Understanding the Performance of Evolutionary Algorithms

AFOSR Workshop
April 2008
How to predict algorithmic performance?

- Competitive testing – Not good
- Controlled scientific testing - Better
- Explanatory models - Best
Problems with Competitive Testing

• Hard to tune parameter settings.
• Random instances are unrealistic.
• Benchmark instances may be unrepresentative.
  – How do we tell what is representative?
  – Instances reflect success of past algorithms.
  – Many instances are proprietary.
• We find out which algorithms are faster, but not why.
Controlled Experimentation

- Get rid of benchmark problems.
- Use a factorial design.
- Control for problem characteristics that may influence performance.
  - Other characteristics random.
- Control for parameter settings.
- Use statistical analysis (ANOVA, etc.)
- Predict performance of an algorithm based on problem characteristics and parameter settings.
Empirical Theory

- Ultimate aim – an empirical theory that predicts algorithmic performance.
  - Empirical $\neq$ nontheoretical
  - Think about quantum electrodynamics.
Example: Branching Rules

• We want to predict performance of branching rules for the propositional satisfiability problem (SAT)
  – Based on Hooker & Vinay (1995)
  – Use a simple branching algorithm (Davis-Putnam-Loveland) to search for feasible solution of a SAT problem, such as

\[
\begin{align*}
  x_1 & \lor \neg x_3 \lor x_4 \\
  x_2 & \lor x_4 \lor \neg x_5 \\
  \neg x_1 & \lor \neg x_2 \lor x_4 \\
  x_2 & \lor x_3 \lor \neg x_4
\end{align*}
\]

  – Apply unit resolution at each node of the search tree.
Example: Branching Rules

- Formulate a Markov chain model of what happens during unit resolution.

\[ Pr(C_i \text{ eliminated}) = \frac{k}{2n}, \]
\[ Pr(C_i \text{ reduced to } k - 1 \text{ literals}) = \frac{k}{2n}, \]
\[ Pr(C_i \text{ unchanged}) = 1 - \frac{k}{n} \]
Example: Branching Rules

- Resulting transition matrix:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & \ldots \\
0 & 1 & 0 & 0 & 0 & 0 \\
\frac{2}{2n} & \frac{2}{2n} & 1 - \frac{2}{n} & 0 & 0 \\
\frac{3}{2n} & 0 & \frac{3}{2n} & 1 - \frac{3}{n} & 0 \\
\frac{4}{2n} & 0 & 0 & \frac{4}{2n} & 1 - \frac{4}{n} \\
\vdots
\end{bmatrix}
\]

- The models predicts performance of several branching rules.
  - Checks out against controlled testing.
  - No theorems – only empirical verification.
  - Leads to design of a superior branching rule.
Evolutionary Algorithms

- View evolutionary algorithms as a “biological” phenomenon.
  - Use *biology* to model the *algorithm*.

- Most existing models of genetic algorithms are not suitable as *empirical* models.
  - Biological models that are inadequate for natural evolution may be suitable for evolutionary algorithms.
Evolutionary Algorithms

• **Some biological models:**
  – Fisher fundamental theorem of natural selection
  – Price equation
  – Haldane principle
  – Haploid/diploid models of natural selection
  – Artificial life models (e.g., Belew & Mitchell 1996)
  – Molecular evolution (Kimura 1983)
Evolutionary Algorithms

• **Modeling EMAS**
  – It is a two-level evolutionary process.
    • Solutions and algorithmic agents.
  – One model: humans who raise cattle.
    • Human behavior and cattle both evolve.
  – Another model: single-level evolutionary process.
    • Reproductive process evolves in the organisms that reproduce.
    • Solutions contain instructions for generating new solutions.
    • These instructions evolve along with the solutions.