

SAT řešič pomocí algoritmů inspirovaných přírodou

Jan Klátil, Milan Rybář

Obsah

1. Základní algoritmy pro SAT
2. SAT Competition
3. SAT a algoritmy inspirované přírodou

Vteřinový úvod do SATu

Rozhodnout zda je formule v CNF tvaru splnitelná.

- formule v CNF: konjunkce klauzulí
- klauzule: disjunkce literálů
- literál: atom nebo negace atomu

Základní algoritmy:

- Úplné: DPLL, CDCL
- Neúplné: lokální prohledávání

Davis-Putnam-Loveland-Logemann Procedure (DPLL)

DPLL (Φ)

/* Input Φ is a set of sets of literals representing a CNF formula */

/* Output is either “unsatisfiable” or “satisfiable” */

If $\Phi = \emptyset$; return “satisfiable”.

If there is a clause containing a single literal l do the following:

Return **DPLL**($\{c \setminus \{\neg l\} : c \in \Phi \text{ and } l \notin c\}$).

If there is a variable v which occurs in Φ only as either a positive literal or as a negative literal do this:

Return **DPLL**($\{c \in \Phi : v \notin c \text{ and } \neg v \notin c\}$).

Choose any variable v from Φ .

$\Phi_0 \leftarrow \{c \setminus \{v\} : c \in \Phi \text{ and } \neg v \notin c\}$

If **DPLL**(Φ_0) == “satisfiable” Return “satisfiable”.

$\Phi_1 \leftarrow \{c \setminus \{\neg v\} : c \in \Phi \text{ and } v \notin c\}$

If **DPLL**(Φ_1) == “satisfiable” Return “satisfiable”.

Return “unsatisfiable”.

úplný a korektní algoritmus

DPLL - Nevýhody / Vylepšení

Chronologický backtracking => backjumping

Zapomíná konflikty => nové klauzule z konfliktu během vyhledávání

Heuristiky pro výběr proměnné / hodnoty

Pravidelný restart vyhledávání (nové klauzule jsou zachovány)

Strategie pro mazání nových kláuzulí (šetření paměti)

...

Soubor těchto vylepšení je znám jako Conflict-Driven Clause Learning (CDCL)

Conflict-Driven Clause Learning (CDCL)

```
CDCL( $\phi$ ,  $v$ )
if (UnitPropagation( $\phi$ ,  $v$ ) == CONFLICT)
    then return UNSAT
 $dl \leftarrow 0$  Decision level
while (not AllVariablesAssigned( $\phi$ ,  $v$ ))
    do ( $x$ ,  $v$ ) = PickBranchingVariable( $\phi$ ,  $v$ )           // Decide stage
         $dl \leftarrow dl + 1$  // Increment decision level due to new decision
         $v \leftarrow v \cup \{(x, v)\}$ 
        if (UnitPropagation( $\phi$ ,  $v$ ) == CONFLICT) // Deduce stage
            then  $\beta =$  ConflictAnalysis( $\phi$ ,  $v$ ) // Diagnose stage
                if ( $\beta < 0$ )
                    then return UNSAT
                else
                    Backtrack( $\phi$ ,  $v$ ,  $\beta$ )
                     $dl \leftarrow \beta$  // Decrement decision level due to backtracking
return SAT
```

úplný a korektní algoritmus

Stochastic local search (SLS)

GSAT

- překlápí hodnotu proměnné, která maximalizuje počet splněných klauzulí, se statickými restarty
- vylepšení: vážené klauzule, náhodná procházka (GWSAT), ...

