T-79.194 Tietojenkäsittelyteorian seminaari
-
SP-toteutuvuusratkaisimen kokeellinen tarkastelu

Martti Meri
mmeri@cs.hut.fi

17. Huhtikuuta 2003

Abstract

sp-toteutuvuustarkastin perustuu esikäsittelyyn, jossa totuusarvoltaan selvästi joko todeksi tai äpätodeksi biasoituvat muuttujat kiinnitetään vaiheittain ja jäännökseksi jäävät alilauseet annetaan varsinaisten toteutuvuustarkistimien ratkaistaviksi, kun biasointi laskelmat konvergoivat ja enempää muuttujia ei voi enää kiinnittää.

Tässä työssä tarkastellaan sp-ratkasimen suorituskykyä suhteessa muuttujien määrään hard-sat alueella ja sen ympäristössä. Päähuomio on satunnaisissa 3SAT ongelmissa, joiden ratkaisemiseen sp onkin ensisijaisesti kehitetty.

Ratkaisimena sp-on oikeellinen (correct) mutta ei täydellinen (noncomplete). Analyysin kannalta kiintoisaa on varsinaisen toteutuvuustarkistimen käyttö sp:n sisällä. sp kutsuu walksat-ratkaisinta varsinaisena toteutuvuustarkistimena. Periaatteessa olisi kiintoisaa kyetä arviomaan mikä osuus lauseista, joille sp antaa arvon PROBABLY UNSAT johtuu siitä, että biasointilaskelmat eivät suppene, ja mikä osuus siitä, että ne suppenevat, mutta walksat ei. Analyysissä käytetään kokeellista lähestymistapaa. Kokeen tarkoituksena on tarkastella sp:n suorituskykyä hard-sat alueella ja faasimuunnoksen vaikutusta suoritusaikoihin ja algoritmin suppenemiseen. Kokeet on suoritettu 10000 muuttujan satunnaisilla 3SAT lauseilla joissa klausuuli/muuttuja-suhde vaihtelee välillä 3.90 - 4.29.

1 Johdanto

SP-ratkaisin [BMZ02] on suunniteltu satunnaisten 3SAT-ongelmien ratkaisuun. Lauseita voidaan esittää tekijägraafeina (factor graph) jotka ovat bipartitegraafeja, joissa on muuttujasolmuja ja lausekesolmuja, sekä kahdenlaisia kaaria näiden välillä. Ensimmäisen tyyppinen kaari lausekesolmusta menee muuttujasolmuun, jos lausekkeessa on muuttujaa vastaava positiivinen literaali ja toisen tyyppinen, jos siinä on vastaava negatiivinen literaali.

Kun suuresta muuttujajoukosta jonka koko on N, arvotaan muuttujakolmikoita tasajakauman todennäköisyydellä lausekkeisiin joita on M ($M = \alpha * N$) kappaletta,esiintyy tiettyä muuttujakolmikkoa ($i < j < k$) vastaava lausekesolmu tekijägraafissa todennäköisyydellä $6 * \alpha / N^2$. Suurelle N:n arvolle muuttuja solmujen asteet ovat *independent identically distributed (iid)* satunnaismuuttujia, jotka ovat Poisson jakautuneita (keskiarvolla 3α) [BMZ02].

2 Uskomusten propagointi

sp:ssä perustuu uskomusten vyöryttämiseen tekijägraafissa [BMZ02]. Ohessa esitettävät tietorakenteet ja pseudoalgoritmit on kuvattu alkuperäisen lähteen pohjalta englanniksi, koska muutoin olisi pitänyt tarkistaa fysiikan sanakirjoista osa termejä. Käsitteistö perustuu osin fysikan sanastoon. Algoritmi käy läpi verkon kaaria satunnaisessa järjestyksessä ja päivittää niille seuraavia ympäristön tilaa kuvaavia muuttujia [BMZ02].

cavity field

$$h_{j\to a} = \sum_{b\in V_+(j)-a} u_{b\to j} - \sum_{b\in V_-(j)-a} u_{b\to j}$$

