI szolgáltatásainak automatizálása script-ek segítségével

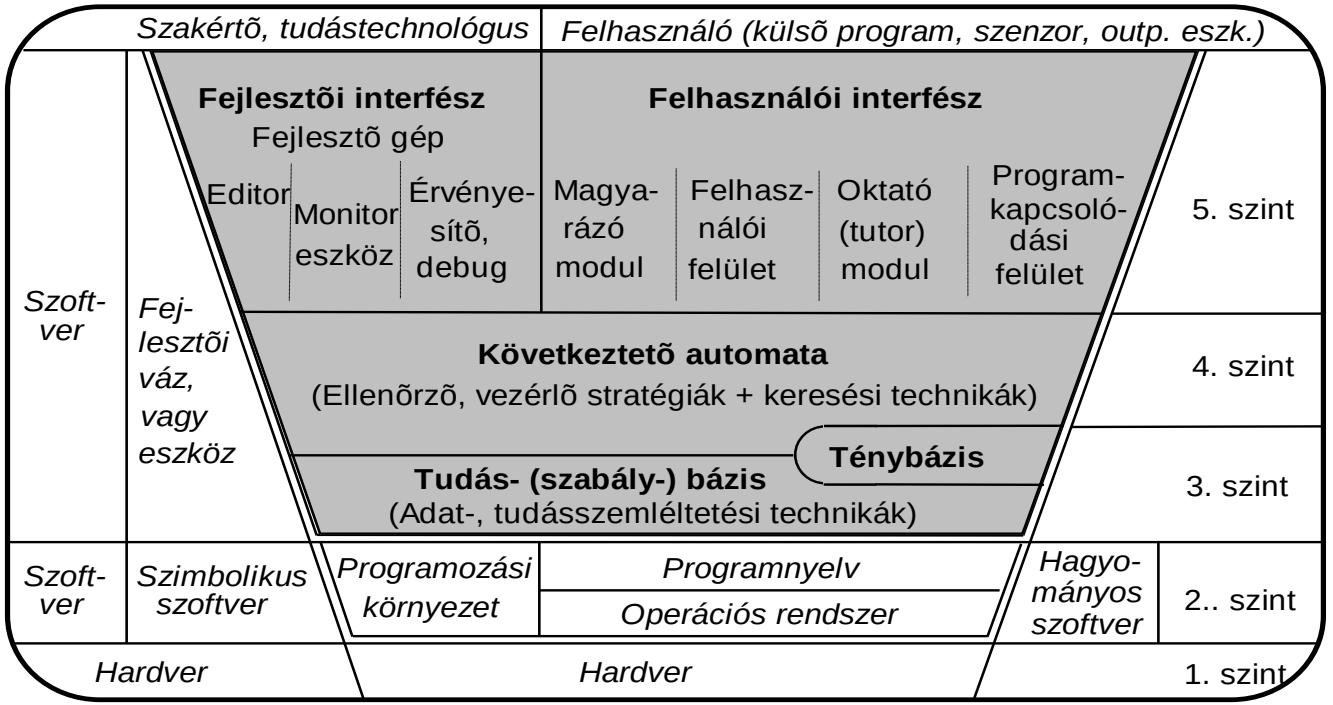
Nincs szükség script írásra, párbeszéd-panelokat kell kitölteni.



dr.Dudás László

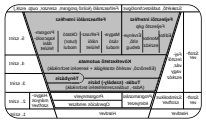
Szakértőrendszerek megvalósítása

Egy szakértőrendszer olyan eszköz, amely problémaspecifikus ismeret megértésére képes, és intelligensen használja a tématerület ismeretanyagát egy tevékenység különböző megvalósítási útjainak felvetéséhez. A szakértőrendszerek nem csak az ismeretátadás technikáit alkalmazzák, hanem analitikus, elemző eszközöket is az ismeret kiértékelésére, valamint tanulási technikákat.



Szakértőrendszer és beágyazó környezetének összetevői

Deborah D. Wolfgram - Teresa J Dear - Craig S. Galbraith: ***Expert Systems for the Technical Professional*** *John Wiley & Sons,New York, 1987.*



Szakértőrendszerek megvalósítása ..

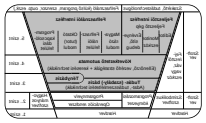
1. szint: a hardver

A szakértőrendszer alkalmazások futtatására szolgáló **dedikált** hardver jelentősen eltér a hagyományos alkalmazások futtatására előnyös hardvertől. A fő eltérések:

- Adattípus szerinti memóriaelérés
- Egyfelhasználós kivitel
- LISP célprocesszor.

