

DNA Algorithms for 3-SAT

Presented by

Sardar Anisul Haque

Presentation Outline

- **Backtracking procedure**
- **Breadth first procedure**
- **Estimating tree size by backtracking procedure**
- **3-SAT problem**
- **DPLL algorithms**
- **DNA computing for 3-SAT**
- **Proposed DNA encoding for formula**
- **Time and space complexity**
- **Proposed variable ordering for 3-SAT**

Backtracking Procedure

- Sudoku Puzzle

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

<http://en.wikipedia.org/wiki/File:Sudoku-by-L2G-20050714.svg>

Backtracking Procedure (cont.)

- A solution

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Backtracking Procedure (cont.)

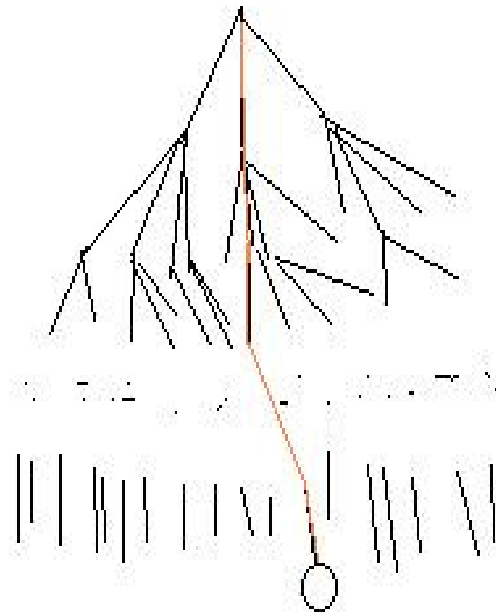
Backtracking procedure to solve Sudoku puzzle

1. Select any empty cell. If no empty cell found then return the solution.
2. Select an permissible integer for that cell and place the integer into it. If no such integer found then change the last assignment.

1. Variable ordering
2. Value ordering

Backtracking Procedure (cont.)

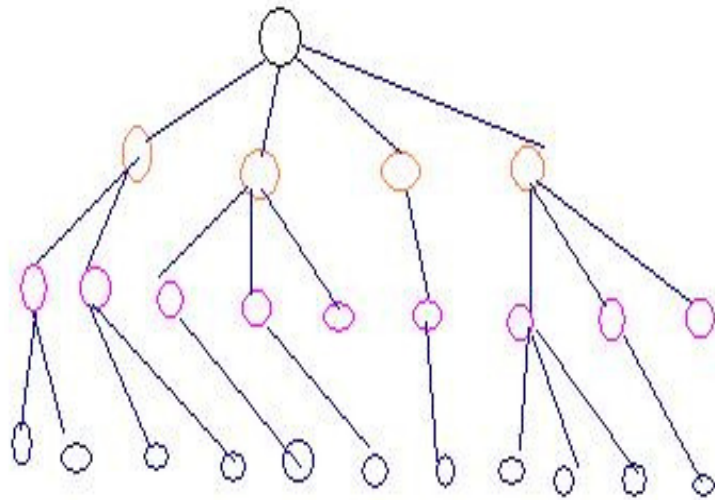
- Importance of variable and value ordering



Backtracking Tree

Breadth First Procedure

- Visiting all nodes of same depth of the backtracking tree in parallel.
- Value ordering is not important.



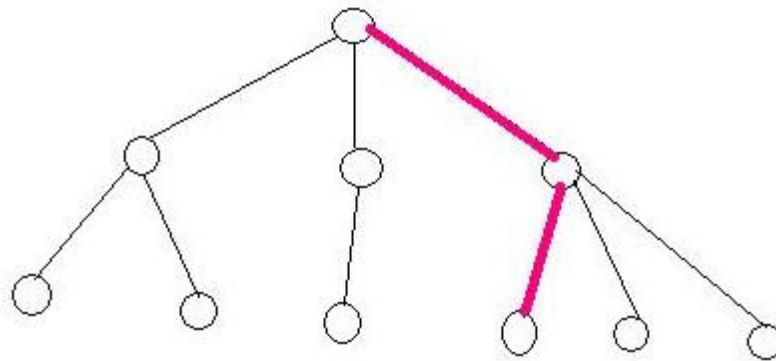
Estimating Size of Backtracking Tree

- Estimating backtracking tree size can help in variable and value ordering.
- Tree estimation by a number of random probing from a fixed node [Knuth, 1975].

Estimating Size of Backtracking Tree (Cont.)

$$|T| = 1 + c_0 + c_0 c_1 + c_0 c_1 c_2 + \dots + c_0 c_1 c_2 \dots c_{n-1}$$

For example: $1 + 3 + 3 \cdot 3$



3-SAT Problem

- n binary variables: $x_0 x_1 x_2 \dots x_{n-1}$
- Literal: Positive (x_i) and Negative ($\sim x_i$) literal
- Clause of Length 3: $(x_i \vee x_j \vee x_k)$
- Formula: A set of clauses.
- Question: Find a truth assignment of variables for which all the clauses of a formula is satisfied.
- NP-Complete Problem

DPLL Algorithms

- Backtracking procedure
- Davis, Putnam [1960] and Davis, Logemann, Loveland [1962]
- Assigning a truth value to a variable($x_i = T$):
Deleting all clauses containing x_i , and deleting literal $\sim x_i$ from clauses.
- Empty formula : satisfiable
- Empty clause: unsatisfiable

DPLL Algorithms (Cont.)

