

Kārlis Freivalds, Pēteris Ručevskis, Paulis Ķikusts

LU Matemātikas un informātikas institūta  
Grafu teorijas un vizuālās informācijas apstrādes grupa  
Piedāvā jaunu metodi

## Automātiska attēlu atpazīšana un analīze

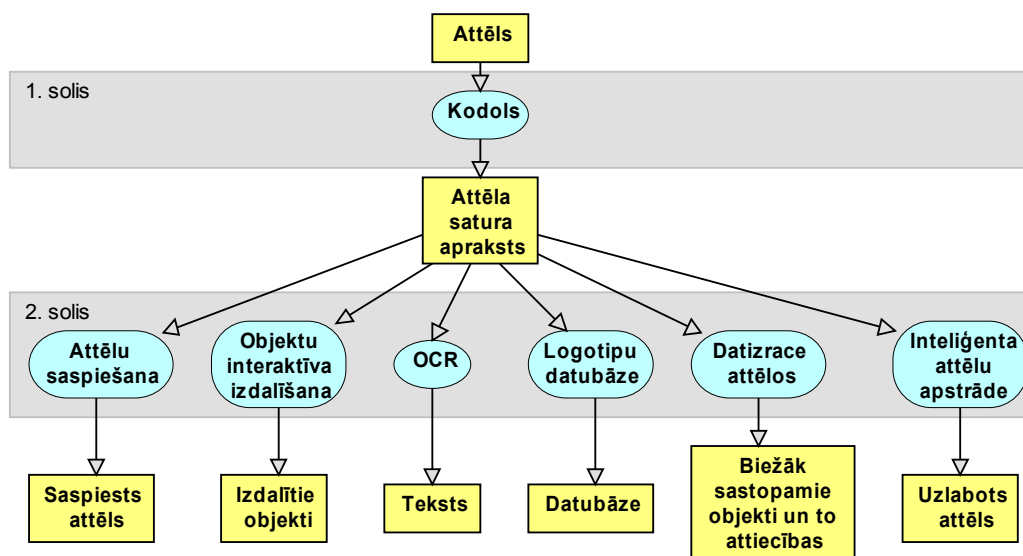
### Fundamentālā attēlu atpazīšanas problēma

Attēli ir it visur. Pirmais jautājums, aplūkojot kādu attēlu, ir “kas šajā attēlā ir redzams?” Interesē arī, kādi objekti tur ir redzami un kādas ir viņu savstarpējās attiecības. Šie ir pamatjautājumi, kas ir jāatbild, lai attēlu varētu izmantot jaunu atziņu gūšanai.

Cilvēkam parasti nesagādā grūtības sniegt atbildi uz šiem jautājumiem, pat tikai pēc paviršas attēla uzlūkošanas. Taču šis ir izrādījies ļoti sarežģīts uzdevums automatiskām sistēmām. Pašlaik ir izstrādāti veiksmīgi risinājumi tikai nedaudzās pielietojumu jomās, piemēram dokumentu automatiskās lasīšanas sistēmām drukātiem burtiem un tabulām. Šajos risinājumos tas tiek panākts, ieliekot tajās daudz specifisku zināšanu, kas ļauj atrisināt tieši šo konkrēto problēmu kopu. Taču, tiklīdz dotais attēls kaut nedaudz neatbilst stingrajiem pieņēmumiem, šādi risinājumi vairs nespēj dot rezultātu. Vispārīgas automatiskas metodes objektu atpazīšanai, aprakstīšanai un analīzei pagaidām nav.

### Mūsu piedāvājums

Mēs esam izstrādājuši jaunu zinātnisku metodi šīs fundamentālās attēlu atpazīšanas problēmas risināšanai. Šī metode ļauj pilnīgi autonomi, neprasot nekādas iepriekšpieņemtās zināšanas par attēlu, iegūt attēla aprakstu, kas satur visus attēlā redzamos objektus un viņu savstarpējās attiecības. Turklāt apraksts ir pilnīgs, t.i., no apraksta attēlu var pilnībā restaurēt. Vispārīgā metodes lietošanas shēma parādīta 1. attēlā.



