Elektrotehniški vestnik 73(4): 215–220, 2006
Electrotechnical Review, Ljubljana, Slovenija

Razdelitev mreže s kolonijami mravelj

Peter Korošec[1], Jurij Šilc[1], Borut Robič[2]

[1] *Institut Jožef Stefan, Odsek za računalniške sisteme, Jamova c. 39, 1000 Ljubljana, Slovenija*
[2] *Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška c. 25, 1000 Ljubljana, Slovenija*
E-pošta: peter.korosec@ijs.si

Povzetek. V sestavku je prikazana uporaba optimizacijske metode s kolonijami mravelj pri praktičnem problemu razdelitve mreže. Za ta problem predlagamo dve različni izvedbi osnovnega algoritma s kolonijami mravelj. Prva je večnivojska, ki je nadgradnja osnovnega algoritma z večnivojsko metodo, druga pa je hibridna, kjer združimo vektorsko razdelitev in osnovni algoritem.

Ključne besede: razdelitev mreže, optimiranje s kolonijami mravelj, večnivojski algoritem, vektorska razdelitev

Mesh Partitioning with Ant Colonies

Extended abstract. Many real-world engineering problems can be expressed in terms of partial differential equations and solved by using the finite-element method, which is usually parallelized, i.e. the mesh is divided among several processors. To achieve high parallel efficiency it is important that the mesh is partitioned in such a way that workloads are well balanced and interprocessor communication is minimized. In this paper we present an enhancement of a technique that uses a nature-inspired metaheuristic approach to achieve higher-quality partitions. We present two heuristic mesh-partitioning methods, both of which build on the multiple ant-colony algorithm in order to improve the quality of the mesh partitions. The first method augments the multiple ant-colony algorithm with a multilevel paradigm, whereas the second uses the multiple ant-colony algorithm as a refinement to the initial partition obtained by vector quantization. The two methods are experimentally compared with the well-known mesh-partitioning programs, p-METIS and Chaco.

Key words: mesh partitioning, ant colony optimization, multilevel algorithm, vector quantization

1 Uvod

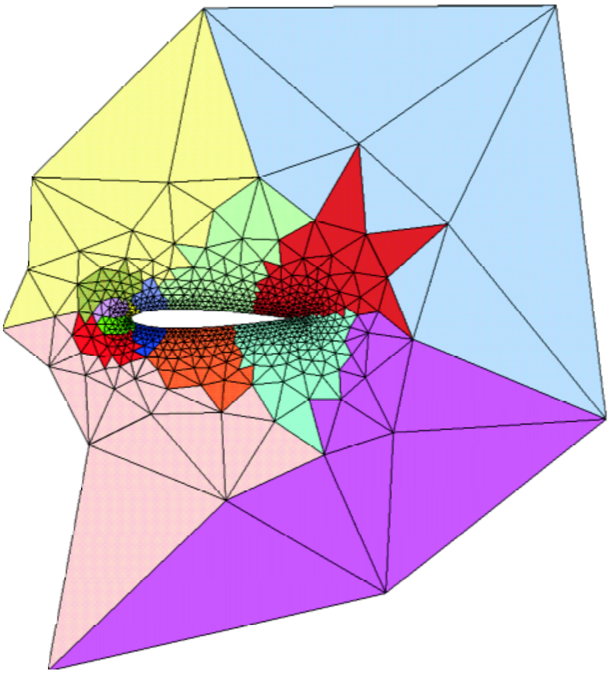
V sestavku bomo pokazali praktičen primer uporabe optimizacijske metode s kolonijami mravelj (ACO iz angl. *Ant-Colony Optimization*) pri modeliranju in simulaciji pri inženirskih problemih. Tovrstni problemi so opisani s fizikalnimi modeli, ki temeljijo na parcialnih diferencialnih enačbah. Za njihovo reševanje so se uveljavile metode končnih elementov. Pomembna slabost metode končnih elementov je njena velika računska zahtevnost – kot posledica uporabe velikih matrik –, kadar ni mogoče najti neke splošne pravilnosti v obravnavanem sistemu. Računanje je mogoče pospešiti tako, da problem razdelimo na podprobleme in le-te rešujemo na vzporednem računalniku. Ko govorimo o t.i. domenski dekompoziciji kot enem izmed načinov paralelizacije metode končnih elementov, imamo v mislih razdelitev mreže končnih elementov na manjša področja in njihovo dodelitev v izračun posameznim procesorjem. Toda povezave, ki potekajo med različnimi področji, zahtevajo, da se med računanjem podatki med procesorji izmenjujejo. Iskanje optimalne razdelitve, take, ki zmanjša medprocesorsko komunikacijo na najmanjšo mero, je zelo pomembno, saj skrajša celoten čas reševanja problema (in s tem možnost reševanja večjih problemov) ter izboljša skalabilnost, saj se z večanjem števila procesorjev v veliko manjši meri veča tudi medprocesorska komunikacija.

