Turing's Legacy and AI
Two perspectives

Barry Richards
17th April 2012

b.richards85@btinternet.com

Nondeterminism

"Nondeterminism is a central concept in complexity theory because of its affinity not so much with computation itself, but with the *applications of computation,* most notably logic, combinatorial optimization, and artificial intelligence."

Christos Papadimitriou, *Computational Complexity*, Addison – Wesley, 1994

Two persisting questions

1. Can the *typical* performance of a ***heuristic search algorithm*** for an NP-complete problem be predicted?
   - Can an instance of such a problem be identified as easy or hard to solve?

2. Is ***intelligent search*** characteristic of human problem solving?
   - Turing's approach
   - The approach of AI-as-symbolic-computation

Turing's legacy and AI: One perspective

- Turing's **question:** *Can computers think?*
  - Is this question just semantic or philosophical?
- Turing's **response**: A question-answer game (Turing Test)
  - Aim
    - To ***turn*** a semantic / philosophical question into a ***scientific*** hypothesis.
  - Hypothesis
    - A computer can play the game so well that one cannot (typically) distinguish its behaviour from that of a human.
  - Assumption
    - Showing that computers can think and behave as humans typically do is a ***scientific*** quest.
  - Research agenda
    - To develop a program that ***simulates*** typical human behaviour
- Will Turing's quest yield a model of ***intelligent search***?

Turing’s legacy and AI: A second perspective

- AI-as-symbolic-computation
  - Problem solving
    - A major driver of AI research
    - A paradigm example of **thinking**
  - Research agenda
    - Design “effective” problem-solving programs

- Do problem-solving programs “embody” models of ***intelligent search***?

Problem solving: A synoptic view

- Assumption
  - Problem solving is essentially a ***search*** process.
    - Programs driven by search
- Problems are typically NP-complete.
  - Search space grows exponentially in the size of the problem.
- Search process
  - Must resort to ***heuristic strategies***
  - Two kinds of heuristic strategy
    - **Intelligent** processes
    - **Stochastic** processes
- Conjecture
  - Some Intelligent-search heuristics reflect human intelligence.

An example: 3-SAT problems

- Definition
  - 3-SAT formula
    - A propositional formula in conjunctive normal form where each conjunct has exactly 3 literals (either a propositional variable or its negation)
  - Task
    - Given a 3-SAT formula σ, determine whether there is an assignment to the variables such that σ is true?
    - This is an NP-complete problem (exponential worst-case complexity).
- Relevance of worst-cast complexity: Two claims

  a. " ... *most* randomly generated SAT instances are actually surprisingly easy to solve ... with the hardest instances only occurring in a rather small range of parameter settings ... "

  b. " ... *many* satisfiable instances in the hard region could still be solved quite efficiently solved ... based on local search techniques."

  Carla P. Gomes, et al.: "Satisfiability Solvers", in F. Van Harmelen, et al., editors, *Handbook of Knowledge Representation,* Elsevier, 2008.

3-SAT problems: <u>Complete</u> systematic search

- Davis-Putnam-Logemann-Loveland (DPLL)
  - Backtrack search over partial assignments
    - Given a 3-SAT formula σ and the empty partial assignment,
      extend the partial assignment successively by selecting an unassigned literal (branching step)
      and seeking an assignment to that literal that yields an assignment satisfying σ.
  - DPLL remains the favoured basic procedure for <u>complete</u> search.
- Some heuristic enhancements
  - Variable / value selection
  - Clause learning (learning no-goods)
  - Conflict-directed backjumping
  - Etc.

Search behaviour

- Search behaviour of *DPLL+heuristics*
  - ***Varies greatly***,
    depending on variable selection and the order of value selection.
  - ***Unpredictable*** on problem instances,
    where the effectiveness of different heuristics may diverge greatly.
- Questions
  - Can the ***behaviour*** of heuristic search algorithms be studied "scientifically"?
  - What do the ***heuristics*** contribute to an understanding of intelligent search?

3-SAT problems: Incomplete search methods

- Stochastic local search
  - Favoured basic procedure for incomplete 3-SAT search

- Examples

GSAT

i. Start with randomly generated total assignment.
ii. Change the assignment to a variable that minimizes the number of unsatisfied clauses.
iii. Continue up to a fixed maximum of changes.
iv. Repeat the process up to a fixed number of repeats.

* C. Papadimitriou and K. Steiglitz: *Combinatorial Optimization.* Prentice-Hall Inc., 1982.
* B. Selman, et.al.: “A new method for solving hard satisfiability problems”. In *10th AAAI, 440.446, San Jose,* 1992.

Walksat and *weight* (variants of *GSAT)*

- Start with randomly generated total assignment.
- Change the assignment to a variable that is selected according to certain CONDITIONS.
- As above in *GSAT*.

* B. Selman, et al.: “Local search strategies for satisfiability testing”, in D. S. Johnson and M. A. Trick, editors, *Cliques, Coloring, and Satisfiability: the Second DIMACS Implementation Challenge,* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1996.

Search behaviour

- Exploring the behaviour / performance of 3-SAT algorithms
  - Random problem generator **G**
    - Generate random 3-SAT formulas with *k* clauses on *n* variables as follows.
      - For each clause randomly select 3 distinct variables from the *n* variables and negate each variable with a probability 0.5.
    - Let Σ be the set of such formulas generated with variables *k* and *n*.