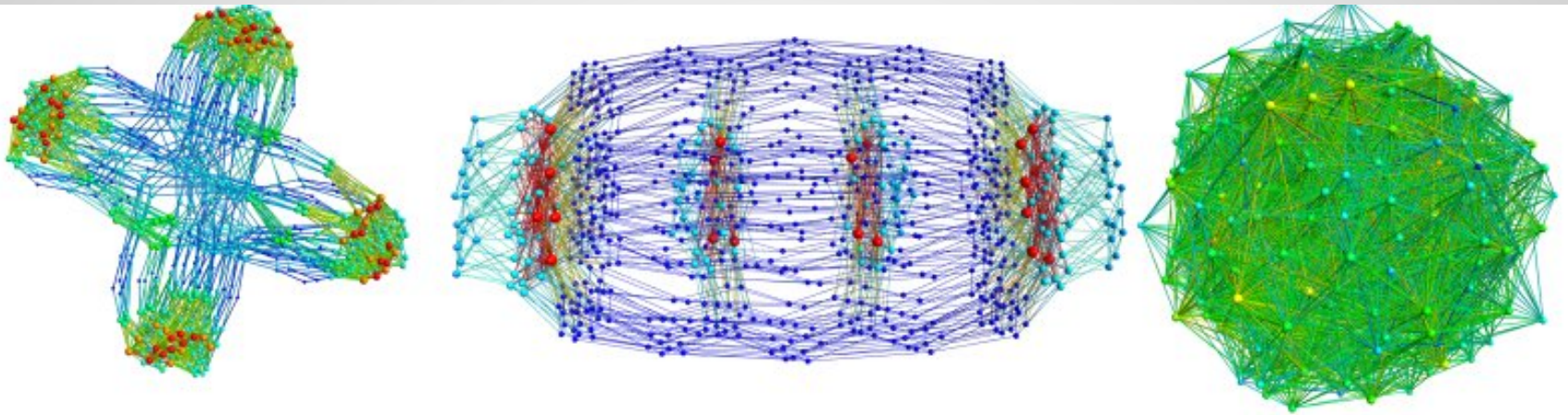
WalkSAT

- minimalizuje počet nesplněných klauzulí
- vylepšení: Tabu search, ...

Obecné SLS metody:

- Simulované žíhání
- Evoluční algoritmy

Neúplné, korektní algoritmy



SAT Competition 2013

2015: 9. ročník (1.: 2002)

2013: https://helda.helsinki.fi/bitstream/handle/10138/40026/sc2013_proceedings.pdf?sequence=2



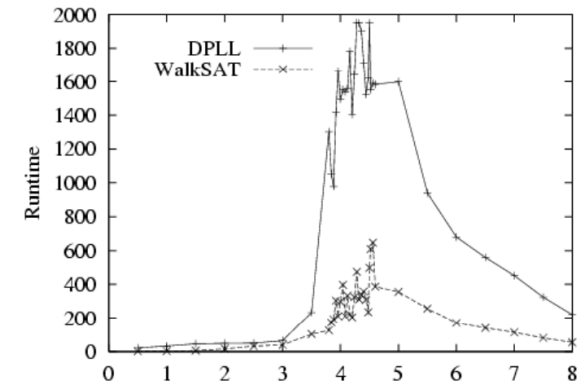
Random

náhodný uniformní k-SAT:

vstup: #neznámých (n), #klauzulí (m), #literálů v klauzuli (k)

alg: vytvářej klauzule:

1. vyber neznámou=>literál
2. přidej pokud není opačný v klauzuli
3. pokud není klauzule ve formuli, přidej jí



SAT:

- prahové: (k=3; r=**4.267**; n=3000 .. 13000), (k=4,..7; n=2300,..140)
- obří: “jako aplikované”

(k=3; r=3.7,..4.2; n=1000000), (k=4,..7; n=500000,..50000)

UNSAT:

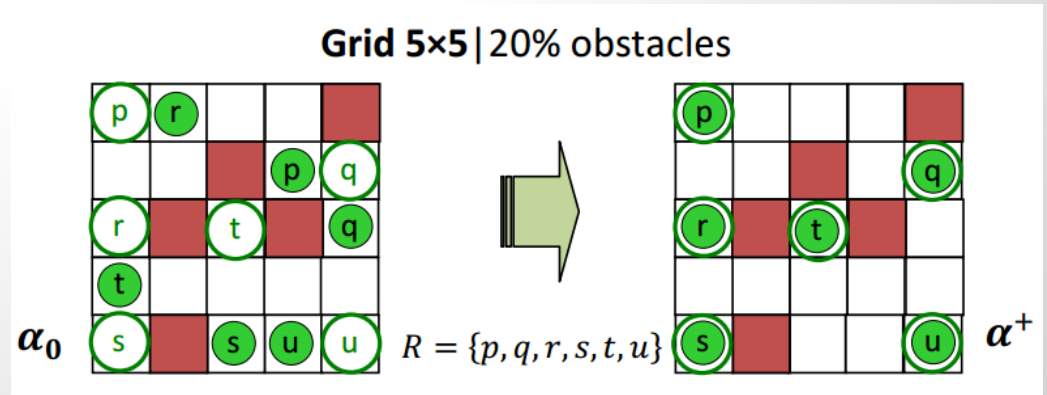
- jen pro úplné SAT řešiče
- pro k=3: n=1800,..2500
- filtrování: lokální SAT solver nenalezl řešení do 10 minut

	10	7		4
	12		14	
	1	20		15

Kombinační

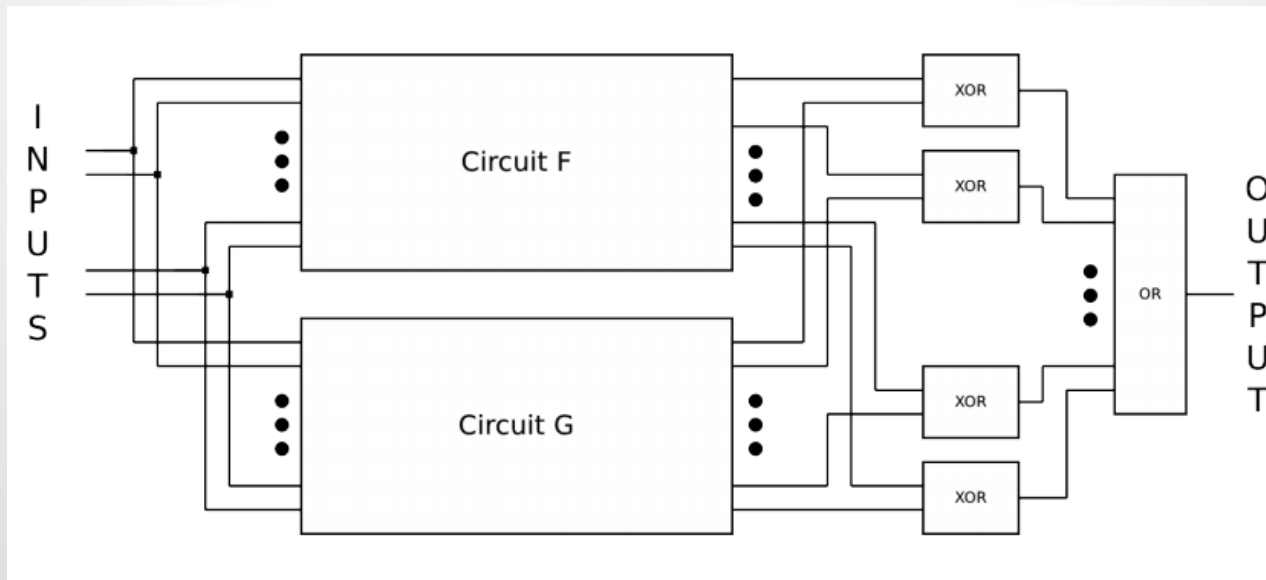
9	10	7	6	4
11	8	13	3	5
25	12	2	14	16
24	1	20	17	15
23	22	21	19	18

- Nová kategorie, dříve ručně vyrobené, hodnocení podle počtu nalezených řešení, 50:50 SAT:UNSAT, hlavně nová zadání
- Kategorie podle nalezení řešení předchozích řešičů od minuty po hodinu
- barvení grafu, izomorfismus grafů, hidoku, rozklad na prvočísla (152411913452483), plánování (Cooperative Path Planning, RNDr. P. Surynek PhD)



