

Bináris tomográfiai rekonstrukció objektum alapú evolúciós algoritmussal*

Gara Mihály, Balázs Péter

Képfeldolgozás és Számítógépes Grafika Tanszék
Szegedi Tudományegyetem
6720 Szeged, Árpád tét 2.
gara, pbalazs@inf.u-szeged.hu

Absztrakt. Korábbi munkánkban [1] kifejlesztettünk egy olyan evolúciós algoritmust, amely a bináris tomográfia egy speciális rekonstrukciós problémájára próbál megoldást szolgáltatni: egy körgyűrű belsejében található körlapokat tartalmazó bináris kép előállítására a vetületeinek ismeretében. Ebben a dolgozatban azt vizsgáltuk, hogy mik ennek az algoritmusnak az optimális paraméterei, mekkora az ideális populáció méret, valamint hány generáció után érdemes leállítani az algoritmust.

1. Bevezetés

A *tomográfia* egy olyan leképező eljárás, amely háromdimenziós objektumoknak állítja elő a kétdimenziós keresztmetszeti képét. A *rekonstrukciós tomográfiában* ezeket a keresztmetszeti képeket a vizsgált objektum vetületeinek segítségével alkotják meg. Számtalan ilyen rekonstrukciós algoritmuson alapuló alkalmazást kidolgoztak már különböző felhasználási területeknek megfelelően. Így például a nemroncsoló anyagtesztelésben, az angiográfiában, a kristályszerkezet-vizsgálatban, az adat titkosításban, a képfeldolgozásban, és így tovább. Ezen alkalmazások legtöbbjében a rekonstruált szürkeárnyaltos képen az előforduló intenzitásértékek száma kevés, de a felhasználható vetületek száma is korlátozott. A *diszkrét tomográfia* [4, 5] azt vizsgálja, hogyan lehet az előbbi feltételeknek megfelelően kialakítani a keresztmetszeti képeket. Egy szintén elterjedt, de még több megszorításon alapuló módszer, az, amikor bináris képeket próbálnak rekonstruálni. Ezt a fajta tomográfiát *bináris tomográfiának* nevezik, ennek is számos alkalmazása létezik.

A következőkben olyan bináris képekkel dolgozunk, melyek tartalmaznak egy körgyűrűt, aminek a vastagsága változhat, továbbá ezen gyűrű belsejében elhelyezkedik néhány egymást nem metsző körlap. Mivel a képeken csak meghatározott alakzatokat engedünk meg, ezért leírhatjuk őket a rajtuk található objektumok alakjellemezőivel, így a rekonstrukció ekkor tulajdonképpen nem más,

* A kutatást részben a Nemzeti Fejlesztési Ügynökség TÁMOP-4.2.2/08/1/2008-0008 és TÁMOP-4.2.1/B-09/1/KONV-2010-0005 programjai támogatták. A második szerző munkáját az MTA Bolyai János Kutatási Ösztöndíj programja is támogatta.

mint a körök középpontjainak és sugarának a legmegfelelőbb elhelyezése a képen. A 1. ábrán egy ilyen képre láthatunk példát.

A dolgozat felépítése a következő: a 2.. fejezetben ismertetjük a legszükségesebb fogalmakat, ezt követően bemutatjuk a problémát és magát az evolúciós algoritmust a 3.. fejezetben. A hiba mérésének módját és annak speciális eseteit a 4.. fejezetben járjuk körül, eredményeinket pedig a 5.. részben írjuk le, melyet a 6.. fejezetben az összefoglaló és a további terveink követnek.

2. Fogalmak, definíciók

A rekonstrukciós tomográfia a keresztmetszeti képeket a vetületek ismeretében állítja elő. Munkánk során a vetületképzéskor folytonos értéket használtunk. Az objektumokat folytonosnak tekintettük, majd egy adott irányból egymással párhuzamosan sugarakat húztunk rajtuk. Az egyes vetületi értékeket az egy-egy vetítősugar által az objektumokból kimetszett szakaszok hosszainak összegei képzik.

