

AUKŠTOS KOKYBĖS SKAITMENINIŲ SPALVŲ ATSPALVIŲ KOMPIUTERINIS GENERAVIMAS

Alfonsas Misevičius, Jonas Blonskis, Vytautas Bukšnaitis,
Evelina Stanevičienė, Tautvydas Želvys

Multimedijos inžinerijos katedra, Kauno technologijos universitetas,
Studentų g. 50–400/416a, LT–51368 Kaunas,
alfonsas.misevicius@ktu.lt, jonas.blonskis@ktu.lt, vytautas.buksnaitis@ktu.lt,
evelinastaneviciene@yahoo.com, gtautvis@gmail.com

Anotacija. Šis straipsnis skirtas aukštos kokybės (galimai optimalių) skaitmeninių spalvų atspalvių formavimui automatinio (t. y. kompiuteriniu) būdu. Straipsnio pradžioje formaliai apibrėžiamas spalvų atspalvių generavimo matematinis modelis — vadinamasis pilkų šablonų formavimo (PŠF) uždavinys. Po to aptariamas euristinis optimizavimo algoritmas PŠF uždavinio sprendimui. Tai nėra tikslus algoritmus, tačiau jis įgalina surasti nepriekaištingos kokybės (galimai optimalius ar jiems artimus) atspalvius per priimtina skaičiavimo laiką, nors PŠF uždavinys ir priklauso sudėtingų, NP-sunkių optimizavimo uždavinių kategorijai. Straipsnyje taip pat yra pateikti kompiuterinių eksperimentų rezultatai, gauti, atlikus tyrimus su didelės apimties PŠF uždavinio testiniais pavyzdžiais. Pateikiamos ir gautų sprendinių grafinės iliustracijos — tai leidžia įsitikinti sugeneruotų šablonų (vaizdų) kokybiškumo lygiu.

Pagrindiniai žodžiai: skaitmeninių spalvų atspalvių (vaizdų) sudarymas, pilkų šablonų formavimo uždavinys, kūrybinė multimedija, optimizavimas, euristiniai algoritmai.

Įvadas

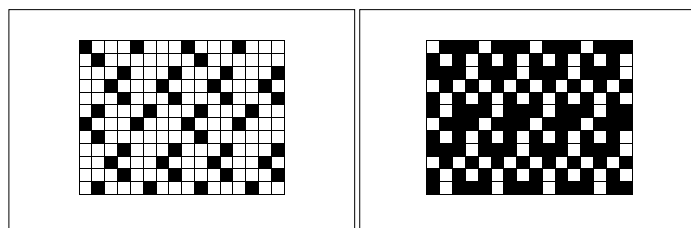
Straipsnyje nagrinėjama, kaip automatinio (t. y. kompiuteriniu) būdu gali būti generuojami aukštos kokybės (galimai optimalūs) spalvų pereinamieji atspalviai (tonai). Remiamasi racionali prielaida, jog skaitmeninių spalvų atspalvių kompiuterinis formavimas gali būti realizuojamas atitinkamo formalaus diskretinio matematinio modelio – vadinamojo pilkų šablonų formavimo (generavimo) (PŠF) uždavinio (*angl.* grey pattern problem) pagalba. Šiuo atveju skaitmenizuotų spalvų atspalvių aukšta, ideali kokybė ir optimalumas išvedami tiesiog iš PŠF uždavinio analitinių sprendinių optimalumo (žr. žemiau).

PŠF uždavinys vaizdų pilkų atspalvių sudarymo kontekste buvo apibrėžtas Taillard'o 1995 metais (Taillard, 1995). Šis uždavinys, nors ir formuluojamas pagal tradiciją tik kaip skirtas pilkų (t. y. juodos spalvos) skaitmeninių atspalvių (tonų) generavimui (*angl.* digital halftoning) (Ulichney, 1987; Lau, Arce, 2008), gali būti pritaikomas iš esmės bet kuriai turimos spalvų paletės diskretinei spalvai (nebūtinai monochromatinei spalvai, bet ir bet kuriai galimai užkoduoti skaitmeniniu būdu spalvos reikšmei (pvz., spalvos reikšmei, koduojamai panaudojant spalvų skaitmeninio kodavimo modelį RGB (http://en.wikipedia.org/wiki/RGB_color_model))). PŠF uždavinio tiksliai formuluotė remiasi žemiau pateikiamais samprotavimais.

1. Skaitmeninių spalvų atspalvių (pilku šablonų) sudarymo modelis

Sakykime, kad yra duotas stačiakampis tinklelis („gardelė“, matrica), kurio matmenys yra $n_1 \times n_2$; kitaip tariant, tinklelį sudaro $n = n_1 \times n_2$ reguliariai išdėstyti „mazgų“ – mažų identiško dydžio kvadratėlių (arba tiesiog diskretinių taškų) (taip pat žr. (Taillard, 1995)). Tinklelio kvadratėlių eilučių skaičius yra lygus n_1 , o stulpelių – n_2 . Neprarandant universalumo, laikoma, kad taškai yra arba baltos spalvos, arba juodos (baltą spalvą tikslinga asocijuoti su fono spalva (*angl.* background), o juodą spalvą – su pagrindine (bazine) spalva (*angl.* foreground)). Tarkime, jog m ($m \leq n$) yra juodos spalvos taškų skaičius; tuomet baltos spalvos taškų skaičius yra $n - m$. Išdėstant reguliaria tvarka tokius iš juodų ir baltų kvadratėlių sudarytus tinklelius, gaunami „pilki šablonai“ („freimai“) (*angl.* grey frames) (žr. 1 pav.). Šablono pilkumo tankumas (intensyvumas, sodrumas) (*angl.* grey density) yra lygus santykiui $\frac{m}{n}$ ($0 \leq \frac{m}{n} \leq 1$), čia $n = n_1 \times n_2$. Suprantama, kad kuo mažesnė reikšmė m (esant fiksuotam n), tuo „silpnesnis“ šablono pilkumas ir tuo artimesnė gautoji spalva baltai (arba atskiru atveju yra tiesiog balta, jeigu $m = 0$); ir atvirkščiai, kuo didesnis skaičius m , tuo labiau pasireiškia artimos juodai (arba tiesiog juodos, kai $m = n$) spalvos efektas.

Turint n vienodo dydžio kvadratėlių (taškų), iš kurių m yra juodi, o likusieji – balti, keliamas tikslas išdėstyti/paskirstyti tinklelyje (matricoje) juodus arba baltus kvadratėlius taip tolygiai, kaip tik tai gali būti įmanoma. Kitaip išsireiškiant, reikia suformuoti šablono piešinį su kiek įmanoma tolygesniu, subtilesniu juodos (baltos) spalvos taškų išsklaidymu. Taigi, turi būti sudarytas kuo taisyklingesnis, tobulesnis pilkų atspalvių šablonas, imituojantis pageidaujamo tankumo (intensyvumo) pilką spalvą. Toks pat principas gali būti pritaikytas ir gaunant bet kokių kitų (tiek grynujų, tiek kombinuotųjų) spalvų tarpinius atspalvius¹.



1 pav. Pilkų atspalvių šablonų pavyzdžiai.

¹ Galima realiai suformuoti iki n duotosios pagrindinės/bazinės spalvos skaitmenizuotų diskretinių atspalvių (tonų), įskaitant pačią pagrindinę spalvą (spalvą be atspalvių). Spalvai be atspalvių atitinka atvejis $m = n$, čia n gali būti parenkamas vartotojo (dizainerio) nuožiūra. Programavimo patogumo dėlei rekomenduotina, jog n būtų lygus skaičiui 2, pakeltam tam tikru laipsniu λ , t. y. 2^λ , pvz., 2^6 , 2^8 ir pan. Akivaizdu, jog turint q pagrindinių/bazinių (nesutampančių su fono spalva) spalvų, maksimalus jų galimų atspalvių skaičius būtų lygus nq ($= 2^\lambda q$). (Šiuo atveju reikšmei λ padidėjus vienetu, atspalvių skaičius padvigubėtų.)