1. attēls. Vispārīgā metodes lietošanas shēma

Metodes pielietošanā var izdalīt divus soļus. Pirmajā solī no ieejas attēlu kopuma tiek veidots to apraksts. Otrajā solī iegūto aprakstu apstrādā pielietojumu modulis. Pielietojumu moduļi var būt dažādi. Tie atfiltrē konkrētajam lietojumam nepieciešamo apraksta daļu un izdod rezultātu lietojumam nepieciešamajā formā.

Šāda apstrādes procesa priekšrocība ir pielietojumu moduļu vienkāršība. To uzdevums ir tikai caurskatīt aprakstu un atlasīt sev svarīgo. Tas ir iespējams, pateicoties mūsu apraksta veidošanas solim (“kodolam”), kas paveic visu sarežģīto apstrādi neatkarīgi. Lietotājam nav nepieciešams ievadīt jebkādas prasības sistēmai. Rezultātā mūsu izgudrojums ir izmantojams gandrīz visās attēlu apstrādes un analīzes jomās, no kurām dažas ir aplūkotas nākamajā nodaļā.

## Iespējamie pielietojumi

1. **Attēlu saspiešana.** Attēla saspiešanas uzdevums ir nokodēt attēlu pēc iespējas īsākā formā, lai tā uzglabāšana un pārraide būtu pēc iespējas efektīvāka. Šim uzdevumam ir daudz populāru metožu. Ar mūsu pieeju var iegūt būtiski labākus saspiešanas rezultātus daudz plašākai attēlu klasei. Tehniski, lai attēlu saspiestu, mums atliek tikai nokodēt ar kodolu iegūto attēla aprakstu failā. Pilnīgums nodrošina, ka izejas attēls būs viennozīmīgi atjaunojams, bet apraksta saturīgums norāda, ka tas būs īss, un dos labu saspiešanas pakāpi. Sīkāku šī pielietojuma analīzi skatīt nākamajā nodaļā.
2. **Datizrace attēlos.** Datizraces uzdevums ir dokumentu atrašana, kas satur uzdotās iezīmes. Piemēram, atrast visus dokumentus, kas satur uzdotās firmas logo. Vai arī, piemēram, var meklēt uzdoto objektu satelīta uzņēmumos. Ar mūsu paņēmienu to veikt ir pavisam vienkārši un ātri. Tehniski to paveic no sākuma dotajiem dokumentiem izveidojot aprakstu un pēc tam aprakstā ņemam objektus, kas atbilst uzdotajām iezīmēm. Tad atlasām visus dokumentus, kuros šādi objekti ir sastopami.
3. **Dokumenta pārvēršana tekstuālā formā (OCR, Optical Character Recognition).** Uzdevums ir ieskanētu dokumentu pārvērst par rediģējamu dokumentu kādā teksta apstrādes sistēmā (piemēram, MSWord). Vai arī no videokameras iegūtiem attēliem automātiski nolasīt auto numurus. Ar piedāvāto metodi to var ērti veikt, pie tam nav nepieciešama konfigurēšana katram konkrētajam dokumentu tipam vai video attēla saturam. Piemēram, teksta orientācijai vai kropļojumiem nav ietekmes uz atpazīšanas kvalitāti (skatīt tālāk kropļojumu ietekmi uz saspiešanu). Izmantojot izstrādāto kodolu, šādas sistēmas izstrāde ir vienkārša: dotam dokumentam izveido aprakstu, atrod objektus, kas atbilst katram burtam, vai pat veseliem vārdiem, un izveido atbilstošu tekstuālo formu.
4. **Logotipu (trademark) datubāze.** Šāda datubāze nodrošina krāsainu attēlu saspiešanu, saglabāšanu datubāzē, automātisku anotēšanu, kategorizēšanu un objektu un teksta atpazīšanu. Nākot klāt jaunam attēlam, var atrast visus viņam līdzīgos. Pašreiz zināmie risinājumi šajā jomā ir ļoti vāji, tie meklē līdzīgos attēlus tikai pēc specifiski uzdotām pazīmēm tādām kā krāsa, tekstūra,

krāsu sadalījums utml. Mūsu pieeja nodrošina meklēšanu pēc būtības, tas ir pēc attēla sastāvā esošiem līdzīgiem objektiem.