A szimbolikus szoftverek futtatására szánt hardvernél kiemelt fontosságú a nagy sebesség.

Napjaink tendenciája, hogy a hagyományos hardver nagyfokú teljesítménynövekedésének, árcsökkenésének és elterjedtségének köszönhetően előtérbe kerül a mesterséges intelligencia alkalmazások céljaira, köztük a szakértőrendszerek futtatására is.



Szakértőrendszerek megvalósítása ..

1. szint: a hardver ..

Hardverplatformok:

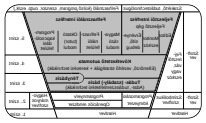
- Dedikált hardver, célszámítógép, elsősorban LISP processzorral
- Mainframe, fürtbe kötött, vagy hálózatban egyesített erőforrások
- Munkaállomás
- Mikroszámítógép

Dedikált hardver alkalmazásának indokai:

- Extrém nagy teljesítményigény
- Eleve erre készült szoftver

- Néhány célhardvert gyártó cég :
  - Symbolics
  - LMI (LISP Machine Inc)
  - Texas Instruments

dr.Dudás László



Szakértőrendszerek megvalósítása ..

2. szint: a szoftver

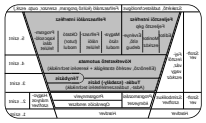
A szoftver adja a rendszer intelligenciáját. Két fajtája:

- Operációs rendszer+ programozási nyelv
- Dinamikus programozási környezet dedikált hardver esetén.

A mesterséges intelligencia szoftverek a hagyományos adatfeldolgozó alkalmazásokkal szemben tudásfeldolgozó alkalmazások. Ebből származik néhány eltérés a hagyományos szoftverekhez képest:

	Szimbolikus	Hagyományos
1.	Heurisztikák	Algoritmusok
2.	Szimbolikus módon elérhető ismeretbázis	Numerikusan címzett adatbázis
3.	Szimbolikus feldolgozás orientált	Numerikus-feldolgozás orientált
4.	Emberközeli interfészek, beszédfeldolgozás	Nagyrészt önálló, interakció nélküli működés
5.	Futásközbeni magyarázatadás	Nincs futásközbeni felhasználóbarát magyarázatadás
6.	A feladat leíró módja az ember által könnyen követhető szimbolizmust alkalmaz	A feladat a programnyelv logikájára van átfogalmazva, csak a nyelvet ismerő ember képes megérteni

dr.Dudás László



Szakértőrendszerek megvalósítása ..

2. szint: a szoftver ..

Szakértőrendszer-szoftverek felosztása:

- Általános célú MI-nyelvek: LISP, Prolog, Poplog, stb.
- Fejlesztőeszközök, szakértőrendszer-vázak, pl: KappaPC, GURU, stb.
- Kulcsrakész szakértőrendszerek, pl. Lending Advisor, stb.

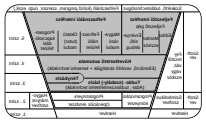
Az **általános célú MI nyelveken** történő fejlesztések időigényesek és költségesek. Napjainkban egyre több rendszer készül C++ nyelven.

Fejlesztőeszközök, szakértőrendszer-vázak

Váz = üres tudás (szabály)bázisú szakértőrendszer. Létrehozását a tudásbázis és a következtető automata éles elkülönülése teszi lehetővé. Új szakértőrendszerek kifejlesztéséhez fél megoldást nyújtanak.

dr.Dudás László

Szakértőrendszerek megvalósítása ..

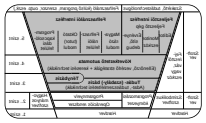


2. szint: a szoftver ..

Szakértőrendszer-**vázak képességei**:

- Tudásbázis fejlesztő **editor** a tudásbázis feltöltésére és karbantartására
- A tudásbázis integritásának, **konzisztenciájának** ellenőrzése, az ellentmondó, redundáns szabályok kiszűrése, a hiányzó információk jelzése (validálás)
- **Szintaktikai**, formai ellenőrzés
- Következtetésindoklás, **magyarázatadás**
- **Hibakeresés**, nyomkövetés, töréspontok elhelyezése teszteléshez
- **Grafikus** felület az információk és kapcsolatrendszerek megjelenítésére
- **Kapcsolódási felület** más programokhoz, pl. adatbázisokhoz, táblázatkezelőkhöz, Internethez, szenzorokhoz és beavatkozó szervekhez
- **Tudáskinyerés** szöveg alapú forrásokból
- Hanggenerálás és **hangfelismerés** az emberközeli kommunikáció elősegítésére természetes nyelvi interfész formájában.

dr.Dudás László



Szakértőrendszerek megvalósítása ..

