Paikalliset (stokastiset) hakumenetelmät

Käsiteltäviä asioita:

1. Paikalliset hakumenetelmät
2. Toteutuvuus haku- ja optimointiongelmana
3. Paikallinen haku toteutuvuusongelmassa
4. Paikalliseen hakuun perustuvia algoritmeja:
   — Anhe paikallinen haku
   — Simuloitu jäähdytys
5. Ahneen paikallisen haun muunnelmia
6. Kokeellisia vertailuja

1. Paikalliset hakumenetelmät

➤ Paikalliset hakumenetelmät kuten *simuloitu jäähdytys* ovat suosittuja kombinatoristen (optimointi)ongelmien ratkaisemisessa.

➤ Paikallisen haun perusideat ovat seuraava:

(i) Aloitetaan satunnaisesti generoidusta ratkaisuehdokkaasta.

(ii) Pyritään etenemään paikallisin muutoksin kohti parempaa ratkaisuehdokasta.

(iii) Tehdään paikallisia muutoksia joskus myös satunnaisesti, mikä saattaa huonontaa ratkaisuehdokasta.

➤ Paikalliset hakumenetelmät ovat *epätäydellisiä*: ratkaisun löytyminen ei ole taattua, vaikka sellainen on olemassa.

2. Toteutuvuus haku- ja optimointiongelmana

➤ Hakuongelma (**SAT**): löydettävä annetulle lausejoukolle malli.

➤ Optimointiongelma (**MAX-SAT**): löydettävä annetulle lausejoukolle totuusjakelu, jossa mahdollisimman moni joukon lauseista on tosi.

➤ Tyypillisesti käsitellään konjunktiivisessa normaalimuodossa olevia lausejoukkoja (klausuulijoukkoja).

Esimerkki. Olkoon joukko $S = \{A \lor B, \neg A \lor B, A \lor \neg B\}$.

➤ Joukon S toteutuvuusongelmalle löytyy ratkaisu $\{A, B\}$.

➤ Joukko $S \cup \{\neg A \lor \neg B\}$ on toteutumaton, mutta $\emptyset$, $\{A\}$, $\{B\}$ ja $\{A, B\}$ ovat vastaavan optimointiongelman ratkaisuja.

3. Paikallinen haku toteutuvuusongelmassa

➤ Paikallinen hakumenetelmä toteutuvuusongelman tapauksessa:

(i) Aloitetaan satunnaisesti generoidusta totuusjakelusta.

(ii) Suoritetaan paikallisia muutoksia vaihtamalla atomilauseiden totuusarvoja (engl. termi *flip*).

(iii) *Ahne haku*: muutetaan totuusarvo atomilauseelta, joka johtaa suurimpaan laskuun epätosien klausuulien määrässä.

➤ **Ongelma:** miten päästään pois *paikallisista minimeistä* (totuusjakeluista, joissa minkään atomilauseen totuusarvon muuttaminen ei laske epätosien lauseiden määrää)?

(a) Haun aloittaminen uudelleen

(b) Sivuttaissiirrot (epätosien klausuulien määrä ei laske)

(c) Satunnaiset siirrot (epätosien klausuulien määrä voi kasvaa)

Esimerkki. Tarkastellaan klausuulijoukon $S = \{A \vee B, \neg A \vee B, A \vee \neg B\}$ osoittamista toteutuvaksi paikallisella haulla:

Satunnainen totuusjakelu:		Aloitetaan uudelleen:	
$\{\neg A, \neg B\}$		$\{\neg A, B\}$	
(1)		(1)	
$\{A, \neg B\}$	$\{\neg A, B\}$	$\{A, B\}$	$\{\neg A, \neg B\}$
(1)	(1)	(0)	(1)
Paikallinen minimi.		Malli löytyi.	