5. **Objektu interaktīva izdalīšana.** Attēlu apstrādes programmās viena no visbiežāk veicamajām operācijām ir interesējošā apgabala iezīmēšana. Bieži interesējošais apgabals ir kaut kāds objekts. Vairumā esošo programmu, lai objektu iezīmētu, tam ir precīzi ar roku jānorāda objekta kontūras, kas ir sarežģīti un darbietilpīgi. Dažas programmas (piem. Adobe Photoshop) piedāvā automātiskus rīkus objekta iezīmēšanai vienkārši noklikšķinot uz tā. Bet esošie šāda veida rīki ir ļoti primitīvi un bieži kļūdās. Piedāvātā metode ļauj pilnīgi automātiski un nekļūdīgi iezīmēt izvēlēto objektu, tikai uzklikšķinot uz tā. Turklāt objekts tiek nekļūdīgi iezīmēts, pat ja to pārklāj citi objekti. Sīkāku demonstrāciju skatīt nākamajā nodaļā.
6. **Inteliģenta attēlu apstrāde.** Mūsu piedāvātā pieeja ļauj veikt attēlu apstrādi jaunā augstākā līmenī. Tradicionāli tādas darbības kā mērogošana, rotācija, trokšņu noņemšana, un citas attēla uzlabošanas funkcijas tiek realizētas apstrādājot attēla pikseļus, un tas noved pie būtiskas attēla degradācijas. Mūsu pieeja ļauj šīs funkcijas veikt izdalīto objektu līmenī, tādējādi nebojājot attēla būtiskās iezīmes. Piemēram, novācot lieko traucējošo informāciju ar tradicionāliem paņēmieniem, attēls tiek izpludināts. Mūsu pieeja ļaus novākt traucējošo informāciju, objektu kontūrām paliekot asām. Līdzīgi ir ar mērogošanu, kur mūsu pieeja ļaus iegūt būtiski asāku attēlu, to palielinot. Papildus attēla uzlabošanai mūsu pieeja nodrošina attēlu salīdzināšanu, segmentāciju, objektu izdalīšanu un vektorizāciju.

## Rezultāti

Lai veiktu metodes lietojamības pārbaudi, tika izstrādāts tās kodola prototips kā arī divi pielietojumu moduļi – attēlu saspiedējs un objektu izdalītājs. Kodola prototips prot apstrādāt binārus attēlus, kas ir tikai baltā un melnā krāsā (šādā formā parasti tiek uzglabāti skanēti dokumenti vai rasējumi).

### *Attēlu saspiešana*

Attēla saspiešanas uzdevums ir nokodēt attēlu pēc iespējas īsākā formā, lai tā uzglabāšana un pārraide būtu pēc iespējas efektīvāka. Tiek aplūkota tikai bezzudumu saspiešana, t.i, tāda, kur no saspiešanas informācijas atjaunotais attēls ir identisks sākotnējam.

Attēla saspiešanas analīze sniedz priekšstatu ne tikai par konkrētā lietojuma efektivitāti, bet parāda metodes efektivitāti vispār, jo saspiešanas pakāpe galvenokārt ir atkarīga no metodes spējas analizēt un saprast attēla saturu un izdalīt būtisko informāciju tajā. Kompaktāka attēla reprezentācija nozīmē, ka iegūts nozīmīgāks un vairāk izsakošs attēla apraksts. Turklāt saspiešana dod kvantitatīvu un izmērāmu salīdzināšanas kritēriju, kas citā veidā ir grūti iegūstams.