A 1. ábrán látható egy példa a körgyűrűn belül egymást nem metsző körlapokat tartalmazó alakzatra, valamint ennek 0, 45, 90 és 135 fokból vett vetületeire.



1. ábra: Vetületképzés négy irányból: hat körlapot tartalmazó kép és annak vetületei 0, 45, 90, 135 fokos irányokból.

Mivel az eredeti képről azt feltételezzük, hogy azon csak körlapok (és egy körgyűrű) találhatóak, ezért ezeket a középpont-koordinátáikkal és a sugarukkal reprezentálhatjuk, a körgyűrű esetén két körlap különbségéként. Ekkor a vetületek kiszámításához elegendő a vetítősugarak által a körökből kimetszett húrok hosszának ismerete. Ezeket a hosszakat a körök sugarainak és középpontjainak ismeretében könnyen kiszámíthatjuk.

3. Körmetszeteket tartalmazó képek rekonstruálása

A bemutatásra kerülő algoritmus olyan képeket rekonstruál, amelyeken található egy körgyűrű, melynek középpontja egybe esik a kép közepével, ezen kívül csak diszjunkt körlapok alkotják, melyek a körgyűrű belsejében helyezkednek el, és nem metszik azt. (1. ábra bal oldali kép.) Ilyen tulajdonságú fantomot korábban

a [6] munkában neutrontomográfias kísérletekhez alkalmaztak. Az ilyen típusú vetületképzés esetén a hagyományos tomográfias eljárások nem alkalmazhatóak, mert az azokhoz szükséges többszáz vetületi kép elkészítésekor a vizsgált anyag jelentős mértékben roncsolódna. Ezen képek esetén a rekonstrukciós feladat nem más mint, a körlapok megfelelő elhelyezkedésének, illetve méretének megtalálása. Azaz, amennyiben a körlapokat és a körgyűrűt is a koordinátáik és sugaraik segítségével reprezentáljuk, akkor a probléma felírható úgy, hogy adjuk meg a középpontok koordinátáinak és a sugarak alkotta rendezett hármasainak azon halmazát, amelyre a vetületi értékek minimálisan térnek el a rekonstruálandó kép vetületeitől.

Az ilyen, objektum alapú reprezentáción alapuló rekonstrukció jelenleg még nem terjedt el a szakirodalomban, a feladat feltehetően többféle algoritmus segítségével megoldható. Azonban a megfelelő körök halmazát jelen munkában egy evolúciós algoritmus [2] segítségével keressük. Választásunk azért esett erre a módszerre, mert meglátásunk szerint így lehet a legjobban kihasználni az objektumok nyújtotta lehetőségeket.

Ez az algoritmus a már meglévő megoldáskezdemények ötvözésével és módosításával állít elő további lehetséges megoldásokat, majd ezeket egy hasznossági – fitness – függvény szerint kiértékeli, és a legjobbakat tartja meg. A végső legjobb konfiguráció lesz az algoritmus által szolgáltatott megoldás. A rekonstrukcióhoz négy – függőleges, vízszintes, és a két átlós – irányból vett vetületeket használunk.

A képen található objektumokat listában tároljuk, ahol az első két számhármassal mindig a körgyűrűt reprezentálja (külső és belső körök), ezeknek a középpontja mindig a kép középpontjával egyezik meg, a többi pedig a belső körlapokat írja le. Ilyen konfigurációra látható néhány példa a 1. táblázatban. Az ilyen képek – valamint az őket leíró konfigurációk – lesznek az evolúciós algoritmus egyedei.