Esant galimybei turėti ne vieną, bet q foninių spalvų, maksimalus įmanomas „susintetintų“ (virtualių) spalvų skaičius galėtų būti padidintas iki $\frac{nq(q-1)}{2}$. Tokiu būdu atsivertų dar didesnės galimybės, suteikiant kompiuteriniams grafiniams vaizdams naujų spalvinių niuansų ir kiekybiškai bei kokybiškai praplečiant vaizdų spalvų gamą ir įvairovę.

2. Pilkų šablonų formavimo uždavinys

2.1. Pilkų šablonų formavimo uždavinio matematinė formuluotė

$$a_{ij} = \begin{cases} 1, & 1 \leq i \leq m \wedge 1 \leq j \leq m, \quad i, j = 1, 2, \dots, n, \quad m \leq n. \\ 0, & \text{priešingu atveju} \end{cases} \quad (1)$$

Iš tiesų, ši matrica, sprendžiant PŠF uždavinį, tampa apleamai nebereikalinga. Pakanka tik žinoti dydžio (parametro) m konkrečią reikšmę. Tuo tarpu matricos $B = (b_{kl})_{n \times n}$ reikšmės apibrėžiamos visoms galimoms taškų poroms tam tikru specialiu būdu, būtent:

$$b_{kl} = b_{(r-1)n_2+s \quad (t-1)n_2+u} = \omega_{rstu}, \quad (2)$$

$$\omega_{rstu} = \max_{w_1, w_2 \in \{-1, 0, 1\}} \frac{1}{(r-t+w_1n_1)^2 + (s-u+w_2n_2)^2},$$

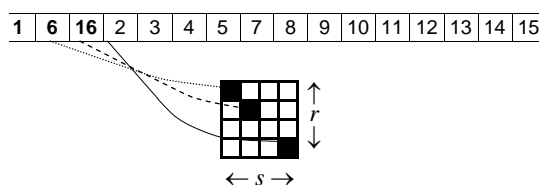
čia $k, l = 1, \dots, n$, $r, t = 1, \dots, n_1$, $s, u = 1, \dots, n_2$, $n_1 \times n_2 = n$.

Reikšmės ω_{rstu} gali būti traktuojamos taip. Sakykime, kad stačiakampio tinklelio mazguose yra n įelektrintų dalelių (elektronų), tarp kurių veikia fizikinės, tiksliau, elektrinės atostūmio jėgos. Tuomet reikšmę ω_{rstu} galime interpretuoti kaip dydį, proporcingą atostūmio jėgai tarp elektronų i ir j ($i, j = 1, \dots, n$), esančių tinklelio pozicijose $k = p(i)$ ir $l = p(j)$, kai pozicijų k ir l koordinatės yra atitinkamai (r, s) ir (t, u) ($r, t = 1, \dots, n_1$, $s, u = 1, \dots, n_2$).

Nagrinėjamu atveju p žymi perstatymą, o $p(i)$, $p(j)$ – perstatymo atitinkamus elementus. Bet kuris korektiškas perstatymas laikytinas PŠF uždavinio galimu sprendiniu. Tuomet tikslas yra surasti geriausią galimą, optimalų sprendinį p , t. y. perstatymo elementus $p(1), p(2), \dots, p(m)$ ($1 \leq p(i) \leq n$, $i = 1, 2, \dots, m$, $m \leq n$) ir tokius, kad suma $\sum_{i=1}^m \sum_{j=1}^m b_{p(i)p(j)}$ (traktuojama kaip PŠF uždavinio tikslo funkcija $z(p)$) būtų minimali (kiek įmanoma mažesnė), t. y.:

$$z(p) = \sum_{i=1}^m \sum_{j=1}^m b_{p(i)p(j)} \rightarrow \text{minimumas}. \quad (3)$$

Ši tikslo funkcija (TF) ir apibrėžia generuojamų spalvų atspalvių kokybiškumą, optimalumą. Optimalią (mažiausią) tikslo funkcijos reikšmę atitinkančio analitinio sprendinio (perstatymo) p^* i -tasis ($i \leq m$) elementas $p^*(i) = (r-1)n_2 + s$ nurodo, kurioje konkrečioje šablono tinklelio pozicijoje turi būti padėtas i -tasis juodas kvadratėlis. (Elementai $p^*(m+1), p^*(m+2), \dots$ ignoruojami.) Kvadratėlio pozicijos koordinatės, t. y. atitinkamos tinklelio eilutės ir stulpelio indeksai (r, s) apskaičiuojami pagal formules: $r = \lfloor (p^*(i) - 1) / n_2 \rfloor + 1$, $s = ((p^*(i) - 1) \bmod n_2) + 1$, $i \leq m$ (žr. 2 pav.). (Operacija \bmod žymi liekaną, gaunamą dalijant vieną skaičių iš kito.)



$$n_1 = 4, n_2 = 4, n = n_1 \times n_2 = 16.$$

2 pav. Analitinio sprendinio (perstatymo) ir grafinio tinklelio pozicijų (juodų taškų) susiejimas.

2.2. Pilkų šablonų formavimo uždavinio pavyzdžiai

Varijuojant reikšmėmis n_1 , n_2 , m , galima sudaryti labai daug ir įvairių PŠF uždavinio pavyzdžių (egzempliorių). Tikslinga ir patogiu turėti fiksuotas, lygias parametrų n_1 , n_2 reikšmes; tuo tarpu reikšmė m , laikytina šioje situacijoje spalvos atspalvio reikšme (eilės numeriu), yra besikeičianti: ji kinta nuo 1 iki n ($n = n_1 \times n_2$). (Jeigu $m = 0$, tai, kaip aukščiau pastebėta, gaunama tiesiog fono spalva.) Svarbu pabrėžti, jog esamoms fiksuotoms parametrų n_1 , n_2 reikšmėms nėra būtina PŠF uždavinį spręsti n kartų. Pakanka išspręsti tik $n/2$ uždavinio egzempliorių ($m = 1, 2, \dots, n/2$). Taip yra todėl, kad šablono atspalviai, pradedant $(n/2 + 1)$ -uoju ir baigiant $(n - 1)$ -uoju, gali būti lengvai gaunami invertuojant šablono fono ir pagrindinę spalvas.

Vienas iš pavyzdinių PŠF uždavinio atvejų yra taškinis tinklelis (matrica), kurį sudaro 8 taškų eilutės ir stulpeliai ($n_1 = n_2 = 8$). Šiuo atveju bendras tinklelio taškų skaičius (n) yra lygus 64 (2^6). Taigi, atspalvio reikšmė (m) gali kisti nuo 1 iki 64 (64 yra duotu atveju maksimalus galimas atspalvių skaičius). Šio tipinio uždavinio pavyzdžio (atvejo $m = 13$) duomenis (matricas A , B) galima rasti internete, KP uždavinio testinių pavyzdžių duomenų bazėje (bibliotekoje) QAPLIB (Burkard ir kt., 1997) (prieiga per internetą: <<http://www.seas.upenn.edu/qaplib/>>). Uždavinio pavyzdžio simbolinis pavadinimas yra Tai64c. Toje pat duomenų bazėje galima rasti ir pavyzdį Tai256c ($n = 256$ (2^8), $n_1 = n_2 = 16$, $m = 92$). Kaip matyti, čia uždavinio apimtis (n) išauga 4 kartus, tuo pačiu atitinkamai padidėja ir galimų atspalvių kiekis, o tai savo ruožtu įgalina sukurti geresnės kokybės spalvų tonus.