Kā pasaulē labākās un populārākās bināru attēlu saspiešanas metodes jāmin PNG, Lossless Jpeg (JPEG-LS) un DjVu. PNG un JPEG-LS formāti ir samērā primitīvi un veic saspiešanu pikseļu līmenī. DjVu ir daudz gudrāks un automātiski veic burtu un citu simbolu atpazīšanu, iegūstot labāku saspiešanas pakāpi. Taču savu efektivitāti tas

var izmantot tikai samērā šauram attēlu klāstam – skanētiem dokumentiem ar noteiktu struktūru.

Mūsu testiem tika izmantoti 3 attēli, kas redzami pielikumā. Attēls *Page* ir teksta lappuse no zinātniska raksta, *Page\_skewed* ir šī paša lappuse pagriezta 45 grādu leņķī, bet attēlā *Random\_circles* ir redzami 100 dažāda lieluma riņķi. Attēlus skatīt pielikumā 1. Sekojošā tabulā ir redzami faila izmēri baitos pēc saspiešanas ar dažādām metodēm.

Nosaukums	<a href="#">PNG</a>	<a href="#">JPEG-LS</a>	<a href="#">DjVu</a>	Piedāvātais (prototips)	Piedāvātais (prognoze)
Page	20936	32606	5206	7219	4125
Page_skewed	36657	44316	9456	7826	4472
Random_circles	22019	24904	7027	1495	854

Tabula. *Dažādu saspiešanas metožu salīdzinājums*

Tā kā piedāvātā metode ir vēl tikai prototipa stadijā, tad iegūtie rezultāti vēl nav optimāli, un, izstrādājot galīgo variantu, tiks veikti dažādi uzlabojumi, kā rezultātā saspiešanas efektivitāte varētu paaugstināties 1.5 – 2 reizes. Kolonnā “Piedāvātais (prototips)” ir parādīts šī brīža rezultāts, bet kolonnā “Piedāvātais (prognoze)” ir prognozētais galīgā varianta rezultāts.

Redzam, ka parasta teksta lappusei DjVu formāts ir stipri pārāks par PNG un JPEG-LS. Mūsu piedāvātā algoritma prototips nedaudz atpaliek no DjVu, taču, realizējot galīgo variantu, tiek prognozēts, ka piedāvātais algoritms pārspēs DjVu uz šāda veida attēliem.

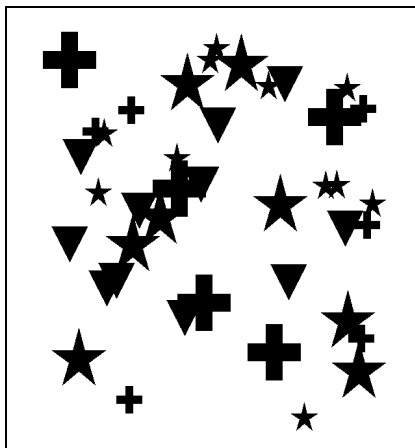
Ja šo pašu teksta lappusi sašķiebj (attēls *Page\_skewed*), tad redzams ka DjVu tāpat kā PNG un GPEG-LS sniegums stipri pavājinās, bet piedāvātā algoritma rezultāti pavājinās pavisam nedaudz. Tas norāda uz to, ka esošo algoritmu pamatā ir stingri iešūtas specifiskas attēla īpašības (piemēram, burtiem jāveido horizontālas rindiņas), kuru neizpildīšanās būtiski ietekmē algoritmu sniegumu. Piedāvātajai metodei nekādu šādu iepriekšdefinētu īpašību nav, tāpēc saspiešanas pakāpe praktiski nemainās, attēlu izkropļojot. Šādi kropļojumi bieži rodas praksē, piemēram, skanēšanas procesā, neakurāti novietojot dokumentu skanerī.

Attēls *Random\_circles* pavisam slikti saspiežas ar esošajiem saspiešanas algoritmiem, jo tie neprot atšifrēt tā struktūru. Piedāvātais algoritms prot atpazīt atsevišķus riņķus šajā attēlā, tāpēc tam izdodas nokodēt attēlu daudz reižu labāk.

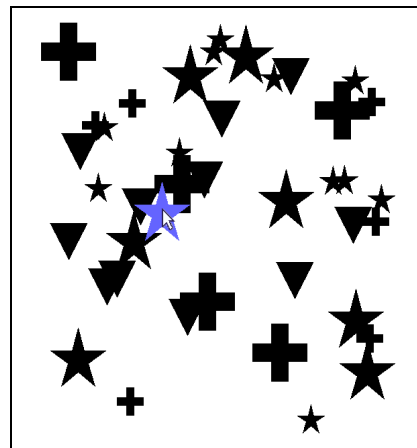
### *Interaktīva objektu izdalīšana*

Objektu izdalīšanas uzdevums ir atrast attēlā objektu, uz kura ir uzklikšķinājis lietotājs. Šāds uzdevums ir svarīgs visās attēlu rediģēšanas programmās un pagaidām veiksmīgu risinājumu nav. Šo uzdevumu var viegli atrisināt ar mūsu pieeju, kur objektu izdalīšana notiek pilnīgi automātiski, un atliek tikai paziņot to objektu, kas atrodas uzrādītājā punktā.

Piemēram, 2. attēlā ir redzams objektu kopums. 3. attēlā redzams, ka lietotājs ir uzklikšķinājis peļi uz objekta apmēram attēla vidusdaļā, un zilā krāsā ir iezīmējies attiecīgais objekts.



2. attēls. *Objektu kopums*



3. attēls. *Izdalītais objekts*

Ir redzams, ka ir izdalīts pareizais objekts, neskatoties uz to, ka tas pārklājas ar vairākiem citiem objektiem. Citas zināmās objektu izdalīšanas metodes nespēj atdalīt pārklājošus objektus.

### **Noslēgums**

Ir izstrādāta un tiek piedāvāta jauna metode, kas ļauj vienotā un ērtā veidā atrisināt daudzus attēlu apstrādes un atpazīšanas uzdevumus. Ir izstrādāts piedāvātās metodes prototips, kas parāda metodes darbību un priekšrocības.

Šī metode tiks patentēta tuvākajā laikā, tāpēc detaļas par metodes realizāciju netiek sniegtas.

Mēs meklējam partnerus metodes un pielietojumu tālākai attīstīšanai, kā arī mārketinga un pārdošanas veikšanai.

## 1. pielikums. Dažādi attēlu veidi.

and since  $t$  was sufficiently small, one of them is  $(A, A')$ .

There are two cases:

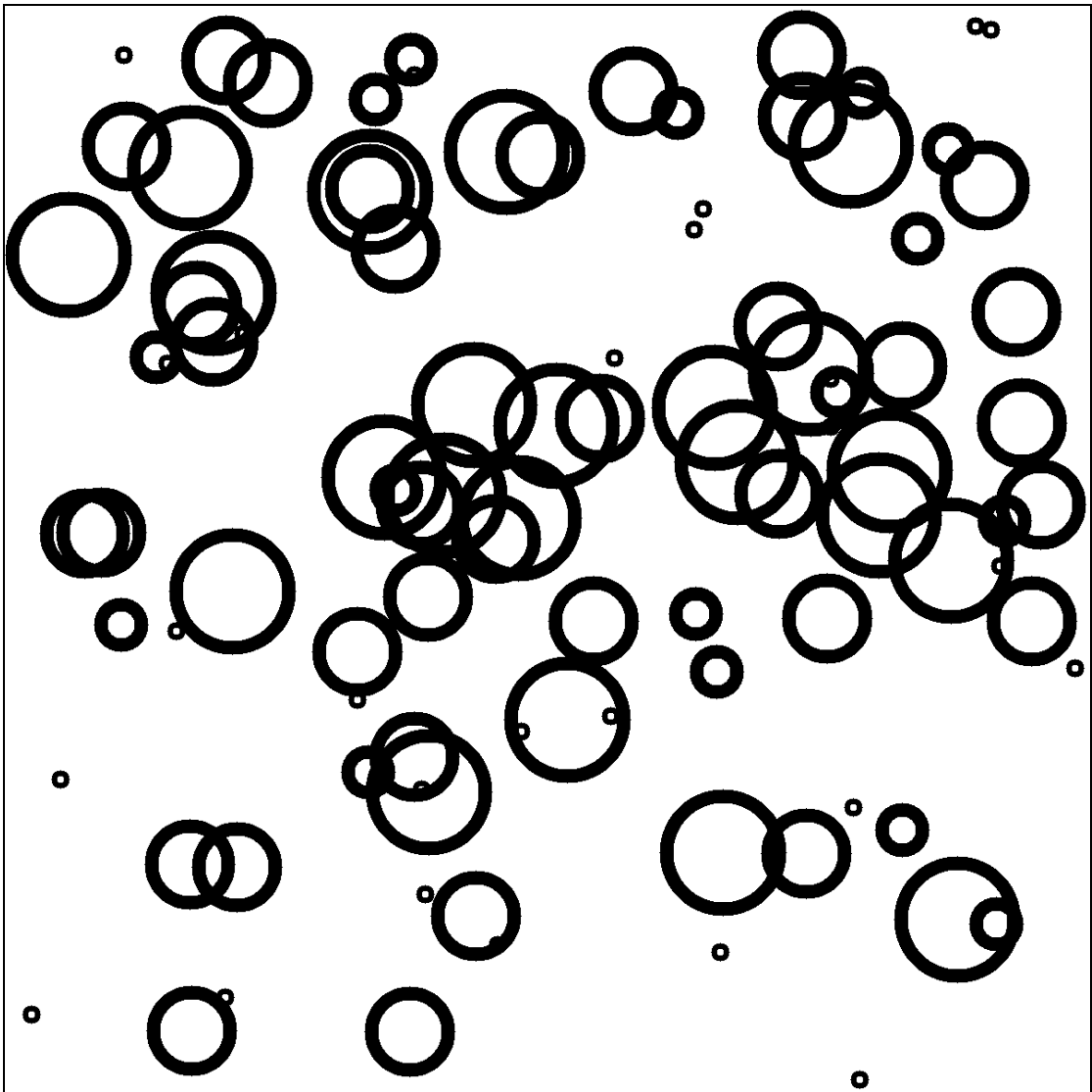
- (i) there exist a positional cut  $R_p(x)$  with value not equal to  $R(A, A')$ . Then by Theorem 1 there is a positional cut with value  $R_p(x) < F(x) \leq F(x^A) = R(A, A')$  which divides the nodes into sets  $U_1$  and  $U_2$ . Since  $t$  was sufficiently small, both  $U_1$  and  $U_2$  cannot contain nodes from both sets  $A$  and  $A'$  i.e. either  $U_1 \subset A$ , or  $U_1 \subset A'$ , or  $U_2 \subset A$ , or  $U_2 \subset A'$  contradicting the local minimality of  $(A, A')$  by Definition 1.
- (ii) all positional cuts of  $x$  are of the same value. Let us choose some positional cut  $(U_1, U_2) \neq (A, A')$ . And again since we can choose  $t$  small enough either  $U_1 \subset A$ , or  $U_1 \subset A'$ , or  $U_2 \subset A$ , or  $U_2 \subset A'$  what contradicts strict local optimality of  $(A, A')$  by Definition 1.