3.1. Az algoritmus áttekintése

Az evolúciós algoritmus egy γ számú egyedet – képkonfigurációt – tartalmazó kezdő populációból kiindulva állítja elő a lehetséges megoldásokat. A következő generáció mindig úgy alakul ki, hogy a populáció egyedei közülük egyenletes eloszlás szerint kiválaszt néhányat *kereszteléshez*, majd az ezzel képzett μ_1 darab *utód* bekerül a populációba. Ezt még egy egyedkiválasztás követi, de ezúttal a $\gamma + \mu_1$ egyed áll már rendelkezésre. A mutációval keletkezett újabb μ_2 darab egyed ugyancsak bekerül a populációba. Ezt követően a $\gamma + \mu_1 + \mu_2$ egyed közül kiválasztjuk a kiértékelő függvény szerint legjobb γ darabot, ezek alkotják majd a következő generációt. Az algoritmus addig folytatódik, amíg a legjobb egyed fitness értéke át nem lép egy meghatározott szintet, vagy a generációk száma el nem éri az előre beállított maximális értékét.

Az egyedek kiértékelése a következő függvénnyel történik:

$$f(p, c) = \lambda_1 \cdot \sum_{i=1}^4 \sum_{j=1}^n |p_{ij} - p_{ij}^*| + \lambda_2 \cdot \varphi(c), \quad (1)$$

ahol p_{ij} a vizsgált konfiguráció vetülete, p_{ij}^* pedig a bemeneti vetületi érték (i egy adott vetületi irány, ezek száma 4, j egy vetítősugár, ezek száma megegyezik a képmérettel, ami n). Lehetőség van az eredeti kép valamely előre ismertnek feltételezett tulajdonságát is bevinni a rekonstrukcióba, erre szolgál a $\varphi(c)$ tag. Ilyen előzetes információ lehet például, hogy közel egyforma méretű körlapokat feltételezünk a képen. Ekkor a nagyobb eltérés esetén nagyobb, kisebb esetén kisebb lesz a $\varphi(c)$ tag. Hasonlóan lehet felhasználni azt a tudást is, hogy hány körlap található a képen. Ennek lehetőségeit az [1] munka tárgyalja. A λ_1, λ_2 pedig alkalmasan választott konstansok, ezek határozzák meg a vetületektől való eltérés és az apriori információ hatásának arányát.

3.2. A keresztezés operátor

A keresztezés során az algoritmus a keresztezendő egyedeket egyenletes eloszlás szerint választja ki: minden egyedhez hozzárendel egy valós számot a $[0, 1]$ intervallumból. Ha ez a véletlen szám kisebb, mint egy előre meghatározott küszöb, akkor az egyed kiválasztja keresztezésre, egyébként nem. Minden kijelölt konfigurációhoz véletlenszerűen kiválaszt egy másik egyed a teljes populációból, majd megpróbálja őket egymással keresztezni. Ezen próbálkozásokat addig folytatja, amíg legalább egy egyed sikeresen létre nem jön, vagy a próbálkozások száma el nem ér egy meghatározott maximumot.

Maga a keresztezés úgy zajlik, hogy a két kiválasztott egyed reprezentáló listát elvágjuk egy-egy véletlenszerűen választott pozíción, majd a vágások mentén az egyik lista elejéhez illesztjük a másiknak a végét, és viszont. Hogy a környűrűt is mindkét esetben megőrizzük, a vágások mindig a második listaelem – tehát a környűrű belső oldalát reprezentáló kör – után és a lista vége között lehetnek. Előfordulhat, hogy a vágás következtében az egyik egyedből az összes elemet örökliz az egyik utód, valamint az is, hogy egyiket sem.

A keresztezést akkor mondjuk sikeresnek, és akkor vesszük fel a keletkezett utódot a populációba, ha az őt meghatározó körlapok duszjunktak, és a környűrűn belül helyezkednek el. A sikeres és sikertelen keresztezésre láthatunk egy-egy példát a 2. ábrán.



2. ábra: Példa keresztezésre. Az ábrán látható képek rendre: két szülő, a sikeres utód, és a sikertelen utód.

Két szülő és utódjaik konfigurációjára a 1. táblázat mutat egy példát. A táblázatban szereplő utódoknál az egyes számhármasknál feltüntettük, hogy azt az első, vagy a második szülőtől örökölte.

