Paikalliset (stokastiset) 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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Toteutuvuusongelma

- 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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Paikallinen haku toteutuvuusongelmassa

- Paikallisen haun suorittaminen toteutuvuusongelman tapauksessa:
  (i) Aloitetaan satunnaisesti generoidusta totuusjakelusta.
  (ii) Suoritetaan paikallisia muutoksia vaihtamalla atomilauseiden totuusarvoja (engl. termi *flip*).
  (iii) *Ahne haku*: muutetaan totuusarvo sellaiselta atomilauseelta, joka johtaa suurimpaan laskuun epätosien lauseiden 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 lauseiden määrä ei laske)
  (c) Satunnaiset siirrot (epätosien lauseiden määrä saattaa kasvaa)

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Tarkastellaan klausuulijoukon $S = \{A \lor B, \neg A \lor B, A \lor \neg B\}$ 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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Tarkastellaan klausuulijoukkoa

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

➤ 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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Paikalliseen hakuun perustuvia algoritmeja

➤ **GSAT** (ahne paikallinen haku)
B. Selman, H.J. Levesque, and D.G. Mitchell [AAAI, 1992]:
A New Method for Solving Hard Satisfiability Problems.

➤ **SA** (simuloitu jäähdytys)
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

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

GSAT (ahne paikallinen haku)

```
Function GSAT(S):
for i := 1 to MAX-TRIES do
  𝒜 := satunnaisesti generoitu totuusjakelu;
  for j := 1 to MAX-FLIPS do
    if 𝒜 ⊨ S then return 𝒜
    else
      Valitaan satunnaisesti jokin atomilause P niistä atomilauseista,
      joiden totuusarvomuutoksella saadaan suurin lasku epätosien
      lauseiden määrässä ja vaihdetaan P:n totuusarvo 𝒜:ssa.
    endif
  endfor
endfor
```

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Tarkastellaan klausuulijoukkoa $S =$

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

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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

GSAT (arviointia)

- 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 lauseiden määrä saattaa jopa kasva.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

SA (simuloitu jäähdytys)

$\mathcal{A}$:= satunnaisesti generoitu totuusjakelu;
repeat
  if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
  else
    Valitse satunnaisesti atomilause $A \in \mathcal{P}$.
    Laske muutos epätosien lauseiden määrässä (ΔA),
    jos atomilauseen A totuusarvo vaihdettaisiin.
    if $\Delta A \leq 0$ (määrä ei kasva) **then**
      vaihda A:n totuusarvo mallissa $\mathcal{A}$
    else vaihda A:n totuusarvo mallissa $\mathcal{A}$ todennäköisyydellä $e^{-\Delta A/T}$.
    endif
  endif
until False

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Palataan edellisen esimerkin klausuulijoukkoon $S =$

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

Simuloituun jäähdytykseen perustuva haku voi edetä esim. seuraavasti:

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

Satunnaissiirtymät eivät tule kysymykseen tässä esimerkissä!

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

SA (arviointia)

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

T	ΔA	$e^{-\Delta A/T}$
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).

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

- Jäähdytysaikatauluja:
  - — Geometrinen aikataulu: $T_{i+1} = c * T_i$, missä $0 < c < 1$.
  - — Lämpötila vakio.
- Äärellinen jäähdytysaikataulu ei takaa ratkaisun löytymistä, joten joudutaan käyttämään uudelleen aloittamista.
- Keskeisiä eroja **GSAT**:iin nähden:
  1. **GSAT** tekee *aina* muutoksen, joka vähentää epätosien lauseiden määrää, jos sellainen on olemassa.
  2. Muuttujan (alustava) valinta on **SA**:ssa satunnainen. Vaikka $\Delta A < 0$, voi löytyä toinen atominen lause B siten, että $\Delta B < \Delta A$. Tällaisessa tilanteessa **GSAT** ei valitsisi A:ta.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Ahneen paikallisen haun muunnelmia

- **GSAT+RW** (satunnainen vaellus)
  B. Selman, and H.A. Kautz [IJCAI, 1993]:
  Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems.
- **GSAT+RN** (satunnainen kohina)
- **WSAT** (satunnainen vaellus II)
  B. Selman, H.A. Kautz, and B. Cohen [DIMACS, 1993]:
  Local Search Strategies for Satisfiability Testing.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

GSAT+RW (satunnainen vaellus)

for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A}$:= satunnaisesti generoitu totuusjakelu;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else Todennäköisyydellä p: valitse satunnaisesti atomilause, joka esiintyy jossain epätodessa lauseessa ja muuta sen totuusarvo. Todennäköisyydellä $1-p$: muuta minkä tahansa atomilauseen totuusarvo, jolla saadaan suurin lasku epätosien lauseiden määrässä.
    endif
  endfor
endfor

☞ Satunnaisen vaeltamisen ja ahneen haun yhdistelmä.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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:

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

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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 lauseiden määrä kasvaa) tapahtuvat siirtymät liittyvät vielä epätosiin lauseisiin.

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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

GSAT+RN (satunnainen kohina)

for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A}$:= satunnaisesti generoitu totuusjakelu;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else
      Todennäköisyydellä p: valitse satunnaisesti jokin atomilause ja muuta sen totuusarvo.
      Todennäköisyydellä $1-p$: muuta minkä tahansa atomilauseen totuusarvo, jolla saadaan suurin lasku epätosien lauseiden määrässä.
    endif
  endfor
endfor

☞ Yhdistää puhtaasti satunnaiset muutokset ja ahneen haun.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

WSAT (satunnainen vaellus II)

for $i := 1$ to MAX-TRIES **do**
  $\mathcal{A}$:= satunnaisesti generoitu totuusjakelu;
  for $j := 1$ to MAX-FLIPS **do**
    if $\mathcal{A} \models S$ **then** return $\mathcal{A}$
    else Valitse satunnaisesti jokin epätosi lause.
      Todennäköisyydellä p: valitse satunnaisesti siitä atomilause ja muuta sen totuusarvo.
      Todennäköisyydellä $1-p$: valitse siitä jokin atomilause, jonka totuusarvon muutoksella saadaan suurin lasku epätosien lauseiden määrässä ja muuta atomilauseen totuusarvo.
    endif
  endfor
endfor

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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:

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

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

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 monissa epätosissa lauseissa.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Kokeellisia vertailuja

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

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Johtopäätöksiä

- Paikalliset hakumenetelmät suoraviivaisia toteutta.
- 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.

© 2003 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio