

BOTH AMBRUS
TDK DOLGOZAT

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
GÉPÉSZMÉRNÖKI KAR
ENERGETIKAI GÉPEK ÉS RENDSZEREK TANSZÉK



TDK DOLGOZAT

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
GÉPÉSZMÉRNÖKI KAR
ENERGETIKAI GÉPEK ÉS RENDSZEREK TANSZÉK

BOTH AMBRUS
TDK DOLGOZAT
Hőkörfolyamat paraméteroptimalizálása
genetikus algoritmussal

Témavezető:
Groniewsky Axel
tudományos segédmunkatárs

Budapest, 2014.10.10.

Szerzői jog © Both Ambrus, 2014.

Szerzői jog © Groniewsky Axel, 2014.

TARTALOMJEGYZÉK

Mottó	vii
Jelölések jegyzéke	viii
1. Bevezetés	1
1.1. Célkitűzések	1
1.2. Áttekintés	1
2. Szakirodalmi áttekintés	2
2.1. Evolúciós algoritmusok	2
2.2. Alkalmazási példák	3
3. A Cycle-Tempo működése	5
3.1. A Cycle-Tempo számításának alapja	6
3.2. Cycle-Tempo szubrutinok	8
3.2.1. APSUB	9
3.2.2. PIPSUB	9
3.2.3. EPILOG	10
3.2.4. Szubrutinok a Cycle-Tempo futásában	10
4. Genetikus algoritmusok	11
4.1. Kanonikus genetikus algoritmus	11
4.1.1. Reprezentáció	11
4.1.2. Jósági függvény (fitness)	12
4.1.3. A kanonikus genetikus algoritmus lefutása	12
4.1.3.1. Inicializálás	13
4.1.3.2. Kiértékelés	13
4.1.3.3. Kiválasztás	13
4.1.3.4. Genetikus műveletek	14
4.1.3.5. Reprodukció	15
4.1.3.6. Kilépési feltétel	15
4.2. Saját genetikus algoritmusom felépítése	16
4.2.1. Reprezentáció	16
4.2.2. Fitness	16
4.2.2.1. Lineáris skálázás	17
4.2.2.2. Szigma csonkolás	19
4.2.2.3. Hatvány alapú skálázás	19
4.2.3. Genetikus algoritmus lefutása	20
4.2.3.1. Inicializálás	20

4.2.3.2. Kiértékelés	20
4.2.3.3. Kiválasztás	21
4.2.3.4. Keresztezés	21
4.2.3.5. Mutáció	22
4.2.3.6. Reprodukció	23
4.2.3.7. Konvergencia kiértékelése	24
5. Genetikus algoritmus működése	26
5.1. Tesztfüggvények	26
5.1.1. Himmelblau-függvény	26
5.1.2. Ugrásfüggvény	27
5.1.3. Ackley-függvény	27
5.1.4. Fletcher-Powell-függvény	28
5.2. Algoritmus működése tesztfüggvényeken	28
5.2.1. Himmelblau-függvény	29
5.2.2. Ugrásfüggvény	30
5.2.3. Ackley-függvény	31
5.2.4. Fletcher-Powell-függvény	33
5.2.4.1. Konvergencia megállapítására alkalmas mennyiségek	33
5.2.5. A vizsgálatok értékelése	35
5.3. Cycle-Tempoval együttműködve	36
5.3.1. Kapcsolat megvalósítása	36
5.3.2. Ellenőrzés	38
5.3.3. Alkalmazás	40
5.3.3.1. A vizsgált rendszer	40
5.3.3.2. Teszt	44
6. Eredmények értékelése	48
6.1. Eredmények	48
6.2. Javaslatok	48
7. Felhasznált források	49
8. Mellékletek	50

MOTTÓ

„Miféle korlátot lehetne szabni egy olyan erőnek, amely hosszú korokon átnyúlva működik, és minden lény alkatát, felépítését és szokásait szigorúan megvizsgálja: támogatja a jókat, elveti a rosszakat? Nem látom, hogy ezt az erőt bármi korlátozhatná abban, hogy az összes élő formát lassan, szépen hozzáalakítsa akár a legbonyolultabb életviszonyokhoz is.”

Charles Darwin: A fajok eredete

JELÖLÉSEK JEGYZÉKE

Latin betűk

Jelölés	Megnevezés, megjegyzés, érték	Mértékegység
f	Célfüggvény	1
ff	Fitness függvény	1
NSS	A keresési tér dimenzióinak száma	1
X	Kromoszóma	1
XLIM	Keresési tér határai	1
XR	Pont a keresési térben	-

Indexek, kitevők

Jelölés	Megnevezés, értelmezés
i	általános futóindex (egész szám)
a	alsó határ
f	felső határ
$\min$	minimum
$\max$	maximum

1. BEVEZETÉS

1.1. Célkitűzések

A munkám célja olyan optimum kereső eljárás készítése, amely képes meghatározni egy hőkörfolyamat paramétereinek optimumát olyan komplex problémák esetén is, ahol a hagyományos eljárások téves eredményekre vezetnek, vagy egyáltalán nem használhatók. Az optimalizálás célja esetemben a hatásfok maximális értékének megtalálása. Mérnöki szemmel tekintve ez a helyes irány, de nem szabad elfeledkezni arról, hogy a valóságban a döntéseknek mindig gazdasági alapja van.

A feladatot a Cycle-Tempo 5.0 hőséma számító program (továbbiakban CT) és az általam írt genetikusan algoritmus (továbbiakban GA) valamint néhány egyéb software együttműködésével hajtom végre. Az optimalizálás során a GA futtatja a CT-t, kiolvassa és kiértékeli a CT számítási eredményeit.

A módszerem legfőbb előnye, hogy a bonyolultabb, pontosabb hőséma számító programokkal szemben jóval gyorsabb a futása. Ezáltal a munkám irányadó jellegű eredményeket szolgáltat elfogadható időn belül, valamint alkalmas az említett pontosabb programokkal együttműködő genetikusan algoritmusok paramétereinek helyes beállítására.

A GA alkalmazásától azt várom, hogy a hagyományos módszerekhez képest legalább 0,1%-pontos nagyságrendű hatásfokjavulást eredményezzen.

1.2. Áttekintés

A dolgozatomban először bemutatom a genetikusan algoritmus alkalmazásának múltját. Ezután a CT működését és a különböző beavatkozási és adatkapcsolati lehetőségeket tárgyalom. Majd bemutatom a genetikusan algoritmus elvét, és az általam használt genetikusan algoritmus részleteit. Ezután leírom a CT és a GA együttműködését, valamint összefoglalom az elért eredményeket és az előttem álló munkát.

2. SZAKIRODALMI ÁTTEKINTÉS

2.1. *Evolúciós algoritmusok*

Az evolúció utánzásának, irányításának gondolata már régóta foglalkoztatja az emberiséget. Elég csak a különböző növény- és állatnemesítésekre gondolni, ahol a szakember kijelöl egy (vagy több) céltulajdonságot, és ez alapján szaporítja az adott faj egyedeit.

Az evolúció műszaki és informatikai problémákra való átültetését az 1950-es években kezdték. Ezt a számítástechnika fejlődése tette lehetővé, mivel minden evolúciós algoritmusnak igen nagy a számítási igénye, és elfogadható időn belül elfogadható eredményt csak gépi számításokkal kaphatunk. Ezek a módszerek műszaki problémák megoldásait–mint egyedeiket– populációkba foglalják és ezeken a populációkon különböző műveleteket végezve keresik a legjobb megoldást. (ÁLMOS ET AL., 2002)

Az egyik első alkalmazás az evolúciós programozás volt, ahol magát a programfejlesztést próbálták automatizálni. Itt még csak a mutáció jelent meg a genetikus műveletek közül, ami a programkód véletlenszerű megváltoztatását jelentette.

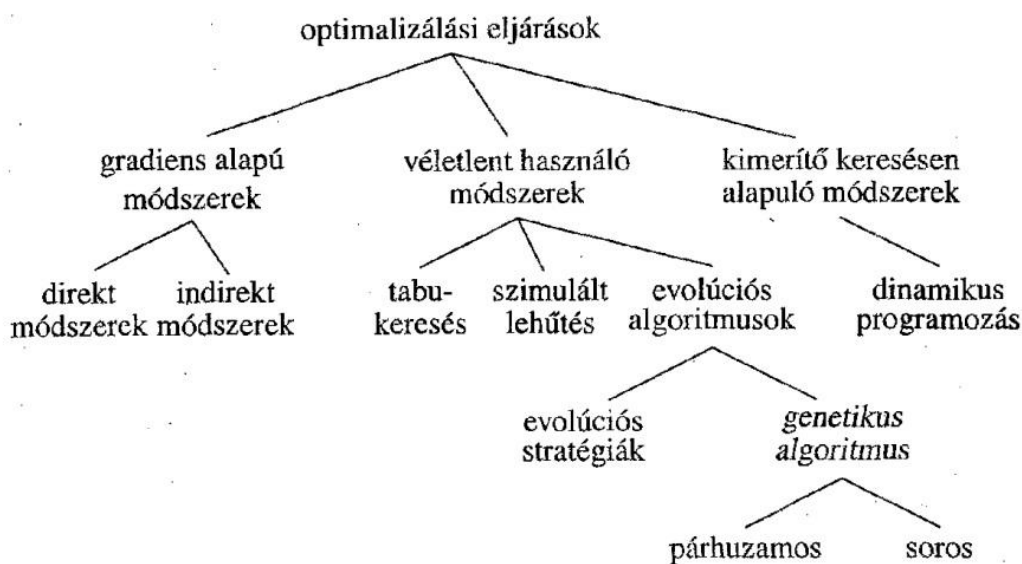
Az algoritmusok csoportosítása –természeti analógiát használva– lehetséges a szülők és utódok száma, és a szülő- és utódgeneráció új generációvá alakításának módja szerint. A jelölések általános esetben: $(\mu+\lambda)$ ha a szülők és az utódok közösen alkotják az új generációt, és (μ,λ) ha az új generáció csak az utódokból áll. $\lambda \geq \mu$

Az evolúciós programozás kutatása során került kifejlesztésre az $(1,\lambda)$ stratégia, ami mutációval dolgozik, és a legjobb utódot teszi meg a következő szülőnek. Alkalmaztak továbbá $(1+1)$ és $(\mu+\mu)$ stratégiát, ahol a szülő(k) és az utód(ok) közül a jobbik fél kerül megőrzésre. Ezek a módszerek nem használták ki két egyed keresztezésének lehetőségét, és az utódok kiválasztási módszerével is akadtak gondok, így ezek az algoritmusok nem tudtak versenyezni a konkurens módszerekkel. (ÁLMOS ET AL., 2002)

Másik fejlesztési terület az evolúciós stratégiáké, melyek a mutációra helyezik a hangsúlyt. A kis mutációk jelentőségét kiemelve nulla várható értékű standard eloszlású mutációt alkalmaznak. Ezzel a módszerrel a generációk számát tekintve létrehoztak $(1+1)$ és $(\mu+1)$ stratégiát, utóbbinál az utód a teljes generáció valamilyen lineáris kombinációja, és ha jobb, mint a legrosszabb szülő egyed, akkor vele helyettesítik. Ennek mintájára létrehoztak $(\mu+\lambda)$ és (μ,λ) stratégiát. (ÁLMOS ET AL., 2002)

Ezzel szemben a genetikus algoritmus a mutációt és a szülők keresztezését egyaránt alkalmazza. Általában $(\mu+\mu)$ vagy (μ,μ) -ként írható le a generációk kapcsolódása. További különbség a kiválasztásban van, ugyanis a genetikus algoritmusok működéséhez elengedhetetlen egy kritériumrendszer, ami alapján *számszerűsíthető* az egyedek jósága (fitnesz.) A genetikus algoritmusoknál a kiválasztás is valószínűségi alapú, így az optimalizálási problémák során csökken lokális szélsőértékek dominanciája és megelőzhető a korai konvergencia.

A genetikus algoritmus optimalizálási módszerek közti besorolását a 2-1. ábra mutatja. A GA alkalmazása akkor indokolt, ha számítási igény és pontosság alapján felül tudja múlni a többi algoritmust.



2-1. ábra. Optimumkereső eljárások csoportosítása
(ÁLMOSE ET AL., 2002, 21.o.)

2.2. Alkalmazási példák

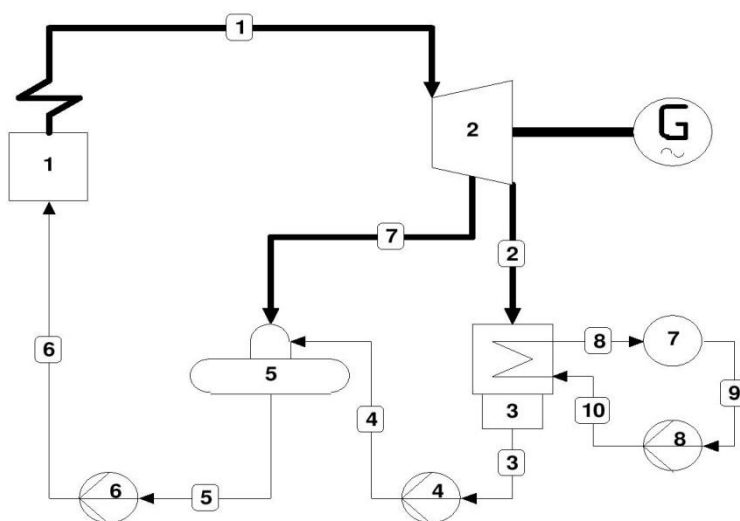
A genetikus algoritmust számos probléma megoldására alkalmazták energetikai területen is. KAVIRI ÉS TÁRSAI (2012) valamint VLADÉS ÉS TÁRSAI (2003) kombináltciklusú erőművet, QIN ÉS TÁRSAI (2003) gőzturbina fokozatot, XI ÉS TÁRSAI (2013) ORC rendszert továbbá SUBBARAJ ÉS TÁRSAI (2008) gazdaságos teherelosztást optimalizáltak genetikus algoritmussal.

Közülük többen is sikeresen használták a valós reprezentációt (kódolást), amit én is alkalmaztam. (SUBBARAJ ET AL., 2008) (XI ET AL., 2013) Ezt a kódolási módszert GAFFNEY ÉS TÁRSAI (2010) is támogatják. Ezen kívül ÁLMOS ÉS TÁRSAI (2002) is megemlíti, hogy ez a lehetőség egyes esetekben eredményes lehet.

SUBBARAJ ÉS TÁRSAI (2008) azonos környezetben hasonlították össze a különböző optimumkeresési eljárásokat. Két szempontot vettek figyelembe, a futási időt és a legjobb elért eredményt. Az általuk fejlesztett valós kódolású alkalmazkodó genetikus algoritmus az összes többinél jobbnak bizonyult mindkét szempontból.

3. A CYCLE-TEMPO MŰKÖDÉSE

A Cycle-Tempo 5.0 egy termodinamikai modellezésre szolgáló szoftver. A modell berendezések csövekkel összekötött hálózatát jelenti. (3-1. ábra) A CT fő célja kiszámolni ezen a hálózaton a csövek energia- és tömegáramait. A felhasználó rendelkezésére álló nagyszámú berendezés modell, és a hálózat topológiájának szabad megváltoztatása miatt a programban szinte korlátlan rugalmassággal kezelhetők a problémák. (TU DELFT, 2007a)



3-1. ábra: Példa egy körfolyamat kapcsolására CT-ban (TU DELFT, 2007b, 2-3. o.)

A CT-t és a GA-t is az alábbi környezetben futtattam, a CT-nak ezek a körülmények megfelelnek, a GA-nak sem okoz problémát az eddigi tapasztalatok szerint:

- Operációs rendszer: Windows 7 Home Premium
- Rendszer típusa: 32 bites operációs rendszer
- Processzor: Intel Pentium ® Dual-Core CPU T4200 @ 2.00 GHz
- Memória (RAM): 2.00 GB

3.1. A Cycle-Tempo számításának alapja

A CT futtatása során a program fő célja az egyes csövek tömegáramainak meghatározása. A csövek közt a kapcsolatot az egyes berendezések teremtik meg. A berendezésekre felírhatók tömeg- és entalpiamérlegek. A tömegáramok kiszámíthatóságának kritériuma, annyi *független* egyenlet felírása ahány ismeretlen tömegáram (azaz ahány cső) van. (TU DELFT, 2007b)

A fent említett mérlegegyenleteket a CT egy mátrixegyenlettel fejezi ki: (TU DELFT, 2007b)

$$\mathbf{S} \cdot \mathbf{m} = \mathbf{e} \quad (3.1)$$

Ahol:

- **S** a rendszermátrix: tömegmérleget kifejező sorok esetén +1;-1 szerepel tekintettel a berendezés ki- és belépő csonkjaira. Entalpiamérleget kifejező sorok esetén a berendezés ki- és belépő csonkjaira számított entalpia szerepel benne megfelelő előjellel.
- **m** a tömegáramok vektora
- **e** a mérlegegyenletek jobb oldalát tartalmazó vektor. Ennek értékei általában nullák, kivéve, ha az adott egyenlet valamilyen kívülről megadott kényszert fejez ki. Ez a kényszer tömegmérlegek esetén tömegforrást vagy -nyelőt jelent, entalpiamérlegek esetén a környezet és a berendezés közti entalpiaforgalmat. (Például turbina belső teljesítményét.)