1. táblázat: Két szülő, valamint a keresztezés operátor segítségével létrehozott utódok konfigurációi (képméret: 100×100). Az első két-két számhármask a körgyűrűt, a többi pedig a körlapokat írja le. Az utódoknál minden számhármask után alsóindexben jelöltük, hogy melyik szülőtől származnak.

Szülő ₁	Szülő ₂	Utód ₁	Utód ₂
(50, 50, 40.01)	(50, 50, 45.6)	(50, 50, 45.6) ₂	(50, 50, 40.01) ₁
(50, 50, 36.16)	(50, 50, 36.14)	(50, 50, 36.14) ₂	(50, 50, 36.16) ₁
(41.29 27.46, 8.27)	(40.33, 24.74, 7.51)	(40.33, 24.74, 7.51) ₂	(41.29 27.46, 8.27) ₁
(24.17, 54.79, 7.59)	(74.35, 46.37, 10.08)	(54.69, 55.8, 5) ₁	(65.12, 47.3, 5.65) ₁
(65.12, 47.3, 5.65)		(56.56, 73.38, 5.04) ₁	(24.17, 54.79, 7.59) ₁
(54.69, 55.8, 5)		(46.49, 67.41, 5) ₁	(74.35, 46.37, 10.08) ₂
(56.56, 73.38, 5.04)			
(46.49, 67.41, 5)			

3.3. A mutáció operátor

A mutáció valójában nem egy, hanem három különböző műveletet takar. Az egyik az, amikor a körök száma változik meg – nő, vagy csökken – a másik az amikor a kör sugara módosul, a harmadik amikor egy kör középpontja mozdul el. Ezek egymástól külön történnek, tehát lehet olyan utód is, amelyen két, vagy három jellemző is megváltozott. Az egyedek kiválasztása a keresztezésnél leírt módon történik, de a háromféle mutációhoz külön-külön küszöbszám tartozhat.

A körök számának módosítása esetén $1/2 - 1/2$ valószínűséggel növeljük, illetve csökkentjük a körök számát. A konfiguráció bővítésekor egy új számhármask kerül be a kör listába. Amennyiben az ez által meghatározott körlap belemetsz valamelyik másikba, vagy a körgyűrűbe, akkor újabb próbálkozás következik. Ez addig folytatódik, amíg nem lesz sikeres utód, vagy a kísérletek száma el nem ért egy maximális értéket. A kör eltávolításakor egy véletlenszerűen választott hármask kerül törlésre, amely azonban nem lehet az körgyűrűt reprezentáló hármaskok egyike sem.

A kör méretének megváltoztatása esetén egy – a listából véletlenszerűen választott – kör sugara módosul. Ez lehet akár a körgyűrű valamely köre is. A módosítás mértékét egy adott intervallumon belüli véletlenszerűen meghatározott érték határozza meg.

Egy körlap pozíciójának megváltoztatása hasonlóan történik, mint a sugár átméretezése (mind az x , mind az y koordináta megváltozhat), de ez nem érintheti a körgyűrűt alkotó köröket.

Az utóbbi két esetben nem ismételjük a módosítást, ha az nem sikeres. A 3. ábra a mutáció mindhárom formájára mutat egy-egy példát.



3. ábra: Példa mutációra. Az ábrán látható képek rendre: eredeti kép (szülő), körlapok számának csökkentése, növelése, mozgatása, átméretezése.

4. Hiba mérés

A képek hibáját relatív átlagos hibaszázalékban adjuk meg (RME) [6]. Ennek számítása a következő formula alapján történik:

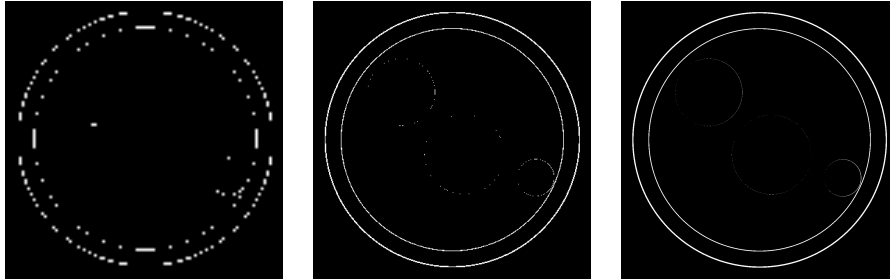
$$RME = \frac{\sum_i |f_i^o - f_i^r|}{\sum_i f_i^o} \cdot 100\%, \quad (2)$$

ahol az f_i^o és az f_i^r az eredeti illetve a rekonstruált kép i -edik pixelét jelenti. Tehát az RME érték az eredeti képen lévő fehér pontokhoz viszonyított hibát adja meg. Minél kisebb ez az érték, annál jobb a rekonstrukció. Érdeemes megjegyezni, hogy amennyiben a rekonstruált képen több hibásan objektumpontnak titulált pont van, mint az eredeti képen, akkor az RME 100%-nál nagyobb értéket is felvehet. Ez azt eredményezi, hogy olyan képek esetén, amelyekben kevés objektumpont (fehér pont) található számszerűen kevesebb pixeleltérés is nagyobb RME értéket fog eredményezni. Ez magyarázhatja azt is, hogy a kevesebb belső kört tartalmazó képek esetén magasabb RME értékeket tapasztaltunk (5. ábra).

Megvizsgáltuk, hogy az ismert hibamérték mennyire függ az alkalmazott felbontástól. Ezt jelen esetben könnyen megtehetjük, mert mind az eredeti, mind a rekonstruált kép előállítható az őket leíró konfigurációkból folytonos képként. Ezekon különböző mintavételezéseket alkalmazva más-más felbontású diszkrét képet kaphatunk.

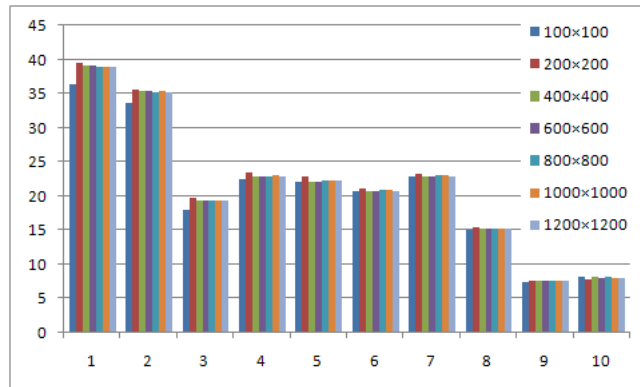
A 4. ábrán látható egy példa, melyben a rekonstruált és az eredeti kép különbsége látható 100×100 , 200×200 és 1600×1600 -as felbontásban. A képek azonos méretre lettek hozva, de így is megfigyelhető, hogy az egyre nagyobb felbontás egyre jobban összeköti a szétszakadozott görbéket, azonban így a hibásan detektált objektum pixelek száma sok esetben növekszik.

A 5. ábra grafikonja azt mutatja, hogy hogyan változik az RME érték a különböző felbontásokon. Egytől tíz darab belső kört tartalmazó képeket vizsgáltunk, minden körszámhoz öt különböző tesztképet, valamint ezek egy-egy rekonstruált képét használtuk. A grafikonon ezen öt-öt eset átlagait tüntettük fel a különböző felbontásoknak megfelelően. Amint az leolvasható tíz esetből nyolcszor a 200×200 -as méret esetén tapasztaltuk a legmagasabb RME értékeket. Azonban ez az eltérés nem tekinthető szignifikánsnak, ezért egyik felbontás sem tekinthető jobbnak, vagy rosszabbnak ebből a szempontból. Itt jegyeznénk meg, hogy ez



4. ábra: Különböző felbontású különbség képek, azonos méretre hozva. A felbontás rendre: 100×100 , 200×200 és 1600×1600 , a hozzájuk tartozó RME értékek pedig: 6,5; 7,8 és 7,9.

volt a vetület számítás felbontása is, tehát a rekonstrukcióhoz négy irányból vettünk 200-200 vetítősugarat. Habár ez a felbontás szolgáltatja a legnagyobb hibaértékeket, mégis ezt fogjuk használni a dolgozat további részeiben, mert így eredményeink összehasonlíthatóak lesznek a más, nem folytonos reprezentációt használó algoritmusokkal is. A továbbiakban ahol külön nem jelezzük, ott az RME értékek a kép valós felbontásával lettek számítva.