Tämä muuttuja ilmaisee sen kuinka suuressa määrin tekijägraafin muuttujasolmun j takaa klausuulisolmun a suunnasta katsoen solmu j on saanut ehdottomia toiveita arvonsa suhteen muilta klausuulisolmuilta. Kun laskenta on korvergoinut saadaan tästä laskettua muuttujan toivottu tila.

cavity bias

$$u_{a\to i} = \prod_{j\in V(a)-i} \theta\left(h_{j\to a} J_j^a\right)$$

Tämä muuttuja ilmaisee hälytyksen klausuulisolmulta muuttujasolmulle. $\theta(x)$ on porrasfunktio joka saa arvon 0 jos $x/leq0$ ja arvon 1 muuten. $J_j^a = 1$ jos muuttuja j esiintyy negatiivisena literaalina lauseessa a, muutoin se saa arvon 0. Intuitiivisesti hälytys markitsee sitä että klausuulin kannalta vastaanottava muuttuja on ainoa joka voi saada lauseen tässä tilanteessa toteutumaan.

contradiction number

$$c_i = 1if \qquad \left(\sum_{b\in V_+(j)-a} u^*_{b\to j}\right) * \left(\sum_{b\in V_-(j)-a} u^*_{b\to j}\right) > 0$$

$$c_i = 0 \qquad muuten$$

Kun laskentaa (solmujen läpikäyntiä) iteroidaan suppenee laskenta joissain tapauksissa ja tällöin kiintopisteessä saadut arvot edellisille toimivat perustan sille kiinnitettävien muuttujien valinnalle. Iteraatiokierrosten maksimimäärä ja kerrallaan kiinnitettävien biasoituneiden muuttujien määrä ovat sp:lle annettavia syöteparametreja.

3 Uskomusten propagoinnin algoritmi

sp alustaa satunaiset arvot $cavitybias$ ja $cavityfield$ muuttujille kullekin kaarelle. Kukin iteraatiokierros käy läpi kaaret satunnaisessa järjestyksessä. Jos laskenta suppenee asetetaan kiintopisteen arvot muuttujien kiinnitystä varten.

BP algoritmi

- INPUT: the factor graph of Boolean formula in CNF ; a maximal number of iterations t_{max}

- OUTPUT: UN-CONVERGED if BP has not converged after t_{max} sweeps. If it has converged: a set of all cavity biases $u^*_{a \to i}$

  1. At time t=0: For every edge $a \to i$ of the factor graph, randomly initialize the cavity bias
  2. For $t = 1$ to $t = t_{max}$:
     (a) sweep the set of edges in a random order and update sequentially the cavity biases on all the egdes of the graph, generating the $u_{a \to i}(t)$, using the equations above
     (b) If $u_{a \to i}(t) = u_{a \to i}(t-1)$ for all edges set $u*$ and goto 3.
  3. If $t = t_{max}$ return UN-CONVERGED. If $t < t_{max}$ set the fixed point cavity biases $u*_{a \to i} = u_{a \to i}(t)$

4 Surveyn inspiroima algoritmi (SID)

Uskomusten vyöryttämisen perustalle rakentuva algoritmi voidaan yleistää koskemaan uskomuksia koskevien todennäköisyysjakaumien vyöryttämistä[BMZ02]. Sovelletaan Min-Sum-algoritmia SAT ongelmaan, joka on rajoittunut versio Sum-Product algoritmista, joka on perustuu siihen että evaluoidaan rajatodennäköisyyksiä vuorovaikutuksessa olevien diskreettien muuttujien osalta yli verkon.

Alla esitetyssä pseudokielisessä ohjelmassa esiintyvä SP on edellä esitetyn BP:n todennäköisyysjakaumien käsittelyyn yleistetty versio.