Plečiantis kompiuterinės technikos galimybėms (didėjant centrinių procesorinių įrenginių spartai, augant operatyviosios atminties dydžiui), tampa įmanoma spręsti vis stambesnes apimties PŠF uždavinius. Šio straipsnio bendraautorių A. Misevičiaus ir T. Želvio dėka sukonstruotas rekordinio dydžio PŠF uždavinio pavyzdys, kurio apimtis (n) lygi 1024 (2^{10}) ($n_1 = n_2 = 32$). (Tuo pačiu tai yra ir pats didžiausias kada nors tirtas kvadratinio paskirstymo uždavinio atvejis.) Be to, pastaruoju atveju parametro m reikšmė nėra surišta su konkrečiu pavyzdžiu, ją pasirenka savo nuožiūra pats tyrėjas/vartotojas kompiuterinės programos vykdymo metu.

3. Pilkų šablonų formavimo uždavinio sprendimo euristinis algoritmas

PŠF uždavinio, kaip vieno iš charakteringų kombinatorinio optimizavimo uždavinių sprendimui galima adaptuoti platų spektrą euristinių algoritmų, pradedant vieną sprendinį naudojančiais lokalsios paieškos algoritmais (Taillard, 1995; Drezner, 2006) ir baigiant

sprendinių populiacijomis operuojančiais metodais (Taillard, 1995; Taillard, Gambardella, 1997; Drezner, 2006). Galima taip pat taikyti ir hibridinius algoritmus (Drezner, 2006; Misevičius, 2006; Misevičius, 2011), integruojančius atskirų sprendinių lokalų pagerinimą ir sprendinių populiacijų evoliuciją. Kaip tik hibridinio optimizavimo panaudojimas PŠF uždaviniui yra viena iš, galima sakyti, „sėkmės istorijų“ efektyviai pritaikant euristinius metodus kombinatorinio tipo uždaviniams, koks yra PŠF uždavinys².

Kalbama konkrečiai apie hibridinį genetinį-evoliucinį algoritmą (HGEA), kuris yra detalai aprašytas A. Misevičiaus straipsnyje (Misevičius, 2011). Panašaus principo algoritmas anksčiau tirtas Z. Dreznerio darbe (Drezner, 2006). Šis algoritmas yra grindžiamas genetinėmis operacijomis (atranka, reprodukuavimu, mutavimu), o jo evoliucinis pobūdis pasireiškia tuo, jog įprastas sprendinių genetinio kryžminimo operatorius yra pakeistas nebūdinga genetiniams algoritmams specifine rekombinuojančia procedūra. Bendruoju požiūriu į šį algoritmą galima žvelgti kaip į sudėtinį procesą. Iš vienos pusės, galima išvelgti vadinamąją „interevoliuciją“, t. y. globalaus pobūdžio evoliucinį vyksmą; kita vertus, interevoliucija turi savyje ir „intraevoliuciją“ — vidinį, lokalų paieškos procesą. Interevoliucija siejasi su paieškos krypčių įvairoviškumu, diversifikuotumu, o intraevoliucijai yra savybingas paieškos koncentruotumas, intensifikacija.

procedure *hibridinis_genetinis_evoliucinis_algoritmas*;

.pradiniai duomenys: n – uždavinio apimtis (bendras taškų/kvadratėlių skaičius), m – „juodų“ taškų/kvadratėlių skaičius,

. PD – populiacijos dydis, N_{gen} – generacijų skaičius,

. N_{rek} – rekombinavimo pakartojimų skaičius, ES – entropijos slenkstis

.rezultatai: p^* – geriausias rastas sprendinys (perstatymas) tikslo funkcijos z atžvilgiu

```

for  $i := 1$  to  $m$  do for  $j := m + 1$  to  $n$  do  $tabu[i, j] := 0$ ;  $steko\_dydis := 0$ ; //  $tabu$ 
sarašo ir sarašo steko dydžio inicializacija
sukurti pradinę optimizuotą populiaciją  $P \subset \Pi_n$  dviem žingsniais:
(1) sugeneruoti populiacijos  $P$  sprendinius atsitiktiniu būdu;
(2) optimizuoti populiacijos  $P$  sprendinius, vykdant pagerinimo procedūrą
    intraevoliucija kiekvienam populiacijos sprendiniui;
    intraevoliucijos procedūra vykdoma, esant padidintam procedūros iteracijų
    skaičiui
for generacija := 1 to  $N_{gen}$  do begin .pagrindinis (generacijų) ciklas
    surūšiuoti populiacijos  $P$  narius, atsižvelgiant į jų kokybę;
    for rekombinavimo_iteracija := 1 to  $N_{rek}$  do begin .rekombinavimo ir intraevoliucijos procedūrų
    ciklas
        atrinkti "sprendinius-tėvus"  $p', p'' \in P$ ;
        sprendiniams  $p', p''$  vykdyti procedūrą rekombinavimas, gauti "sprendinį-palikuonį"
         $p'''$ ;
        sprendiniui  $p'''$  vykdyti pagerinimo procedūrą intraevoliucija, gauti (pagerintą) sprendinį
         $p^*$ ;
         $P := P \cup p^*$ 
    end;
    atnaujinti populiaciją  $P$ , paliekant joje  $PD$  geriausių sprendinių;
    jeigu populiacijos  $P$  entropija (sprendinių įvairovė) yra mažesnė negu
    entropijos slenkstis  $ES$ , tai įvykdyti populiacijos  $P$  restartą ("invazija")
    dviem žingsniais:
    (1) atlikti makromutavimą, t.y. visų populiacijos sprendinių mutavimą,
        išskyrus geriausią sprendinį;
    (2) optimizuoti populiacijos  $P$  sprendinius vykdant procedūrą intraevoliucija
     $p := \underset{p \in P}{\operatorname{argmin}}(z(p))$  mutuotam sprendiniui;
end.
    
```

Pastaba. Populiacijos entropija apskaičiuojama pagal formules, aprašytas A. Misevičiaus publikacijoje (Misevičius, 2011).

3 pav. Hibridinio genetinio-evoliucinio algoritmo pseudokodas.

² Nežiūrint to, jog PŠF uždavinys yra redukuotas, paprastesnis kvadratinio paskirstymo uždavinio specialus atvejis, jis vis dėlto yra NP-sunkus. Iš tiesų, PŠF uždavinys gali būti traktuojamas kaip kito kombinatorinio uždavinio, būtent komivojažieriaus uždavinio, kuris savo ruožtu yra NP-sunkus, bendresnis atvejis.

```

procedure intraevoliucija;
.pradiniai duomenys:  $p$  – esamas sprendinys (perstatymas),
 $\mu_{min}$ ,  $\mu_{max}$  – minimalus ir maksimalus mutavimo lygis (stiprumas),  $T$  –
iteracijų skaičius
.rezultatai:  $p^*$  – geriausias rastas sprendinys

vykdyti procedūrą randomizuota_tabu_paiška, esant pradiniam sprendiniui  $p$ , ir tegul gautas
sprendinys yra  $p^*$ ;
 $p := p^*$ ;  $p^* := p^*$ ;  $\mu := \mu_{min} - 1$ ;  $\mu$  yra esamasis mutavimo lygis
for iteracija := 1 to  $T$  do begin .pagrindinis intraevoliucinio proceso (iteratyviosios
randomizuotos tabu paieškos) ciklas
 $p := p^*$ ; // sprendinio-kandidato mutavimo procedūrai atrinkimas
if  $\mu < \mu_{max}$  then  $\mu := \mu + 1$  else  $\mu := \mu_{min}$ ; // mutavimo lygio pakeitimas
sprendiniui  $p$  vykdyti procedūrą mutavimas (esant mutavimo lygiui  $\mu$ ),
gaunamas mutuotas sprendinys  $p^*$ ;
mutuotam sprendiniui  $p^*$  vykdyti procedūrą randomizuota_tabu_paiška,
gauti naują (pagerintą) sprendinį  $p^*$ ;
if  $z(p^*) < z(p)$  then begin .išimenamas geriausias rastas sprendinys, atstatomas
minimalus mutavimo lygis
 $p^* := p^*$ ;  $\mu := \mu_{min} - 1$  end
end.