2 Problem razdelitve mreže

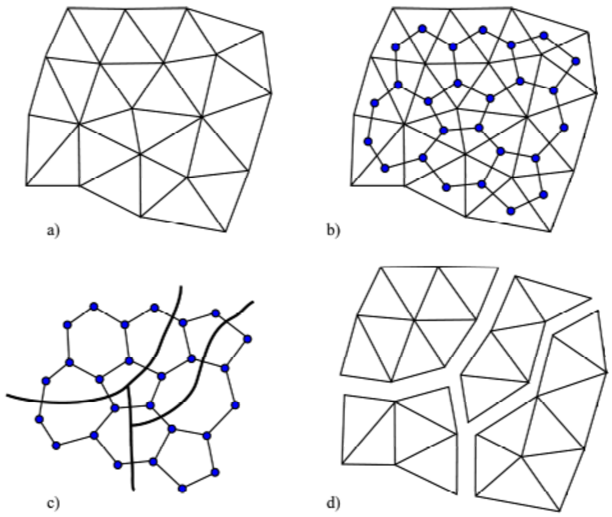
Razdelitev grafa je pomembna prvina razdelitve mreže pri dekompoziciji domen. Žal je iskanje optimalne razdelitve mreže pri nekaterih omejitvah NP-težak problem, kar nam onemogoči, da bi našli rešitev v času, ki je polinomski v odvisnosti od velikosti danega problema (razen, če velja P = NP). Kot rezultat tega se pri reševanju problema razdelitve mrež ponavadi uporabljajo hevristični načini.

Na tem mestu se s samim generiranjem mrež [1], ki je prav tako zahtevno delo, ne bomo posebej ukvarjali, temveč se bomo osredotočili predvsem na razdelitev samo. Problem razdelitve nestrukturirane mreže končnih elementov (slika 1) je lahko formuliran kot problem razdelitve grafa (slika 2). Mreža končnih elementov je ponazorjena z grafom $G(V, E)$, ki je sestavljen iz vozlišč in povezav, ki povezujejo posamezna vozlišča med seboj. Vsako vozlišče v ima težo 1 in se sklada z elementom

Prejet 10. februar, 2006
Odobren 7. september, 2006



Slika 1. Razdelitve mreže Figure 1. Mesh-partitioning



Slika 2. Razdelitve grafa Figure 1. Graph-partitioning

iz mreže končnih elementov. Povezava med dvema vozliščema v in u nam pove, da sta dani vozlišči sosedi. Obstaja nekaj meril, po katerih se določa, ali sta dva elementa soseda [2]. Merilo, ki ga bomo uporabili, je: dva elementa mreže končnih elementov sta soseda, če imata skupno stran (v 3D) ali skupni rob (v 2D).

Optimizacijska oblika problema razdelitve grafa je naslednja. Naj bo $G(V, E)$ neusmerjen graf. Razdelitev D grafa G je sestavljena iz k medsebojno disjunktnih podmnožic $D_1, D_2, \ldots, D_k$ (domen) množice V, katerih unija je množica V. Množica povezav, ki povezujejo različne domene razdelitve D, se imenuje rez povezav (angl. *edge-cut*). Razdelitev D je uravnotežena, če so velikosti domen približno enake; to pomeni, da velja $b(D) = \max_{1\leq i\leq k}|D_i| - \min_{1\leq i\leq k}|D_i| \approx 0$. Problem razdelitve grafa je najti uravnoteženo razdelitev z minimalnim rezom $\zeta(D)$, tj. minimalnim številom prerezanih povezav.