SID algoritmi

- INPUT: the factor graph of Boolean formula in CNF ; a maximal number of iterations t_{max} and a precision ϵ used in SP
- OUTPUT: satisficing truth assignment, or SP UN-CONVERGED, or PROBABLY UNSAT

  1. Random initial condition for the surveys
  2. Run SP
  3. Evaluate all the biases $W_i^{(+)}, W_i^{(-)}, W_i^{(0)}$ and the complexity $\sum$
  4. Simplify and Search
     (a) If non-trivial surveys are found , then fix the variable with the larges $|W_i^{(+)} - W_i^{(-)}|$ and **simplify** the problem.
     (b) If all surveys are trivial then output the simplified sub-formula and run on it a local search process (e.g. walksat).
     (c) If SP does not converge, output PROBABLY UNSAT and stop (or restart, i.e. go to 1)
  5. If the problem is solved completely by unit clause propagation, then output SAT and stop. If no contradiction is found then continue the decimation process on the smaller problem (go to 2.) else (if a contradistion is reached) stop.

Mikään ei kai suoranaisesti edellytä juuri paikalliseen satunnaiseen haluun perustuvan ratkaisimen käyttöä suppenemista ja muuttujien kiinnitystä seuraavassa vaiheessa, jossa alilause jää ratkaistavaksi.

Toteutus yhdistää kiinnittämällä saadut muuttujat ja jäännöslauseen ratkaisun yhdeksi malliksi joka uudelleen tarkistetaan suhteessa alkuperäiseen kokonaislauseeseen. Tämän toimenpidesarjan tarpeellisuus tuntuu kyseenalaiselta koska sekä iterointivaihe että walksat ovat oikein toimivia.

5 Menetelmät

Laajin käyttöohje sp-ratkaisimen käytöstä löytyy kun ohjelmalta kysyy neuvoa sen käytöstä (listaus alla).

```
mmeri@kass:~/report$ sp -h
sp [options]

  formula
        -n <numvars>
        -m <numclauses>
        -a <alpha>
        -l <filename> reads formula from file
  solving
        -d <danger> fix this much PAP spins (experimental)
        -f <fixes> per step
        -% <fixes> per step (%)
        -e <error> criterion for convergence
        -i <iter> maximum number of iterations until convergence
  stats
        -p <filename> output a magneticity plot
        -r <filename> replay file
        -c computes complexity
        -F print fields
        -v increase verbosity
  misc
        -s <seed> (0=use time, default=1)
        -/ stop after first convergence
        -h this help
```

Ratkasimen kokeelliseen tutkimiseen liittyy kolme keskeistä kysymystä:

- Mitkä ovat kullekin ongelmainstanssityypille (ja pahimmassa tapauksessa instanssille) ne lähes optimaaliset parametrin arvot, joilla kokeita tulee tehdä jotta ratkaisimien vertailu tapahtuu tasavertaiselta pohjalta?
- Miten ratkaisimen parametrit ovat vertailtavissa muun tyyppisten, kuten systemaattisten tai paikalliseen satunnaiseen hakuun perustuvien ratkaisimien parametreihin?
- Miten sp ja sen sisällä oleva 'oraakkeli'- ratkaisin (tässä walksat) suhteutuvat toisiinsa parametrisoinnin suhteen?

Ensimmäisellä kysymyksellä on suurta käytännön merkitystä sekä tulosten että kokeiden suorituksen kannalta.

Toisella kysymyksellä on suurta merkitystä esimerkiksi analysoitaessa ssymptoottista käyttäytymistä iteraatiomäärien (ja ajoajan) suhteen. Tällä hetkellä ilmeisestikin iteraatiomäärät ja kiinnitettävien muuttujein määrät pitää selvittää kokeellisesti ja muiden ratkaisin tyyppien menetelmätutkimus ei tällä alueella anna perustaa parametrien virittämiseen.