```

4 pav. Intraevoliucijos (iteratyviosios randomizuotos tabu paieškos) procedūros pseudokodas.

```

procedure randomizuota_tabu_paiška;
.pradiniai duomenys:  $n$  – uždavinio apimtis/taškų skaičius,  $m$  – „juodų“ taškų skaičius,  $B$  – PŠF uždavinio duomenų („astumų“) matrica,
 $p$  – esamas sprendinys,  $\tau$  – tabu paieškos gylis (apimtis),  $h_{min}$ ,  $h_{max}$  – apatinis ir
viršutinis tabu (draudimo) periodo rėžiai,
 $\alpha$  – randomizacijos koeficientas,  $\varpi$  – alternatyvaus intensifikavimo
periodo parametras
.rezultatai:  $p^*$  – geriausias rastas sprendinys

// tabu sąrašo paruošimas, pagalbinės atminties/masyvo  $C$  inicializacija
for  $i := 1$  to steko_dydis do tabu[stekas1[ $i$ ], stekas2[ $i$ ]] := 0;
for  $i := 1$  to  $n$  do begin  $c_i := 0$ ; for  $j := 1$  to  $m$  do  $c_i := c_i + b_{ip(j)}$  end;
 $p^* := p$ ;  $w := 1$ ;  $w^- := 1$ ; steko_dydis := 0; pagerinta := FALSE;  $h := \text{random}(h_{min}, h_{max})$ ;
while ( $w \leq \tau$ ) or (pagerinta = TRUE) then begin pagrindinis tabu paieškos ciklas
 $\Delta_{min} := \infty$ ;
for  $i := 1$  to  $m$  do
for  $j := m+1$  to  $n$  do begin .nagrinėjama  $m(n - m)$  esamo sprendinio „kaimynų“
 $\Delta := 2(c_{p(j)} - c_{p(i)} - b_{p(i)p(j)})$ ;
uždrausta := iif( $(\text{tabu}[i, j] \geq w)$  and  $(\text{random}() \geq \alpha)$ , TRUE, FALSE);
aspiracija := iif( $(z(p) + \Delta < z(p^*))$  and uždrausta, TRUE, FALSE);
if  $(\Delta < \Delta_{min})$  and not(uždrausta) or aspiracija then begin  $\Delta_{min} := \Delta$ ;  $k := i$ ;
 $l := j$  end end;
if  $\Delta_{min} < \infty$  then begin .esamas sprendinys pakeičiamas nauju; modifikuojamas tabu sąrašas, stekas ir pagalbinė atmintis/masyvas
 $p := p \odot \phi_{k1}$ ; tabu[ $k$ ,  $l$ ] :=  $w + h$ ; steko_dydis := steko_dydis + 1;
stekas1[steko_dydis] :=  $k$ ; stekas2[steko_dydis] :=  $l$ ;
for  $i := 1$  to  $n$  do  $c_i := c_i + b_{ip(k)} - b_{ip(l)}$  end;
pagerinta := iif( $\Delta_{min} < 0$ , TRUE, FALSE);
if (pagerinta = TRUE) and  $(w - w^- \geq \varpi)$  then begin
vykdyti procedūrą determinuota_lokalioji_paiška pradedant nuo sprendinio  $p$ ;
grąžinti (pagerinta) sprendinį  $p^*$ ;
 $p := p^*$ ;  $w^- := w$  end;
 $w := w + 1$ ; if  $z(p) < z(p^*)$  then  $p^* := p$  .išimenamas geriausias rastas sprendinys
end.

```

Pastabos. 1. Funkcijos $\text{random}(x, y)$, $\text{random}()$ grąžina pseudoatsitiktinį skaičių atitinkamai iš intervalų $[x, y]$, $[0, 1)$. 2. Funkcijos $\text{iif}(sąlyga, x, y)$ rezultatas yra lygus x , jeigu *sąlyga* patenkina, arba y , jeigu *sąlyga* nepatenkinta. 3. $\text{not}(\cdot)$ yra loginio neigimo funkcija. 4. Formalizuota išraiška $p \odot \phi_k$ reiškia, kad esamam sprendiniui (perstatymui) p yra taikomas pertvarkymo operatorius ϕ_k , kuris sukeičia vietomis k -tąjį ir l -tąjį esamo perstatymo elementus.

5 pav. Randomizuotos tabu paieškos procedūros pseudokodas.

```

procedure determinuota_lokalioji_paiška;
.pradiniai/bendri duomenys:  $n, m, B, p$  – esamas sprendinys,  $C$  – pagalbinė atmintis/masyvas tikslo funkcijos
pokyčiams,
.w – esamas tabu paieškos iteracijos numeris,  $h$  – esamas tabu (draudimo) periodas,
.tabu, steko_dydis, stekas1, stekas2 – tabu sąrašas, steko kintamieji
.rezultatai:  $p^*$  – naujas (pagerintas) sprendinys

```

```

repeat .ciklas vykdomas, kol randamas lokaliai optimalus sprendinys
 $\Delta_{min} := 0$ ;
for  $i := 1$  to  $m$  do
for  $j := m+1$  to  $n$  do begin  $\Delta := 2(C_{p(j)} - C_{p(i)} - b_{p(i)p(j)})$ ; if  $\Delta < \Delta_{min}$  then begin
 $\Delta_{min} := \Delta$ ;  $k := i$ ;  $l := j$  end end;
if  $\Delta_{min} < 0$  then begin
.esamas sprendinys pakeičiamas geresniu, modifikuojamas tabu sąrašas, stekas ir pagalbinė atmintis/masyvas
 $p := p \odot \phi_{k,l}$ ;  $tabu[k, l] := w + h$ ;  $steko\_dydis := steko\_dydis + 1$ ;
 $stekas_1[steko\_dydis] := k$ ;  $stekas_2[steko\_dydis] := l$ ;
for  $i := 1$  to  $n$  do  $c_i := c_i + b_{ip(k)} - b_{ip(l)}$  end
until  $\Delta_{min} = 0$ ;
 $p := p$ .

```

6 pav. Determinuotos lokalsiosios paieškos procedūros pseudokodas.

```

procedure mutavimas;
.pradiniai duomenys:  $n, m, p$  – esamas sprendinys,  $\mu$  – mutavimo lygis (stiprumas); rezultatai:  $p^*$  – mutuotas sprendinys

```

```

for mutavimo_iteracija := 1 to  $\mu$  do begin
sugeneruoti du atsitiktinius skaičius  $i, j$  ir tokius, kad  $1 \leq i \leq m, m + 1 \leq j \leq n$ ;
 $p^* := p \odot \phi_{i,j}$ ; //perstatyme  $p$  sukeičiami vietomis  $i$ -tasis ir  $j$ -tasis elementai
.TF reikšmė  $z$  atitinkanti sprendinį  $p^*$ , perskaičiuojama panaudojant  $O(m)$  operacijų, t.y.:
 $z(p^*) = z(p) + 2 \sum_{k=1, k \neq i} (b_{p^*(j)p^*(k)} - b_{p^*(i)p^*(k)})$ 
 $p := p^*$ 
end.