2. szint: a szoftver ..

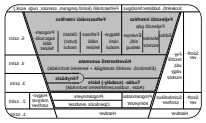
Szakértőrendszer-**vázak alkalmazásának**

- **előnye:** idő- és költségmegtakarítás
- **hátránya**: eleve adott tudásszemléltetési és következtetési technika. Ez a probléma nem merül fel a hibrid rendszereknél, pl: KES, PICON, GURU, KappaPC.

- Ismertebb szakértőrendszer-vázak: PLUS, KES, KEE, ART, RULEMASTER, EMYCIN.
- Alkalmazható szabályok száma: 100 – 10000.

Kulcsrakész szakértőrendszerek

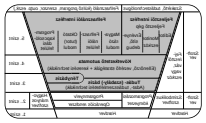
- Nem igényelnek fejlesztőmunkát, rögtön használhatók



Szakértőrendszerek megvalósítása ..

Kulcsrakész rendszer vásárlása előtt mérlegelendő

- **Létezik-e** a piacon a problémára megfelelő szakértőrendszer?
- Ki az a **szakértő**, akinek a tudását hordozza? Mások által is elfogadott szintet képvisel-e?
- **Illeszkedik-e** a tárolt tudás a megoldandó problémákhoz?
- **Megfelelő** adatszerkezeteket, keresési módszereket, **következtetési eljárásokat** alkalmaz-e?
- Kellően **teljes-e** az adott szakterület tudását illetően?
- Rugalmasan **bővíthető**, módosítható-e?
- Rendelkezik-e **interfész** felülettel más alkalmazásokhoz, hálózathoz?
- Illeszkedik-e a meglévő **hardverhez**?
- A **tudásbázis mérete** elegendően nagy-e?
- Belefér-e az **anyagi lehetőségekbe**?



Szakértőrendszerek megvalósítása ..

3. szint: a tudásbázis

A tudásbázis tartalmazza az ismeretet és szakértelmet. Minél teljesebb a tudás, annál erőteljesebb a szakértőrendszer. Fontos a tudás szemléltetésének módja:

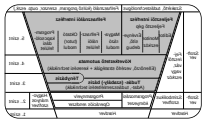
- Predikátum logika
- Szemantikus háló
- Keret
- Szabályalapú
- Hibrid modell.

4. szint: a következtető automata

Ez teszi **aktívvá** a tudásbázisban tárolt ismeretet.

Fontos jellemzője a következtetési folyamat **vezérlés stratégiája**.

Másik összetevője a **keresési módszer**, mely meghatározza, hogy milyen módon járja be a tudásbázist a következtető automata. **Lehet**: előrehaladó, visszafelé haladó, kétirányú, szembehaladó és a legkisebb kötelezettség elvén működő.



Szakértőrendszerek megvalósítása ..

5. szint: az interfész

Az interfész jelenti a kapcsolatot a külvilág felé.

- A szakértővel
- A tudástechnológussal

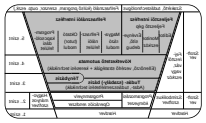
} *Fejlesztői interfész*

- A felhasználóval
- A szenzorokkal
- A beavatkozószervekkel
- Más alkalmazásokkal.

} *Felhasználói interfész*

A fejlesztői és a felhasználói interfész közrefogja a szakértőrendszer fejlesztésének folyamatát..