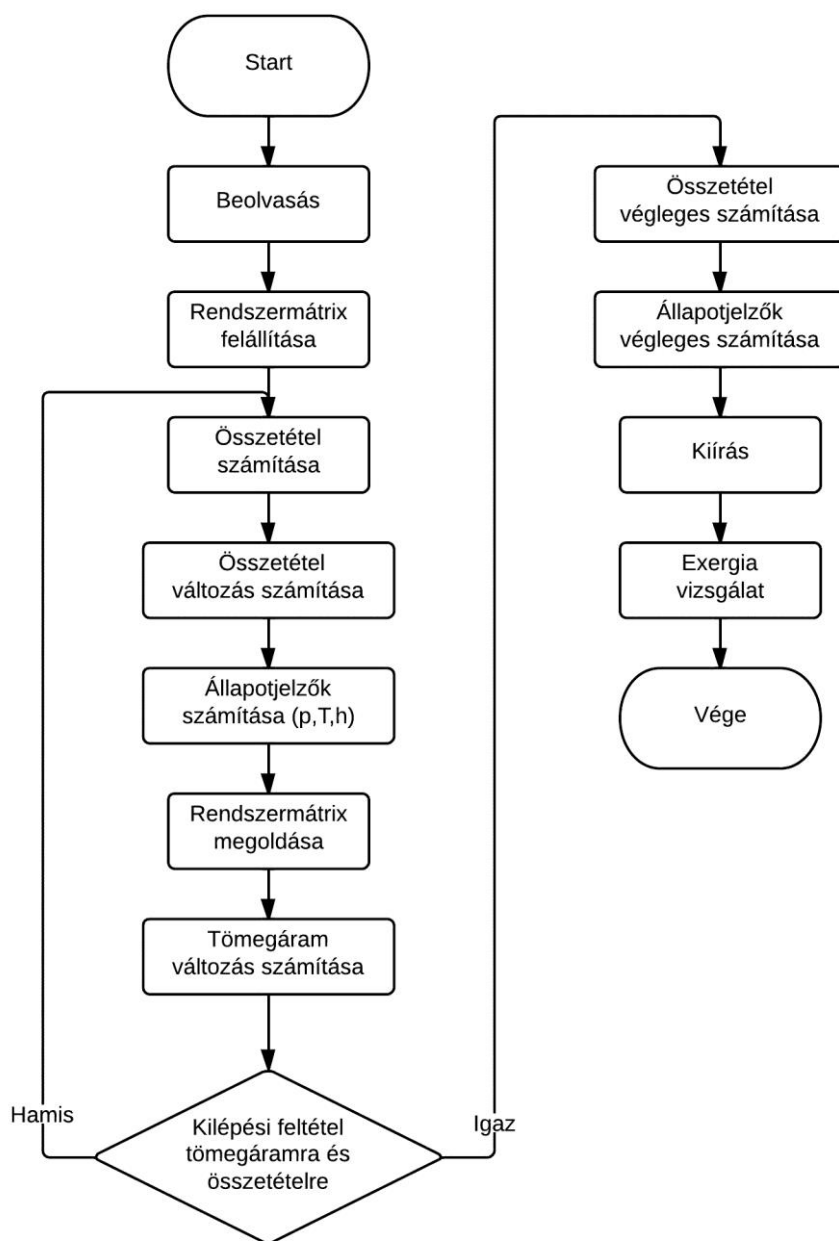
Erre láthatunk példát a 3-2. ábrán, ami a 3-1. ábrán látható kapcsolásra írtak fel. Az első 7 sor a tömegmérlegeket fejezi ki, az utolsó 3 sor pedig az entalpiamérlegeket.

		Cső sorszáma →												
Egyenlet eredete	nr.	1	2	3	4	5	6	7	8	9	10			
Kazán	1	-1					-1					m_1		0
Turbina	2	1	-1					-1				m_2		0
Kondenzátor	3		1	-1					-1		1	m_3		0
Kondenzátor	3								-1		1	m_4		0
GTT	5				1	-1		1				m_5		0
Tápszivattyú	6					1	-1					m_6		0
Hűtővíz szivattyú	8									1	-1	m_7		0
GTT	5				h_4	$-h_5$		h_7				m_8		0
Kondenzátor	3		h_2	$-h_3$					$-h_8$		h_{10}	m_9		0
Turbina	2	h_1	$-h_2$					$-h_7$				m_{10}		P_{tur}

3-2. ábra. Egy példa a rendszermátrixra (TU DELFT, 2007b, 2-3. o.)

A megoldást a rendszer iteratív úton végzi, ennek folyamatát a 3-3. ábra mutatja. A CT az állapotjelzők számítását a berendezések viselkedését leíró egyenletekből számítja. Majd a kiszámolt entalpiákat a rendszermátrixba helyettesíti, és azt Gauss eliminációval oldja meg. (TU DELFT, 2007b)

A kilépési feltétel a koncentráció- és tömegáram változások egy adott szint alá csökkenése minden egyes csőben. Ha a konvergencia nem következik be, akkor egy megadott iterációszám után a program kilép a ciklusból.



3-3. ábra: A CT számítási folyamata (TU DELFT, 2007b, 2-8. o.)

A körfolyamatok modellezése során előfordulhat olyan eset, amikor elegendő számú független egyenlet írható fel, de a megoldást jelentő állapotban az egyes berendezések nem képesek üzemelni. Ilyenkor a CT általában figyelmeztetést (warning) ad a felhasználónak. Ilyen eset például, amikor csak úgy oldható meg a rendszer mátrix, hogy egy hőcserélőben a hőmérsékletprofilok metszik egymást, vagy egy csőben a tömegáram negatív lesz. Ezeket az eseteket a GA-nak tudnia kell kezelni.

3.2. *Cycle-Tempo szubrutinok*

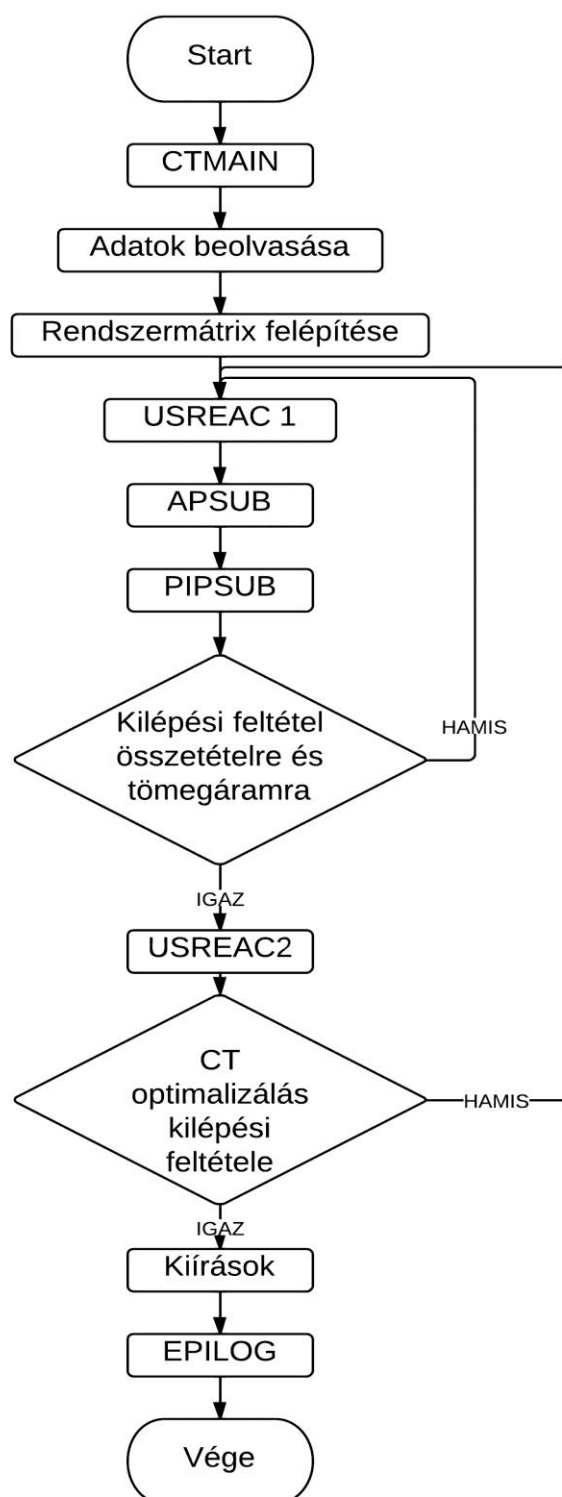
A genetikus algoritmus alkalmazásához szükséges, hogy a CT paramétereit kívülről lehessen megadni, ezért fontos a szubrutinok megértése. A szubrutinok használatáról szóló dokumentum sok tekintetben pontatlan, így a szubrutinok működésének felderítése jelentős időbe telt a munkám során.

A CT szubrutinjai olyan alprogramok, melyeket a főprogram meghív, végrehajtja az utasításait, majd visszatér. A CT-t Fortran programnyelven írták, így a szubrutinokat is Fortranban kell megírni. A Fortranban szubrutin és a főprogram között csak a ki- és bemeneti változók teremtenek kapcsolatot. Ebből következik, hogy a CT működésére csak korlátozott befolyással van a szubrutin.

Az alábbi szubrutinok állnak rendelkezésre: (TU DELFT, 2007c)

- CTMAIN: a rendelkezésre álló memóriát változtatja
- APSUB: az egyes berendezések viselkedését írja felül
- PIPSUB: a csövek viselkedését módosítja
- USREAC: felhasználó által adott kémiai reakció leírására alkalmas
- FUNCOP: a beépített optimalizálás célfüggvényét módosítja
- EPILOG: utómunkálatokra, adatkezelésre alkalmas szubrutin

Ezek közül az APSUB, a PIPSUB és az EPILOG szubrutinokat használok a GA-sal való együttműködéshez.



3-4. ábra. Szubrutinok a CT futása során
(TU DELFT, 2007c)

3.2.1. APSUB

Ez a szubrutin a berendezések viselkedését módosítja. Azokat a paramétereket lehet általa megváltoztatni, amit a berendezés felhasználói felületén is meg lehet adni. Az esetleges részsámításokhoz rendelkezésre áll a berendezéshez kapcsolódó összes csőre a tömegáram és az állapotjelzők. (TU DELFT, 2007c)

A GA-t tekintve ezt a szubrutint a berendezések paramétereinek teljes megváltoztatására használom. Az adatkapcsolat egy fájlön keresztül valósul meg, amiből az APSUB szubrutin kiolvassa a kívánt paramétereket, és a berendezés paramétereit azzal helyettesíti.

3.2.2. PIPSUB

Ez a szubrutin az egyes csővekre van hatással. Itt az adott csőre vonatkozó tömegáram, állapotjelző és összetétel adatok állnak rendelkezésre, és az állapotjelzők megváltoztatására van lehetőség. (TU DELFT, 2007c)

A GA szempontjából szerepe hasonló az APSUB-éhoz.

3.2.3. EPILOG

Az EPILOG szubrutinban a legutolsó programrészlet a CT futtatása során. Ez azt jelenti, hogy csak bemenő paraméterei vannak, amit szabadon fel lehet használni utólagos számításokra. (TU DELFT, 2007c) Valamint ezzel a szubrutinnal hozzá lehet férni a CT által automatikusan elkészített kimeneti fájlokhoz is, amikből például a warningok számát és más származtatott adatokat lehet kiolvasni megspórolva ezeket a számításokat.

A GA szempontjából ez a végeredmények fájlba való kiírására lesz alkalmas, ami a GA bemenő adataként szolgál.

3.2.4. SZUBRUTINOK A CYCLE-TEMPO FUTÁSÁBAN

A szubrutinok adta lehetőségek megértéséhez szükséges tudni, hogy a főprogram hol hívja meg azokat. Ezt a 3-4. ábra szemlélteti.

Az APSUB és PIPSUB szubrutinok az állapotjelzők és az összetétel számításába tudnak beavatkozni (3-3. ábra). Ez azt jelenti, hogy a rendszermátrix felépítésének a szubrutinok nélkül is megvalósíthatónak kell lennie, azaz a szubrutinok használatától függetlenül az adatokat meg kell adni a CT-ban is.

4. GENETIKUS ALGORITMUSOK

Ebben a fejezetben leírom a kanonikus genetikusan felépítését az általam írthoz képest fennálló különbségekre koncentrálni, majd ismertetem a saját programom is.

4.1. Kanonikus genetikusan algoritmus

Az 1975-ben John Holland által javasolt, darwini evolúciót és a genetika alapjait magába foglaló klasszikus eljárást nevezzük kanonikus genetikusan algoritmusnak. (ÁLMOS ET AL., 2002) A Holland-féle genetikusan algoritmus megértéséhez előbb szükséges tisztázni néhány fontos kérdést.

4.1.1. REPRESENTÁCIÓ

A GA-ban az egyedek a keresési tér vizsgált pontjait jelölik. Az egyed az algoritmus egy kromoszómaként képezi le. Ez a *kromoszóma* Holland javaslatára egy bináris sztring, mivel ez kellőképpen egyszerű, és valamilyen mértékben hasonlít a DNS felépítésére. A bitsztring egyes elemeit ugyanezen okból géneknek nevezzük. A 4-1. ábrán erre láthatunk példát.

0	1	1	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---

4-1. ábra. Példa bináris sztringre

A reprezentáció kulcskérdése, a keresési tér és a kromoszóma közti kapcsolat, azaz a kódolás leírása. Bináris kromoszóma esetén ennek legszélesebb körben használt típusa a Gray-kód, ami a keresési tér szomszédos pontjaihoz olyan kromoszómákat rendel, amik csak egy génben térnek el. (GAFFNEY ET AL., 2010)

A bináris reprezentáció mellett szól a Holland által levezetett skémaelmélet, amelynek egyik következtetése, hogy egy genetikusan algoritmus teljesítménye annál rosszabb, minél több értéket vehet fel egy gén. (ÁLMOS ET AL., 2002)

FRIESE ÉS TÁRSAI (1998) vizsgálataik során kanonikus genetikusan algoritmust alkalmaztak, és úgy találták, hogy nagyméretű folytonos keresési terekre a bináris reprezentáció felbontása problémákat okoz, mivel az algoritmus így csak megközelíti az optimumot, de nem találja

meg a globális optimum pontos helyét. Épp ezért ők gradiens alapú módszerekkel ötvözték a genetikus algoritmust.

Számos gyakorlati példa –többek közt ez a munka is– bizonyítja, hogy a bináristól eltérő reprezentáció is jól teljesít. (GAFFNEY ET AL., 2010) (XI ET AL., 2013) (SUBBARAJ ET AL., 2008) (ÁL-MOS ET AL., 2002)

4.1.2. JÓSÁGI FÜGGVÉNY (FITNESZ)

Az evolúciót modellezve a GA-nak képesnek kell lennie az egyedeket (megoldásokat) egy jósággal jellemezni. Ez alapján tesz különbséget az algoritmus az egyedek közt. A különbségtétel elengedhetetlen az evolúció megvalósulásához.

Az optimalizálási problémák általában egy *célfüggvény* (objective function) szélsőértékének megkeresését jelentik. Ez a célfüggvény a keresési tér pontjainak egyértékű függvénye. Azonban a GA működéséhez nem szükséges, hogy minden pontban értelmezett legyen a célfüggvény és a folytonosság sem elvárás.

A GA-ban az egyedek megkülönböztetésére nem a célfüggvényt használjuk, hanem valamilyen átalakítást végzünk rajta azért, hogy jobban megfeleljen a különbségtételre. Ezt az átalakítást *skálázásnak* nevezzük, skálázott értéket pedig jóságnak vagy *fitnesznek*. (ÁL-MOS ET AL., 2002) A fitnesszel szembeni legfontosabb elvárás, hogy a jobb egyedekhez magasabb fitness tartozzon, valamint a GA többi művelete szempontjából lényeges, hogy pozitív legyen.

Kanonikus genetikus algoritmus esetén a skálázás lineáris. Az 4.1 egyenlet alapján:

$$ff_i = \frac{f_i}{\bar{f}} \quad (4.1)$$

Ahol $\bar{f}$ az aktuális generációban a célfüggvény átlaga.

4.1.3. A KANONIKUS GENETIKUS ALGORITMUS LEFUTÁSA

A kanonikus genetikus algoritmus folyamatát a 4-2. ábra mutatja. A generáció kifejezés mind az iterációs ciklus sorszámára mind az egyedek csoportjára utalhat, utóbbit esetenként populációnak is nevezik. Az alábbiakban röviden leírom az egyes részfolyamatok működését.

4.1.3.1. Inicializálás

A GA működéséhez szükséges egy populáció, amin végezheti a különböző genetikus műveleteket. A kezdeti populációt az inicializálás során adjuk meg. Ez általában véletlenszerűen történik. Mivel a GA-t általában olyan problémákra alkalmazzák, amik túl bonyolultak a kisebb számítási igényű módszerek alkalmazására, valamint a széleskörű a priori ismeretek sem jellemzik ezeket a problémákat, így az inicializálás külső irányítása általában megjósolhatatlan következményekkel jár. (ÁLMOS ET AL., 2002)

Bináris kódolás használatával a véletlen inicializálás azt jelenti, hogy a létrehozott kromoszómák minden egyes génjét 50%-50% valószínűséggel 0 vagy 1 értékre állítjuk. FRIESE ÉS

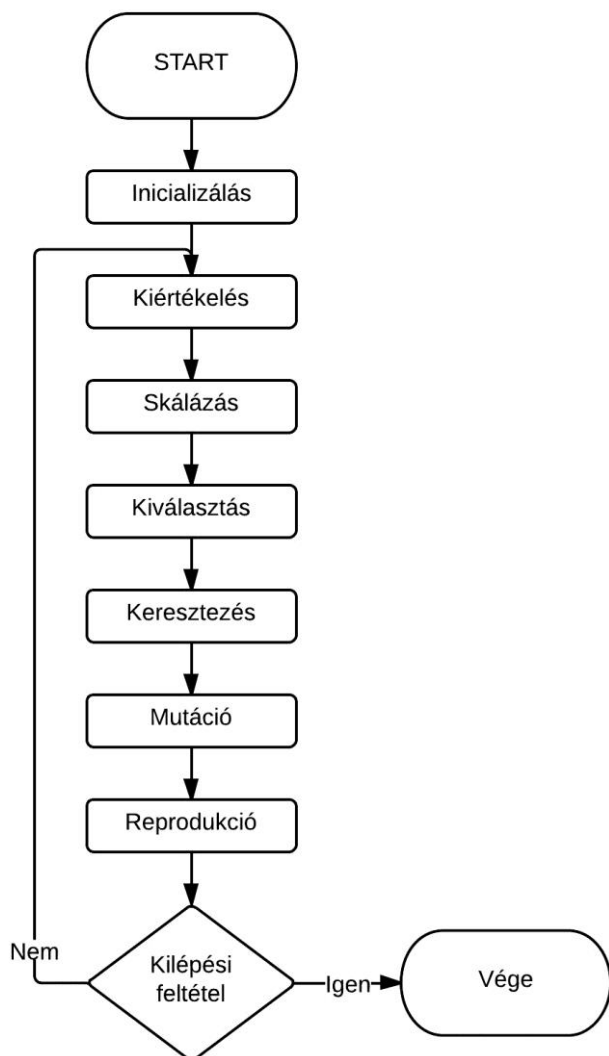
TÁRSAI (1998) szerint problémát jelent, hogy ez a módszer előnyben részesíti azokat a kromoszómákat, melyekben hozzávetőleg egyenlő a 0 és az 1 értékű gének száma, és elnyomja azokat, amelyek az egyikből keveset tartalmaznak.

4.1.3.2. Kiértékelés

A kanonikus genetikus algoritmusban a célfüggvény hozzárendelése a kromoszóma keletkezésétől függetlenül történik, azaz előbb létrehoz egy generációnyi egyedet, és aztán kiértékeli az egyedeket. Egyes műveletek elvégzéséhez ez hátrányos lehet.

4.1.3.3. Kiválasztás

A kiválasztás művelete biztosítja, hogy a nagyobb fitnessű egyedek nagyobb eséllyel jutnak tovább a következő műveletekre. A kiválasztási módszernél emellett figyelembe kell venni azt is, hogy ne forduljon elő, hogy egy egyed teljesen dominánssá válik,



4-2. ábra. Kanonikus genetikus algoritmus folyamatábrája

mert a változatosság fenntartása biztosítja, hogy az algoritmus nem ragad lokális optimumba.

A kanonikus genetikus algoritmus esetén a kiválasztás véletlent használó művelet, az egyes egyedek kiválasztásának valószínűsége arányos a fitnessükkel. (fitnessarányos vagy ruletkerék kiválasztás) (ÁLMOS ET AL., 2002)

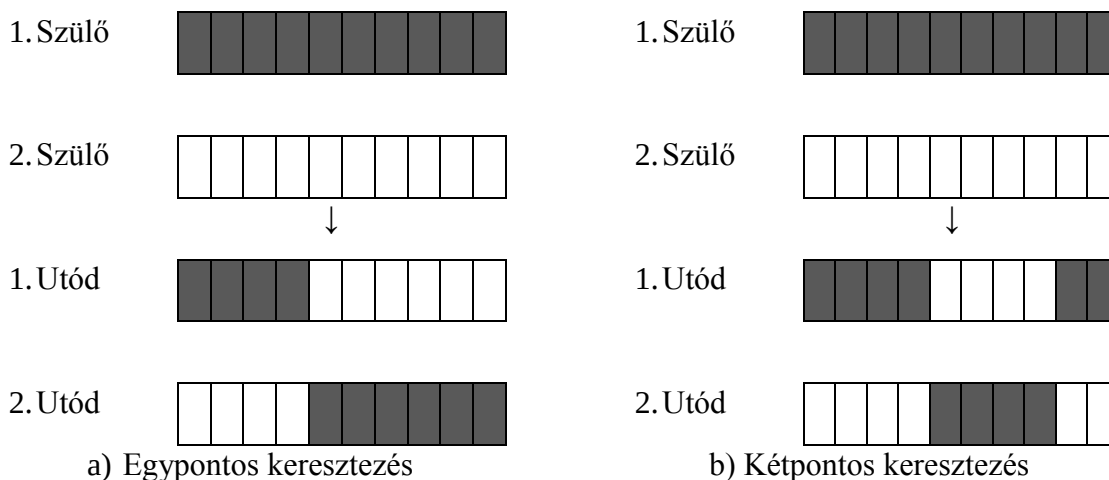
Ez a kiválasztási mód lehetőséget ad arra, hogy egy egyed többször is szerepeljen a következő műveletek során, így a skálázás megválasztásán múlik, hogy el lehet-e kerülni a domináns egyed kialakulását, és a korai konvergenciát. A kiválasztott elemek halmazát köztes generációnak nevezzük.

4.1.3.4. Genetikus műveletek

Alapvetően a keresztezést és a mutációt jelentik, de egyes források ide sorolják a kiválasztást is. Az evolúció modellezésében ez a két művelet biztosítja, hogy a populáció megváltozzon, de a változás irányát nem befolyásolja, azaz a fejlődést a kiválasztás biztosítja. A genetikus műveletek a köztes generáció egyedeit módosítják.

4.1.3.4.1 Keresztezés

A keresztezést a valós kromoszómák rekombinációjának mintájára hozták létre. Lényege, hogy két kromoszóma közt egyes gének cseréjét valósítja meg. Kanonikus genetikus algoritmus esetén ez általában egy pontos keresztezést jelent két szülő közt, ami úgy valósul meg, hogy a kromoszómákat egy génnél kettévágjuk és a levágott részt felcseréljük (4-3. ábra, a) Ezen kívül esetenként alkalmaznak több pontos keresztezést is. Páros számú keresztezési pont esetén a kromoszóma eleje és vége nagy valószínűséggel marad együtt, így a páratlan számú keresztezések kedvezőbbek. (4-3. ábra, b) (ÁLMOS ET AL., 2002)



4-3. ábra. Keresztezés

A keresztezés működési mechanizmusa, hogy a jobb kromoszómarészleteket képes összegyűjteni egy egyedbe. Ez alapján a keresztezés hatékonyságát elveszti, ha a gének számával összemérhető a keresztezések száma, mert ilyenkor csak a két szülő keverékét állítja elő. (ÁLMOS ET AL., 2002)

A GA ciklusában a két szülőre a keresztezés művelete egy adott valószínűséggel kerül alkalmazásra. Ha nincs keresztezés, akkor az utódok a szülőkkel egyenlők. A keresztezési helyek kiválasztása is véletlenszerűen történik.

4.1.3.4.2 Mutáció

A kanonikus genetikusan algoritmus a mutációt minden egyes kromoszóma minden egyes génjére függetlenül alkalmazza egy bizonyos valószínűséggel. A mutáció fő feladata a változatosság fenntartása. (ÁLMOS ET AL., 2002) A mutáció esetén az adott bit az ellenkező bitre változik.

4.1.3.5. Reprodukció

A GA-ban a reprodukció a köztes és szülőgenerációból valamilyen logika mentén létrehozza az új generációt, ami a következő ciklusban a szülők helyére lép.

A kanonikus genetikusan algoritmus a legegyszerűbb megoldást alkalmazza: az utódok alkotják az új generációt. Esetenként ez kiegészülhet a legjobb szülőegyed továbbvitelével, ami az egyik (esetleg a legrosszabb) utódegyed helyére kerül. Ezt elitizmusnak nevezzük. (ÁLMOS ET AL., 2002)

4.1.3.6. Kilépési feltétel

Az iterációs ciklus leállítását több tényező szabályozhatja: (ÁLMOS ET AL., 2002)

- Maximális generációs szám elérése. A végtelen ciklusok elkerülése érdekében a GA mindig tartalmazza ezt a feltételt.
- Populáció jellemzőinek vizsgálata. Például a legjobb és az átlagos fitness aránya, vagy a fitness szórása.
- Legjobb egyed fitnessze.

4.2. Saját genetikus algoritmusom felépítése

A következőkben leírom az általam készített genetikus algoritmus elemeit. A programba több választási lehetőséget is beépítettem, hogy a különböző módszerek hatékonyságát megvizsgáljam.

4.2.1. REPRESENTÁCIÓ

A genetikus algoritmus alapvető feladata a CT-val való együttműködés. Ennek tükrében észszerű megoldás, hogy bináris reprezentáció helyett a gének valós értékűek legyenek, hiszen a CT bemeneti paraméterei is zömmel valós számok.

Hogy a GA ugyanolyan rugalmasan kezelje a problémákat, mint maga a CT, ezért minden egyes gén egy $[0,1]$ intervallumbeli valós szám. A keresési tér egy dimenziójához így elegendő egy gént rendelni. A keresési tér határait külön tömb tárolja, és a GA csak a kiértékelés erejéig számítja ki a változók valódi értékét. (4.2)

$$\mathbf{XR}_i = \mathbf{XLIM}_{a,i} + (\mathbf{XLIM}_{f,i} - \mathbf{XLIM}_{a,i}) \cdot \mathbf{X}_i \quad (4.2)$$

Ennek a reprezentációnak egyik előnye, hogy orvosolja a FRIESE ÉS TÁRSAI (1998) által vázolt felbontás-problémát. Az általuk használt bináris reprezentáció a keresési tér egy dimenzióját 15 biten ábrázolta. Ezzel szemben a GA valós (kétszeres pontosságú, double precision) génjei 64 biten ábrázolnak egy dimenziót.

Az, hogy minden gén egyformán $[0,1]$ intervallumbeli valós szám, azzal az előnnyel jár, hogy a keresési térrel kapcsolatos információkra csak a kiértékelés során van szükség, így a GA többi lépésében a problémától függetlenül végezhetők a műveletek.

4.2.2. FITNESZ

A GA célfüggvény CT- val való együttműködés során valamilyen hatásfok jellegű mennyiség, de hogy robosztusabb eszköz legyen az algoritmus, nem lehet azzal a megkötéssel élni, hogy a célfüggvény értékek $[0,1]$ intervallumbeli számok. Azaz olyan skálázás szükséges, ami alkalmas tetszőleges valós értékű célfüggvényekből pozitív valós fitneszt előállítani.

Az algoritmus többféle skálázással is futtatható, de nem mind alkalmas tetszőleges célfüggvény értékek kezelésére.

4.2.2.1. Lineáris skálázás

Az általam is alkalmazott lineáris skálázást a (4.3) egyenlet írja le. (ÁLMOS ET AL., 2002)

$$\overline{ff} = a \cdot \overline{f} + b \quad (4.3)$$

Ahol a és b konstansokat az alább leírt elvek mentén lehet meghatározni.

1. A lineáris skálázás alapgondolata, hogy az adott populációra a fitness átlaga egyezzen meg a célfüggvény átlagával. Ez azt eredményezi, hogy fitnessarányos kiválasztás esetén az átlagos egyedek átlagosan egyszer kerülnek kiválasztásra.
2. A konstansok meghatározásához szükséges másik megkötés többféle lehet.
 - a. Alapvetően a legjobb egyed várható kiválasztási számát korlátozza, azaz a legjobb fitness az átlagos fitness $c = 1,2..2$ -szerese, ahol c a legjobb egyed kiválasztásának várható értéke.
 - b. Ha az előbbi módszert alkalmazva negatív fitnessok is létrejönnek, akkor a második megkötés helyett azt alkalmazzuk, hogy a legrosszabb egyed fitnessa legyen nulla.

Az első feltételt (4.3) egyenlet átlagát képezve lehet kifejezni:

$$\begin{aligned} \overline{\overline{ff}} &= \overline{a \cdot \overline{f} + b} \\ \overline{\overline{ff}} &= a \cdot \overline{\overline{f}} + b \end{aligned} \quad (4.4)$$

Az első feltevést alkalmazva (4.4)-re:

$$\overline{\overline{ff}} = \overline{\overline{f}} \quad (4.5)$$

$$b = \overline{\overline{f}} \cdot (1 - a) \quad (4.6)$$

A (4.6) egyenlet minden esetben igaz lesz. Attól függően, hogy minimum vagy maximumkeresésről van szó, és hogy a második feltételek közül melyiket kell alkalmazni, más lesz a értéke.

4.2.2.1.1 Maximumkeresés

Maximumkeresés esetén a legjobb egyed a legnagyobb célfüggvényű. Ennek fitnessa a 2./a feltétel értelmében az átlagos célfüggvény c -szerese. A (4.3) és (4.6) egyenleteket a legjobb egyedre alkalmazva a (4.7) egyenlet adódik.

$$ff_{\max} = c \cdot \bar{f} = a \cdot f_{\max} + (1-a) \cdot \bar{f}$$

Azaz

$$a = \frac{(c-1) \cdot \bar{f}}{f_{\max} - \bar{f}}$$
(4.6)

Ha ezzel a módszerrel létrejön negatív fitness érték, akkor a 2./b feltételt kell alkalmazni, amit a (4.7) egyenlet szemléltet.

$$ff_{\min} = 0 = a \cdot f_{\min} + (1-a) \cdot \bar{f}$$

Azaz

$$a = \frac{\bar{f}}{\bar{f} - f_{\min}}$$
(4.7)

4.2.2.1.2 Minimumkeresés

Minimumkeresésnél a legkisebb célfüggvényű a legjobb egyed. A 2./a feltétel alkalmazását a (4.8) egyenlet mutatja.

$$ff_{\max} = c \cdot \bar{f} = a \cdot f_{\min} + (1-a) \cdot \bar{f}$$

Azaz

$$a = \frac{(c-1) \cdot \bar{f}}{f_{\min} - \bar{f}}$$
(4.8)

Negatív fitness esetén e helyett a 2./b feltételt alkalmazva következik a (4.9) összefüggés.

$$ff_{\min} = 0 = a \cdot f_{\max} + (1-a) \cdot \bar{f}$$

Azaz

$$a = \frac{\bar{f}}{\bar{f} - f_{\max}}$$
(4.9)

A lineáris skálázás előnye a működési logikája, de súlyos hátránya, hogy nem képes minden célfüggvény halmazt kezelni. Ahol a célfüggvények átlaga negatív ott viselkedése nem teljesíti a skálázással szemben támasztott követelményeket. Ez alapján hatásfokmaximum-keresésre alkalmas, de szélesebb körben aggályos az alkalmazhatósága.

4.2.2.2. Szigma csonkolás

A szigma csonkolás logikája, hogy az aktuális populációból átlagosnál kedvezőtlenebb egyedek egy részének ad csak pozitív fitnesszt, a többit „életképtelennek” ítéli, azaz 0 fitnesszt ad. A határ a szórás (σ) függvénye a (4.10) (4.11) egyenletek szerint. A módszer lényege, hogy nagy szórású populációkban értékeli a változatosságot, de ha az egyedek közti különbségek kicsik, akkor az átlagon aluliakat nem tolerálja.

Ennek a felépítése is függ attól, hogy minimum- vagy maximumkeresés a feladat.

4.2.2.2.1 Maximumkeresés

$$ff = \begin{cases} f - (\bar{f} - c \cdot \sigma) & \text{ha ez pozitív} \\ 0 & \text{egyébként} \end{cases} \quad (4.10)$$

Ahol $c = 1..3$ azt a toleranciasávot jelöli ki, ahol még átlag alatti célfüggvényekhez is pozitív fitnesszt rendel a skálázás.

4.2.2.2.2 Minimumkeresés

$$ff = \begin{cases} -f + (\bar{f} + c \cdot \sigma) & \text{ha ez pozitív} \\ 0 & \text{egyébként} \end{cases} \quad (4.11)$$

Ahol $c = 1..3$ azt a toleranciasávot jelöli ki, ahol még átlag feletti célfüggvényekhez is pozitív fitnesszt rendel a skálázás.

A szigma csonkolás csak bizonyos célfüggvény halmazokra alkalmazható, így ez a módszer szintén korlátolt.

4.2.2.3. Hatvány alapú skálázás

A hatvány alapú skálázást az előbbiekkal szemben úgy alakítottam ki, hogy alkalmas legyen bármilyen célfüggvény halmaz kezelésére. A fitnessz itt a legkedvezőtlenebbtől való eltérés pozitív kitevőjű hatványa. (4.12) (4.13)

4.2.2.3.1 Maximumkeresés

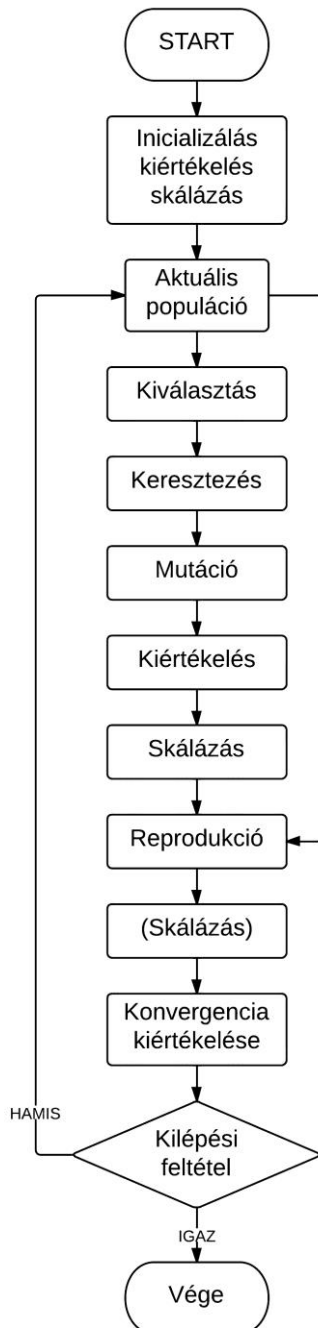
$$ff = (f - f_{\min})^\alpha \quad (4.12)$$

4.2.2.3.2 Minimumkeresés

$$ff = (f_{\max} - f)^\alpha \quad (4.13)$$

Ahol α a skálázási kitevő. ($\alpha > 0$)

- Ha $0 < \alpha < 1$, akkor a skálázás a változatosságot helyezi előnybe.
- Ha $1 < \alpha$, akkor a skálázás a különbségeket emeli ki.



4-4. ábra. Genetikus algoritmus folyamatábrája

4.2.3. GENETIKUS ALGORITMUS LEFUTÁSA

Az én genetikus algoritmusom lefutása némiképp eltér a kanonikus genetikus algoritmusétól. (4-4. ábra.) A különbségek ez egyes műveletek indokolják.

4.2.3.1. Inicializálás

A számos random számot használó művelet miatt, fontos, hogy a random számok valóban véletlenszerűek legyenek. A Fortran beépített DRAND függvényét használva ez úgy érhető el, hogy a futtatás elején a rendszeridő alapján indul a random szám generátor.

A DRAND függvény értékészlete a $[0,1]$ intervallum, így a kezdeti populáció minden kromoszómájának minden génjének közvetlenül lehet a DRAND által visszaadott értéket adni.

4.2.3.2. Kiértékelés

A kiértékelés mindig közvetlenül a kromoszóma létrejötte után történik.

A kiértékeléshez a GA az adott kromoszómát visszaalakítja a keresési tér egy pontjává a (4.2) egyenlet szerint. Az így kapott pont már alkalmas a célfüggvénybe való behelyettesítésre, vagy egy programnak való átadásra, ami behelyettesíti, mint ahogy azt a CT is teszi.

A CT-val való együttműködés során a futási idő jelentős részét a kiértékelés teszi ki, így egy *elő-kiértékelést* alkalmazok, ami a nyilvánvalóan életképtelen megoldásokat szűri ki. Ez a megközelítés jelentősen gyorsíthatja az algoritmust, de növeli a problémával kapcsolatos szükséges előismeretek mennyiségét.

Szintén a CT teszi szükségessé az utókiértékelést, ugyanis annak ellenére, hogy egy megoldás megvalósíthatatlan, még rendelhet hozzá hatásfokot, így a visszakapott warningok számának függvényében érdemes szankcionálni az ilyen megoldásokat.

4.2.3.3. Kiválasztás

A kiválasztásnál egyszerű fitnessz arányos (rulettkerék) módszert alkalmaztam.

Ezt úgy lehet leírni, mintha a fitnessz értékeknek megfelelő hosszúságú szakaszokat mérnénk fel egymás után egy számegyenesre. Ezután az összes szakasz által a számegyenesből lefedett tartományon generálunk egy random számot, és amelyik szakaszra esik, az annak megfelelő egyed lesz a kiválasztott.

4.2.3.4. Keresztezés

A genetikus algoritmusomban két féle keresztezést alkalmaztam:

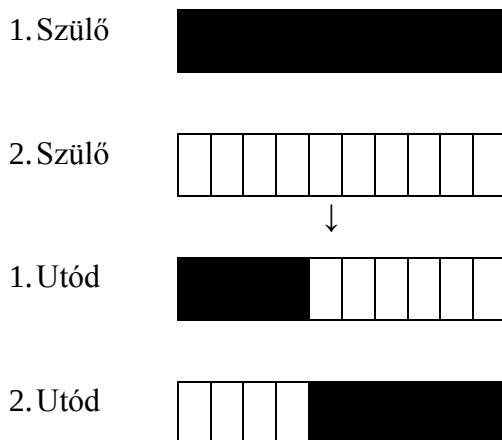
- Egyszerű keresztezést
- Aritmetikai keresztezést (XI ET AL., 2013)

Az egyszerű keresztezés a (4-3. ábra) szerint mód, azaz egyszerűen a gének felcserélését jelenti. Mivel a kanonikus genetikus algoritmussal ellentétben a keresési tér egy dimenzióját én egy génnel kódolom, ezért a keresztezés hatása különbözik. Ezt hivatott orvosolni az aritmetikai keresztezés, ami a keresztezési pont után a gének lineáris kombinációját veszi a (4.14) egyenlet szerint, ahol R $[0,1]$ intervallumbeli random szám.

$$\begin{aligned} \mathbf{X}_{1,i}^{\times} &= R \cdot \mathbf{X}_{1,i} + (1 - R) \cdot \mathbf{X}_{2,i} \\ \mathbf{X}_{2,i}^{\times} &= (1 - R) \cdot \mathbf{X}_{1,i} + R \cdot \mathbf{X}_{2,i} \end{aligned} \quad (4.14)$$

Az aritmetikai keresztezést szimmetrikusan történik, hogy az utódok mindkét szülőt egyforma mértékben képviseljék. Az aritmetikai keresztezést és az egyszerű keresztezés közti különbséget a 4-5. ábra szemlélteti.

A keresztezési pontok száma tetszőleges lehet (amennyiben értelmes) és kiválasztásuk véletlenszerűen történik.



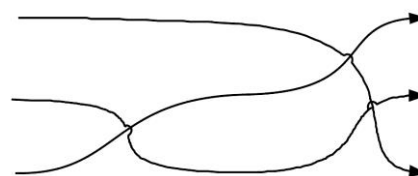
b) Egyszerű keresztezés



b) Aritmetikai keresztezés

4-5. ábra. Egyszerű és aritmetikus keresztezés

Felmerült az algoritmus készítése közben, hogy a valószínűségi elvonatkoztatva lehetséges lenne kettőnél több „szülő” alkalmazása is. (4-6. ábra) Ezekre szintén az egyszerű és az aritmetikai keresztezés alkalmazható, de az adatcsere itt már nem feltétlenül kölcsönös és így a szimmetria sem áll fenn. Egy egyed a keresztezési alatt álló géncsoportokon egy másik egyedtől kaphat értéket és akár egy harmadiknak adhat tovább. Az aritmetikai keresztezés itt az adott sorszámu egyed eredeti génjei és a vele összerendelt egyed génjei közt zajlik. A keresztezési pontokon az egyedek véletlen sorrendjének frissítése történik, tehát a páros sorszámu keresztezési pontokon nem feltétlenül áll helyre az eredeti sorrend.



4-6. ábra. Több szülős keresztezés

A keresztezés egy bizonyos valószínűséggel következik be a szülőkön. Lehet egyszerű, aritmetikai vagy a kettő együtt. Utóbbi esetben az egyszerű módszer választásának valószínűsége adható meg.

4.2.3.5. Mutáció

A mutáció egyetlen génre van hatással. Erre a műveletre is két módszert dolgoztam ki. Első az egyszerű mutáció, amikor a gén új random értéket kap a teljes $[0,1]$ intervallumról. Második a korlátozott mutáció, amikor az eredeti érték egy ε sugarú környezetében kaphat új értéket a gén véletlenszerűen választva. Utóbbi létrehozhat a keresési téren kívüli egyedeket. Ha a keresési tér határai szigorú feltételeket jelentenek, akkor ezt az elő- és utókiértékelés során az ilyen

egyedeket szankcionálni kell. Ha a keresési tér határai nem jelentenek valós korlátokat, akkor ezek az egyedek is elfogadhatók.

A mutáció adott valószínűséggel következik be. Alkalmazható egyszerű és korlátos mutáció vagy a kettő együtt is. Utóbbi esetben az egyszerű mutáció választásának valószínűségével szabható meg az arányuk.

Valós reprezentáció esetén a mutáció jelentősége nagy, ha mellette csak egyszerű keresztezés van, mivel ebben az esetben csak a mutáció képes új géneket létrehozni. Aritmetikai reprezentáció alkalmazásával a jelentősége abban rejlik, hogy a keresztezés „átlagoló” jellegét ellensúlyozva fenntartsa a változatosságot.

A korlátos mutáció célja, hogy az optimum megközelítése után a kis elmozdulások nagyobb valószínűségűek legyenek, így jobban megközelíthető a globális optimum. Ugyanakkor az ε helyes megválasztásával alkalmazható a lokális szélsőértékek elkerülésére is.

4.2.3.6. *Reprodukció*

A reprodukcióra is több módszer alkalmazható, és van néhány opció, ami a módszertől függetlenül befolyásolhatja az új generációt.

Az alkalmazott módszerek:

- Egyszerű reprodukció
- *Átlapolt* reprodukció
 - A legjobb egyedekkel
 - Random egyedekkel

Az opciók:

- Elitizmus
 - Legrosszabb egyedet helyettesítve
 - Random egyedet helyettesítve
- Random *bevándorló*
 - Legrosszabb egyedet helyettesítve
 - Random egyedet helyettesítve

Az egyszerű reprodukció esetén az utódok alkotják a következő populációt. Az átlapolt keresztezés esetén az új generációt részben a szülők, részben az utódok alkotják. Ekkor szülőpopulációból továbbvitt egyedek részaránya adott. A két részpopulációt alkothatják a legjobb egyedek, vagy random kiválasztott egyedek.

Az elitizmus a szülőgeneráció legjobb egyedének továbbvitelét jelenti. A random bevándorló egy teljesen újonnan generált egyed új generációba illesztését jelenti. Mindkettő lehetséges a legrosszabb vagy egy véletlenszerű egyed helyettesítésével. Az elitizmus célja a visszalépések elkerülése. A random bevándorló a változatosságot tartja fenn.

Fontos tényező, hogy a populáció megváltoztatása esetén mindig szükséges újraszámolni, és a random bevándorló alkalmazásával egy kiértékelés is szükséges.

4.2.3.7. Konvergencia kiértékelése

A GA megadott generációs szám elérésekor mindenképp kilép a ciklusból, ezt a határt úgy érdemes megválasztani, hogy mindenképp bekövetkezzen a konvergencia. Sok idő spórolható, ha a konvergencia bekövetkeztekor kilép.

A konvergenciát több tényező vizsgálatával igyekszem leírni, ezek a következők:

1. Minimális/maximális célfüggvény érték elérése
2. Átlagos célfüggvény értékek mozgóátlagának megváltozása
3. *Súlypont* elmozdulása
4. *Súlypont* elmozdulások mozgó átlaga
5. Legjobb egyed célfüggvény értékének megváltozása

A minimális/maximális célfüggvény érték elérése olyan bonyolult problémák esetén lehet jó választás, ahol egy elfogadható megoldás megkeresése a cél véges számítási időn belül. Mivel a CT elég gyorsan eredményt ad, ezért ilyen igény nem merül fel.

A súlypontot úgy definiáltam, hogy az egyedeket tömegpontoknak tekintetem, amiknek tömege a fitness értékük. A súlypont a pontthalmaz súlypontja a keresési térben.

Az átlagos célfüggvény értékek mozgóátlagának megváltozása akkor mutatná helyesen a konvergenciát, ha az optimum megtalálásakor az összes egyed az optimum közelében helyezkedne el. Ha nagy az egyszerű mutáció valószínűsége, vagy van random bevándorló, akkor a teljes populáció nem feltétlenül közeledik az optimumhoz. Ennek kiküszöbölésére dolgoztam ki a súlypontot használó módszereket, mivel a súlypont kevésbé érzékeny a populáció gyengébbik felére, így lehetséges, hogy alkalmasabb a konvergencia leírására.

Az 5. feltétel esetén a GA legjobb egyed célfüggvény értékének minimumát és maximumát vizsgálja adott számú generációra visszatekintve, a kettő különbségére adott egy kritérium. Ha a kritériumérték nulla, akkor a kilépés feltétele, hogy adott számú generációig a legjobb egyed ne változzon.

5. GENETIKUS ALGORITMUS MŰKÖDÉSE

5.1. Tesztfüggvények

A tesztfüggvények olyan függvények, melyeknek ismertek a szélsőérték helyei és értékei és megfelelő bonyolultságúak ahhoz, hogy az optimum kereső eljárás minőségét meg lehessen határozni velük.

Általánosan az alábbi nehézséget jelentő tulajdonságokat használják az eljárással szemben:

- Lokális szélsőértékek
- Szakadások
- Konstans értékű tartományok

A tesztfüggvényekkel szemben támasztott további kritérium, hogy tetszőleges dimenziójú keresési téren értelmezettek legyenek. Az alábbiakban bemutatok néhány tesztfüggvényt, majd a tesztek eredményeit ismertetem.

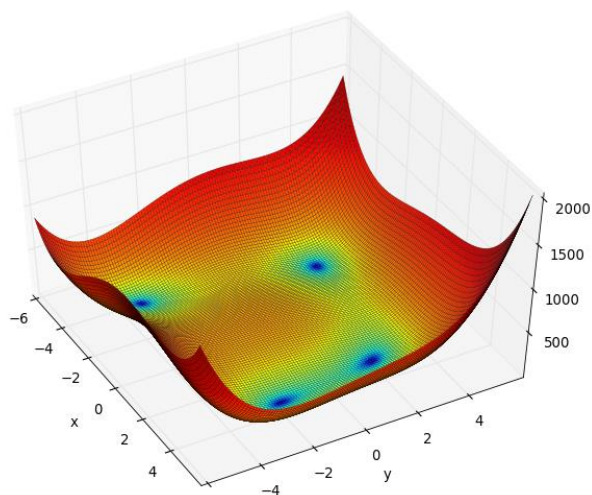
5.1.1. HIMMELBLAU-FÜGGVÉNY

$$\begin{aligned} \mathbf{xR}_i &\in [-6;6] \\ f(\mathbf{xR}_1, \mathbf{xR}_2) &= (\mathbf{xR}_1^2 + \mathbf{xR}_2 - 11)^2 + (\mathbf{xR}_1 + \mathbf{xR}_2^2 - 7)^2 \end{aligned} \quad (5.1)$$

Ez a függvény csak két dimenzióra értelmezett a $[-6,6]^2$ tartományon.(5.1) (DEAP, 2014)

Négy globális minimuma van, ahol a függvényérték 0, ahogy azt az 5-1. ábra mutatja.

- (3;2)
- (-2,805118; 3,131312)
- (-3,779310;-3,283186)
- (3,584428;-1,848126)



5-1. ábra. Himmelblau-függvény (DEAP, 2014)

5.1.2. UGRÁSFÜGGVÉNY

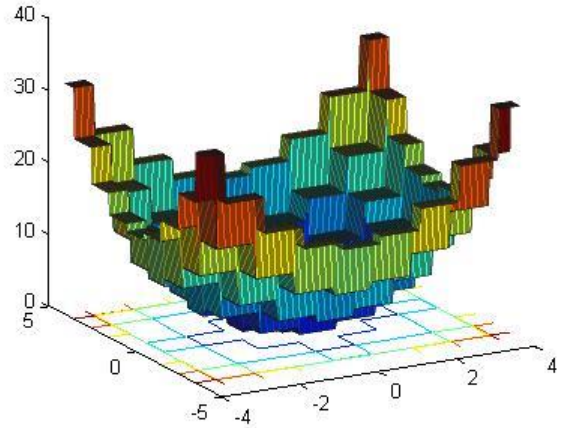
$$\begin{aligned} \mathbf{XR}_i &\in [-4;4] \\ f(\mathbf{XR}) &= \sum_{i=1}^{NSS} [\mathbf{XR}_i + 0,5]^2 \end{aligned} \quad (5.2)$$

Az ugrásfüggvény (ÁLMOS ET AL., 2002) már alkalmazható tetszőleges számú dimenzióra. Lényege, hogy számos szakadást tartalmaz. A vizsgálatot $[-4,4]^{NSS}$ tartományon végeztem. (5.2)

Ahol $\lfloor \cdot \rfloor$ az egészrészt jelöli.

Az ugrásfüggvénynek globális minimuma van a $[-0,5;0,5]^{NSS}$ intervallumon.

Az ugrásfüggvény felépítését két dimenzióra az 5-2. ábra szemlélteti.

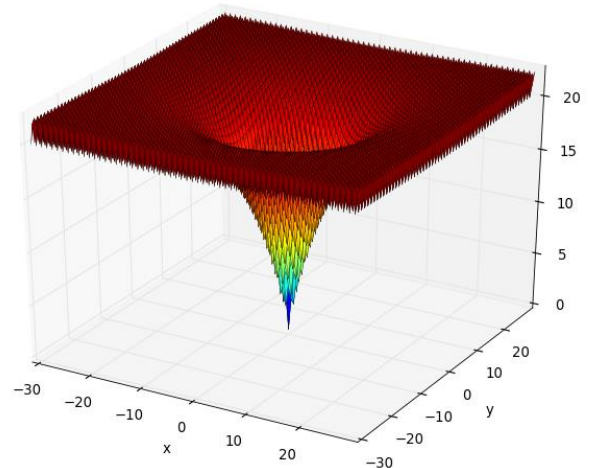


5-2. ábra. Ugrásfüggvény két dimenzióban

5.1.3. ACKLEY-FÜGGVÉNY

$$\begin{aligned} \mathbf{XR}_i &\in [-30;30] \\ f(\mathbf{XR}) &= -20 \cdot e^{-0,2 \cdot \sqrt{\frac{1}{NSS} \sum_{i=1}^{NSS} \mathbf{XR}_i^2}} - e^{\frac{1}{NSS} \sum_{i=1}^{NSS} \cos(2\pi \cdot \mathbf{XR}_i)} + 20 + e \end{aligned} \quad (5.3)$$

Az Ackley-függvénynek (5.3) számos lokális szélsőértéke van és egy globális minimuma, ami megadja a lehetőségét, hogy az optimum kereső eljárás lokális szélsőértékben ragadjon. (GAFFNEY ET AL., 2010) Szintén tetszőleges dimenziójú keresési térben értelmezett. A vizsgálatot a $[-30;30]^{NSS}$ intervallumon végeztem, mivel a külső területeken a lokális szélsőérték helyek kialakításában már a koszinusz tag dominál, és így megnehezíti a minimum megtalálását.



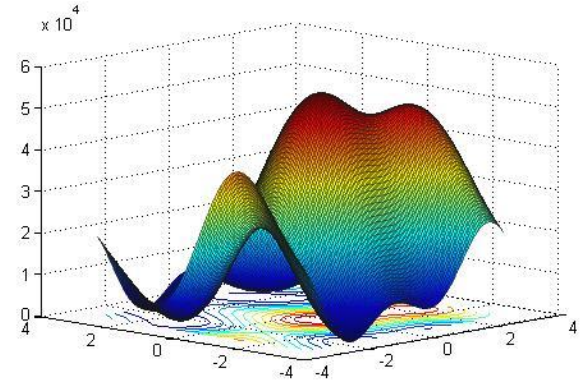
5-3. ábra. Ackley-függvény két dimenzióra (DEAP, 2014)

Ahogy az 5-3. ábrán látható, az Ackley-függvénynek globális minimuma van az origóban, ahol értéke 0.

5.1.4. FLETCHER-POWELL-FÜGGVÉNY

$$\begin{aligned}
 \mathbf{XR}_i &\in [-\pi; \pi] \\
 \mathbf{A}_i &= \sum_{j=1}^{NSS} (\mathbf{a}_{ij} \cdot \sin(\alpha_j) + \mathbf{b}_{ij} \cdot \cos(\alpha_j)) \\
 \mathbf{B}_i &= \sum_{j=1}^{NSS} (\mathbf{a}_{ij} \cdot \sin(\mathbf{XR}_j) + \mathbf{b}_{ij} \cdot \cos(\mathbf{XR}_j)) \\
 f(\mathbf{XR}) &= \sum_{i=1}^{NSS} (\mathbf{A}_i - \mathbf{B}_i)^2
 \end{aligned} \tag{5.3}$$

Az eddigiekkel ellentétben a Fletcher-Powell-függvény (FLETCHER AND POWELL, 1963) nem mutat szimmetriát. Számos lokális szélsőértéke van a keresési térben véletlenszerűen elosztva, ami egyes eljárásokkal szemben kihívást jelent. Az $\mathbf{a}, \mathbf{b}, \alpha$ tömbök véletlen értékeket tartalmaznak: $\mathbf{a}, \mathbf{b}$ elemei a $[-100, 100]$ intervallumból kerülnek ki, α elemei pedig a $[-\pi; \pi]$ intervallumból. A függvénynek nulla értékű minimuma van az $\mathbf{XR} = \alpha$ helyen. A függvénynek összesen legfeljebb 2^{NSS} szélsőértéke van, mind $[-\pi; \pi]$ intervallumon. A kétdimenziós Fletcher-Powell-függvényre egy példát a 5-4. ábrán mutat.