5. ábra: Az átlagos RME értékek (függőleges tengely) a képet alkotó körök számának függvényében (vízszintes tengely), különböző felbontásokon.

5. Eredmények

5.1. Az algoritmus paraméterezése

A kiindulási populáció legenerálására a DIRECT-rendszer [3] generáló algoritmusát használtuk. Az egyes küszöbértékek a keresztezéshez 0.05, a mutáció esetén a körök számának változtatásához 0.05, a kör méretének, illetve pozíciójának változtatásához 0.5-0.5 voltak. A keresztezés esetén 50, a körök számának növelésekor 1000 volt a megengedett próbálkozások száma. Mivel ezen kísérleteinkben nem használtunk prior információt, ezért a λ_1 és λ_2 értékei 1, illetve 0 voltak. Annyi kikötés volt csak, hogy a rekonstruált képen legalább egy, de legfeljebb tizenkét körlap található. Az algoritmus végfeltételeként az iterációk számát szabtuk meg. 200×200 -as méretű képekkel dolgoztunk, melyek vetületeit 0, 45, 90 és 135 fokból képeztük, a vetítésugarak száma minden irányból 200-200 volt.

Tesztjeinkben ötven darab képet rekonstruáltunk az algoritmus különböző beállításával. Ezen tesztképek között öt-öt darab volt az 1, 2, ..., 10 körlapot tartalmazó kép.

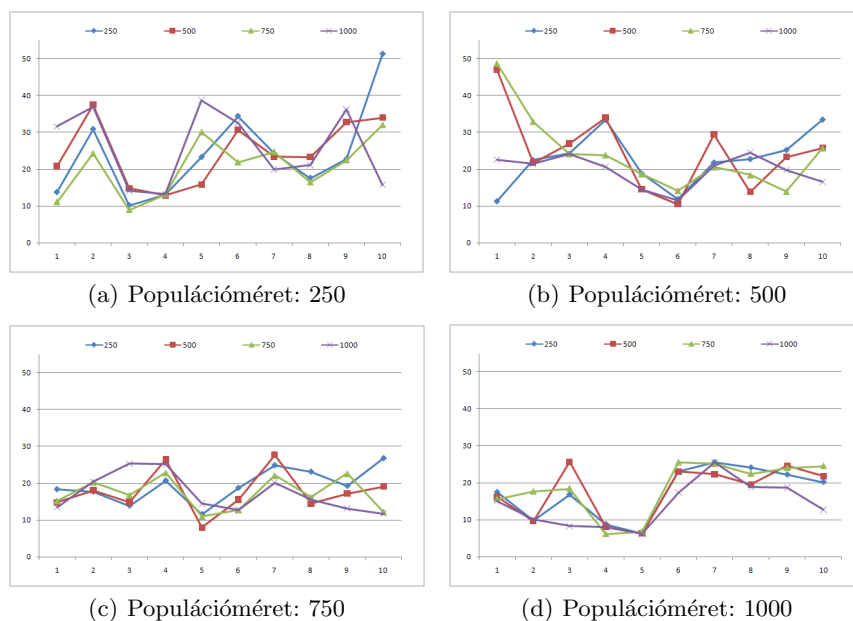
Kísérleteink során a generációk száma 250, 500, 750 és 1000 volt. A kezdő populáció mérete szintén 250, 500, 750 és 1000 voltak. Ezen halmazok mind a tíz féle körszámot tartalmazó képből egyforma arányban tartalmaztak (tehát például 250-es méret esetén 25 darab egy, 25 darab kettő, ..., 25 darab tíz korongot tartalmazó volt).