Secondly we prove that each strict local minimum  $x$  of (2, 3, 4) is in the form of a characteristic vector of a strictly locally minimal cut. From Theorem 2 we have that each strict local minimum is in the form of a characteristic vector  $x^A$  of some cut  $(A, A')$ . We need to show that this cut is strictly locally minimal. Assume to the contrary that there exist  $S \subset A, S \neq \emptyset, A \setminus S \neq \emptyset$  for which  $R(A \setminus S, A' \cup S) \leq R(A, A')$ . Denote  $U = A \setminus S$ . Let us repeat the construction from the proof of Theorem 1 taking the sets  $U_1, U_2$  and  $U_3$  as  $S, U, A'$  correspondingly. We obtain equations (9 – 12) and  $y$  with  $y_1 = y_2$ . From the Definition 5 each small move increases the function. Let us take such feasible move  $t$  that do not destroy the groups  $U_1, U_2$  and  $U_3$  but separates  $U_1$  from  $U_2$ . Let  $t'$  be the projection of  $t$  onto the sets  $U_1, U_2, U_3$  i.e.  $t' = (t_k | k \in U_1, t_p | p \in U_2, t_r | r \in U_3)$ . Let us denote  $y^m = y + t', y^e = y + \alpha t'$  where  $\alpha$  is chosen such that  $y^{e_2} = y^{e_3}$ . Since the constraints (10, 11) are linear and they have the same value for  $y$  and  $y^m$ , they will have the same value for  $y^e$ , which is a linear combination of  $y$  and  $y^m$ . Obviously  $y^e$  satisfies the constraint (12). Similar arguments hold for the function: since  $F_2(y^m) > F_2(y)$ , also  $F_2(y^e) > F_2(y)$  from the linearity of (9). From  $y^e$  we can go back to the original problem and obtain  $x^e$  which is a characteristic vector of the cut  $(A \setminus S, A' \cup S)$ . Since  $H(x^e) = H_2(y^e) = 1$  and  $F(x^e) = F_2(y^e) > F_2(y) = R(A, A'), R(A \setminus S, A' \cup S) > R(A, A')$  which contradicts our assumption.  $\square$

We shall note that also for each non-strictly minimal ratio cut its characteristic vector  $x^A$  gives a local minimum of the function, however the converse is not true. There exist non-strict local minima of (2, 3, 4) with the function value not equal to any locally minimal cut value. In Fig. 1 a graph and their node coordinates in a local minimum of (2, 3, 4) is shown. It is a local minimum because no small move gives a decreased positional cut or a better function value. But none of its positional cuts is locally minimal.

Attēls *Page*

and since  $t$  was sufficiently small, one of them is  $(A, A)$ . Then by Theorem 1 there is a  
 There are two cases:  
 (i) There exist a positional cut  $R(C)$  with value not equal to  $R(A, A)$  which divides the nodes into sets  $C_1$  and  $C_2$  and  
 i.e. either  $C_1 \subset A$  or  $C_2 \subset A$  or  $C_1 \subset A'$  or  $C_2 \subset A'$  contradicting the local minimality of  $(A, A)$ .  
 (ii) And again since we prove that each strict local minimum  $x$  of  $(A, A)$  by Definition 1. We need to show that this cut is strictly locally  
 what contradicts our assumption. Let us repeat the construction from the proof of Theorem 1 taking the sets  $C_1, C_2$   
 Secondly, we prove that each strict local minimum  $x$  of  $(A, A)$  is in the form of a characteristic  
 vector of a characteristic vector  $x^a$  of some cut  $S \subset A$ . Let us take such feasible move  $t$  that do not destroy the groups  
 and  $C_2$  as  $S, C_1 \setminus S, C_2$  but separates  $C_1$  from  $C_2$ . Let us denote the projection of  $t$  onto the sets  $C_1, C_2$  i.e.  $t_{C_1} = y_{C_1}$   
 each small move increases the function. Let us take such feasible move  $t$  that do not destroy the groups  
 $C_1, C_2$  but separates  $C_1$  from  $C_2$ . Let us denote the projection of  $t$  onto the sets  $C_1, C_2$  i.e.  $t_{C_1} = y_{C_1}$   
 Since the constraints (10, 11) are linear and they have the same value for  $\alpha$  is chosen such that  $y_{C_1} = y_{C_2}$   
 From  $y_{C_1} \cup S$ . Since  $f(x^a) = f_3(C)$  which is a characteristic vector of the cut  $(C)$   
 which contradicts our assumption.  
 We shall note that also for each non-strictly minimal ratio cut its characteristic vector  $x^a$  gives a local  
 minimum of the function, however the converse is not true. There exist non-strict local minima of  $(A, A)$   
 coordinates in a local minimum of  $(A, A)$  is shown. It is a local minimum because no small move gives  
 a decreased positional cut or a better function value. But none of its positional cuts is locally minimal.



Attēls *Random\_circles*