Kolmanteenkin kysymykseen liittyy lukuisia avoimia seikkoja. Yksikkövyörytys (unit clause propagation) kannattaa varmasti tehdä ylätasolla arkkitehtuurissa, kuten ohjelmassa tehdään, mutta on hieman epäselvää missä konvergoinnin jälkeen lauseeseen jäisi sellaisia yksikkölauseita, joita ei voitaisi kiinnittää. Kaiken kaikkiaan tutkimuksen kannalta olisi varmaan hyvä, jos tätä biasointien konvergoittamisperiaatetta voisi tarkastella sekä riippumatta muista ratkaiuista, että yhdessä muiden ratkaisun osioiden kanssa, hakien erilaisten yhdistelmien keskinäistä optimaalista integraatiota.

Yleiset benchmarkit ja järjestelyt erilaisten ratkaisimien vertailuun ovat olleet kehitteillä maailmalla. Seminaari tarjoaa benchmark-valikoiman ja kyseessä onkin ratkaisimien suorituskyvyn vertailu. Joten vaikka sp ei olekaan tarkoitettu muiden kuin satunnaisten 3SAT ongelmien ratkaisemiseen on sitä tässä pyritty ajamaan myös annetuilla muilla testiongelmilla.

Se luodaanko satunnaisia 3SAT instansseja hard-sat alueelle $\alpha_d < \alpha < \alpha_c$ vai sen ylä tai alapuolelle on kokeen kannalta eräs varioitavissa olevista parametreista. Lähteen α_d on noin 3.921 ja α_c on noin 4.267. Tältä pohjalta arvo 4.26 on valittu koesarjan perustarkastelupisteeksi vaikeiden lauseiden l. kriittisen klausuulien ja muuttujien määrien suhteen synnyttämisessä. Näin voidaan tutkia sp:n toimintaa faasinmuunnosvaiheessa, jossa tilanteesta jossa on merkittävä osa toteutuvia instansseja, siirrytään suhteen hieman kasvaessa tilanteeseen, jossa niitä ei juurikaan ole.

Koska satunnaiset ongelmat voivat samoilla parametreilla vaihdella hyvin vaikeista hyvin helppoihin, pitäisi koeinstanssijoukon koko olla mieluusti satoja instansseja. Satunnaiseen lokaaliin hakuun perustuvia ratkaisimia pitäisi lisäksi samalla instanssilla ajaa useaan kertaan johtuen ratkaisimen sisäisestä satunnaisuudesta.

Sat-ratkaisimien lokaalien hakualgoritmien kokeellisen tutkimuksen menetelmäkehityksen [HS00] puolella kokeiden suoritukselle annetaan seuraavia suuntaviivoja:

- kokeiden ja koeinstannien pitäisi soveltua joukolle ratkaisimia
- kokeiden olisi tapahduttava kunkin ratkaisimen lähes optimaalisilla parametreilla
- yhdelle instanssille kokeet suoritetaan riittävällä testiajojen määrällä

Parametrit eri ratkaisimissa ovat toisistaan poikkeavia. Uudelleen käynnistyksen määrä ja muuttujan kääntö-iteraatioiden määrä voivat olla periaatteessa samankaltaisessa roolissa.

Lokaaliin hakuun perustuvissa menetelmissä on käytössä alla esitettyjä ajo- ja kohinaparametreja.

Ajoparametrit

maxTries maksimimäärä sallittavia uudelleen käynnistyksiä

maxSteps maksimimäärä sallittavia hakuaskeleita

sp:ssä on mekanismi käynnistää ajo uudelleen satunnaisesti alustetuilla alkuarvoilla. sp:ssä iteraatioiden maksimimäärä ja kerralla kiinnitettävien muuttujien määrä ovat sukua maxTries ja maxSteps - parametrityypille, vaikka eroaakin tämän tyypin parametreista monessa suhteessa.

Kohinaparametrit:

wp todennäköisyys että iteraatioaskeleella GSAT askel eikä satunnaisvaellusaskel

tl kierrosmäärä jonka aikana ei saa kääntää samaa muuttujaa

p historiaan taletukseen perustuvissa muuttujan valintaa ohjaavissa menetelmissä todennäköisyys että tilanteessa jossa muuttuja jolla on korkein lukema (score) mutta joka on viimeksi käännetty käännetään uudestaan.