```

7 pav. Mutavimo procedūros pseudokodas.

```

procedure rekombinavimas;
.pradiniai duomenys:  $n, m, p', p''$  – sprendiniai-tėvai; rezultatai:  $p'''$  – sprendinys-palikuonis

```

```

 $\rho_1 := \max(3, \lfloor 0.05m \rfloor)$ ;  $\theta := \lfloor 0.02n \rfloor$ ; gauti aibes  $A_1, A_2$  atitinkamai iš tėvų  $p', p''$  pirmųjų  $m$ 
elementų; //  $|A_1| = |A_2| = m$ 
if not ( $A_1 \equiv A_2$ ) then begin // aibės  $A_1$  ir  $A_2$  yra skirtingos
 $A_3 := A_1 \cap A_2$ ;  $m^\wedge := |A_3|$ ;  $A_4 := A_1 \cup A_2$ ;  $A_5 := A_4 \setminus A_3$ ; if  $2m - m^\wedge + \rho_1 \leq n$  then  $\rho := \rho_1$ 
else  $\rho := n - 2m + m^\wedge$ ;
iš  $\rho$  perstatymo  $p'$  elementų, nesančių aibėje  $A_4$ , suformuoti aibę  $A_6$ ;
 $A_7 := A_5 \cup A_6$ ;  $m^* := |A_7|$ ; //  $m^\wedge \geq 0, |A_4| = 2m - m^\wedge, |A_5| = 2m - 2m^\wedge$ ,
 $m^* = \min\{2m - 2m^\wedge + \rho, n - m^\wedge\}$ 
suformuoti aibę  $A_8$  iš visų aibės  $A_3$  elementų ir  $m - m^\wedge$  atsitiktinių elementų iš aibės
 $A_7$ ;
if  $m^* - (m - m^\wedge) > 0$  then begin
gauti perstatymą  $\hat{p}$  iš aibės  $A_8$  elementų; gauti perstatymą  $\check{p}$  iš  $A_7 \setminus A_8$ ;
apjungti  $\hat{p}, \check{p}$  ir gauti perstatymą  $\hat{p}$ ;
if  $m^* - (m - m^\wedge) \leq \theta$ 
then sprendiniui  $\hat{p}$  vykdyti procedūrą determinuota_lokalioji_paiška, gauti
(pagerintą) sprendinį  $\hat{p}$ 
else sprendiniui  $\hat{p}$  vykdyti procedūrą randomizuota_tabu_paiška, gauti (pagerinta)
sprendinį  $\hat{p}$ 
.pagerinimas vykdomas tik aibės  $A_7$  elementams (aibės  $A_3$  elementai nenaudojami)
end
else sprendinys  $\hat{p}$  suformuojamas iš aibės  $A_8$  elementų
end
else begin // aibės  $A_1$  ir  $A_2$  yra ekvivalenčios
 $\rho_2 := \max(1, \lfloor 0.1m \rfloor)$ ; iš  $\rho_2$  perstatymo  $p'$  elementų, nesančių aibėje  $A_1$ , suformuoti aibę
 $A_9$ ;
pašalinti atsitiktinai parinktus  $\rho_2$  elementus iš aibės  $A_1$ ; gauti sprendinį  $\hat{p}$  iš aibių  $A_1, A_9$ 
elementų
end;
gauti  $\bar{p}$  iš aibės  $\{1, 2, \dots, n\} \setminus \{\hat{p}(1), \hat{p}(2), \dots, \hat{p}(m)\}$  elementų; apjungti  $\hat{p}, \bar{p}$  elementus, suformuoti
iš jų  $p'''$ .

```

8 pav. Sprendinių rekombinavimo procedūros pseudokodas.

Pagrindinės algoritmo HGEA struktūrinės dalys yra tokios: 1) pats hibridinis algoritmas, 2) intraevoliucijos komponentas, 3) sprendinių rekombinavimas. Savo ruožtu, intraevoliucijos komponentas apima randomizuotos tabu paieškos ir sprendinių mutavimo procedūras. Be to, randomizuotos tabu paieškos procedūroje dar yra determinuotos lokalsios paieškos subprocedūra.

Detalizuotam HGEA ir jo komponentų pateikimui naudojamės pseudokodu — formaliu aprašu panaudojant algoritmo pagrindinių dalių (etapų, žingsnių) conceptualaus aprašymo (pseudo)kalbą, primenančią populiarią programavimo kalbą Pascal. Šie pseudokodai yra pateikti 3–8 pav.

Vienas esminių algoritmo HGEA aspektų yra papildančių vienas kitą interevoliucinio ir intraevoliucinio procesų efektyvus suderinimas. Interevoliucinio proceso struktūra yra analogiška įprastai genetinio/evoliucinio algoritmo schemai. Iš pradžių suformuojama pradinė pagerinta populiacija, kurios dydis PD. Laikomasi nuostatos, jog svarbiau yra populiacijos narių kokybė (genetinės informacijos elitiškumas), o ne populiacijos dydis (genetinės medžiagos gausumas). Tolesnėje algoritmo vykdymo eigoje iteraciniu būdu atliekamas sprendinių-tėvų atrinkimas reprodukcijai, atrinktų tėvų genų kryžminimas, t. y. rekombinavimas bei populiacijos atnaujinimas, t. y. populiacijos narių atrinkimas kitai algoritmo iteracijai (generacijai). Kiekvienas po tėvų rekombinavimo gautas sprendinys yra optimizuojamas, panaudojant autonominę intraevoliucijos procedūrą. Sprendinių mutavimas vykdomas intraevoliucijos proceso metu. Papildomas visos populiacijos narių mutavimas (makromutavimas) vykdomas, kai populiacijos entropija tampa mažesnė už nustatytą entropijos slenkstį (ES). Algoritmo vykdymo trukmė valdoma vartotojo parenkamu algoritmo generacijų skaičiaus parametru (Ngen). Vietoje įprasto kryžminimo, pagrįsto tėvų chromosomų genų pozicijų išlaikymu, algoritme HGEA pasinaudota nestandartine genetinės informacijos rekombinavimo procedūra, kurioje operuojama ne tiesiogiai su perstatymais, bet su atitinkamomis aibėmis, sudarytomis iš perstatymų m pirmųjų elementų. Pats rekombinavimas remiasi aibių operacijomis (sąjunga, sankirta, atimtimi), be to, leidžiama infiltruoti „svetimus“ (ne tėvų) genus, t. y. genus, esančius pozicijose: $m + 1$, $m + 2$, ..., n (taip pat žr. (Drezner, 2006; Misevičius, 2011)). Intraevoliucinio sprendinių pagerinimo proceso vaidmenyje yra iteracinės tabu paieškos algoritmas, kuriame apjungta randomizuotos tabu paieškos procedūra ir adaptyvus sprendinių mutavimas. Specifinė PŠF uždavinio struktūra leidžia tabu/lokalsios paieškos proceso sudėtingumą sumažinti nuo $O(n_3)$ (tai yra standartinio KP uždavinio atvejis) iki $O(mn)$ (neskaitant duomenų matricos įvedimo).

4. Kompiuteriniai eksperimentai

4.1. Kompiuterinių eksperimentų vykdymas

Kompiuterinis skaitmenizuotų spalvų atspalvių šablonų generavimas buvo vykdytas panaudojant x86 serijos asmeninius kompiuterius su greitaeigiais (3 GHz ir didesnio dažnio) centriniais procesoriniais įrenginiais ir MS Windows šeimos operacine sistema bei daugiafunkciniu programų paketu MS Office. Eksperimentuose buvo naudojamas aukščiau minėtas PŠF uždavinio testinis pavyzdys, kuris sukurtas panaudojant T. Želvio sudarytą

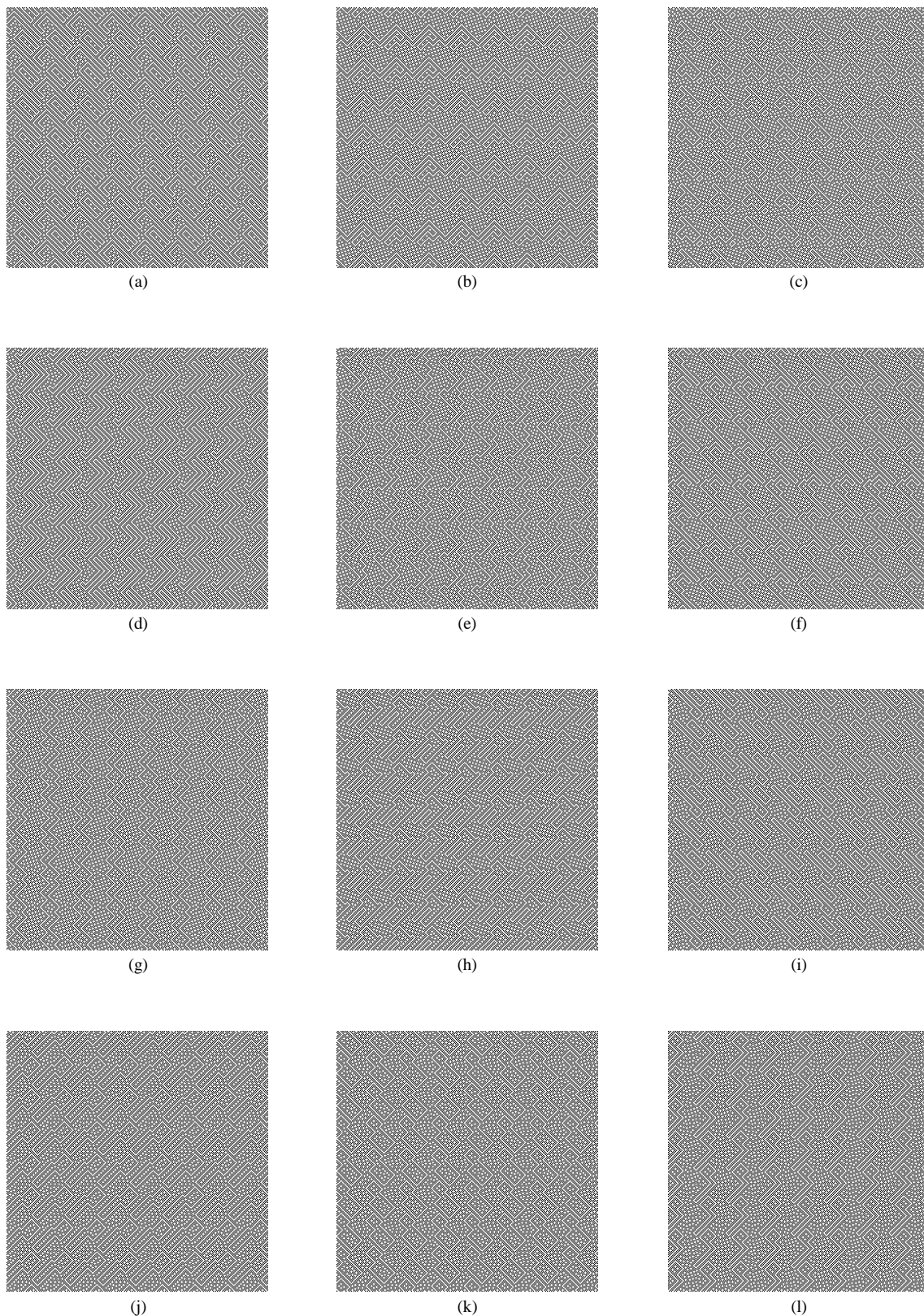
programą. Uždavinio apimtis (n) lygi 1024; šablonus sudarančių tinklelių matmenys yra 32×32 ($n_1 = n_2 = 32$). Sugeneruoto uždavinio pavyzdžio sprendimo euristinis hibridinis genetinis-evoliucinis algoritmas (HGEA) buvo programuojamas pasitelkiant į pagalbą programavimo kalbą Free Pascal, o analitinių sprendinių konvertavimui į grafinius šablonus ir vizualiniam pateikimui, taip pat vartotojo sąsajos realizavimui panaudota kalba C ir integruota programų kūrimo aplinka Qt Creator. Atliekant eksperimentus naudotos tokios konkrečios algoritmo HGEA valdymo parametrų reikšmės: $PD = 10$; $N_{rek} = 10$; $ES = 0,0001$; $\mu_{min} = \lfloor 0.05n \rfloor$; $\mu_{max} = \lfloor 0.07n \rfloor$; $T = 50$; $\tau = 0.5n$; $h_{min} = \lfloor 0.03n \rfloor$; $h_{max} = \lfloor 0.05n \rfloor$; $\alpha = 0.95$; $\varpi = \lfloor 0.01n \rfloor$. (Šios parametrų reikšmės parinktos, atsižvelgus į preliminarinių bandymų rezultatus.) Su kiekviena iš 512 m reikšmių algoritmas buvo pakartotinai įvykdomas 10 kartų. Jei būdavo gaunama bent viena skirtinga galutinė tikslo funkcijos (žr. (3) formulę) reikšmė, algoritmas vėl būdavo vykdomas iš naujo 10 kartų, tačiau esant padidintam generacijų skaičiui N_{gen} . Pradinė N_{gen} reikšmė buvo lygi 100, papildomuose eksperimentuose N_{gen} reikšmė būdavo padidinama dydžiu, lygiu 100. Eksperimentų metu taip pat buvo apskaičiuojamas ir gautų tikslo funkcijos reikšmių vidurkio santykinis nuokrypis (procentais) ($\bar{\delta}$) nuo viršutinės tikslo funkcijos reikšmės ribos. Nuokrypio apskaičiavimo formulė yra tokia: $\bar{\delta} = 100(\bar{z} - \hat{z})/\hat{z}$ [%], čia \bar{z} yra gautų TF reikšmių vidurkis, o \hat{z} žymi TF viršutinę ribą, apskaičiuojamą pagal formulę $\sum_{i=1}^m \sum_{j=1}^m b_{ij}$ (b_{ij} yra atitinkami PŠF uždavinio duomenų matricos B (žr. aukščiau) elementai). Eksperimentai tokiu būdu buvo vykdomi tol, kol visos 10 gautų tikslo funkcijos ir jos nuokrypių reikšmių tapdavo tarpusavyje lygios. Esant visoms šioms reikšmėms identiškomis, traktuojama, jog euristinis algoritmas HGEA konverguoja į galimai optimalų sprendinį (ar bent jau į sprendinį, labai artimą optimaliam).