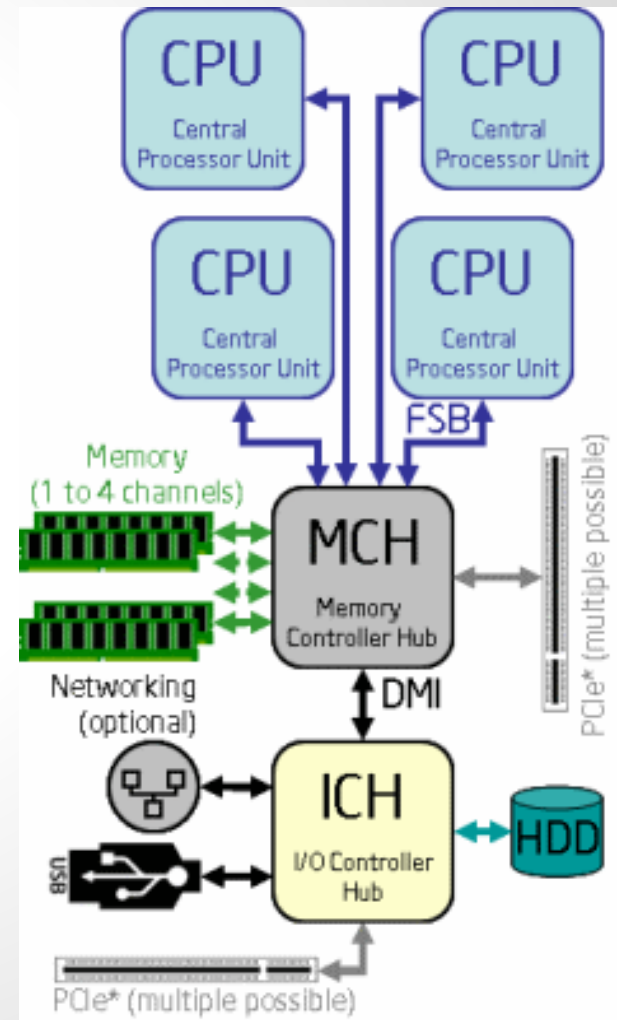
Aplikovaná

- rozvrhování a plánování, kryptografie (sha, aes, md5)
- výrobní linka aut, jízdní řády vlaků, ověřování obvodů
- polovina zadání je recyklována



Paralelismus, miniSAT hack

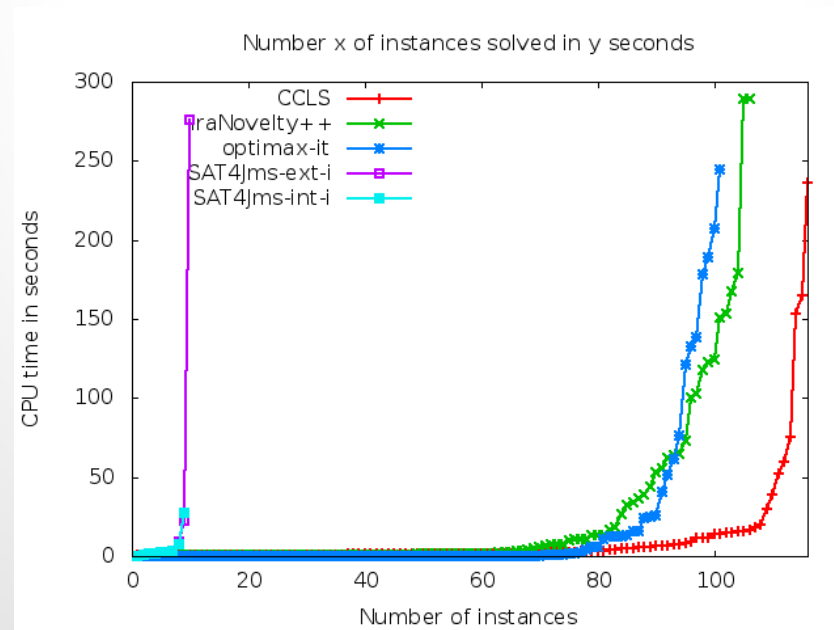
- sekvenčně: 1 jádro, 7.5GB RAM, 50GB disk, 1 hodina a 20 minut
- paralelně: 8 jader, 15GB RAM, 100GB disk, 1 hodina a 20 minut
- 2x Quad-Core Intel Xeon E5440, 2.83 GHz with 16GB RAM
- během soutěže:
 - spotřeboval 6MWh~spotřeba 5 lidí/rok
 - 3t CO₂ (váha autorů SAT řešičů)
- úprava miniSATu do 1000 znaků