- Unit clause rule
- Monotone literal fixing
- Subsumption clause rule
- Variable ordering: Require computing real values. Some variable ordering: Zeroslow-Wang, C-Sat, DSJ rule.

DNA Computing for 3-SAT

- Lipton [1995]
- Ogiwara [1996]
- Chen and Ramachandran [2001]
- Braich, Chelyapov, Johnson, Rothemund and Adleman [2002]
- Wang, Bao, Hu, Wang, and Zhan [2007]

Lipton [1995]

- A number of single strand DNA strands are created. Each of which represents one single truth assignment. All possible truth assignments are encoded.
- For each clause a number of DNA strands are identified and deleted that do not satisfy the clause.
- Time complexity: $4m + 4n + 1$
- Space complexity: 2^n

Ogihara [1996]

- Breadth first Approach
- All partial truth assignments at a certain level are encoded.
- At each level, test for satisfiability and unsatisfiability are done.
- Unit clause, and Monotone literal fixing are applied.
- Time Complexity: $O(n \cdot \max\{m, n\})$
- The number of clauses could be very high compare to n

Chen and Ramachandran [2001]

- Randomized Algorithm
- Partial truth assignments are extended randomly
- Unlike Ogihara[1996] approach, not all branches are explored.

Braich, Chelyapov, Johnson, Rothemund and Adleman [2002]

- 20 variables, 24 clauses
- One unique solution
- Their algorithm is the same as Lipton's algorithm..

Wang, Bao, Hu, Wang, and Zhan [2007]

- Use ligase chain reaction
- Partial truth assignments are extended using ligase chain reaction
- Unsuccessful partial truth assignments are removed.

Features of DNA computers for Satisfiability

- All possible truth assignments creation has limitation of $n = 60$
- Extension of partial truth assignments require 2^n space complexity (upper bound) and experimentally $\sim 2^{0.5n}$

Proposed DNA Encoding for Formula

- Instead of truth assignment, the formula is encoded.
- All clauses have equal length (k-SAT can be converted to 3-SAT)
- A special sequence for header 'h', clause separator 'cs', literal separator 'ls', formula end sequence 'es' and truth assignment separator 'ts' are used

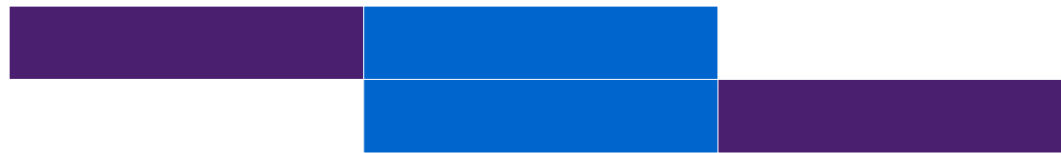
Proposed DNA Encoding for Formula (cont..)

- Header= red, clause separator = yellow, literal separator= sky blue, formula end = green truth assignment separator = deep blue



Assignment of Truth Values

- Initially only the formula is in the test tube
- X_i and $\sim X_i$ are complement
- Literal $X_i = T \Rightarrow$ pour \sim ls X_i and double strand (\sim ts $\sim X_i$) in upper and (X_i ts)



Assignment of Truth Values (cont..)

- $X_i = T$ means all the $\sim X_i$ in clauses are attached with X_i and create double strand. And the corresponding truth assignment is attached.

Checking for unsatisfiability

- DNA sequences representing formula with partial truth assignment have some piecewise double strands
- Denaturation and looking for all single strand DNA ($X_i \sim \text{Is} X_j \sim \text{Is} X_k \sim \text{Is}$). We call it forbidden sequence.
- Extract the formulas that have partial truth assignment $X_i = T$, $X_j = T$, and $X_k = T$

Algorithm

- Let we have a static variable ordering 'SVO'
- Repeat the following procedure n times
- Select the variable X_i according to SVO
- Divide the DNA strands into two test tubes
- Assign $X_i = T$ in one and $X_i = F$ in the other
- Check for unsatisfiability and delete separate the unsatisfiable formulas.
- Merge two test tubes.

Analysis

Time complexity :

- For assigning $X_i = T$ in the formula = 1 step
- For unsatisfiability checking: converting to single strand, detecting forbidden sequences, separating unsatisfiable formulas: 5 steps
- Detecting the successful truth assignment (if any)

$$\sim 6n + 1 = O(n)$$

Analysis

Space complexity

- Depends on the number of leaves of the breadth first tree
- The space complexity of Ogihara's experiment was expected to be $2^{0.5n}$
- We can also expect the same space complexity

Variable Ordering

- We are looking for a variable ordering (static) which can be implemented easily
- Such that the space complexity of our proposed algorithm should depend on the hardness of the problem

Variable Ordering (Cont..)

We can run the Lipton's algorithm with some modifications:

- Create a number of truth assignments fixing the truth value of one variable X_i .
- Select the clause randomly while deleting truth assignments.
- Count how many clauses need to be checked (let m_{X_i}) before reaching an empty test tube.

Variable Ordering (Cont..)

- The weight of a variable $m_{x_i} + m_{\sim x_i}$
- Order the variables according to their weight. Lower weight means higher priority.
- We can get a solution by chance.

Thank you

?