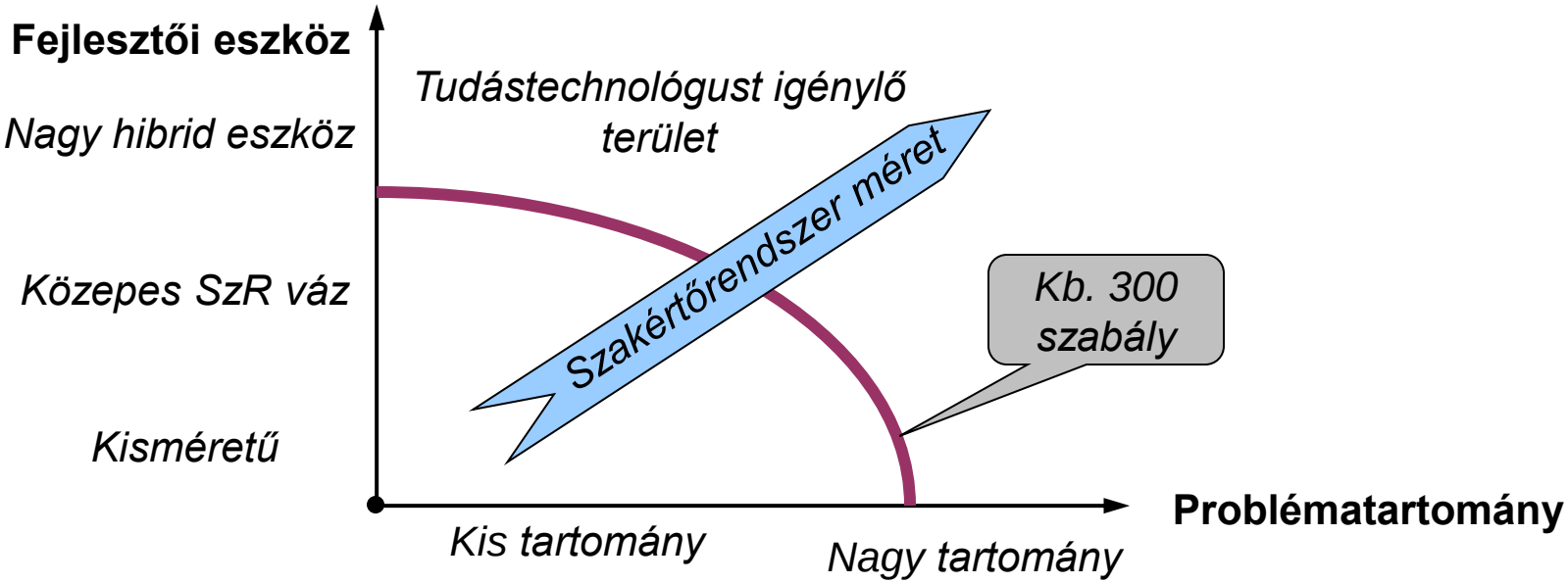
dr.Dudás László

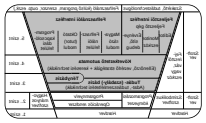


Szakértőrendszerek kifejlesztése

Méretfüggő fejlesztési megközelítés:

- Kis rendszereket: programozással nem hivatásszerűen foglalkozók, pl. a szakértő
- Közepes rendszereket: nem MI specialista programozók
- Nagy rendszereket: hivatásosak, tudástechnológus és szakértő, team.

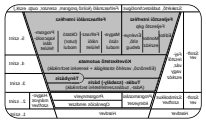




Szakértőrendszerek kifejlesztése ..

