Analysis of Estimation of Distribution Algorithms and Genetic Algorithms on NK Landscapes

Martin Pelikan

Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
University of Missouri, St. Louis, MO
http://medal.cs.umsl.edu/
pelikan@cs.umsl.edu
Motivation

Testing evolutionary algorithms

- Adversarial problems on the boundary of design envelope.
- Random instances of important classes of problems.
- Real-world problems.

NK landscapes

- Additively decomposable with subproblems of bounded order.
- Highly multimodal and complex landscape, NP completeness.
- Complements prior work on random additively decomposable problems which are polynomially solvable.

Purpose

- Generate large number of NK landscape instances.
- Solve generated instances with branch-and-bound.
- Test various EDAs and GAs on the generated problems.
Outline

1. NK landscapes.
2. Branch and bound for NK landscapes.
3. Experiments.
4. Summary and conclusions.
NK Landscape

NK landscape

- Model of rugged landscape and popular test function.
- An NK landscape is defined by
  - Number of bits, $n$.
  - Number of neighbors per bit, $k$.
  - Set of $k$ neighbors $\Pi(X_i)$ for $i$-th bit, $X_i$.
  - Subfunction $f_i$ defining contribution of $X_i$ and $\Pi(X_i)$.
- The objective function $f_{nk}$ to maximize is then defined as

$$f_{nk}(X_0, X_1, \ldots, X_{n-1}) = \sum_{i=0}^{n-1} f_i(X_i, \Pi(X_i)).$$
NK Landscape

Properties

- NK landscapes are additively decomposable.
- Subproblems overlap in a complex way.
- Subproblems themselves are complex (look-up tables).
- High multimodality, complex structure.

NK landscape instances in this work

- Neighbors for each bit chosen randomly from other bits.
- Subfunctions $f_i$ represented by look-up tables with uniformly distributed numbers from $[0, 1]$. 
Branch and Bound

Basic idea

- Traverse the entire search space (try all binary strings).
- Each level decides on one bit (0 or 1).
- Each leaf encodes a unique binary string.
- Branches that lead to provably suboptimal solutions are cut.

Why?

- BB is inefficient, but can verify the global optimum.
Objectives of Experiments

Objectives of experiments

- Study influence of $n$ on performance.
- Study influence of $k$ on performance.
- Compare different GAs and EDAs.

Important

- Consider a large number of random instances.
- Ensure that the global optimum is obtained.
Test Instances and Compared Algorithms

Description of NK instances

- Use $k = 2, 3, 4, 5, 6$.
- Use $n = 20, 22, \ldots$ (limited by BB complexity).
- 10,000 instances for each combination of $n$ and $k$.
- Total of 600,000 instances.

Compared algorithms

- Genetic algorithm
  - Uniform crossover and bit-flip mutation.
  - Two-point crossover and bit-flip mutation.
  - Bit-flip mutation but no crossover.
- Estimation of distribution algorithms (EDAs)
  - Hierarchical BOA (hBOA).
  - Univariate marginal distribution algorithm (UMDA).
- Local search
  - Hill climbing (omitted due to intractable computation).
Experimental setup

- Binary tournament selection.
- Restricted tournament replacement.
- Run bisection to determine appropriate population size; ensure 10 successful runs out of 10 independent runs.
- Bound the number of iterations by $n$.
- Probability of crossover in GA = 0.6.
- Probability of flipping bit with mutation in GA = $1/n$.
- Deterministic hill climber used to improve every solution.
Results: hBOA

![Graph showing the number of evaluations (hBOA) vs problem size (n) for different values of k.]

- Problem size, n
- Number of evaluations (hBOA)

- k=6
- k=5
- k=4
- k=3
- k=2

Martin Pelikan
Analysis of EDAs and GAs on NK Landscapes
Results: UMDA

Number of evaluations (UMDA) vs. Problem size, n

- k=6
- k=5
- k=4
- k=3
- k=2
Results: GA (uniform)

Problem size, n

Number of evaluations (GA, uniform)

k=6
k=5
k=4
k=3
k=2
Results: GA (two-point)
Results: GA (no crossover)

```
<table>
<thead>
<tr>
<th>Problem size, n</th>
<th>Number of evaluations (GA, no crossover)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>160</td>
</tr>
<tr>
<td>20</td>
<td>320</td>
</tr>
<tr>
<td>40</td>
<td>640</td>
</tr>
<tr>
<td>80</td>
<td>1280</td>
</tr>
<tr>
<td>160</td>
<td>2560</td>
</tr>
<tr>
<td>320</td>
<td>5120</td>
</tr>
<tr>
<td>640</td>
<td>10240</td>
</tr>
</tbody>
</table>

k=6
k=5
k=4
k=3
k=2
```

Martin Pelikan
Analysis of EDAs and GAs on NK Landscapes
Analyzing Distribution of the Number of Evaluations/Flips

Distribution analysis

- Observed extreme-value distributions in results.
- Variance was large and tails were fat.

Why is this important?

- This can give us lot of important input.
- Analysis can be used to predict appropriate parameters for solving larger problems reliably.
Head-to-Head: GA (uniform) vs. GA (two-point)

Uniform outperforms two-point
Head-to-Head: GA (uniform) vs. GA (no crossover)

Uniform outperforms no crossover
Head-to-Head: hBOA vs. UMDA

hBOA outperforms UMDA

**Analysis of EDAs and GAs on NK Landscapes**
Head-to-Head: hBOA vs. GA (uniform)

hBOA outperforms GA (uniform)

Problem size

Num. GA (U) evals / num. hBOA evals

- k=6
- k=5
- k=4
- k=3
- k=2
Discussion of Results

Performance with respect to $n$ and $k$

- Worse-than-polynomial complexity with respect to $n$.
- Exponential complexity with respect to $k$.

Operators and algorithms

- Crossover-based GA outperforms mutation-only GA.
- Uniform crossover preferable to two-point crossover.
- But too much crossover hurts (UMDA).
- Hill climbing performs the worst.
- Linkage learning outperforms other alternatives but the differences are not dramatic.
Comparison with Other Problem Classes

Comparision with other problem classes

- Superiority of linkage learning more substantial for difficult polynomially solvable problems (e.g., 2D spin glass or random additively decomposable problems).

- Results confirm that overlap leads to inferior performance of local operators and superior performance of crossover-based search.
Summary and Conclusions

Summary

▶ Generated large number of random NK instances.
▶ Solved all instances with the branch-and-bound solver.
▶ Applied various EDAs and GAs to the resulting instances.
▶ Analyzed performance and compared the algorithms.

Conclusions

▶ Time complexity grows worse than polynomially with \( n \) and \( k \).
▶ Crossover-based GAs superior to mutation-only GAs.
▶ Stronger crossover better than weaker crossover.
▶ Linkage learning beneficial, but not as much as in many other similar problem classes.
▶ Hill climbing works worst.
▶ NK instances with guaranteed optima available for testing.
Acknowledgments

- NSF; NSF CAREER grant ECS-0547013.
- U.S. Air Force, AFOSR; FA9550-06-1-0096.
- University of Missouri; High Performance Computing Collaboratory sponsored by Information Technology Services; Research Award; Research Board.