4.2. Kompiuterinio spalvų atspalvių generavimo rezultatai: galimai optimalūs sprendiniai ir jų vizualizacija

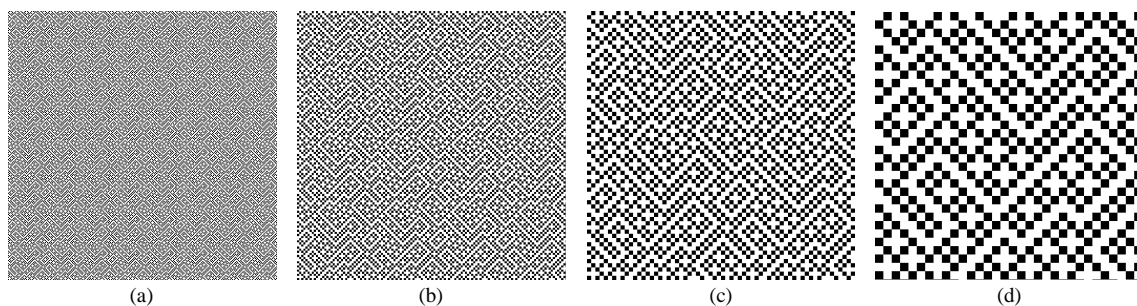
Atliktų nuodugnių kompiuterinių eksperimentų rezultatai pirmiausiai pateikiami 1 lentelėje, kur parodomi galimai optimalūs sprendiniai visoms 512 parametro m reikšmių. Kompaktiškumo dėlei, pateikiamos tik galimai optimaliems gautiems analitiniams sprendiniams atitinkančios tikslo funkcijos absoliutinės skaitinės reikšmės (laikytinos (šiuo metu) geriausiai žinomomis tikslo funkcijos reikšmėmis (GŽTFR)).

Pastebėtina, kad GŽTFR nereiškia optimumo, tai yra tik kaip tam tikra atskaitos reikšmė, kai reikia palyginti gaunamus sprendinius. Nesant įrodomų tikslų sprendinių, o tai būdinga euristiniams algoritmams, GŽTFR yra vienintelis orientyras kiekybiškai įvertinant gaunamus sprendinius ir algoritmo efektyvumą.

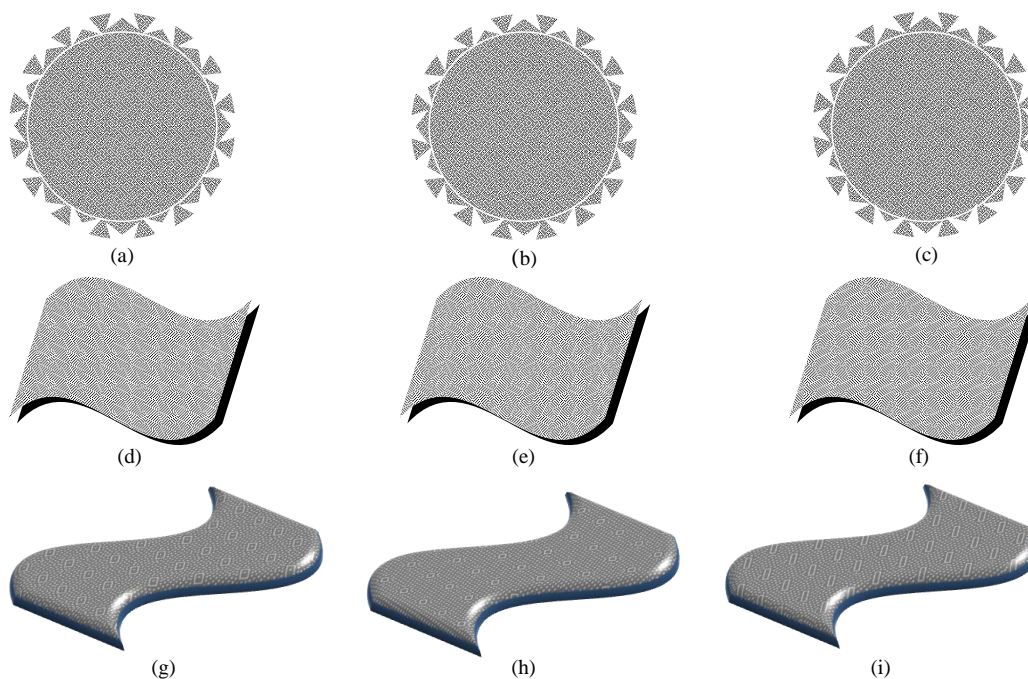
Konvertavus analitinius sprendinius į jų atitinkamus grafinius ekvivalentus, galima vizualiai įsitikinti pastarųjų kokybiškumu, unikalumu, estetiškumu. Grafiniai vaizdai, liudijantys puikią sugeneruotų atskirų pilkų atspalvių šablonų/tekstūrų kokybę, iliustruojami 9–12 pav.



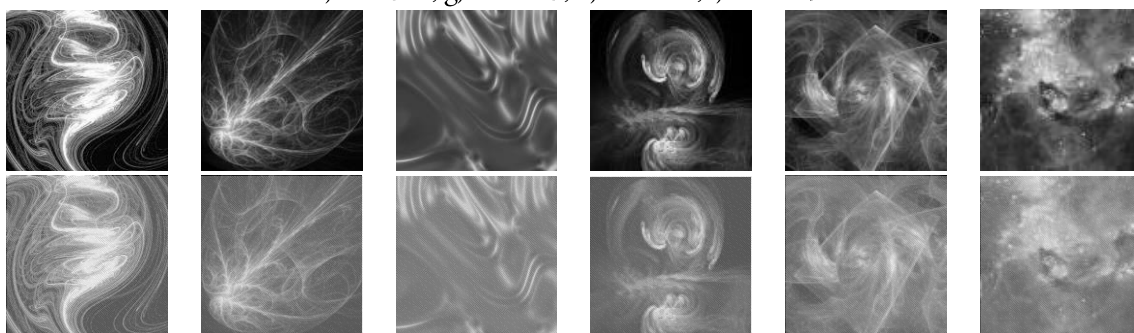
9 pav. Galimai optimalių pilkų atspalvių šablonų pavyzdžiai, kai $n = 1024$:
a) $m = 408$, b) $m = 409$, c) $m = 410$, d) $m = 411$, e) $m = 412$, f) $m = 413$,
g) $m = 414$, h) $m = 415$, i) $m = 416$, j) $m = 417$, k) $m = 418$, l) $m = 419$.



10 pav. Galimai optimalaus šablono pavyzdys ($n = 1024$, $m = 392$), atvaizduotas skirtingais masteliais: a) 1 : 1, b) 2 : 1, c) 4 : 1, d) 8 : 1.



11 pav. Dvimačių ir pseudotrimačių vaizdų su panaudotomis galimai optimalių atspalvių tekstūromis pavyzdžiai ($n = 1024$): a) $m = 384$, b) $m = 385$, c) $m = 386$, d) $m = 322$, e) $m = 323$, f) $m = 324$, g) $m = 443$, h) $m = 444$, i) $m = 445$.



12 pav. Galimai optimalių atspalvių šablonų panaudojimas sudarant subtilius skaitmeninius vaizdų tonus (Čia matomi originalūs juodai-balti paveikslėliai ir jų atitinkami pilki atspalviai (atspalvio reikšmė (m) lygi 322, $n = 1024$). Originalių paveikslėlių šaltinis: 123RF, NASA (prieiga per internetą: <<http://www.123rf.com/>>, <<http://www.nasa.gov/>>).)

Šiame darbe neturime galimybių pailiuoti kitų galimų spalvų atspalvius, nors galima potencialiai sudaryti iki 1024 kiekvienos RGB paletės spalvos skirtingų atspalvių. Planuojama bent dalį reprezentatyvių spalvų atspalvių šablonų (tekstūrų) vizualiai pateikti internete (internetinio tinklalapio adresas: <<http://www.personalas.ktu.lt/~alfmise/>>).

Išvados

Šiame straipsnyje nagrinėjamas aukštos kokybės (galimai optimalių) skaitmeninių spalvų atspalvių generavimas automatinio (t. y. kompiuteriniu būdu). Kompiuterinis atspalvių generavimas realizuotas atitinkamo matematinio modelio — pilkų šablonų formavimo uždavinio pagalba.