- Yllä totuusjakeluille $\mathcal{A}$ on käytetty *literaaliesitystä* $\mathrm{lit}(\mathcal{A}) = \mathcal{A} \cup \{\neg A \mid A \in \mathcal{P} - \mathcal{A}\}$.
- Joukon S epätosien klausuulien lukumäärä on annettu suluissa.

Esimerkki. Tarkastellaan klausuulijoukkoa

$$S = \{ \quad A \vee \neg B,\ A \vee \neg C,\ B \vee \neg A,\ B \vee \neg C,\ C \vee \neg A,\ C \vee \neg B,\ \neg A \vee \neg B \quad \}.$$

- Totuusarvomuutokset voivat johtaa aina huonompiin malleihin:

	$\{A, B, C\}$	
	(1)	
$\{A, \neg B, C\}$	$\{\neg A, B, C\}$	$\{A, B, \neg C\}$
(2)	(2)	(3)

☞ Turvaudutaan tarvittaessa satunnaiseen siirtymään.

4. Paikalliseen hakuun perustuvia algoritmeja

- *Ahne* paikallinen haku (**GSAT**)

  B. Selman, H.J. Levesque, and D.G. Mitchell [AAAI, 1992]:
  A New Method for Solving Hard Satisfiability Problems.

- *Simuloitu jäähdytys* (**SA**)

  D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon [Operations Research, 1991]:
  Optimization by Simulated Annealing: an Experimental Evaluation; part I, Graph Coloring and Number Partitioning .

  W.M. Spears [Techical Report, 1993]:
  Simulated Annealing for Hard Satisfiability Problems.

Joitain määritelmiä

Olkoon $\mathcal{A}$ totuusjakelu, S klausuulijoukko ja P atomilause.

- $\mathrm{Hb}(S)$ on klausuulijoukossa S esiintyvien atomilauseiden joukko.
- $\mathrm{Flip}(\mathcal{A}, P)$ on totuusjakelu, joka eroaa totuusjakelusta $\mathcal{A}$ ainoastaan atomilauseen P osalta.
- $\mathrm{Random}(p)$ on funktio, joka palauttaa totuusarvon tosi todennäköisyydellä $0 \leq p \leq 1$.
- $\mathrm{RC}(J)$ on funktio, joka palauttaa satunnaisen alkion joukosta J.
- $\mathrm{False}(S, \mathcal{A}) = |\{C \in S \mid \mathcal{A} \not\models C\}|$, joten $\mathrm{False}(S, \mathcal{A}) = 0 \implies \mathcal{A} \models S$.
- $\Delta(\mathcal{A}, P) = \mathrm{False}(S, \mathrm{Flip}(\mathcal{A}, P)) - \mathrm{False}(S, \mathcal{A})$.

Ahne paikallinen haku (GSAT)

Function GSAT(S):
for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A} := \mathrm{RC}(\mathbf{2}^{\mathrm{Hb}(S)})$;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else
      $P := \mathrm{RC}(\{Q \in \mathrm{Hb}(S) \mid \Delta(\mathcal{A}, Q) = \min_{R \in \mathrm{Hb}(S)} \Delta(\mathcal{A}, R)\})$;
      $\mathcal{A} := \mathrm{Flip}(\mathcal{A}, P)$
    endif
  endfor
endfor

Huom! $\mathrm{RC}(\mathbf{2}^{\mathrm{Hb}(S)})$ voidaan toteuttaa lineaarisessa ajassa ja tilassa.

Esimerkki. Tarkastellaan klausuulijoukkoa $S =$

$$\{\ A \lor B \lor C,\quad A \lor B \lor \neg C,\quad A \lor \neg B \lor C,\quad A \lor \neg B \lor \neg C,$$
$$\neg A \lor B \lor C,\quad \neg A \lor \neg B \lor C,\quad \neg A \lor B \lor \neg C\ \}.$$

Ahne paikallinen haku voi edetä esim. seuraavasti:

$$\begin{array}{cc}
\{\neg A, \neg B, \neg C\} & (1) \\
\mid (A) & \\
\{A, \neg B, \neg C\} & (1) \\
\mid (B) & \\
\{A, B, \neg C\} & (1) \\
\mid (C) & \\
\{A, B, C\} & (0)
\end{array}$$

☞ GSAT-algoritmi mahdollistaa sivuttaissiirrot.