Parametri wp on tärkeä, koska jos se on suurempi kuin nolla, lokaaleista minimeistä päästään karkuun jollain nollaa suuremmalla todennäköisyydellä.

Satunnaisuus sp:ssä ilmenee alustusvaiheessa ja siinä missä järjestyksessä kullakin iteraatiokierroksella kaaret käydään läpi.

6 Huomioita

Tuntuu hieman yllättävältä, että kun Max-Sum-menetelmä on laajassa käytössä kombinatorisen laskennan puolella tekijägraafeja [KFL01] läpikäytäessä, niin että sp:n kannalta keskeistä pulmaa , eli tekijägraafin solmujen kautta syntyvää riippumattomuutta, ei olisi tarkemmin selvitetty.

Nyt SID algoritmi yleistetään BID algoritmista ja keskeinen teoria jälkimmäiseen nojaa läpikäytävän verkon puumaisuuteen. Perusteluna esitettävät todennäköisyydet riittävän suurten 3-SAT ongelmien osalta yhden solmun naapureiden riippumattomuudesta (tarkastelusolmussa tapahtuva cut saa naapurit riippumattomiksi) on kyllä järkevä oletus.

SP on tekijöiden mukaan kovin heuristinen, mikä tarkoittaa kaiketi paitsi sitä, mitä tekijät sanovat, että heillä ei ole todistusta konvergenssista, myös mielestäni sitä, että toimintaperiaate on järkeenkäypä: iteroiden käydään kullakin iteraatiokerralla verkko läpi ja propagoidaan kuhunkin solmuun ympäristöstä todennäköisyysjakaumat naapureiden saamista varoituksista. Näistä muodostetaan käsitys biasoitumisesta. Kun iteraatio konvergoi, katsotaan mitä solmuja voidaan alkaa kiinnittää arvon suhteen.

Artikkeli [BMZ02] jättää arvioimatta false negative tulosten osuuden ja syntymekanismit. Olisi hyödyllistä kyetä aavistamaan, kuinka monia ratkaisuja ja miten jä löytämättä. Biasointi voi olla **under constained** , **biased** tai **balanced** . Algoritmi kiinnittää biasoituneita muuttujia, mikä tarkoittaa toimimista varmalla pohjalla.

Artikkeli mainitsee mahdollisuuden kiinnittää balanced-muuttujia, mikä tarkoittaa siis niitä muuttujia, joille haun tulokset ilmaisevat olevan erisuuntaista painetta (joidenkin kannalta olisi parempi ottaa totuusarvo 0 ja joidenkin kannalta 1). Näiden muuttujien kiinnittäminen tarkoittaisi siis hakuavaruuden puolittamista kahtia, jolloin puolet ratkaisuista oletettavasti jäisi puoliskoon, jota ei käydä läpi.

Table 1: Satunnais-3SAT kokeet

ajo	vars	suhde	toteutuvia	ajoaika	keskihajonta
1	10000	3.90	200/200	5.18	0.800
2	10000	4.20	178/200	48.1	78,74
3	10000	4.22	126/200	103	121
4	10000	4.23	67/200	190	157
5	10000	4.24	41/200	265	171
6	10000	4.25	16/200	334	145
7	10000	4.26	9/200	389	135
8	10000	4.29	1/200	529	114

Table 2: Kirjastobenchmarkit

testi	tulos	aika (s)
bmc-ibm-1.cnf	ei konvergoi	440
bmc-ibm-2.cnf	ei konvergoi	20
bmc-ibm-3.cnf	ei konvergoi	780
bmc-ibm-4.cnf	ei konvergoi	1060
bmc-ibm-5.cnf	ei konvergoi	350
bmc-ibm-6.cnf	ei konvergoi	2850
bmc-ibm-7.cnf	ei konvergoi	290
bmc-galileo-8.cnf	ei konvergoi	4780
bmc-galileo-9.cnf	ei konvergoi	5570
bmc-ibm-10.cnf	ei konvergoi	3370
bmc-ibm-11.cnf	ei konvergoi	2310
bmc-ibm-12.cnf	ei konvergoi	2
bmc-ibm-12.cnf	ei konvergoi	770

7 Kokeet

Satunnaisten 3SAT ongelmien koesarja on suoritettu iteraatioparametrin arvolla 10000 ja kiinnittämällä kerrallaan 100 muuttujaa (mikä on prosentti 10000 muuttujasta).