Za boljšo ponazoritev zahtevnosti problema si bomo ogledali naslednji primer. Naj obstaja graf $G(V, E)$ ter naj bo $n = |V|$ število vozlišč. Za vsak par vozlišč $v_i, v_j \in V$ naj bo c_{ij} cena povezave $e_{ij} = (v_i, v_j)$. (Opomba. Cena med nepovezanima vozliščema ni definirana.) Množico V moramo razdeliti na $k \geq 2$ disjunktnih podmnožic. Predpostavimo, da je n mnogokratnik števila k, torej velja $n = kp$ za neki $p \in I\!N$. Potem obstaja $\binom{n}{p}$-načinov za izbiro prve množice, $\binom{n-p}{p}$-načinov za izbiro druge množice in tako naprej. Ker je premeščanje podmnožic nepomembno, je število razdelitev enako $\frac{1}{k!}\binom{n}{p}\binom{n-p}{p}\cdots\binom{2p}{p}\binom{p}{p} > \sqrt{k^{2n-k-1}n^{1-k}}$. Za večino vrednosti n, k in p ta izraz pomeni zelo veliko število; npr. pri $n = 40, p = 10$ in $k = 4$ je to število večje od 10^{20}. Iz poenostavitve zgornjega izraza izhaja, da je v prostoru dopustnih rešitev našega problema vsaj $\Omega(kn)$ elementov. Problem razdelitve je v odločitveni obliki NP-poln [3], kot optimizacijski problem pa seveda s tem NP-težak [4].

3 Algoritmi s kolonijami mravelj

3.1 Osnovni algoritem

Osnovna ideja uporabe mravelj pri razdeljevanju grafa je zelo preprosta [5,6]. Imamo dve ($k = 2$) ali štiri ($k = 4$), med seboj za hrano tekmujoče kolonije mravelj. V našem primeru je hrana vozlišče v grafu. Ideja je v tem, da mravlja pobere hrano (in tako pridruži vozlišče k enemu od mravljišč – domen). To je dalo algoritmu MACA tudi ime (iz angl. *Multiple Ant-Colony Algorithm*).

Graf najprej preslikamo na polje celic, ki pomenijo okolje, po katerem se premikajo mravlje. Obstaja veliko različnih možnosti, kako preslikati grafe, a za naš primer bomo izbrali kar naključno preslikavo. Mravlje postavimo v mravljišča, ki se nahajajo v celicah polja. Od tod hodijo nabirat hrano. Mravlja se lahko po polju premika iz celice v celico v treh smereh (naprej, levo in desno). Odločitev, v katero izmed smeri se bo mravlja premaknila, je določena z verjetnostjo premika. Pri odločanju o smeri premika je uporabljena kumulativna verjetnostna porazdelitev. Da mravlja ne bi zapustila polja, ji tedaj, ko doseže celico ob robu, dovolimo le še premikanje stran od roba; v obe smeri z enako verjetnostjo. Ko mravlja najde hrano, jo poskuša dvigniti. Najprej preveri, ali je količina trenutno nabrane hrane v mravljišču prekoračila "kapaciteto skladišča" (kapaciteta skladišča je omejena zaradi zahtevanih omejitev problema). Če zmogljivost ni dosežena, se izračuna teža hrane glede na vrednost reza (tj. števila prerezanih povezav v grafu), ki bi nastala ob dodelitvi izbranega vozlišča k določeni domeni. Ta domena je v zvezi z mravljiščem trenutne mravlje. V primeru, ko je kapaciteta dosežena, se mravlja premakne v naključno izbrano smer. Če je hrana za eno mravljo pretežka (in ni pretežka za več mravelj), pošlje mravlja signal SOS v nekaj bližnjih celic. Torej, če je v okolici kakšna mravlja, bo ta priskočila na pomoč in pomagala nesti hrano v mravljišče. Med vračanjem v mravljišče odlaga mravlja feromone. Tako lahko druge mravlje sledijo tej feromonski sledi in naberejo dodatno hrano iz iste ali pa sosednjih celic. Ko mravlja doseže mravljišče, spusti hrano na prvo prosto mesto v okolici mravljišča. Seveda lahko mravlje nabirajo/kradejo hrano tudi iz drugih mravljišč. S tem zelo pripomorejo k izboljšavi trenutne rešitve.