5.2. Kísérleteink eredménye

A rekonstrukciók RME értékeit a 6. ábra mutatja. Az egyes grafikonok rendre a 250, 500, 750 és 1000 egyedből álló populációk esetén számított átlagértékeket mutatják. Ezen átlagokat 5-5 azonos számú kört tartalmazó kép egy-egy rekonstrukciójának RME értékeinek átlagaként képeztük. A grafikonokról leolvasható, hogy egy adott populáció méret esetén nincs igazán különbség a különböző generációs szám esetén kialakult hibaértékek között, hiszen ezek a görbék többnyire együtt mozognak. Nagyobb eltérést tapasztalunk, ha a grafikonokat egymással hasonlítjuk össze. Ekkor azt tapasztaljuk, hogy ugyan kis mértékben, de a nagyobb populáció méret kisebb RME-t eredményez.

Sajnos nincs lehetőség eredményeink korrekt összevetésére más módszerekkel, mert ilyen jellegű képekkel csak a már említett [6] munkában foglalkoztak, azonban ott csak egyetlen képre közöltek RME hibaértéket. Ott egy négy körlapot tartalmazó képet rekonstruáltak pixel alapú rekonstrukcióval, melyhez nyolc vetületet használtak. Ennek RME hibája 12,57% volt.

A 2. táblázat azt mutatja, hogy a rekonstruált képeken levő körök száma hány esetben nem egyezik meg az eredeti képen levő körök számával. Minden esetben 50 darab kép rekonstrukcióját végeztük, így például az 5-ös érték 10%-os eltérést jelent. A sorokban az egyes generációs számokhoz, míg az oszlopokban egy adott populáció mérethez tartozó értékek láthatók. Látható, hogy mind a populáció méret, mind a generációk számának növelése kedvezően hat a megfelelő számú



6. ábra: A rekonstrukciók hibája. A grafikonok az egyes tesztek RME hibáit mutatják a különböző populációméretek esetén. Az egyes görbék a generációk számának megfelelő RME értékeket ábrázolják a körlapok számának függvényében. (A különböző színű görbék különböző generáció számokat jelentenek: kék – 250, piros – 500, zöld – 750, lila – 1000)

körlapot tartalmazó eredménykép megtalálására. Azonban mind a kettőnek a növekedésekor jelentősen nőhet az algoritmus futási ideje.

Összehasonlításként megemlítjük, hogy a korábbi kísérleteinkben [1] vizsgáltuk, hogy a vetületi információkat felhasználva döntési fákra alapuló tanulással meghatározható-e a képen található körlapok száma. Ekkor tíz osztályra a legjobb osztályozás hibája is 50% körül volt. Ha ezt összevetjük a most kapott 10% körüli eredménnyel, akkor elmondhatjuk, hogy az előbbieken bemutatott evolúciós algoritmus megfelelhet akár előfeldolgoásként is más rekonstrukciós algoritmusokhoz.

Megvizsgálva az eredmény képeket arra jutottunk, hogy a rekonstrukciós hiba nagy részét a körgyűrű teszi ki. Erre látható egy példa a 7. ábrán. Ebből arra következtetünk, hogy a későbbiekben érdemes lehet a rekonstrukciót úgy kiegészíteni, hogy az evolúciós algoritmus megoldását alapul véve egy egyéb kereső módszerrel csak a körgyűrű méretét változtatva tovább javítjuk a fitness értéket.

A futási időkre vonatkozóan a 3. táblázatban látható néhány eredmény. Ezek az adatok csak tájékoztató jellegűek, nem végeztünk még pontos méréseket, valamint a jelenlegi programkód sem sebességre lett optimalizálva. Azonban

2. táblázat: Az eredeti és a rekonstruált képeken levő körlapok számának eltérése különböző méretű populációk (oszlopok) és generációs számok esetén (sorok).