5-4. ábra. Fletcher-Powell-függvény két dimenzióra

5.2. Algoritmus működése tesztfüggvényeken

A tesztelés során a célom az volt, hogy bemutassam, hogy létezik olyan paraméter halmaz, ami mellett a GA minőségileg jobb megoldást ad, mint egy alapesetnek tekintett egyszerű algoritmus. Minden paraméterhalmazt három futás átlagával jellemeztem. Mivel a GA heurisztikus módszer, így ez a 3 futás csak minőségi, összehasonlító jellegű kiértékelésre ad lehetőséget. A GA további vizsgálatára csak az Ackley-függvényen került sor.

Tehát a vizsgálatok során alkalmaztam egy egyszerű paraméterhalmazt, amit az 5-1. táblázat tartalmaz.

5-1. táblázat. A GA egyszerű felparaméterezése

Paraméter neve	Paraméter értéke
Populáció mérete	30
Skálázás	Lineáris
Legjobb egyed kiválasztásának valószínűsége	1,6 (ÁLMOS ET AL., 2002)
Keresztezés típusa	Egyszerű
Keresztezésben résztvevő szülők száma	2
Keresztezések száma	1
Keresztezés valószínűsége	0,4 (XI ET AL., 2013)
Mutáció típusa	Egyszerű
Mutáció valószínűsége	0,2 (XI ET AL., 2013)
Reprodukció típusa	Egyszerű
Elitizmus	Van, legrosszabb egyedet helyettesítve

Majd kerestem olyan paramétereket, amik mellett a GA átlagosan jobb eredményeket ad, illetve gyorsabban konvergál.

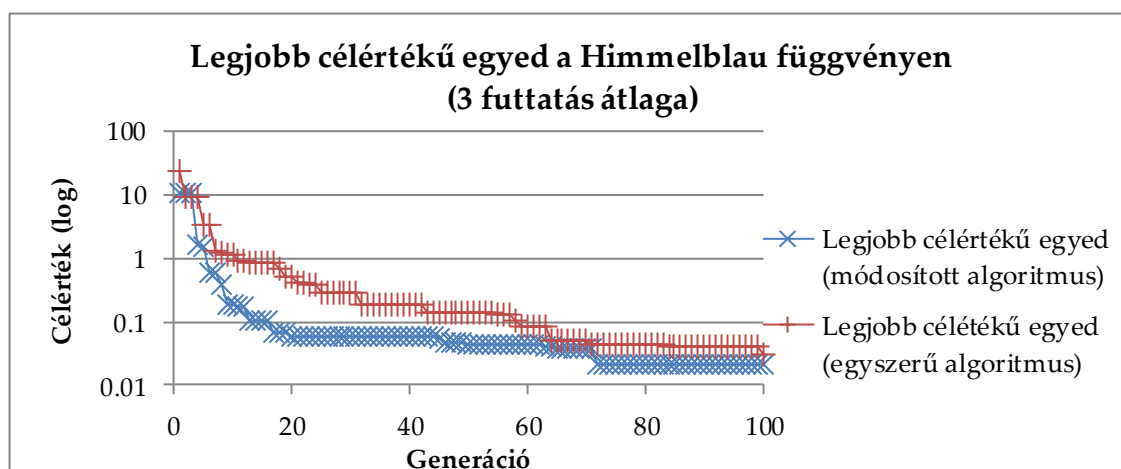
A skálázási eljárásokkal szemben támasztott monotonitási követelményt az összes módszer kielégíti, így a minimum és maximumkereséseket jelen munkában nem vizsgáltam külön.

5.2.1. HIMMELBLAU-FÜGGVÉNY

Ezt a függvényt csak két dimenzióra értelmezett, így kevesebb lehetőséget ad. Jelentősége, hogy négy egyforma szélsőértéke van, így az algoritmus nem feltétlenül találja meg mind a négyet.

Az egyszerű paraméterekkel végzett futtatások során az egyik optimum közeli gén a futtatás végére mindig nagyobb számban van jelen. Ezt a változatosság növelésével lehet ellensúlyozni, azaz keresztezési és mutációs valószínűségek növelésével, a random bevándorló alkalmazásával, és a hatvány alapú skálázás –egy alatti kitevővel való– alkalmazásával az algoritmus minősége jelentősen javult. (1. Melléklet) Három futás átlagában a módosításokkal 17

generáció, az egyszerű paraméterekkel 59 generáció volt szükséges ahhoz, hogy a függvény-érték 0,1 alá csökkenjen. (5-5. ábra)



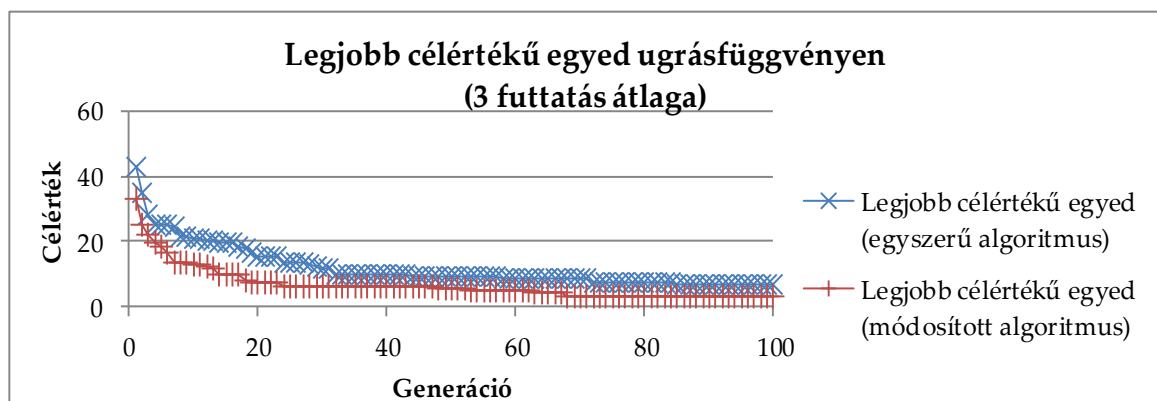
5-5. ábra. Genetikus algoritmus viselkedése Himmelblau-függvényen

5.2.2. UGRÁSFÜGGVÉNY

Az ugrásfüggvény már alkalmas több dimenzió kezelésére, ezért ennek megfelelően a tesztet 15 dimenziós keresési térben végeztem. Ennek a függvénynek az „erősségét” a szakadások jelentik, de mivel a genetikus algoritmus nem alkalmaz gradiens alapú módszereket, ezért elméletileg érzéketlenebb rá.

A viszonylag nagyszámú dimenzió miatt a globális optimumot sem az egyszerű sem a megfelelően módosított algoritmus nem találta meg 100 lépésen belül. Az egyszerű paraméterek esetén itt inkább az jelenti a problémát, hogy az optimális gének jelen voltak az egyedekben, de nem kerültek összerendezésre. Ezért megemeltem a keresztezési pontok számát, és a keresztezés valószínűségét. Továbbá egy feletti kitevőjű, hatvány alapú skálázást alkalmaztam a szigorúbb kiválasztásért, valamint a korlátos mutációt is alkalmaztam az egyes gének javítására. (2. Melléklet)

A módosítás a konvergenciát gyorsította és jobb célfüggvényt is eredményezett, ahogy az az 5-6. ábrán is látszik.



5-6. ábra. Genetikus algoritmus viselkedése ugrásfüggvényen

5.2.3. ACKLEY-FÜGGVÉNY

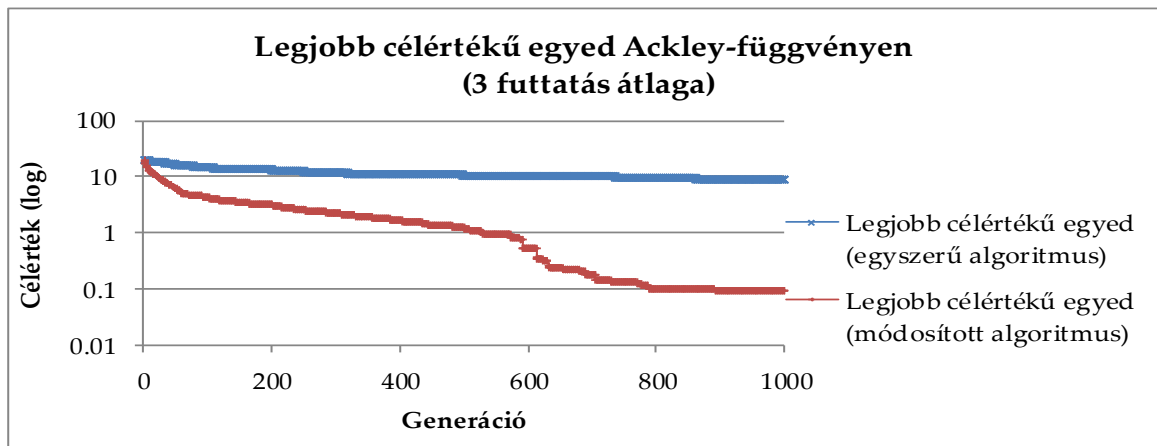
Az Ackley-függvénynek sok lokális szélsőértéke van, így a GA-ra tekintve komolyabb kihívást jelent, mint az ugrásfüggvény. Ezzel szintén 15 dimenzióra teszteltem a GA-t. A probléma bonyolultsága miatt a maximális generációs számot 1000-re állítottam.

Ebben az esetben az egyszerű paraméterekkel meglehetősen messze marad az algoritmus a lokális optimumtól. Úgy találtam, a hiba az, hogy a kedvező gének ki sem alakulnak. A módosítás hatásának jelentős részét a korlátozott mutáció fejtette ki. A helyes ε megválasztása fontos eleme ennek a problémának. A jobb egyedek nagyobb számban való megőrzésére átlapolított reprodukciót alkalmaztam a legjobb egyedek továbbvitelével. (3. Melléklet)

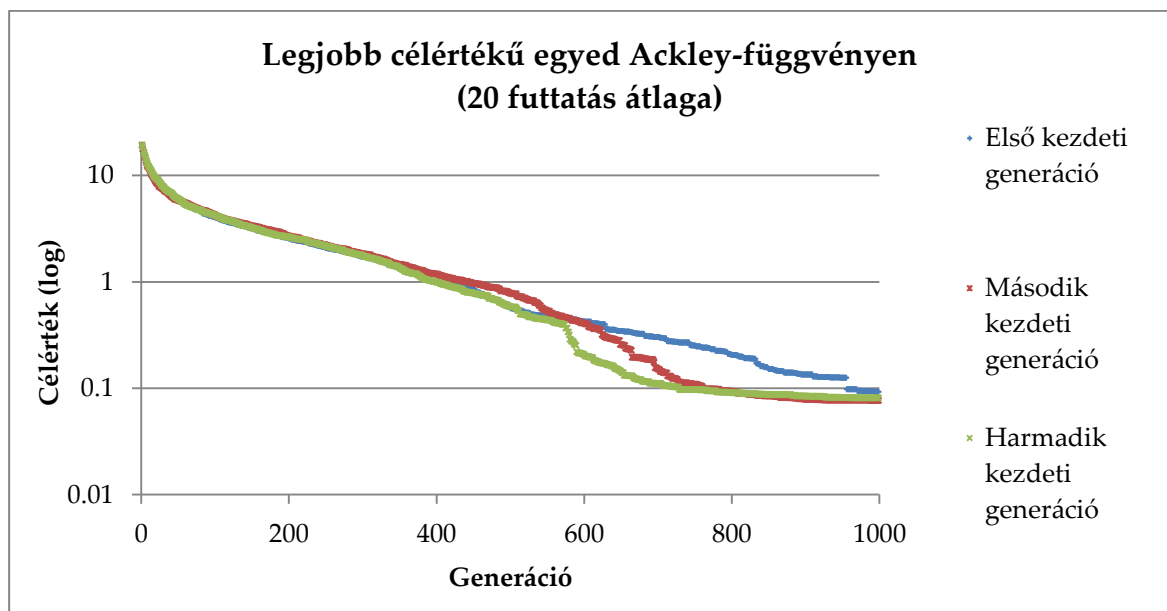
A paraméterek módosításával két nagyságrenddel közelebb került a legjobb egyed célfüggvény értéke az optimumhoz. (5-7. ábra)

Ugyanezen paraméterekkel tovább vizsgáltam a genetikus algoritmus teljesítményét. Három rögzített kezdeti populációról indítottam 20-20 futtatást. A vizsgálat eredményeit az 5-8. ábra és az 5-9. ábra szemlélteti. Előbbi diagramon az látható, hogy sem a kezdeti populáció sem az algoritmus által elvégzett véletlen műveletek nincsenek hatással a végeredményre. Utóbbi tény az 5-9. ábra megerősíti, ami a második kezdeti populációról indított futások átlagát és a szórással növelt átlagot, mint egyfajta felső korlátot ábrázolja. Ezen látszik, hogy a köztes generációkban az egyes futtatások közti eltérések nagyok, de a nagyjából 900. generációtól a különbségek lecsökkennek. Megjegyzendő, hogy a nagy kezdeti szórásokat a logaritmikus skála miatt nem látni.

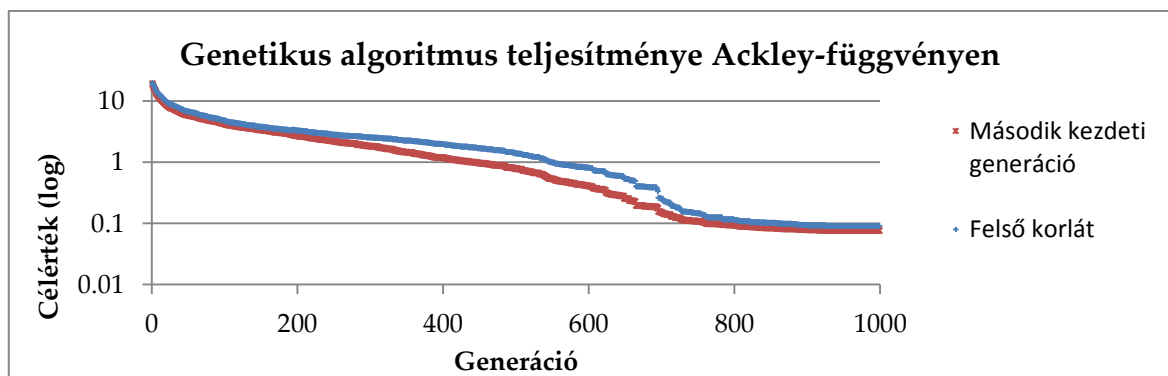
Az algoritmus által kiadott adatokon jól megfigyelhető, hogy minden futtatás legjobb egyede a globális optimum közvetlen környezetébe kerül.



5-7. ábra. Genetikus algoritmus viselkedése Ackley-függvényen



5-8. ábra. Részletes vizsgálat eredményei Ackley-függvényen
Különböző kezdeti generációk



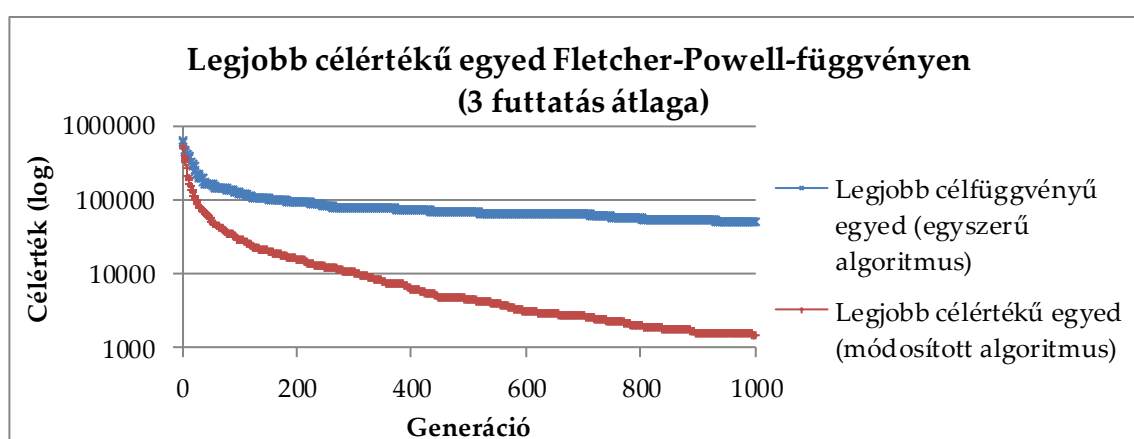
5-9. ábra. Ackley-függvényen a közös kezdeti populációról indított futtatások legjobb egyedének célfüggvény értékei, és annak egy felső korlátja.

5.2.4. FLETCHER-POWELL-FÜGGVÉNY

A Fletcher-Powell-függvényt szintén 15 dimenzióra alkalmaztam, a függvényhez szükséges együtthatókat véletlenszerűen generáltam, majd ezeket az együttható tömböket alkalmaztam minden futtatásnál. Ez a függvény számos lokális szélsőértékkel és aszimmetrikus felépítéssel teszi próbára a GA-t.

A paramétereket az Ackley-függvényénél tapasztaltakhoz hasonlóan alakultak. A korlátos mutáció és az aritmetikai keresztezés, valamint a keresztezési pontok számának növelése vezetett eredményre. Érdekes megjegyezni, hogy a Fletcher-Powell-függvényénél a szigma csonkolás jobb célfüggvény értékhez vezetett, mint a hatvány alapú skálázás. (4. Melléklet)

A Fletcher-Powell-függvényen a paraméterek megfelelő módosításával nagyjából két nagyságrenddel jobb célfüggvényt lehetett elérni (5-10. ábra).