Kot smo že omenili, smo v algoritem MACA vgradili določene omejitve. Kot prvo omenimo kapaciteto skladišča. S tem smo preprečili možnost, da bi ena kolonija nabrala vso hrano, ter ohranjamo primerno ravnotežje med velikostmi domen. Ko intenzivnost feromonov v določeni celici pade pod dovoljeno vrednost, se intenzivnost feromonov v tej celici povrne na prvotno vrednost. S tem ohranjamo visok nivo preiskovanja. V vsako celico lahko damo le določeno količino hrane in tudi vsaka mravlja lahko nosi le omejeno količino hrane. Opis osnovnega algoritma MACA je naslednji:

Inicializacija(G)
algoritem MACA(graf)
  while pogoj za končanje ni dosežen **do**
    for vse mravlje **do**
      if nosi hrano **then**
        if doseže mravljišče **then** *Spusti_hrano*()
        else *Premik_k_mravljiščun*()
      else if najde hrano **then**
        Poberi_hrano()
      else if hrana je naprej **then**
        Premik_naprej()
      else if doseže mravljišče **then**
        Spusti_feromone_po_poti()
      else if SOS signal **then**
        Premik_k_SOS_signalu()
      else *Naprej_po_najbogatejši_feromonski_sledi*()
    endfor
    for vse celice **do**
      Izhlapevanje_feromonov()
    endfor
  endwhile
end

3.2 Večnivojski algoritem

Uveljavljeni način pospešitve in globalne izboljšave razdelitvenega postopka je uporaba t.i. večnivojske metode. Osnovna ideja je v združevanju vozlišč v grozde, ki definirajo nov graf. V naslednjih korakih rekurzivno ponavljamo to krčenje, dokler ne dosežemo neke vnaprej določene grobosti (velikosti) grafa. Temu sledi zaporedno drobljenje tako dobljenih grobih grafov. Ta postopek je znan kot večnivojska metoda. To zamisel sta prva predložila *Barnard in Simon* [7] kot metodo za pospešitev spektralne bisekcije.

Postopek je izveden v dveh delih: v prvem poteka krčenje grafa, v drugem pa drobljenje le-tega z ustrezno razdelitvijo na vsakem nivoju ℓ.

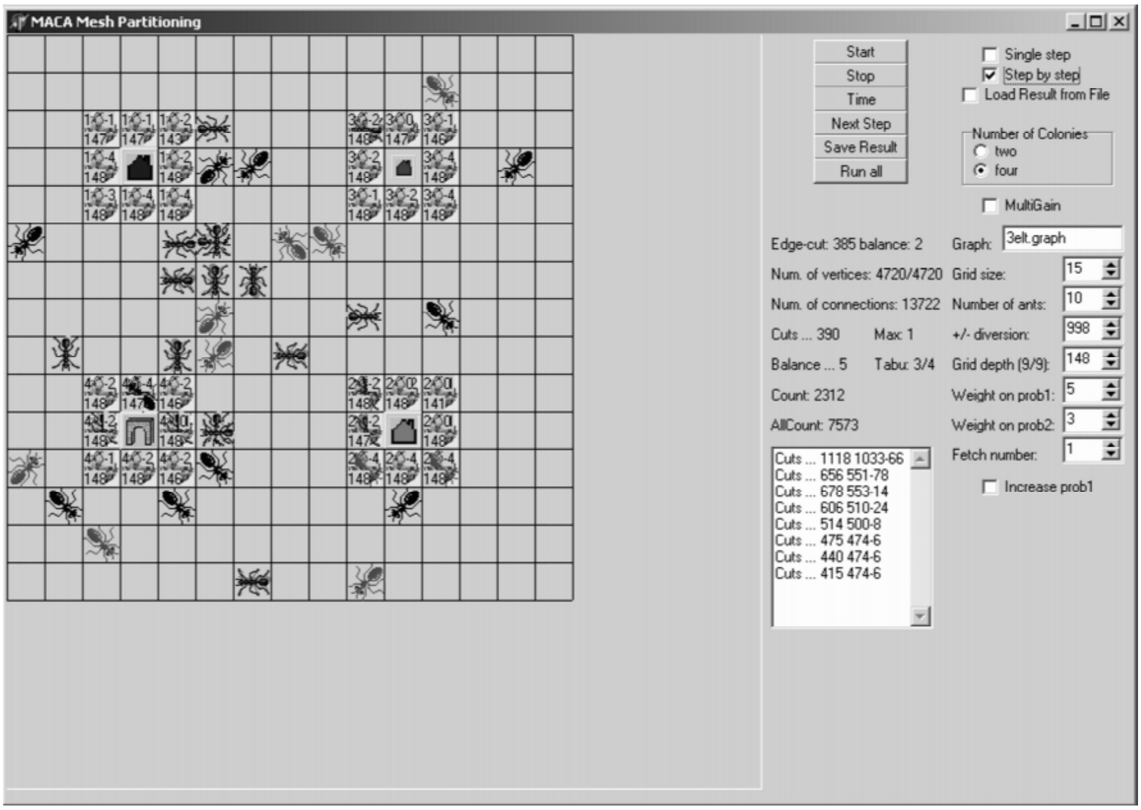
Krčenje – Na vsakem nivoju $\ell + 1$ naredimo iz grafa $G_\ell(V_\ell, E_\ell)$ bolj grob graf $G_{\ell+1}(V_{\ell+1}, E_{\ell+1})$, in sicer z iskanjem največje neodvisne podmnožice povezav (angl. independant connection subset) grafa G_ℓ. Vsako izbrano povezavo uničimo in vozlišča $v_1, v_2 \in V_\ell$, ki sta na obeh koncih te povezave združimo v novo vozlišče $v \in V_{\ell+1}$, ki ima težo $|v| = |v_1| + |v_2|$. Povezave, ki niso bile uničene, so podedovane v novem grafu $G_{\ell+1}$ in povezave, ki so postale podvojene, se združijo v eno, katere teža se sešteje. Zaradi dedovanja ostane skupna teža grafa enaka, skupna teža povezav pa se zmanjša za težo uničenih povezav. Cel postopek krčenja nima nobenega vpliva na samo neuravnoteženost razdelitve $b(D)$ in rez $\zeta(D)$.

Drobljenje – V drugem delu na vsakem nivoju ℓ drobimo že optimirano razdelitev (z algoritmom MACA) grafa G_ℓ. Optimirana razdelitev mora biti preslikana na njen starševski graf $G_{\ell-1}$. Zaradi preprostega krčenja grafa (v prvem delu) je preslikava preprosta. Torej, če je vozlišče $v \in V_{\ell+1}$ in pripada domeni D_i, potem je ustrezen par $v_1, v_2 \in V_\ell$, ki pomeni vozlišče v, tudi v domeni D_i. Graf postopoma drobimo do njegove prvotne velikosti, tako da na vsakem nivoju ℓ uporabimo algoritem MACA in opisano drobljenje. Temu pravimo tudi večnivojski algoritem ali krajše m-MACA:

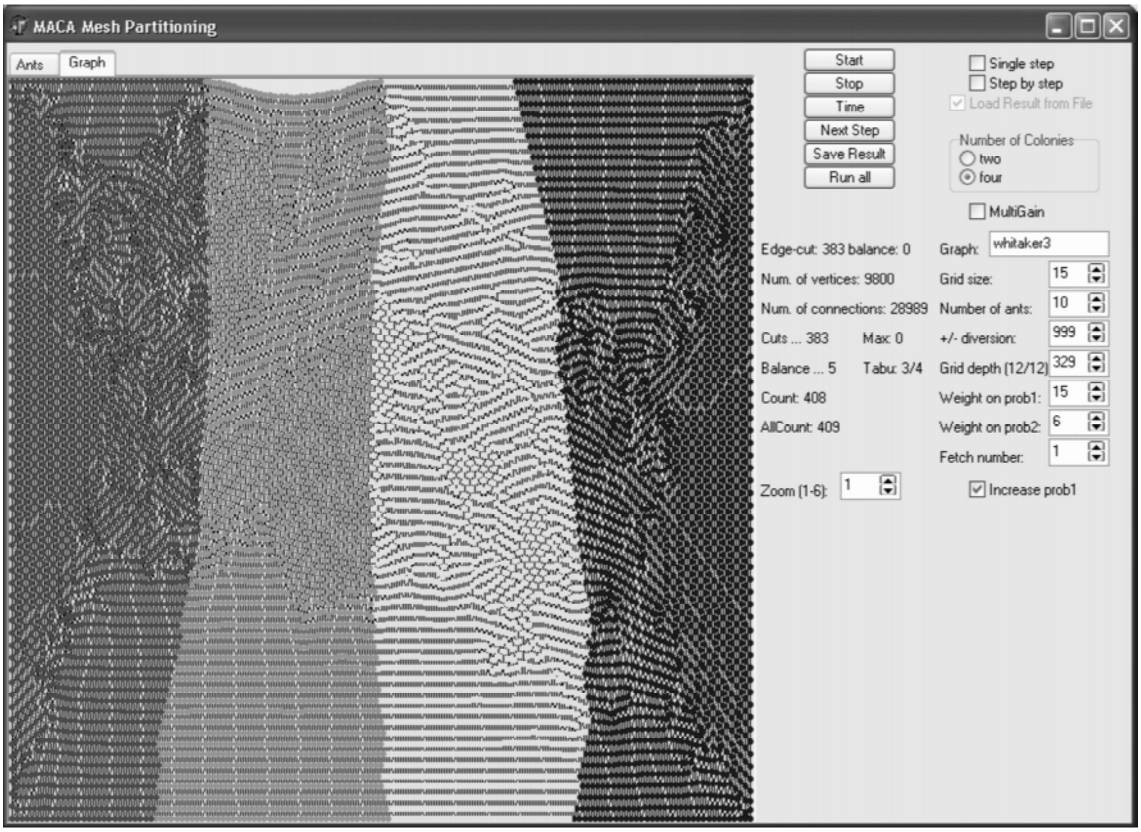
algoritem m-MACA
  Inicializacija(G)
  for nivo = 1 **to** ℓ **do**
    Krčenje(G)
  endfor
  for nivo = ℓ **downto** 1 **do**
    MACA(G)
    Drobljenje(G)
  endfor
end

3.3 Hibridni algoritem

Pomemben element hibridnega algoritma je t.i. vektorska razdelitev (VQ iz angl. *Vector Quantization*) [8], tj. hevristična metoda, ki je zaradi načina delovanja primerna predvsem pri raznih simulacijah (npr. delovanje možganov). Srečamo jo tudi v podsistemih pri zahtevnejših praktičnih aplikacijah, kot so premikanje robota, kompresija slik in govora, prepoznavanje govora in kodiranje (stiskanje) vektorjev.



Slika 3. Uporabniški vmesnik Figure 3. User interface



Slika 4. Vizualizacija razdelitve mreže Figure 4. Mesh-partitioning vizualization

Graf	$\lvert V\rvert$	$\lvert E\rvert$	Chaco 2.0	p-Metis 4.0	m-MACA	h-MACA
grid1	252	476	26	20	**18**	**18**
grid1_dual	224	420	16	16	16	**10**
netz4504_dual	615	1171	21	21	**19**	**19**
U1000.5	1000	2394	10	**1**	**1**	2
U1000.10	1000	4696	115	56	**39**	39
U1000.20	1000	9339	294	253	**220**	221
ukerb1_dual	1866	3538	25	25	**22**	**22**
netz4504	1961	2578	25	26	**22**	24
grid2_dual	3136	6112	35	**32**	**32**	**32**
grid2	3296	6432	38	37	**34**	**34**
airfoil	4253	12289	82	85	**74**	81
3elt	4720	13722	124	98	**90**	**90**
ukerb1	5981	7852	30	28	**27**	28
airfoil_dual	8034	11813	60	40	**37**	40
3elt_dual	9000	13287	70	70	**44**	45
whitaker3	9800	28989	135	128	**126**	127
crack	10240	30380	209	206	**184**	196
big	15606	45878	242	165	141	**139**
whitaker3_dual	19190	28581	82	74	**64**	65
crack_dual	20141	30043	130	101	**80**	87
big_dual	30269	44929	92	92	78	**77**