	250	500	750	1000
250	22	19	12	12
500	16	17	8	15
750	13	11	5	6
1000	14	8	6	5



7. ábra: Eredeti és rekonstruált kép, valamint a kettő különbsége. A rekonstruált kép RME értéke 26,6.

az leolvasható, hogy a generáció szám növekedésével a futási idők sokkal gyorsabb ütemben növekednek, mint a populációméret növelésével. Az egyes futási időket egy 1.86 GHz órajelű Intel Xeon processzorral és 4 GB memóriával rendelkező számítógépen mértük, a program a négy processzormagból csak egyet használt. Továbbá a méréseinkből az is kiderül, hogy a legtöbb időt a az egyes egyedek vetületi értékeinek kiszámítása veszi igénybe. Mivel ez a folyamat jól párhuzamosítható, ezért feltehetően a mind a négy processzormag, vagy esetleg egy megfelelő grafikus processzor alkalmazásával a futási idők jelentősen csökkenthetők lesznek.

3. táblázat: A rekonstrukció futási ideje másodpercben kifejezve egy 12 körlapot tartalmazó kép esetén a különböző populációméret (oszlopok) és generációs számok esetén (sorok).

	250	500	750	1000
250	2	3	5	7
500	3	6	12	22
750	5	17	85	68
1000	24	44	146	89

6. Összefoglalás, további tervek

Eredményeinkből arra következtethetünk, hogy nem érdemes túl nagyra választani sem a populáció méretet, sem pedig a generációk maximális számát. Célszerű

lehet 750 egyedből álló populációt és 750 iterációt használni, hiszen ekkor az algoritmus már hozzávetőleg jó RME értékű eredményképeket szolgáltat, valamint a körlapok száma is elég sok esetben megegyezik az eredeti képen található korongok számával. Továbbá ez esetben a műveletigény is kevesebb, tehát gyorsabban juthatunk eredményhez.

Az itt bemutatott módszerhez hasonló, más objektumokon alapuló képek rekonstrukciós algoritmusok főleg az ipari alkatrész tesztelésben, valamint a nemroncsoló anyagtesztelésben lehet hatékonyan használni.

Terveink közt szerepel az algoritmus átalakítása úgy, hogy az előzőekben bemutatott módon készítenénk egy *elfogadható* megoldáskezdeményt, majd iteratív módon módosítanánk az ezt alkotó körök méretét és pozícióját.

Céljaink között szerepel a rekonstrukció futási idejének csökkentése a vetületképzés és az egyedkiértékelés párhuzamosításával. Ennek megvalósítása elengedhetetlen feltétele az evolúciós algoritmus optimális paraméterezésének megtalálásához is, hiszen ez egy igen összetett, nehezen automatizálható időigényes, regeteg tesztelést igénylő feladat.

További lehetséges továbbhaladási irány lehet az algoritmus kiterjesztése három dimenzióba, valamint más, hasonlóan jól reprezentálható objektumokból álló képek vizsgálata (ilyenek lehetnek például a konvex sokszögek).

Irodalom

1. Balázs, P., Gara, M.: An evolutionary approach for object-based image reconstruction using learnt priors. *Lecture Notes in Comput. Sci.*, **5575** (2009) 520–529
2. Bäck, Th., Fogel, D.B., Michalewicz, T.(eds.): *Evolutionary Computation 1*. Institute of Physics Publishing, Bristol and Philadelphia, 2000.
3. <http://www.inf.u-szeged.hu/~direct>
4. Herman, G.T., Kuba, A. (eds.): *Discrete Tomography: Foundations, Algorithms and Applications*. Birkhäuser, Boston, 1999
5. Herman, G.T., Kuba, A. (eds.): *Advances in Discrete Tomography and its Applications*. Birkhäuser, Boston, 2007
6. Kiss, Z., Rodek, L., Kuba A.: Image reconstruction and correction methods in neutron and X-ray tomography *Acta Cyb.*, **17** (2006) 557–587