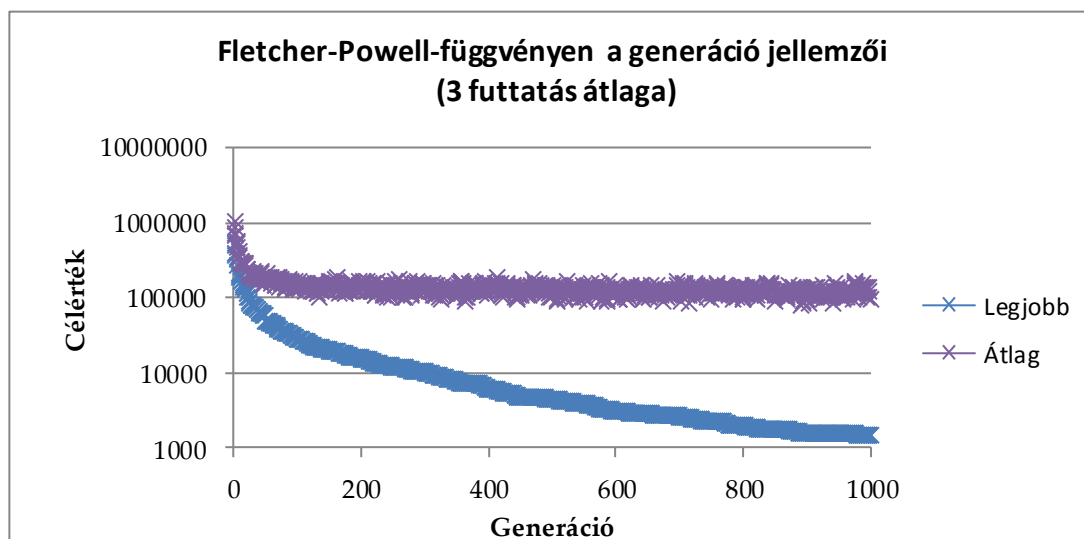
5-10. ábra. Genetikus algoritmus viselkedése Fletcher-Powell-függvényen

5.2.4.1. Konvergencia megállapítására alkalmas mennyiségek

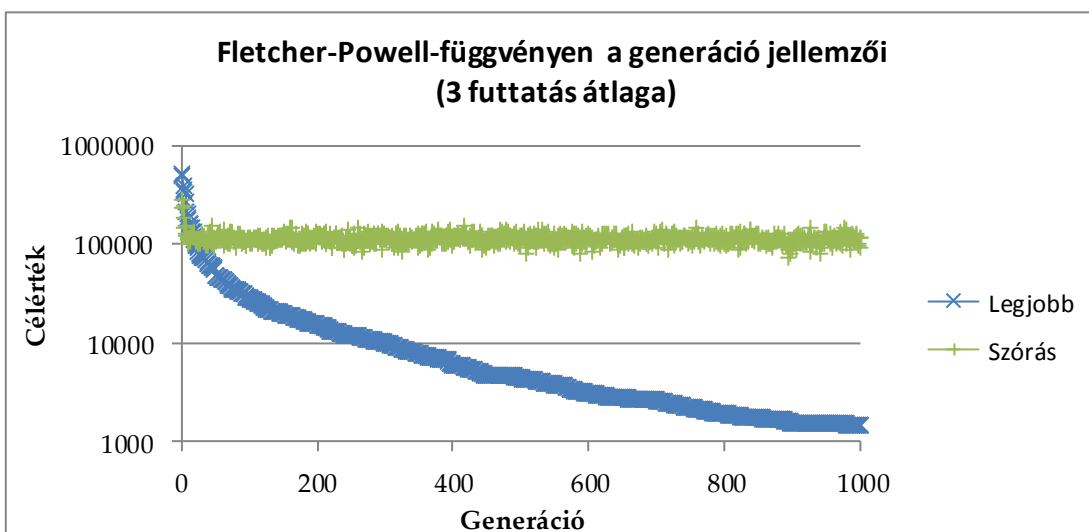
A Fletcher-Powell-függvény esetén vizsgáltam a konvergencia leírására alkalmasnak vélt mennyiségeket. Az eredmény gyakorlatilag az, hogy egyik remélt jellemző sem alkalmas a konvergencia leírására, mert értékük rendszertelenül ingadozik, és amíg a legjobb egyed folyamatosan javul, addig a jellemzők már korábban konvergálnak. Ez a jelenség annak köszönhető, hogy a változatosság fenntartása érdekében az algoritmus nem hagyja, hogy az egyedek a feltételezett optimumra koncentrálódjanak. (5-11. ábra) (5-12. ábra) (5-13. ábra) (5-14. ábra)

A konvergencia leírására ezért a legjobb egyed célértékének megváltozását használom. Adott számú generációra visszatekintve a legjobb egyed minimális és maximális célfüggvény

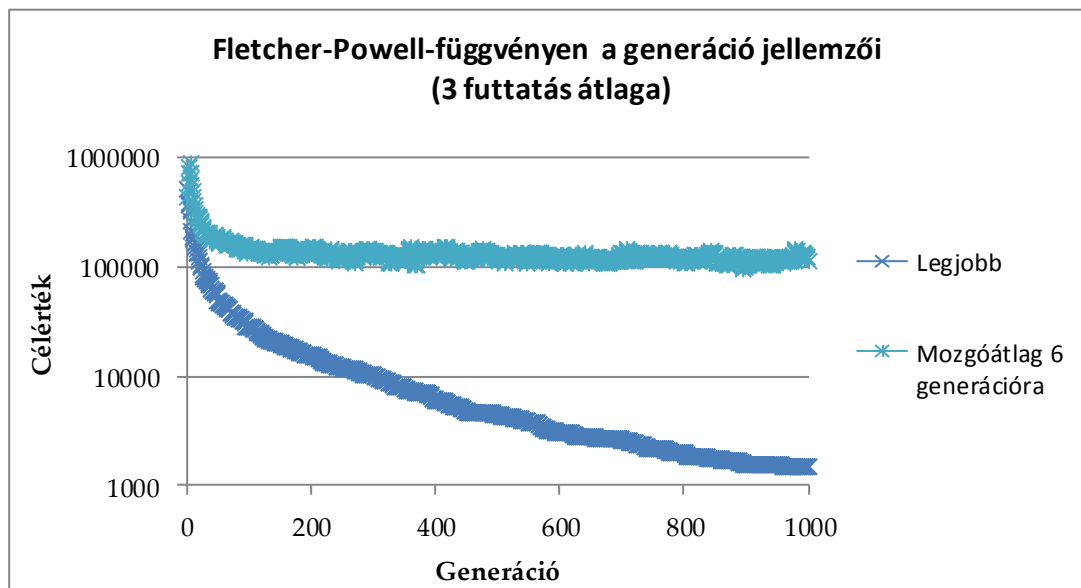
értékének különbségére lehet megadni egy kritériumot, ha ez a kritérium nulla, akkor a kilépési feltétel a legjobb célfüggvény változatlansága egy adott generációs számra.



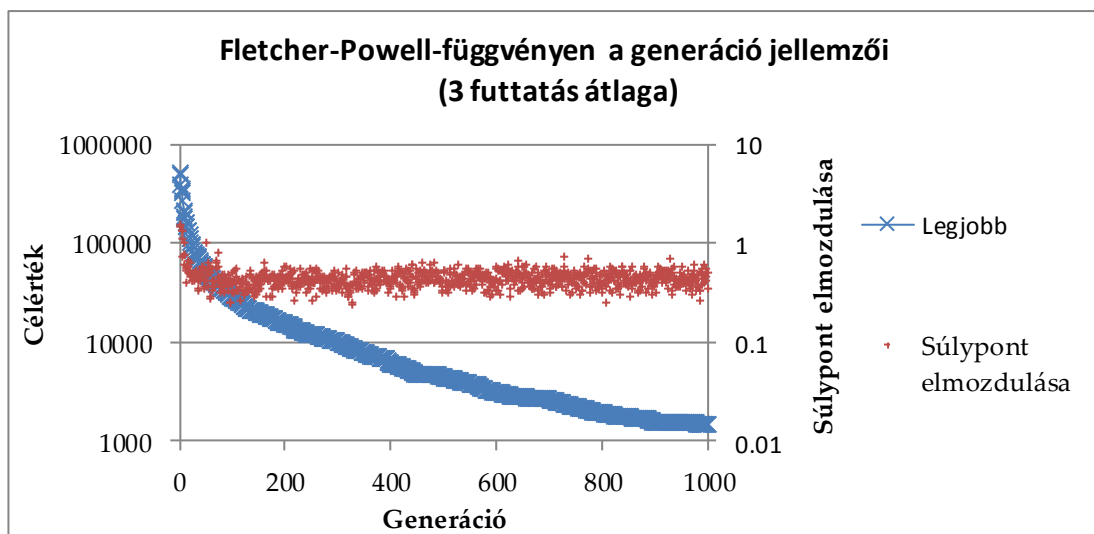
5-11. ábra. Célértékek átlaga Fletcher-Powell-függvényen



5-12. ábra. Célértékek szórása Fletcher-Powell-függvényen



5-13. ábra. Célértékek mozgóátlaga Fletcher-Powell-függvényen



5-14. ábra. Generáció súlypontjának elmozdulása Fletcher-Powell-függvényen

5.2.5. A VIZSGÁLATOK ÉRTÉKELÉSE

A vizsgálatok alkalmat adtak arra, hogy megfigyeljem a különböző beavatkozások hatását. Ez általánosításokra ad lehetőséget, így lehet tudni, hogy az egyes konvergencia hibák kiküszöbölésére melyik paramétert milyen irányban kell módosítani.

Nem megfelelő skálázás esetén előfordulhat, hogy egy kromoszóma túlzottan elterjed. Ekkor a változatosság növelése érdekében érdemes a mutáció valószínűségét megemelni és a random bevándorlót alkalmazni.

Nagy keresési terek esetén problémát jelent, hogy optimálisabb kromoszóma összerakható a rendelkezésre álló génekből, de az egyponos keresztezés erre képtelen, ekkor érdemes növelni a keresztezési pontok számát, és a keresztezés valószínűségét.

A fent említett drasztikus beavatkozások mellett a jobb egyedek nagyobb számban való megőrzésére érdemes alkalmazni a legjobb egyedeket továbbító átlapolt reprodukciót. Ez úgy működik, mintha több egyedre lenne érvényes az elitizmust.

Az aritmetikai keresztezés és a korlátos mutáció alkalmazása növeli a változatosságot, ugyanakkor ezek a szülőegedek környezetében keresnek, azaz az optimumot pontosabban megtalálják. Megfelelően alkalmazva növelik a végeredmény pontosságát.

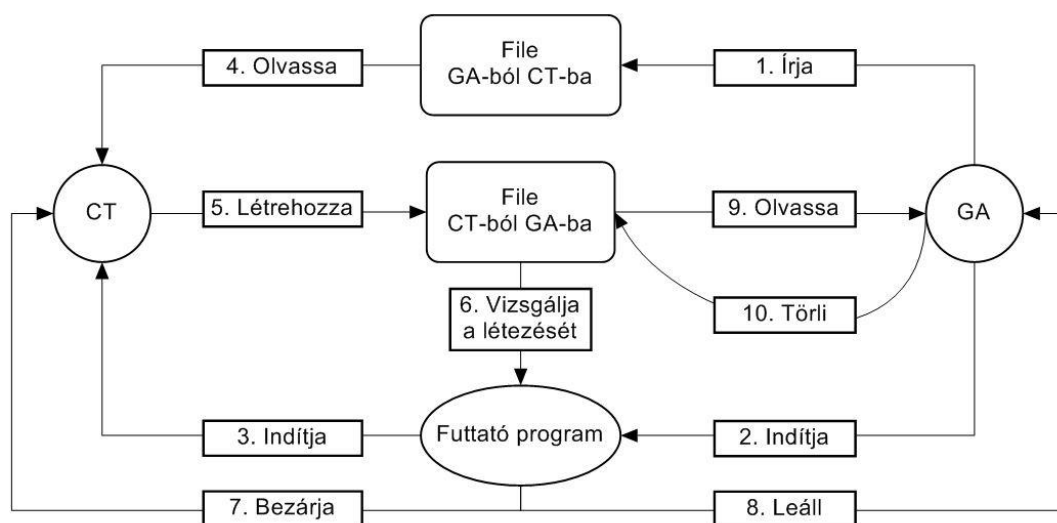
5.3. Cycle-Tempoval együttműködve

A GA alkalmazására végső soron a Cycle-Tempo-val együttműködve kerül sor. Az egyedek génjei egy adott kapcsoláshoz tartozó paramétereket jelölik. A célfüggvény a nettó hatások.

5.3.1. KAPCSOLAT MEGVALÓSÍTÁSA

A CT szubrutinjainak limitált a hatása a CT számítására, ezért a genetikus algoritmussal kapcsolatban csak az adatkapcsolat megvalósítására használhatók. Azaz a GA a CT-t csak egy függvényként látja, amelynek átadja a kromoszóma által reprezentált adatokat és visszakapja a célfüggvény értéket.

Az adatkapcsolatot egy harmadik program tartja fenn, amit a GA minden egyes kiértékelésnél elindít és vár a lefutására. Ez a program futtatja a CT számítását, majd bezárja a számításról szóló ablakot, hogy a következő kiértékelendő pontnál is lehessen futtatni. A GA-ból a CT-ba és viszont két külön fájlön keresztül valósul meg az adatcsere.



5-15. ábra. Cycle-Tempo GA-ból való futtatásának folyamata

A CT futtatása (5-15. ábra) a következőképp zajlik:

1. GA kiírja **XR** vektort a megfelelő fileba.
2. GA elindítja a futtató programot.
3. A futtató program futtatja a CT-t.
4. A CT beolvassa a GA kimeneti fileját.
5. A CT futása végeztével létrehozza a saját kimeneti fileját.
6. A futtató program már a CT futása alatt is 50 ms-os időközönként vizsgálja a file létrejöttét, de legfeljebb 30 s-ot vár.
7. Ha létrejött a file (azaz véget ért a CT futása), vagy ha letelt az időkeret, akkor a futtató program bezárja a CT felugró ablakát.
8. A futtató program leáll, és a GA folytatódik ott, ahol meghívta a futtató programot.
9. A GA beolvassa a CT-ből származó adatokat (ha van file).
10. A GA törli a CT által létrehozott file-t.

Az **XR** vektor értékeinek behelyettesítését az APSUB és PIPSUB szubrutinok végezhetik. Fontos megjegyezni, hogy a szubrutinok nincsenek hatással a rendszermátrixra, tehát valamilyen kezdeti értéket mindenképp meg kell adni a CT-n keresztül a változtatott jellemzőknek úgy, hogy a CT el tudja kezdeni az iterációt, és meghívja az APSUB és PIPSUB szubrutinokat, amik felülírják ezeket a jellemzőket. (3-4. ábra).

A CT kimeneti fileját az EPILOG szubrutin hozza létre. Mivel az EPILOG szubrutin az utolsó programrészlet, amit a CT végrehajt (3-4. ábra), így ez az EPILOG által létrehozott file

létezésének vizsgálata alkalmas arra, hogy jelezze a CT futásának végét. Hogy a számítás biztonságban leálljon még 50 ms-ot vár a futtató program, mielőtt bezárná a felugró ablakot.

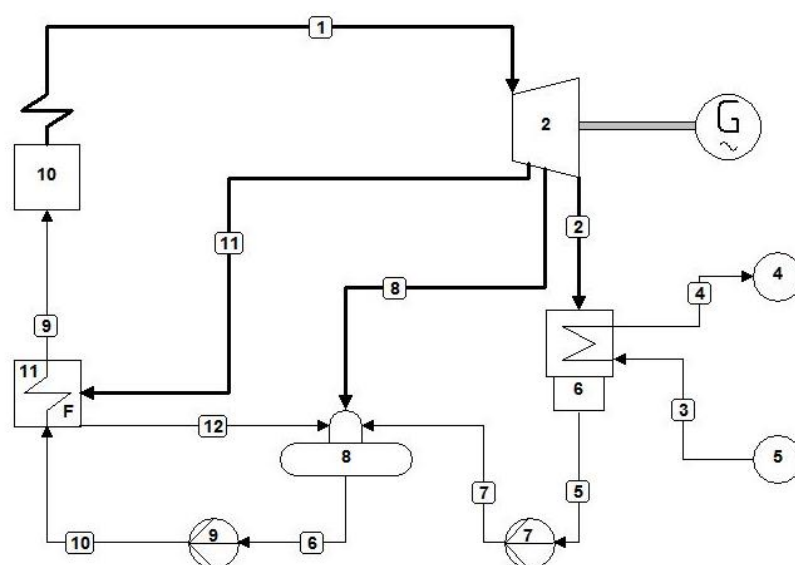
Lehetséges, hogy az **XR** vektor olyan értékek tartalmaz (például korlátos mutációnál létrejövő negatív értékek eredményezhetnek negatív elemeket az **XR** vektorban), hogy a CT error hibaüzenettel áll le. Ekkor az EPILOG szubrutin semmiképp sem fut le, így nem jön létre a kimeneti file. Ezt az esetet hivatott kezelni a futtató programba épített időkorlát. A probléma a GA-ra is kihat, mivel az megpróbálja megnyitni a nem létező filet, tehát ezt az esetet is kezelni kell.

A létrehozott file tartalmazza a célfüggvény értékét (nettó hatásfok, vagy esetleg saját definíció alapján számított célfüggvény) valamint tartalmazza a kapott warningok számát. Ennek a két adatnak a kiértékelését a GA végzi. A warningokat büntetéssel (célérték csökkentésével) veszi figyelembe.

A célfüggvényt kiértékelő programrészletbe érdemes a CT futtatása előtt előkiértékelést végezni, ha rendelkezésre állnak olyan előismeretek, amelyek alapján biztosan kizárhatók egyes megoldások. A kizárt egyed célfüggvény értéke 0.