Tabela 1. Eksperimentalni rezultati ($k = 2$) Table 1. Experimental results ($k = 2$)

Graf	$\lvert V\rvert$	$\lvert E\rvert$	Chaco 2.0	p-Metis 4.0	m-MACA	h-MACA
grid1	252	476	48	40	**38**	**38**
grid1_dual	224	420	37	**35**	**35**	**35**
netz4504_dual	615	1171	54	49	**44**	**44**
U1000.5	1000	2394	20	6	**7**	12
U1000.10	1000	4696	200	108	**99**	107
U1000.20	1000	9339	554	515	546	**497**
ukerb1_dual	1866	3538	56	51	52	**48**
netz4504	1961	2578	66	62	50	**49**
grid2_dual	3136	6112	99	91	**90**	96
grid2	3296	6432	106	121	94	**92**
airfoil	4253	12289	182	179	**176**	190
3elt	4720	13722	258	252	252	**212**
ukerb1	5981	7852	82	64	63	**61**
airfoil_dual	8034	11813	111	84	**80**	110
3elt_dual	9000	13287	130	120	**112**	154
whitaker3	9800	28989	439	424	**383**	**383**
crack	10240	30380	457	458	377	**371**
big	15606	45878	416	405	**354**	382
whitaker3_dual	19190	28581	251	210	200	**195**
crack_dual	20141	30043	228	201	169	**164**
big_dual	30269	44929	219	**196**	215	222

Tabela 2. Eksperimentalni rezultati ($k = 4$) Table 2. Experimental results ($k = 4$)

Vektorska razdelitev je tehnika, ki pri svojem delovanju izkorišča osnovno strukturo vhodnih vektorjev. Povedano bolj natančno, vhodni prostor je razdeljen v določeno število različnih področij in za vsako področje je definiran t.i. rekonstrukcijski vektor. Razdelilnik (ta je lahko preslikovalna funkcija ali naprava) določi za vsak vhodni vektor območje, kateremu vektor pripada, in ga predstavi s pripadajočim rekonstrukcijskim vektorjem za to območje. S tem ko uporabimo za shranjevanje namesto originalnega vhodnega vektorja rekonstrukcijski vektor, ki je kodiran, pridobimo na prostoru ali pasovni širini prenosa. Seveda je slaba stran uporabe rekonstrukcijskega vektorja, da lahko pride do manjšega popačenja. Zbirki vseh rekonstrukcijskih vektorjev pravimo kodirna knjiga, njenim elementom pa kodirne besede.

Težava algoritma MACA je, kako dobiti dobro začetno rešitev, iz katere bi lahko dosegli globalni optimum. Ena od možnosti je, da z VQ naredimo relativno dobro začetno rešitev, ki jo nato izboljšamo z osnovnim algoritmom MACA. Tako dobimo hibridni algoritem h-MACA:

algoritem h-MACA
  VQ(G)
  Inicializacija(G)
  MACA(G)
end

S takšnim hibridnim algoritmom občutno zmanjšamo razpršenost dobljenih rešitev, pri čemer dosežemo podobna kakovost najboljših rešitev kot pri večnivojskem algoritmu.

4 Ocena učinkovitosti algoritmov

4.1 Eksperimentalno okolje

Vsi prej omenjeni algoritmi so bili programirani v jeziku Borlandr Delphi™. Eksperimenti so bili narejeni na računalniku z AMD Athlon™XP 1800+ procesorjem in operacijskim sistemom Microsoftr Windowsr XP Professional. Eksperimentalno okolje daje tudi ponazoritev trenutnega stanja in dogajanja algoritma ter orodja, s katerimi lahko uporabnik poljubno nastavlja parametre algoritma. Slika 3 ponazarja potek reševanja problema, na sliki 4 pa je vizualizacija razdelitve mreže. Desno od polja celic so instrumenti, s katerimi nadzorujemo potek algoritma. Z njimi lahko nastavimo velikost polja celic, število mravelj v eni koloniji, želeno neuravnoteženost, največje mogoče število ponovitev brez izboljšave na enem nivoju, pomembnost hevristične informacije, največje število nabranih koščkov hrane in dolžino tabu seznama. Vsi drugi podatki pomenijo trenutno stanje algoritma.