MAX-SAT 2014

$$(x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee x_1) \wedge (\neg x_0 \vee \neg x_1)$$

- maximální počet ohodnocených klauzulí, NP-úplné
- MAX-SAT, vážené klauzule, částečný MAX-SAT (část klauzulí povinná, max ostatních)
- kategorie náhodných, ručně vyrobených, z praxe
- kategorie **neúplných** řešičů



Algoritmy inspirované přírodou

Algoritmy inspirované přírodou:

- Evoluční algoritmy
- Roje, mravenci, kukačky, světlušky a další...

Hlavní problémy:

- Lokální prohledávání, neúplné
- V praxi používané pro nalezení suboptimálního řešení
 - pro SAT potřeba optimálního řešení
 - výhodnější pro MAX-SAT úlohu
- Definice fitness funkce pro prohledávaný prostor
 - uvíznutí v lokálním optimu, cyklení
- Problém explorace vs exploatace
- Velký prohledávací prostor

Plán

- Vyzkoušení ne moc používaných algoritmů lokálního prohledávání na SAT
- Cílem není vyhrání soutěže
 - nebudeme extrémně optimalizovat kód
 - ani používat speciální moderní náročné postupy
- Cílem je
 - zvolení vhodného algoritmu, vhodné reprezentace CNF a prohledávacího prostoru (definice fitness funkce)
 - porovnání s ostatními SAT řešiči používající lokálního prohledávání a MiniSATem

Aktuálně uvažované (nepromyšlené) přístupy:

- Hiearchistická (více populační) EVA
- Multikriteriální optimalizace
- Roje
- Kombinace více přístupů
- Adaptivní parametry
- ...



Q & A

Děkujeme za pozornost

V příštích dílech uvidíte:

- miniSAT poražen
- Nejlepší zaostávají
- Výhra v SAT Competition

Zdroje

- **SAT Basics.** <http://gauss.ececs.uc.edu/SAT/articles/sat.pdf>
- **History of Satisfiability**, John Franco and John Martin. Handbook of Satisfiability, Chapter 1, IOS Press, 2009. <http://gauss.ececs.uc.edu/SAT/articles/FAIA185-0003.pdf>
- **Conflict-Driven Clause Learning SAT Solvers**, Joao Marques-Silva, Ines Lynce and Sharad Malik. Handbook of Satisfiability, Chapter 4, IOS Press, 2009. <http://gauss.ececs.uc.edu/SAT/articles/FAIA185-0131.pdf>
- **(MiniSAT) An Extensible SAT-solver**, Niklas Eén, Niklas Sörensson. <http://minisat.se/downloads/MiniSat.pdf>
- **SAT competition proceedings: 2013** https://helda.helsinki.fi/bitstream/handle/10138/40026/sc2013_proceedings.pdf
- **Towards Optimal Cooperative Path Planning in Hard Setups through Satisfiability Solving**, P. Surynek, 2012 http://ktiml.mff.cuni.cz/~surynek/publications/files/SurynekPavel_Cooperative-Compression-SAT_PRICAI-2012.pdf