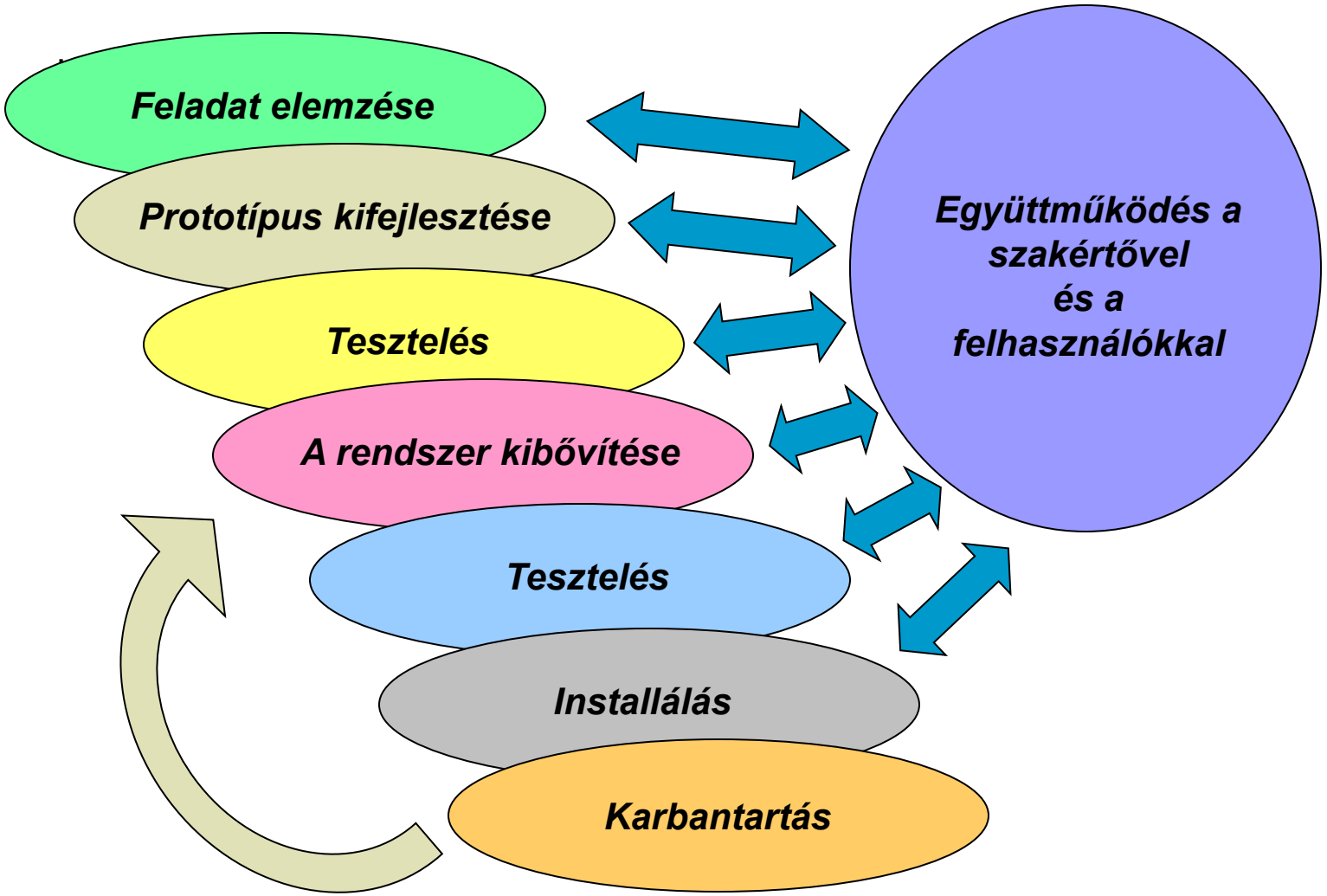
Szakértőrendszer kifejlesztése előtt mérlegelendő szempontok

- Alkalmas-e a feladat szakértőrendszerrel történő megoldásra?
- Létezik-e a feladatra kulcsrakész szakértőrendszer, vagy megfelelő paraméterekkel bíró szakértőrendszer váz?
- A szakértőrendszer kifejlesztési időigénye belefér-e a megrendelő elvárásaiba?
- Elérhető-e a kívánt színvonalú szakértői tudás?
- Rendelkezésre állnak-e a szükséges erőforrások (pénz, idő, hardver, szoftver, személyek) ?
- Eredményez-e a szakértőrendszer megvalósítása gazdasági előnyt ?
- Közepes és nagyméretű szakértőrendszerek csak tudástechnológus közreműködésével hozhatók létre.

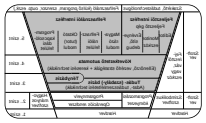


Szakértőrendszerek kifejlesztése ..

Kommunikáció a szakértőrendszer kifejlesztése közben



Szakértőrendszerek kifejlesztése ..



Szakértőrendszer kifejlesztésének lépései

Forrás: Paul Harmon - Rex Maus - William Morrissey:
Expert Systems: Tools & Applications
John Wiley & Sons, New York, 1988. p289.

1. Fejlesztői (Front End) elemzés
   - A megfelelő probléma beazonosítása
   - A gazdaságosság mérlegelése
   - A vezetés támogatásának megnyerése
2. Feladatelemzés
   - A feladat beazonosítása
   - Az új rendszer viselkedésének behatárolása
   - A szükséges tudás behatárolása
3. Prototípus kifejlesztése
   - Esettanulmányok készítése (kritériumok felállítása)
   - Egy kisméretű rendszerrel bizonyítani az elv helyességét és tapasztalatot szerezni
4. A rendszer kifejlesztése
   - Újragondolni az átfogó struktúrát, ha szükséges
   - Újabb tudással bővíteni a rendszert
5. Tesztelés
   - A rendszer tesztelése valódi felhasználókkal
   - Felülvizsgálat, ha szükséges
6. Installálás
   - Telepíteni a rendszert a működési környezet hardverén
   - Betanítani a felhasználókat
7. Karbantartás
   - A rendszer aktualizálása, ha szükséges.

dr.Dudás László