Arviointia (GSAT)

- Valittava parametrit MAX-TRIES ja MAX-FLIPS.
- Atomilauseiden satunnainen valinta tekee silmukat epätodennäköisiksi.
- Suorituskyky ilman sivuttaissiirtoja selvästi huonompi.
- Ahneet siirrot voivat viedä harhaan; pahimmillaan päädytään toistuvasti samaan paikalliseen minimiin aloitettaessa haku uudelleen.

☞ Tulisi sallia myös satunnaiset siirrot, joilla epätosien klausuulien lukumäärä $\mathrm{False}(S, \mathcal{A})$ saattaa jopa kasvaa.

Simuloitu jäähdytys (SA)

Function SA(S, T):
$\mathcal{A} := \mathrm{RC}(\mathbf{2}^{\mathrm{Hb}(S)})$;
repeat
  if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
  else
    $P := \mathrm{RC}(\mathrm{Hb}(S))$;
    if $\Delta(\mathcal{A}, P) \leq 0$ **or** $\mathrm{Random}(e^{-\Delta(\mathcal{A}, P)/T})$ **then**
      $\mathcal{A} := \mathrm{Flip}(\mathcal{A}, P)$
    endif;
    $T := \mathrm{Cool}(T)$
  endif
until False

Esimerkki. Palataan edellisen esimerkin klausuulijoukkoon $S =$

$$\{ \quad A \vee B \vee C, \quad A \vee B \vee \neg C, \quad A \vee \neg B \vee C, \quad A \vee \neg B \vee \neg C, \quad \neg A \vee B \vee C, \quad \neg A \vee \neg B \vee C, \quad \neg A \vee B \vee \neg C \quad \}.$$

Simuloituun jäähdytykseen perustuva haku voi edetä esim. seuraavasti:

$$\begin{array}{cl} \{A, B, \neg C\} & (1) \\ \mid (B) & \Delta(\{A,B\}, B) = 0 \\ \{A, \neg B, \neg C\} & (1) \\ \mid (A) & \Delta(\{A\}, A) = 0 \\ \{\neg A, \neg B, \neg C\} & (1) \\ \vdots & \end{array}$$

Satunnaissiirtymät eivät tule kysymykseen tässä esimerkissä!

Arviointia (SA)

➤ Parametri T (lämpötila) määrää todennäköisyyden sille, milloin huonompaan suuntaan johtava muutos tehdään.

T	$\Delta(\mathcal{A}, P)$	$e^{-\Delta(\mathcal{A},P)/T}$
5	1	0.819
0.5	1	0.135
0.5	2	0.0183
0.2	1	0.00674
0.2	2	0.0000454

➤ Valittava tapa lämpötilan T vähentämiseksi (jäähdytysaikataulu).

➤ Jäähdytysaikatauluja:

— Geometrinen aikataulu: $\mathrm{Cool}(T) = c * T$, missä $0 < c < 1$.

— Lämpötila vakio: $\mathrm{Cool}(T) = T$.

➤ Äärellinen jäähdytysaikataulu ei takaa ratkaisun löytymistä, joten joudutaan käyttämään uudelleen aloittamista.

➤ Keskeisiä eroja **GSAT**:iin nähden:

1. **GSAT** tekee *aina* muutoksen, joka vähentää epätosien klausuulien määrää, jos sellainen on olemassa.
2. Muuttujan (alustava) valinta on **SA**:ssa satunnainen. Vaikka $\Delta(\mathcal{A}, P) < 0$, voi löytyä toinen atomilause Q siten, että $\Delta(\mathcal{A}, Q) < \Delta(\mathcal{A}, P)$. Tällaisessa tilanteessa **GSAT** ei valitsisi atomilausetta P.