5.3.2. ELLENŐRZÉS

A genetikusan algoritmus és a Cycle-Tempo együttműködésének ellenőrzését egy egyszerű modellen végeztem el. (5-16. ábra) Az optimalizálandó jellemzők a tápvíz előmelegítő és a gáztalanító táptartály megcsapolási nyomásai voltak. Mindkettő optimumát 1,013 bar és 80 bar között kereste az algoritmus. Az állandó paramétereket az 5-2. táblázat tartalmazza. Az optimalizálás célfüggvényének itt a bruttó hatásfokot tekintettem.



5-16. ábra. A vizsgált modell

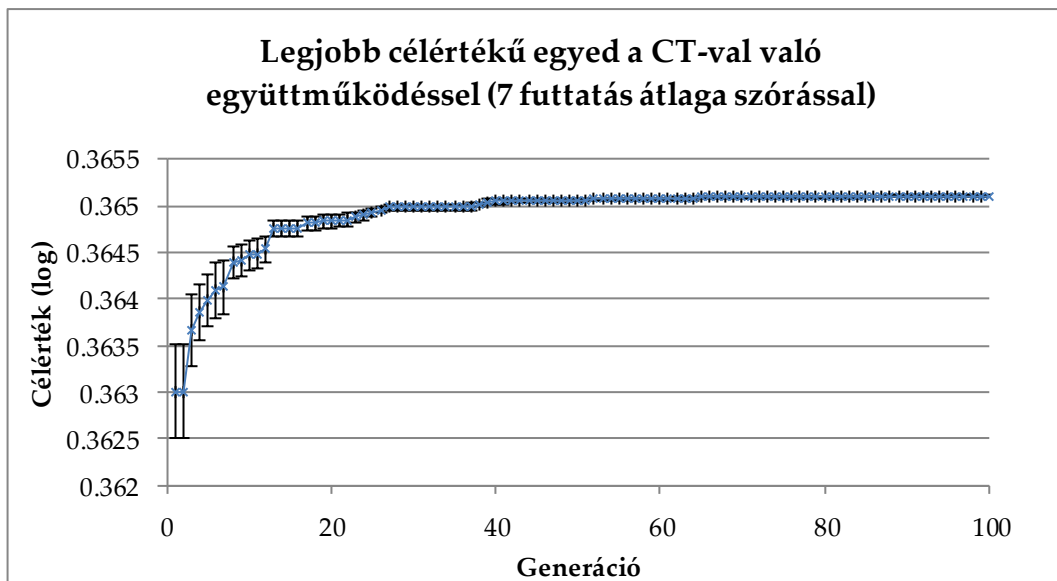
5-2. táblázat. A modellhez felvett paraméterek

Paraméter neve	Paraméter értéke
Frissgőznyomás	120 bar
Frissgőz hőmérséklet	420 °C
Teljesítmény	25 MW
Táp- és kondenzátum szivattyú hatásfok	80%
Turbina hatásfok	80%
Kondenzátornyomás	0,04 bar
Hűtővíz hőmérséklet	20 °C
Hűtővíz felmelegedés	5 °C
Nyomáskeresés a kazánon	10 bar
Nyomáskeresés a tápvíz előmelegítőn	0,4 bar

A modellen beállított értékek irrelevánsak, a teszt célja az algoritmus helyes lefutásának ellenőrzése volt, nem pedig az algoritmus alkalmazása. A GA paramétereinek az az 5-1. táblázatban felvett értékeket alkalmaztam. Az 5-17. ábrán, látható hét futtatás átlaga és a szórások. Az ábrán látszik, hogy a GA alkalmas az egyre jobb hatásfokok elérésére. A GA által talált legjobb paramétereket az 5-3. táblázat foglalja össze.

5-3. táblázat. A genetikus algoritmus által talált legjobb paraméterek

Paraméter neve	Paraméter értéke
Gáztalanító táptartály nyomása	3,953 bar
Tápvíz előmelegítő nyomása	21,106 bar
Hatásfok	36,511%



5-17. ábra. A legjobb paraméterű modell hatásfoka a generációk során

5.3.3. ALKALMAZÁS

A genetikus algoritmus jelenleg vizsgált végső alkalmazási területe a Láng-BBC 215 MW-os gőzturbina Cycle-Tempoban elkészített termodinamikai modelljének optimalizációja.

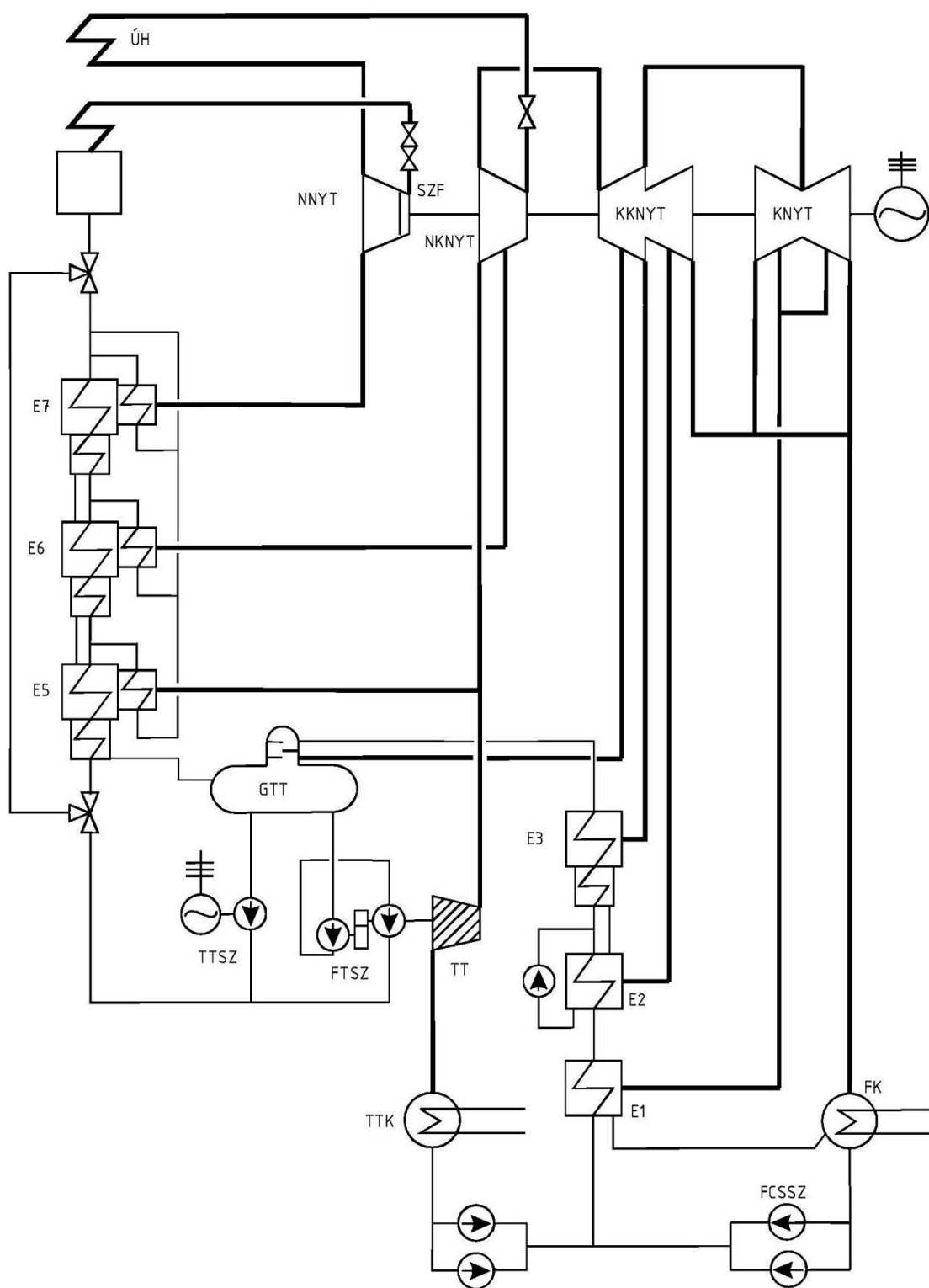
A kvantitatív elemzés a szakdolgozatom részét képezi. Ebben a dolgozatban csak a Cycle-Tempo modell és a Genetikus Algoritmus együttműködésének teszteredményeit közlöm.

5.3.3.1. A vizsgált rendszer

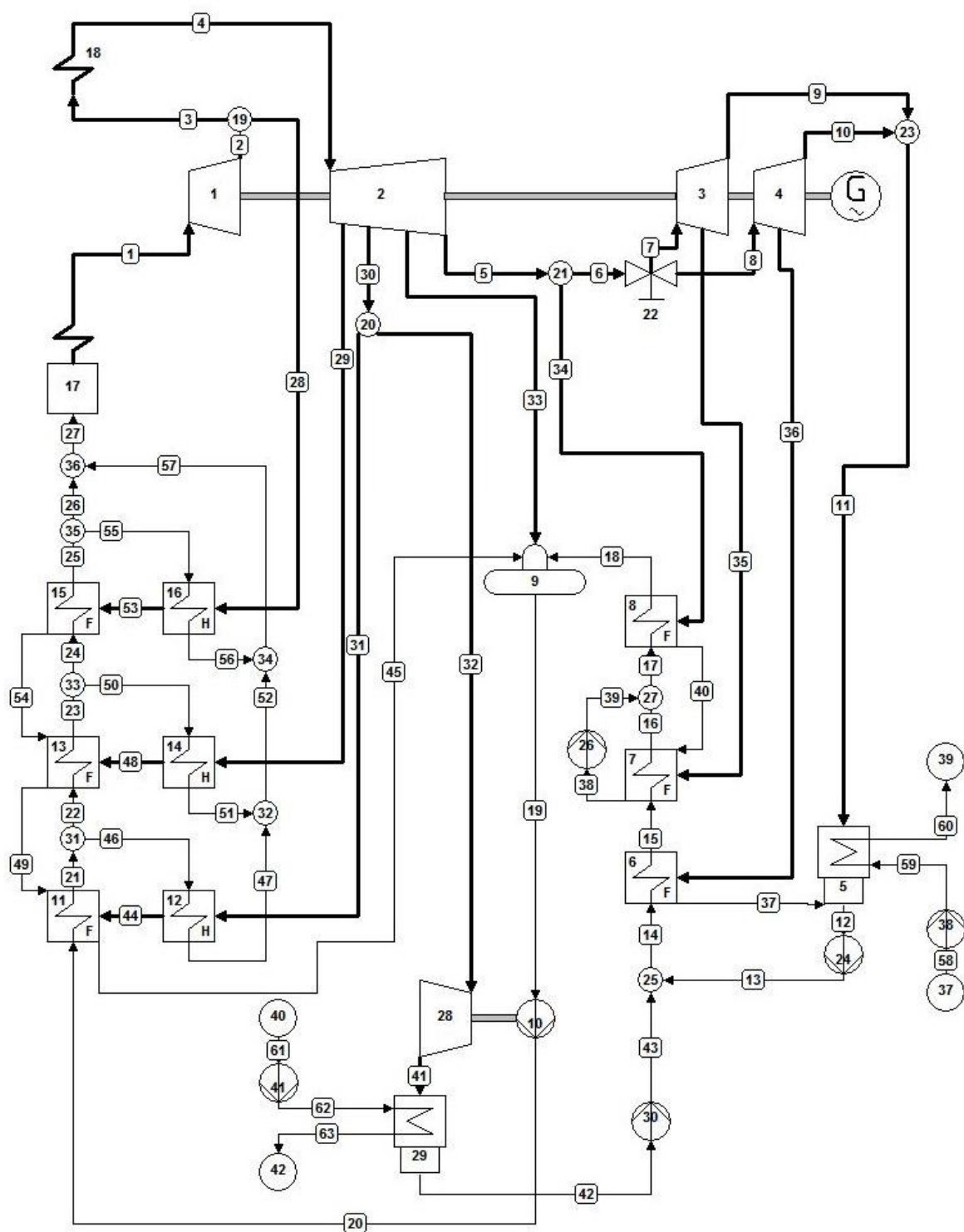
A Láng-BBC 215MW-os blokk jelentősége abban rejlik, hogy a hazai erőműparkban a legnagyobb számban előforduló egység, és számos hasonló rendszer bizonyos átalakításokkal máig üzemel.

A rendszer kondenzációs körfolyamatot valósít meg. A kazánból névleges állapotban $186 \frac{kg}{s}$ frissgőzt termel 164.75 bar , 545°C gőzparaméterekkel. A frissgőz a turbina nagynyomású házán expandál 38 bar nyomásig, a nagynyomású ház után a gőzt újrahevítik, illetve egy része az utolsó tápvízelőmelegítőt fűti. A frissgőz-hőmérsékletig újrahevített gőz a nagyközépnomású, kisközépnomású és kisnyomású turbinarészekén expandál. Itt további 6 helyen van megcsapolás a turbinán a tápvízelőmelegítő-rendszer és a tápturbina ellátására. A kondenzáció frissvízhűtéssel valósul meg. A kondenzátumot melegítő kisnyomású tápvízelőmelegítő-rendszerben vegyesen van jelen kaszkád és előrekeveréses kapcsolás. A nagynyomású tápvízelőmelegítő-rendszer Nicolni-Ricard kapcsolású, azaz mellékáramú gőzhűtőkkel rendelkezik. (BIHARI, 2002) A modell az 5-17. ábrán bemutatott kapcsolással egyenértékű topológiával rendelkezik.

A modellezés lépéseit nem részletezve referencia adatként figyelembe lehet venni, hogy a modell által szolgáltatott nettó hatásfokot: 43,28%. A genetikus algoritmus által elért eredményeket ennek tükrében lehet értelmezni. A modell kapcsolását 5-18. ábra mutatja.



5-17. ábra. A vizsgált blokk kapcsolása((BIHARI, 2002): 117. o. 3-22. ábra)



5-18. ábra. A vizsgált modell kapcsolása.

5.3.3.2. Teszt

Ahogy fent említettem jelen munka a genetikus algoritmus és a végleges modell együttes viselkedésének vizsgálatáig terjed.

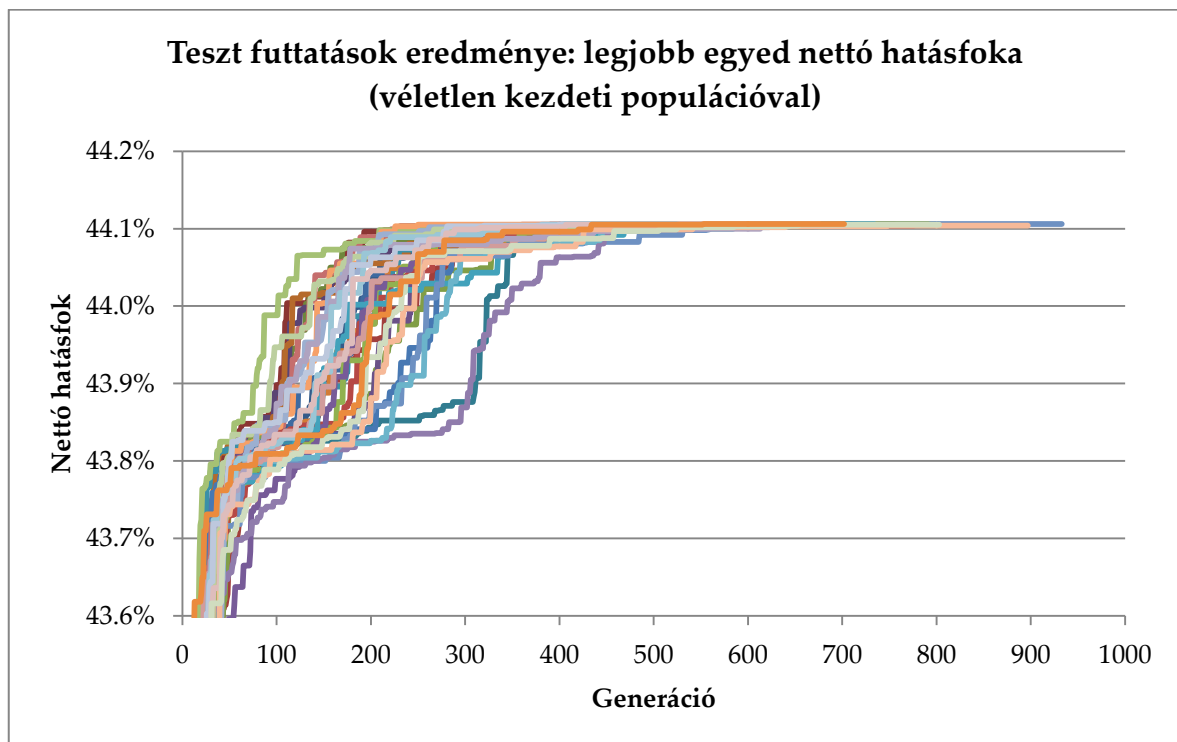
A futtatások során eddigiektől eltérő események is előfordultak, ilyenek a különböző felugró ablakokkal járó hibák, és a futtatás időigényének megnövekedése. Ezeket sikeresen kezeltem, így a Genetikus Algoritmus már az esetek 99%-ában hiba nélkül fut, a maradék hibák okai a fejlesztett programokon kívül álló okokra vezethetők vissza. (például frissítési kérések). A számítás nagy időigénye miatt, a futtatásokat megszakítom, ha 150 generáción keresztül nem változik a legjobb egyed célfüggvény értéke.