Ajojen yhteenveto on esitetty taulukossa (Table 1.). Kuhunkin otokseen kuului 200 instanssia. Taulukossa on esitetty SAT-tulosten osuus instansseista, keskimääräinen ajoaika otoksissa ja tämän keskihajonta.

Tekijöiden haun osalta kokeet jäivät puutteellisiksi ja mitään johtopäätelmiä ei tässä ole syytä esittää.

Kirjastoesimerkkien ajojen yhteenveto on esitetty taulukossa (Table 2.)

Koska toisella ajokerralla testi jäi suorittamaan bmc-ibm-6.cnf ongelmaa yli kuudeksi tunniksi (cpu-aikaa) näyttää siltä, että propagoinnin järjestys tai muu seikka vaikuttaa voimakkaasti aikaan, joka vaaditaan, jotta sp palauttaa tiedon että haku ei konvergoi.

8 Yhteenveto

Menetelmältään sp on kiintoisa ja lupaava ratkaisin. Kokeet osoittavat huomattavaa suorituskykyä. Analyysi ei oikein, johtuen ohjelmiston tulosteiden palautusarvojen vaikealukuisuudesta kyennyt pureutumaan PROBABLY UNSAT ajojen sisäiseen jakaumaan.

Ajot pienemmillä muuttujamäärillä ja sekä kirjasto-instanssien että tekijöiden ajon osalta suoritettiin puutteellisin parametrein, joten niiden perusteella on vaikea sanoa mitään. Näyttää kuitenkin vahvasti siltä että biasoinnit rakenteisissa ongelmissa menevät niin ristiin, että menetelmä ei niihin pure. Pienempien instanssien osalta saatiin viitteitä siitä että tutkimuksen johdannossa viitattu seikka, eli se, että pienet 3SAT instannit ovat suhteessa vaikeampia ja niiden joukkoon sattuu enemmän unsat-instansseja voi vaikuttaa asiaan ja että sp:llä on luonnollinen alaraja instanssien koon suhteen.

Kokeellisen tutkimusmetodiikan kannalta tämä tarkastelu tuo esiin tarpeita hybridiratkaisimien, kuten sp on, käsittelyyn erilaisista näkökulmista. Kun arkkitehtuuri koostuu erilaisista osista, on sen virittäminen vaikeampaa, ja voidaan kysyä minkälaisista osista se tulisi erilaisiin tarpeisiin koota. Olisi mielenkiintoista, jos sp:n rajapinta lokaaliin satunnaishakuun käytettävään ohjelmaan olisi spesifioitu ja standardoitu niin, että erilaisia ratkaisimi voisi siihen kytkeä. Edelleen kehiteltynä tämä saattaisi johtaa protokollaan, joka saa muuttujien kiinnittämisen ylätasolla ja lokaalin haun alatasolla jollain tavoin sellaisiksi, että niiden välillä voisi älykkäästi hypätä (intelligent backjumping). Mutta kaikenkaikkiaan erilaisten menetelmien yhdistäminen on varmasti alue, jossa riittää tutkimussarkaa.

References

[BMZ02] A. Brauenstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *CC/0212002*, 2002.

[HS00] Holger H. Hoos and Thomas Stutzle. Local search algorithms for SAT: An empirical evaluation. *Journal of Automated Reasoning*, 24(4):421–481, 2000.

[KFL01] F. Kschischang, B. Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), February 2001.