5. Ahneen paikallisen haun muunnelmia

➤ **GSAT** ja satunnainen *vaellus* (**RW**)

B. Selman, and H.A. Kautz [IJCAI, 1993]: *Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems.*

➤ **GSAT** ja satunnainen *kohina* (**RN**)

➤ **GSAT** ja satunnainen vaellus II (**WSAT**)

B. Selman, H.A. Kautz, and B. Cohen [DIMACS, 1993]: *Local Search Strategies for Satisfiability Testing.*

`http://www.cs.washington.edu/homes/kautz/walksat/`
(nykyinen versionumero 45)

GSAT ja satunnainen vaellus (RW)

Function GSAT+RW(S,p):
for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A} := \mathrm{RC}(\mathbf{2}^{\mathrm{Hb}(S)})$;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else
      if $\mathrm{Random}(p)$ **then** $P := \mathrm{RC}(\mathrm{Hb}(\{C \in S \mid \mathcal{A} \not\models C\}))$
      else $P := \mathrm{RC}(\{Q \mid \Delta(\mathcal{A},Q) = \min_R \Delta(\mathcal{A},R)\})$ **endif**;
      $\mathcal{A} := \mathrm{Flip}(\mathcal{A},P)$
    endif
  endfor
endfor

☞ Satunnaisen vaeltamisen ja ahneen haun yhdistelmä.

Esimerkki. Olkoon $S = \{A \vee E,\ \neg A \vee B,\ \neg B \vee C,\ \neg C \vee D \vee E\}$.

Kun $p = 1$ (ei ahneita siirtoja), haku voi edetä esim. seuraavasti:

$$\begin{array}{cc}
\{\neg A, \neg B, \neg C, \neg D, \neg E\} & (1) \\
\mid (A) & \\
\{A, \neg B, \neg C, \neg D, \neg E\} & (1) \\
\mid (B) & \\
\{A, B, \neg C, \neg D, \neg E\} & (1) \\
\mid (C) & \\
\{A, B, C, \neg D, \neg E\} & (1) \\
\mid (E) & \\
\{A, B, C, \neg D, E\} & (0)
\end{array}$$

GSAT+RW (arviointia)

- Vaatii parametrinä olevan todennäköisyyden p valinnan.
- Tyypillisesti $0.5 \leq p \leq 0.6$.
- Huomaa satunnaisen vaeltamisen tavoitehakuisuus: mahd. huonompaan suuntaan (epätosien klausuulien määrä kasvaa) tapahtuvat siirtymät liittyvät vielä epätosiin klausuuleihin.

Esimerkki. Olkoon

$$S = \{A \vee \neg B, A \vee \neg C, B \vee \neg A, \neg B \vee \neg C\}.$$

Totuusjakelun $\{A,B,C\}$ tapauksessa satunnainen siirtymä on mahdollinen vain atomilauseille B ja C.

GSAT ja satunnainen kohina (RN)

Function GSAT+RN(S,p):
for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A} := \mathrm{RC}(\mathbf{2}^{\mathrm{Hb}(S)})$;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else
      if $\mathrm{Random}(p)$ **then** $P := \mathrm{RC}(\mathrm{Hb}(S))$
      else $P := \mathrm{RC}(\{Q \mid \Delta(\mathcal{A},Q) = \min_R \Delta(\mathcal{A},R)\})$ **endif**;
      $\mathcal{A} := \mathrm{Flip}(\mathcal{A},P)$
    endif
  endfor
endfor

☞ Yhdistää puhtaasti satunnaiset muutokset ja ahneen haun.