A keresési teret a futtatások teszt jellege miatt intuitív módon vettem fel, ennek elemeit az 5-4. táblázat tartalmazza:

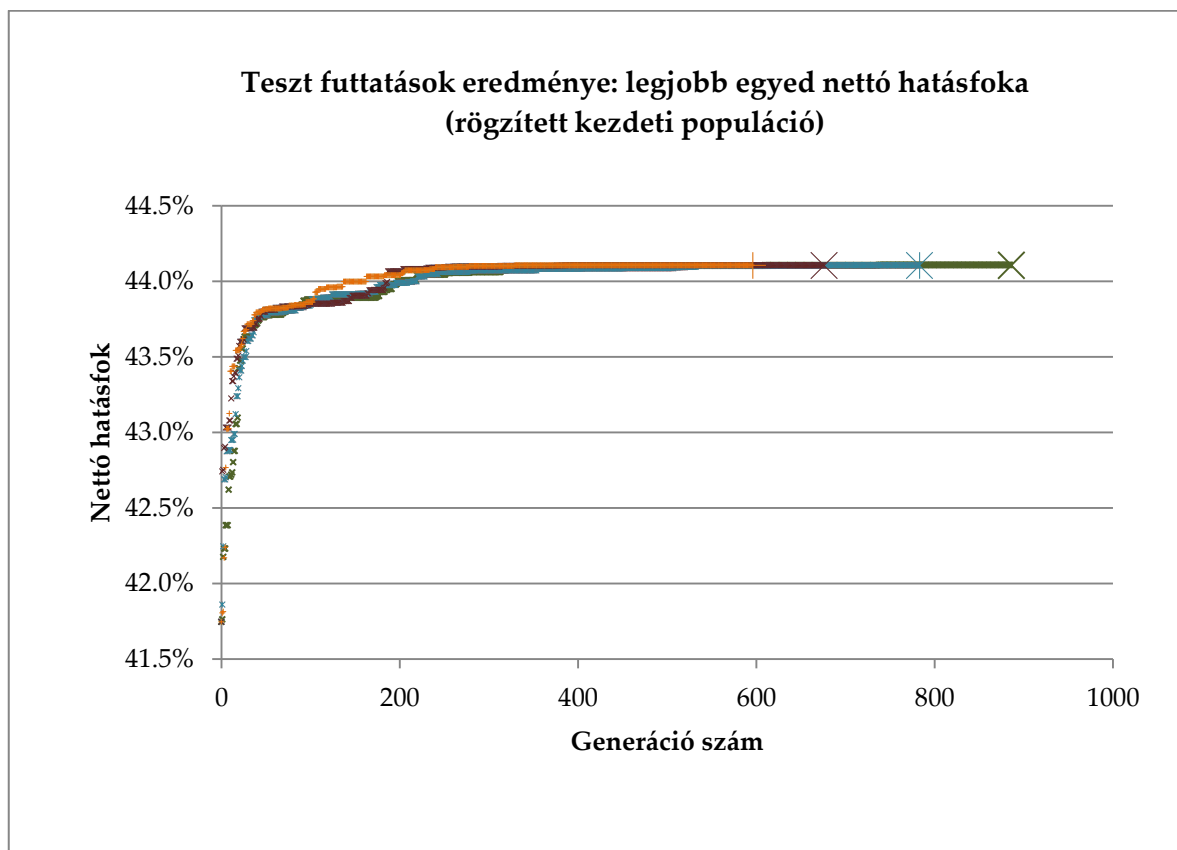
5-4. táblázat. Keresési tér a tesztek alatt

Sorszám	Paraméter neve	Paraméter alsó korlátja	Paraméter felső korlátja
1	E1 előmelegítő fűtőgőzének nyomása (bar)	0,05	3
2	E2 előmelegítő fűtőgőzének nyomása (bar)	0,1	7
3	E3 előmelegítő fűtőgőzének nyomása (bar)	0.5	10
4	GTT fűtőgőzének nyomása (bar)	1.1	15
5	E5 előmelegítő fűtőgőzének nyomása (bar)	3	30
6	E6 előmelegítő fűtőgőzének nyomása (bar)	7	50
7	E7 előmelegítő fűtőgőzének nyomása (bar)	10	60
8	Frissgőznyomás (bar)	120	164.75
9	Frissgőz hőmérséklet (°C)	400	540

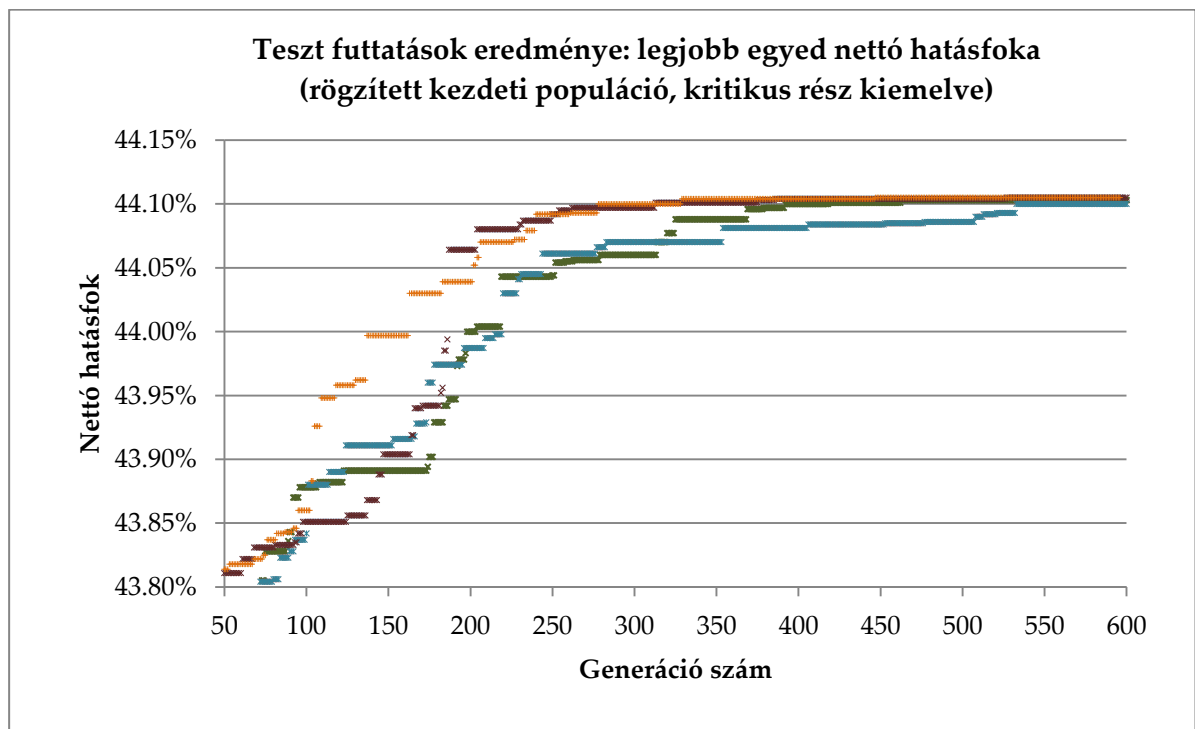
A fent meghatározott keresési téren alkalmaztam a Genetikus Algoritmust az 5. mellékletben jelzett paraméterekkel. Vizsgáltam véletlen és rögzített kezdeti populáció esetén is az eredményeket. (5-19. ábra, 5-20. ábra, 5-21. ábra)



5-19. ábra. Legjobb egyed nettó hatásfoka véletlen kezdeti populációról indított esetben (33 futtatás)



5-20. ábra. Legjobb egyed nettó hatásfoka rögzített kezdeti populációról indított esetben (4 futtatás)



5-21. ábra. Legjobb egyed nettó hatásfoka rögzített kezdeti populációról indított esetben (4 futtatás)
Csak a köztes generációkban alakulnak ki nagyobb eltérések.

Az 5-20. ábrán és 5-21. ábrán jól látszik, hogy adott kezdeti populáció esetén csak a köztes eredmények mutatnak nagy eltérést, a végeredmény minden esetben ugyanaz. 5-19. ábra alapján elmondható, hogy a kezdeti populáció hatása jelentős a köztes eredményekre, de a végeredményre nincs hatással.

A Cycle-Tempo modell teszteléskor használt paraméterei esetén a Genetikus Algoritmus minden esetben megtalálta ugyanazt az optimumot. Ez alapján kijelenthető, hogy a GA alkalmas a további vizsgálatok elvégzésére. Az elért 0,82% pont –tervezési állapotbeli– nettó hatásfok növekmény nagyságrendjében várhatók eredmények a későbbi alkalmazás során.

Az optimalizált paraméterekre a 5-5. táblázat ad egy példát.

5-5. táblázat. Kezdeti paraméterek és eredmények

Sorszám	Paraméter neve	Tervezési paraméterek	Optimalizált paraméterek
1	E1 előmelegítő fűtőgőzének nyomása (bar)	0,165	0,133
2	E2 előmelegítő fűtőgőzének nyomása (bar)	0,625	0,259
3	E3 előmelegítő fűtőgőzének nyomása (bar)	1,46	0,50
4	GTT fűtőgőzének nyomása (bar)	4,30	3,84
5	E5 előmelegítő fűtőgőzének nyomása (bar)	8,81	13,55
6	E6 előmelegítő fűtőgőzének nyomása (bar)	17,91	33.63
7	E7 előmelegítő fűtőgőzének nyomása (bar)	37,46	60.00
8	Frissgőznyomás (bar)	164.75	164.75
9	Frissgőz hőmérséklet (°C)	540	540
Nettó hatásfok		43,283%	44,106%

Látható, hogy az algoritmus több helyen „felütközött” a keresési tér határára. Ez a frissgőz paraméterek esetén triviális. Az E3 jelű kisnyomású előmelegítő a jövőben esetén érdemes kiszélesíteni a keresési teret.

Az E7 jelű nagynyomású előmelegítő esete külön megfontolást igényel. Az algoritmus azért találja optimálisnak a nyomás növelését ebben a pontban, mert topológiából adódóan ez a nyomás határozza meg az újrahevítési nyomást is. A tesztek során a Cycle-Tempo beépített turbina-hatásfok számítási módszereit alkalmaztam. Ez különböző turbina-hatásfokokat eredményezett az 1-es jelű nagynyomású turbinát és a 2-es jelű nagyközép- és kisközépnyomású turbinát modellező elemek esetén. A számított turbina-hatásfok a 2-es elem esetén volt nagyobb, így evidens, hogy az algoritmus igyekszik az expanzió nagyobb részét a nagyobb hatásfokú elemen megvalósítani, az az újrahevítési nyomás növeléséhez vezetett. A jövőben ennek elkerülésére állandó fokozati hatásfok mellett végzem majd az optimalizációt.

Minden futtatás az 5-5. táblázatban bemutatott Optimalizált értékek környezetében találta meg az optimális paramétereket.

6. EREDMÉNYEK ÉRTÉKELÉSE

6.1. *Eredmények*

A genetikus algoritmus és a Cycle-Tempo együttműködése megvalósítható. A futtatásuk más hőséma számító programokhoz képest rövid időn belül végbemegy. Ez alapján a módszer alkalmas arra, hogy belátható időn belül a kiinduló paramétereket felülmúlja.

6.2. *Javaslatok*

A továbbiakban a genetikus algoritmust alkalmazom a végleges modellen. A kapott adatokat elemzem és vizsgálom az optimalizált erőmű viselkedését részterhelésen.

A konvergencia gyorsításának érdekében megvizsgálom az adaptív mutáció (irányítottan változik a mutáció valószínűsége generációról generációra) alkalmazásának lehetőségét.

Természetesen meg kell jegyezni, hogy egy erőmű tervezési pontban vett hatásfoka nem jelent jó célfüggvényt a gyakorlatban. Azonban az alkalmazott módszerek felhasználásával alkotható lenne olyan algoritmus, ami például egy tervezés alatt álló rendszerre keres gazdasági optimumot. A fejlesztett szoftverek alapján olyan algoritmus is alkotható lenne, ami egy meglévő erőmű retrofitjének előkészítésében játszhatna szerepet például a menetrend ismeretében minimalizálva az átlagos tüzelőanyag fogyasztást.

7. FELHASZNÁLT FORRÁSOK

1. ÁLMOS ATTILA, GYÓRI SÁNDOR, HORVÁT GÁBOR, VÁRKONYINÉ KÓCZY ANNAMÁRIA (2002): *Genetikus algoritmusok*. Typotex, Budapest
2. BIHARI PÉTER (2002): *Erőművek*. Jegyzet
3. DEAP (DISTRIBUTED EVOLUTIONARY ALGORITHMS IN PYTHON) (2014): *Continuous Optimization*. Elérhető: <http://deap.gel.ulaval.ca/doc/0.9/api/benchmarks.html#continuous-optimization>
4. FLETCHER R. AND POWELL M. J. D. (1963): *A rapidly convergent descent method for minimization*. Computer Journal
5. FRIESE T., ULBIG P. AND SCHULZ S. (1998): *Use of evolutionary algorithms for the calculation of group contribution parameters in order to predict thermodynamic properties. Part 1: Genetic algorithms*. Computers Chem. Engng
6. GAFFNEY J., GREEN D.A., PEARCE C.E.M. (2010): *Binary versus real coding for genetic algorithms: A false dichotomy?* Anziam
7. KAVIRI ABDOLSAEID GANJEH, JAAFAR MOHAMMAD NAZRI MOHD., LAZIM THOLUDIN MAT (2012): *Modeling and multi-objective exergy based optimization of a combined cycle power plant using a genetic algorithm*. Energy Conversion and Management
8. QIN XIAOYONG, CHEN LINGEN, SUN FENGRUI, WU CHIH (2003): *Optimization for a steam turbine stage efficiency using a genetic algorithm*. Applied Thermal Engineering
9. SUBBARAJ P., RENGARAJ R., SALIVAHANAN S. (2008): *Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm*. Applied Energy
10. TU DELFT (2007a): *Introduction*. Elérhető: <http://www.cycle-tempo.nl/>
11. TU DELFT (2007b): *Technical Notes*. Elérhető: <http://www.cycle-tempo.nl/>
12. TU DELFT (2007c): *User Subroutines*. Elérhető: <http://www.cycle-tempo.nl/>
13. VALDES MANUEL, DURAN MA DOLORES, ROVIRA ANTONIO (2003): *Thermoeconomic optimization of combined cycle gas turbine power plants using genetic algorithms*. Applied Thermal Engineering
14. XI HUAN, LI MING-JIA, XU CHAO, HE YA-LING (2013): *Parametric optimization of regenerative organic Rankine cycle (ORC) for low grade waste heat recovery using genetic algorithm*. Energy

8. MELLÉKLETEK

1. melléklet. Himmelblau-függvény esetén használt paraméterek

Paraméter neve	Paraméter értéke
Populáció mérete	30
Skálázás	Hatvány alapú
Skálázási kitevő	0,4
Keresztezés típusa	Egyszerű és aritmetikai
Keresztezésben résztvevő szülők száma	2
Keresztezések száma	1
Keresztezés valószínűsége	0,6
Egyszerű keresztezés várható részaránya	0,5
Mutáció típusa	Egyszerű és korlátos
Mutáció valószínűsége	0,4
Egyszerű mutáció várható részaránya	0,5
Korlátos mutáció sugara	Keresési intervallum 10%-a
Reprodukció típusa	Átlapolt a legjobb egyedekkel
Szülőgeneráció részaránya	0,1
Elitizmus	Nincs
Random bevándorló	Van, legrosszabbat helyettesíti

2. melléklet. Ugrásfüggvény esetén használt paraméterek

Paraméter neve	Paraméter értéke
Populáció mérete	30
Skálázás	Hatvány alapú
Skálázási kitevő	3,9
Keresztezés típusa	Egyszerű és aritmetikai
Keresztezésben résztvevő szülők száma	2
Keresztezések száma	3
Keresztezés valószínűsége	0,6
Egyszerű keresztezés várható részaránya	0,7
Mutáció típusa	Egyszerű és korlátos
Mutáció valószínűsége	0,4
Egyszerű mutáció várható részaránya	0,3
Korlátos mutáció sugara	Keresési intervallum 30%-a
Reprodukció típusa	Átlapolt a legjobb egyedekkel
Szülőgeneráció részaránya	0,1
Elitizmus	Van, egy random egyedet helyettesít
Random bevándorló	Van, legrosszabbat helyettesíti

3. melléklet. Ackley-függvény esetén használt paraméterek

Paraméter neve	Paraméter értéke
Populáció mérete	30
Skálázás	Hatvány alapú
Skálázási kitevő	0,3
Keresztezés típusa	Egyszerű és aritmetikai
Keresztezésben résztvevő szülők száma	2
Keresztezések száma	6
Keresztezés valószínűsége	1,0
Egyszerű keresztezés várható részaránya	0,2
Mutáció típusa	Egyszerű és korlátos
Mutáció valószínűsége	0,7
Egyszerű mutáció várható részaránya	0,2
Korlátos mutáció sugara	Keresési intervallum 0,1%-a
Reprodukció típusa	Átlapolt a legjobb egyedekkel
Szülőgeneráció részaránya	0,5
Elitizmus	Nincs
Random bevándorló	Van, legrosszabbat helyettesíti

4. melléklet. Fletcher-Powell-függvény esetén használt paraméterek

Paraméter neve	Paraméter értéke
Populáció mérete	30
Skálázás	Szigma csonkolás
Tolerancia tartomány	1,2
Keresztezés típusa	Egyszerű és aritmetikai
Keresztezésben résztvevő szülők száma	2
Keresztezések száma	6
Keresztezés valószínűsége	0,8
Egyszerű keresztezés várható részaránya	0,2
Mutáció típusa	Egyszerű és korlátos
Mutáció valószínűsége	0,6
Egyszerű mutáció várható részaránya	0,2
Korlátos mutáció sugara	Keresési intervallum 1%-a
Reprodukció típusa	Átlapolt a legjobb egyedekkel
Szülőgeneráció részaránya	0,2
Elitizmus	Van, második legrosszabbat, vagy a random bevándorlót helyettesíti
Random bevándorló	Van, legrosszabbat helyettesíti

4. melléklet. A Cycle-Tempoval való együttműködésnél használt paraméterek

Paraméter neve	Paraméter értéke
Populáció mérete	30
Skálázás	Szigma csonkolás
Tolerancia tartomány	1,2
Keresztezés típusa	Egyszerű és aritmetikai
Keresztezésben résztvevő szülők száma	2
Keresztezések száma	1
Keresztezés valószínűsége	0,7
Egyszerű keresztezés várható részaránya	0,35
Mutáció típusa	Egyszerű és korlátos
Mutáció valószínűsége	0,5
Egyszerű mutáció várható részaránya	0,1
Korlátos mutáció sugara	Keresési intervallum 5%-a
Reprodukció típusa	Átlapolt a legjobb egyedekkel
Szülőgeneráció részaránya	0,1
Elitizmus	Nincs
Random bevándorló	Nincs