Pagrindinis atliktų tyrimų mokslinis reikšmingumas yra tas, jog buvo sudarytas naujas ekstra dydžio PŠF uždavinio testinis pavyzdys, kuris įgalina sugeneruoti aukštesnės kokybės skaitmenizuotų spalvų atspalvių šablonus. PŠF uždavinys spręstas panaudojant efektyvų euristinį optimizavimo metodą — hibridinį genetinį-evoliucinį algoritmą (HGEA). Nors šis algoritmas galbūt ir negarantuoja teoriniu požiūriu globalių optimalių sprendinių suradimo visiems PŠF uždavinio testiniams pavyzdžiams, tačiau jis leidžia žymiai sumažinti skaičiavimų apimtį; kita vertus, gaunami analitiniai sprendiniai ir jų grafiniai atitikmenys — spalvų atspalvių šablonai — vis tiek yra nepriekaištingos kokybės. Tikėtina, jog jie savo kokybe tikrai pranoktų žmonių sukurtus analogiškus šablonus ir turėtų pilnai patenkinti potencialius tokių šablonų užsakovus (vartotojus).

Apskritai, šiame darbe siekta tiek teorinių, tiek praktinių tikslų. Teorinė pusė susijusi su efektyvių euristinių algoritmų prototipų sukūrimu iš pradžių sudėtingų teorinių-matematinių uždavinių sprendimui. Vėliau šie prototipai adaptuojami įgyvendinant konkrečius praktinius tikslus (sprendžiant iškilusius aktualius praktinius-techninius uždavinius). Hibridinio genetinio-evoliucinio algoritmo pritaikymas pilkų šablonų formavimo uždaviniui ir spalvų atspalvių generavimui yra tokio sėkmingo teorinių modelių perkėlimo į praktinę sferą tinkamas pademonstravimas.

Ateietyje galėtų būti toliau tiriamos galimybės realizuojant PŠF uždavinio taikymus meninių tekstūrų kūrime, modernioje kompiuterinėje grafikoje bei multimedijoje, taip pat praplėstoje/virtualioje realybėje, praturtintoje naujomis susintetintomis spalvomis ir vaizdais.

Literatūra

- Burkard, R., Karisch, S., Rendl, F. (1997). "QAPLIB — a quadratic assignment problem library," *Journal of Global Optimization* 10: 391–403. [Žr. taip pat prieigą per internetą: <<http://www.seas.upenn.edu/qaplib/>>.]
- Çela, E. (1998). *The Quadratic Assignment Problem: Theory and Algorithms*. Dordrecht: Kluwer.
- Drezner, Z. (2006). "Finding a cluster of points and the grey pattern quadratic assignment problem," *OR Spectrum* 28: 417–436.
- Lau, D. L., Arce, G. R. (2008). *Modern Digital Halftoning*, Sec. Ed. (Signal Processing and Communications Series, Liu, K. J. R. (ed.)). New York-Basel: Marcel Dekker.

- Loiola, E. M., De Abreu, N. M., Boaventura-Netto, P. O., Hahn, P., Querido, T. (2007). "A survey for the quadratic assignment problem," *European Journal of Operational Research* 176: 657–690.
- Misevičius, A. (2006). "Experiments with hybrid genetic algorithm for the grey pattern problem," *Informatika* 17: 237–258.
- Misevičius, A. (2011). "Generation of grey patterns using an improved genetic evolutionary algorithm: some new results," *Information Technology and Control* 40: 330–343.
- Rendl, F. (2002). "The quadratic assignment problem," Iš Drezner, Z., Hamacher, H. (ed.), *Facility Location: Applications and Theory*. Berlin: Springer, 439–457.
- Taillard, É. D. (1995). "Comparison of iterative searches for the quadratic assignment problem," *Location Science* 3: 87–105.
- Taillard, É. D., Gambardella, L. M. (1997). "Adaptive memories for the quadratic assignment problem," *Tech. Report IDSIA-87-97*, Lugano, Switzerland.
- Ulichney, R. A. (1987). *Digital Halftoning*. London: MIT Press.

A. Misevičius gimė 1962 m. Lietuvoje, Marijampolės rajone. 1986 m. su pagyrimu baigė tuometinį Kauno politechnikos institutą. 1996 m. Kauno technologijos universitete (KTU) apgynė techn. m. daktaro disertaciją. Nuo 1998 m. – docentas, nuo 2011 m. – KTU Informatikos fakulteto Multimedijos inžinerijos katedros profesorius. Yra publikavęs arti 100 mokslo ir mokymo metodikos darbų kompiuterių mokslo, optimizavimo algoritmų temomis. Moksliniai interesai: dirbtinis intelektas; optimizavimas ir euristiniai optimizavimo algoritmai; genetiniai/evoliuciniai algoritmai.

J. Blonskis gimė 1945 m. Lietuvoje, Ukmergės rajone. 1968 m. baigė Kauno politechnikos institutą. 1973 m. apgynė techn. m. daktaro disertaciją. Šiuo metu J. Blonskis yra KTU Informatikos fakulteto Multimedijos inžinerijos katedros docentas. Per daugiau kaip 40 savo darbinės veiklos metų yra publikavęs virš 50 mokslo darbų kompiuterių mokslo, programavimo metodikos tematika. Yra keliolikos vadovėlių bei mokomųjų knygų autorius ir bendraautorius. Moksliniai interesai: programavimas; programavimo metodologija; programavimo edukologiniai aspektai.

V. Bukšnaitis gimė 1959 m. Lietuvoje, Kauno rajone. 1981 m. baigė Kauno politechnikos institutą. Nuo 1990 m. V. Bukšnaitis yra KTU Informatikos fakulteto lektorius. Yra publikavęs daugiau kaip 20 mokslo ir mokymo metodikos darbų kompiuterių mokslo, programavimo temomis. Moksliniai interesai: programavimas; programavimo mokymas; algoritimizavimas.

E. Stanevičienė gimė 1981 m. Lietuvoje, Kaune. 2012 m. baigė KTU Informatikos fakultetą ir įgijo informacinių technologijų magistro laipsnį. Šiuo metu yra KTU Informatikos fakulteto doktorantė. Moksliniai interesai: multimedija; algoritmų kūrimas multimedijos uždaviniams.

T. Želvys gimė 1989 m. Lietuvoje, Kaune. 2012 m. KTU Informatikos fakultete įgijo informatikos bakalauro laipsnį. Šiuo metu yra KTU Informatikos fakulteto Multimedijos inžinerijos katedros magistrantas. Moksliniai interesai: kompiuterinis projektavimas; programavimo inžinerija ir technologijos.

COMPUTATIONAL GENERATION OF HIGH-QUALITY DIGITAL HALFTONES (GREY/COLOUR PATTERNS)

**Alfonsas Misevičius, Jonas Blonskis, Vytautas Bukšnaitis,
Evelina Stanevičienė, Tautvydas Želvys**

Summary

The purpose of this paper is to describe the computational algorithmic generation of the high-quality digital halftones (grey/colour patterns). At the beginning, the formal model for generation of the digital halftones, the so-called grey pattern problem (GPP) is introduced. Then, the heuristic algorithm for the solution, in particular, of the grey pattern problem is discussed. Although the algorithm employed does not guarantee the optimality of the solutions found, still superior-quality, near-optimal (and in some cases probably optimal) solutions can be achieved within reasonable computation time. Further, we provide the results of the extensive computational experiments with the newly proposed, extra-large size instance (data set) of the GPP — which is the main contribution of this work. As a confirmation of the quality of the solutions produced, we also give the visual representations of several fine-looking halftone patterns and the reader can judge about the perfection of the images obtained.

Key words: creation of digital color shades (images), gray pattern formation task, creative multimedia, optimization, heuristic algorithms.