GSAT ja satunnainen vaellus II (WSAT)

Function WSAT(S, p):
for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A} := \mathrm{RC}(\mathbf{2}^{\mathrm{Hb}(S)})$;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else
      $C := \mathrm{RC}(\{D \in S \mid \mathcal{A} \not\models D\})$;
      if $\mathrm{Random}(p)$ **then** $P := \mathrm{RC}(\mathrm{Hb}(C))$
      else $P := \mathrm{RC}(\{Q \in \mathrm{Hb}(C) \mid \Delta(\mathcal{A}, Q) = \min_{R \in \mathrm{Hb}(C)} \Delta(\mathcal{A}, R)\})$ **endif**;
      $\mathcal{A} := \mathrm{Flip}(\mathcal{A}, P)$
    endif
  endfor
endfor

Esimerkki. Olkoon

$$S = \{A \vee E,\ E \vee B,\ \neg B \vee C,\ \neg A \vee D,\ \neg E \vee C \vee D\}.$$

Kun $p = 1$ (ei ahneita siirtoja), **WSAT** voi edetä esim. seuraavasti:

$$\begin{array}{cl}
\{\neg A, \neg B, \neg C, \neg D, \neg E\} & (2) \\
\mid (E) & A \vee E \text{ epätosi} \\
\{\neg A, \neg B, \neg C, \neg D, E\} & (1) \\
\mid (C) & \neg E \vee C \vee D \text{ epätosi} \\
\{\neg A, \neg B, C, \neg D, E\} & (0)
\end{array}$$

WSAT (arviointia)

- Vaatii parametrinä olevan todennäköisyyden p valinnan.
- **WSAT** yhdistää satunnaisen vaeltamisen ja ahneen haun.
- Huomaa ero verratuna algoritmiin **GSAT+RW**:
  Esim. kun $p = 1$ (ei ahneita siirtoja), algoritmit toimivat eri tavoin: **WSAT** suosii satunnaisissa siirroissa atomilauseita, jotka esiintyvät useissa epätosissa klausuuleissa.

6. Kokeellisia vertailuja

- Vertailut perustuvat usein satunnaisesti generoituihin esimerkkiluokkiin.
- Parhaat variantit **GSAT+RW** ja **WSAT** [SKC, 1993] (**WSAT** näyttäisi olevan näistä kahdesta parempi).
- Kumpikin tehokkaampia kuin **SA** (simuloitu jäähdytys), joka on tehokkaampi kuin **GSAT** [Spears, 1993].
- **GSAT+RW** parempi kuin **GSAT+RN**.
- Joskus paikalliset hakumenetelmät ratkaisevat paljon suurempia ongelmia kuin täydelliset menetelmät (esim. Davis-Putnam).

Johtopäätöksiä

- Paikalliset hakumenetelmät suoraviivaisia toteuttaa.
- Toteutuvuusongelmaan suunnitelluilla menetelmillä parempi suorituskyky kuin yleisemmillä.
- Hyvät parametrien arvot oleellisia suorituskyvyn kannalta.
  Esim. **WSAT**:issa satunnaisen muutoksen todennäköisyys p tai SA:ssa jäähdytysaikataulu (*parametrien viritysongelma*).
- Paikallisilla hakumenetelmillä ongelmia, kun tehtävässä rakennetta (atomilauseiden välillä monimutkaisia riippuvuuksia).
- *Paikalliset hakumenetelmät eivät ole täydellisiä*:
  (a) ne eivät takaa ratkaisun löytymistä;
  (b) ne eivät kykyne päättämään, ettei ratkaisua ole olemassa;
  (c) ne eivät takaa ratkaisun optimaalisuutta.

Oppimistavoitteet

- Ymmärrät paikallisen haun perusperiaatteet
- Tunnet muutamia perusalgoritmeja, jotka ratkaisevat lauselogiikan toteutuvuusongelmaa näillä periaatteilla
- Osaat simuloida näiden algoritmien toimintaa
- Tiedät mitä algoritmien epätaydellisyydestä seuraa
- Tiedät mistä algoritmien toteutuksia on saatavissa