- Two observations arising from generator **G**

  a) 3-SAT instances with few solutions typically coincide with a ***peak*** in search effort.
     - These are thought to be among the most difficult.

  b) The search-peak typically coincides with a certain ***clause / variable ratio***.
     - Search effort peaks at ratio 4.26.
     - That ratio appears to be same for all algorithms (though the height of the peak varies).

learn-SAT: A 3-SAT algorithm

- *learn-SAT*
  - A complete, non-systematic local search algorithm
    - Learning no-goods
    - Learning-by-merging
  - Heuristics include,
    - Forward checking
    - Binary ordering
    - 3rd order learning

  * Richards, E.Thomas and Barry Richards: “Nonsystematic search and no-good learning”, *Journal of Automated Reasoning, 24*: 483-533, 2000.

- Exploring the search behaviour of *learn-SAT*
  - Three types of randomly generated problems
    i. Problems with a **single** solution (AIM problems)

       * Asahiro, Y. et al.: “Random generation of test instances with controlled attributes”, in D. S. Johnson and M. A. Trick, editors, *Cliques, Coloring, and Satisfiability: the Second DIMACS Implementation Challenge,* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1996.

    ii. **Unsolvable** problems

       * Bayardo, R. J. and Schrag, R.: “Using look-back techniques to solve exceptionally hard SAT instances”, in *Plroceedings of CP-96*.

    iii. Large-scale problems with **many** solutions

       * DIMACS 1998 benchmarks (Center for Discrete Mathematics & Theoretical Computer Science), Rutgers University.

learn-SAT: Search comparison

- Two observations arising from generator **G** (recalled)
  a) 3-SAT instances with <u>few</u> <u>solutions</u> typically coincide with a ***peak*** in search effort.
  b) The search-peak typically coincides with a fixed ***clause / variable ratio***.
     - Search effort peaks at ratio 4.26.
     - That ratio is the same for all algorithms (though the height of peak varies).

- <u>Single-solution</u> problems

  learn-SAT compared to *weight* (best *GSAT* variant then)
  - Results are exceptions to (a).
    - <u>Some</u> single-solution problems (AIM)) are solved with relatively little search.
    - <u>Others</u> are solved only after a lot of search.
    - Algorithms vary widely.
  - Results are exceptions to (b).
    - Hardest instances for ***weight*** (best version of GSAT) lie at clause/variable ratio 2.6.
    - Hardest instances of ***learn-SAT*** lie at clause/variable ratio 3.4.

learn-SAT: Search comparison (cont.)

- More "observations" arising from generator **G**
  c) The search effort to resolve <u>unsolvable</u> 3-SAT problems typically peaks at ratio 4.26.
  d) The search effort becomes progressively less at lower or higher ratios.

- <u>Unsolvable</u> problems

  learn-SAT compared to *relsat* (complete search - see reference above)

  - Results are exceptions to (c) and (d).
    - Hardest instances for ***relsat*** typically lie at clause/variable ratio 4, though sometimes at 3.
    - Hardest instances of ***learn-SAT*** lie sometimes at clause/variable 5, sometimes at 7.

Folklore of NP-complete problems

- Two beliefs
  e) Non-systematic local search algorithms are,
    1. poorly suited to large-scale problems with **few solutions**,
    2. either inefficient or inapplicable to **unsolvable** problems.

  f) Complete backtrack-search techniques are <u>always</u> much more efficient than complete local search algorithms for,
    - unsolvable problems.

- *learn-SAT* performance-comparisons
  - Results challenge (e1) and (e2) .
  - Results challenge (f) at some clause/variable ratios.

Hypothesis and observations

- On deriving insight search behaviour

  A. **Hypothesis**: Understanding search behaviour requires knowing at least 3 things.

     1. Some ***characteristic parameters*** of the problem,
        - E.g. constraint density (clause / variable ratio)
     2. Something about the ***solution space***,
        - E.g. number and distribution of solutions
     3. Some measure that ***connects*** (1) and (2) to performance,
        - E.g. constraint checks

  B. **Observations**: For most problems, except perhaps for 3-SAT,

     - Little is known about (1) and (2).
     - Nothing is known about (3),

- On developing effective search programs

  - **Observations**
     - Programs are currently developed on a ***case-by-case, trial-and-error*** basis.
     - The cumulative experience has yielded only a ***programming "cookbook"***.

More observations

- Heuristic search is the “mechanism” of problem solving.
  - Problem solving is quintessentially
    - An **intelligent** activity
    - An expression of **thinking**.
- *DPLL+heuristics* and *learn-SAT*
  - Both implement **search** that is,
    - Rational
    - Consistent
    - Relatively efficient
  - Both arguably **act intelligently** in solving 3-SAT problems.
  - Both are **far better** than any human at solving 3-SAT problems.
    - Challenge any attempt to abstract a model of how humans **actually** solve such problems.
- Question
  - What do these observations say about algorithms (programs) and ***human intelligence***?

Two questions: Revisited

1. Can the *typical* performance of a ***heuristic search algorithm*** for an NP-complete problem be predicted?
   - Not very likely, except perhaps with one exception

2. Is ***intelligent search*** characteristic of human problem solving?
   - Turing's approach
     - Intelligent searching is whatever **people** actually do when problem solving.
     - What people actually do is a matter for **empirical** investigation.
   - The approach of AI-as-symbolic-computation
     - Intelligent searching is what **programs** ought to do to act intelligently in problem solving.
     - What a program ought to do to act intelligently seems a **philosophical** matter.