4.2 Rezultati

Za preskus smo vzeli grafe s spletne strani Graph Collection z Univerze v Paderbornu. Da bi ocenili algoritem MACA (tako večnivojski, kakor tudi hibridni), smo letega primerjali z znanima razdelitvenima algoritmoma p-METIS 4.0 [9] in Chaco 2.0 z večnivojsko razdelitveno metodo Kernighan-Lin [10]. Vsakega izmed preskusnih grafov smo razdelili na dve in štiri domene ($k = 2$ in $k = 4$). Algoritme smo ocenjevali z rezom $\zeta(D)$, pri čemer je pomembno omeniti, da je bila neuravnoteženost $b(D)$ vedno manj kot 0,2% od $|V|$. Rezultati so prikazani v tabelah 1 in 2. Ugotavljamo, da nam v večini primerov vrneta najboljši rezultat prav algoritma MACA.

5 Sklep

Sklenem lahko, da sta tako večnivojski kakor tudi hibridni algoritem s kolonijami mravelj zelo obetavna, vendar ju je treba še zelo natančno raziskati. Prva in verjetno najbolj očitna izboljšava bi bila njuna združitev. Da bi to naredili, bi lahko uporabili vektorsko razdelitev za gradnjo začetne razdelitve, nato bi graf do določene mere skrčili (a veliko manj kot pri originalni večnivojski metodi) in uporabil večnivojsko metodo za drobljenje in razdeljevanje.

6 Literatura

[1] K. Ho-Le, K, Finite-element mesh generation methods: a review and classification, *Comp. Aided Design 20*, 1988, pp. 27-38.

[2] N. Bouhmala, X. Cai, Partition of unstructured finite-element meshes by a multilevel approach, *Lect. Notes Comput. Sc. 1947*, 2001, pp. 187-195.

[3] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of Incompleteness*, W. H. Freeman and Company, 1979.

[4] T. N. Bui, C. Jones, Finding good approximate vertex and edge partitions is NP hard, *Infor. Proc. Lett. 42*, 1992, pp. 153-159.

[5] A. E. Langham, P. W. Grant, Using competing ant colonies to solve k-way partitioning problems with foraging and raiding strategies, *Lect. Notes Comput. Sc. 1674*, 1999, pp. 621-625.

[6] P. Korošec, J. Šilc, B. Robič, Solving the mesh-partitioning problem with an ant-colony algorithm, *Parall. Comput. 30*, 2004, pp. 785-801.

[7] S. T. Barnard, H. D. Simon, A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems, *Concurrency-Pract. Ex. 6*, 1994, pp. 101-117.

[8] Y. Linde, R. M. Gray, An algorithm for vector quantizer design, *IEEE Trans. Commun. 28*, 1980, pp. 84-95.

[9] G. Karypis, V. Kumar, Multilevel k-way partioning scheme for irregular graphs, *J. Parallel Dist. Comput. 48*, 1998, pp. 96-129.

[10] B. Hendrickson, R. Leland, A multilevel algorithm for partitioning graphs, *Proc. Supercomputing '95*, 1995.

Peter Korošec je mladi raziskovalec na Institutu Jožef Stefan v Ljubljani. Njegovo raziskovalno področje je uporaba metahevrističnih optimizacijskih metod pri numeričnem in kombinatoričnem optimiranju.

Jurij Šilc je višji znanstveni sodelavec na Odseku za računalniške sisteme Instituta Jožef Stefan v Ljubljani. Raziskovalno se ukvarja s procesorsko arhitekturo, visokonivojsko sintezo in metahevrističnim optimiranjem.

Borut Robič je redni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer predava na dodiplomskem in podiplomskem študiju. Področja, za katera se raziskovalno zanima, zajemajo algoritme, zahtevnost računanja in vzporedno računanje.