Dynamic Local Search for SAT

By Dave Tompkins
Ph.D. Student davet@ece.ubc.ca
Co-Supervisor: Dr. Holger Hoos

1 SATisfiability:

A Satisfiability (or SAT) problem is usually defined as one large logical statement. The objective is to find a solution that makes the entire statement true, or in other words, to SATisfy the statement. SAT is important because it can represent many different problems from a wide variety of disciplines. SAT problems can be very deceiving: a problem may be very easy to describe, but extremely difficult to solve. Some statements are impossible to satisfy, in which case we often focus on satisfying it as best as possible, which is known as MAX-SAT.

What does a logic statement look like?

```
Alice is a Vegetarian
Bob wants to go for a Steak or Thai food
Carol won't attend if Dan is there
The Thai restaurant has a Vegetarian menu
... and so on. This gets translated as:
(V or not A) and (S or T or not B) and
(D or not C) and (V or not T) ...
```

For example, A is a variable that is True if Alice attends dinner

All statements can be arranged into normal form:

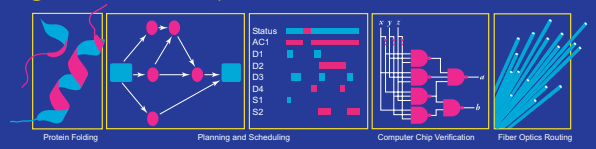
$$(V + \overline{A})\,(S + T + \overline{B})\,(D + \overline{C})$$

clause; literals

The statement is satisfied when all clauses are True

For N variables, there are actually 2^N different combinations that could be a solution.
For a small problem of N = 100, there are 1,267,650,600,228,229,401,496,703,205,376 possible combinations!

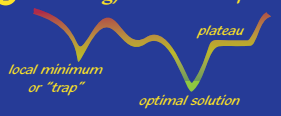
Thousands of other problems can be reduced to a SAT Problem...



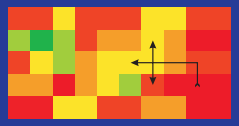
2 Local Search:

With most SAT problems, it is impractical to test every possible combination to see if it satisfies the statement, because there are just too many possible combinations. We like to think of all these combinations as points in a large "search space". For any given point in the space, it will have "neighbouring" points that are nearly identical combinations. With a local search strategy, we "search" through the space by stepping through adjacent ("local") neighbours until we find a solution. Usually, we think of this as a minimization problem: in the case of SAT, we try to minimize the number of unsatisfied clauses. There are a wide variety of local search algorithms that have been developed to tackle SAT and other combinatorial problems.

Terminology of a Search Space:



A 2-D diagram gives a better picture of what is involved:



These diagrams help us to visualize the space, but for SAT it's actually an N-dimensional Hyper-Cube!

Local search methods are often called "Stochastic" local search algorithms because their search strategy usually includes a random component, which helps the algorithm explore the space and escape from traps.

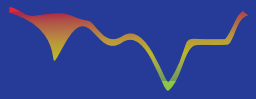
Famous Local Search Methods:

- *Simulated Annealing*
- *Genetic Algorithms*
- *Ant Colony Optimization*
- *Evolutionary Computing*
- *TABU Search*

3 Dynamic:

One type of a local search strategy is known as Dynamic Local Search (DLS). In DLS, the algorithm dynamically modifies the space as it searches. Typically, when a DLS algorithm encounters a local minimum (or a "trap") it modifies the search space so that it is less likely to fall in that trap again.

Some find it helpful to think of DLS as a way of "filling in traps"



but the truth is, we don't have a good understanding of how DLS algorithms warp their hyper-cube search spaces

At UBC, we've developed a DLS algorithm known as Scaling and Probabilistic Smoothing (SAPS) which is one of the fastest SAT solvers available!

What are we working on? We are continuing to create and evaluate new SAT solvers. By analyzing how DLS algorithms warp their search space, we hope that we can gain the insight required to develop the next generation of SAT solvers.

Event: BC Advanced Systems Institute — The AI Exchange 2003

Affiliations: UBC — ICICS — Binformatics, and Empirical & Theoretical Algorithmics Laboratory

Funding: Natural Sciences and Engineering Research Council of Canada