Evaluating the Performance of Heuristics for NP-hard Problems: Are We Ready for a New Treatment?

Franc Brglez
Collaborative Benchmarking Laboratory (CBL)
Dept. Comp. Science, NC State University
http://www.cbl.ncsu.edu/

Outline

• Motivation: Design of Experiments
• Formalized Models for DoE
• Graph Equivalence Classes
• Towards Equivalence Class Synthesis
• Experimental Results
• Conclusions
  • Role of the Internet: TOCO Example
Background and Motivation

Experimental design methodology (DoE):
- **inputs:** instances of subjects representing a well-defined population class
- **objectives:** specific treatment optimizations
- **outputs:** strict comparisons of ALL observations (measurements) after each treatment, supported by statistical tests of significance

in contrast to

Traditional ‘benchmarking’ (of algorithms):
- **inputs:** unrelated (loosely classified) instances of data sets (e.g. graphs)
- **objectives:** specific cost function minimization(s)
- **outputs:** ‘selective’ comparisons of unrelated cost minimizations, unsupported by statistical tests of significance

Tutorial on DoE from U. Oklahoma Medical School

- **population:** ALL subjects to whom a treatment might possibly be applied. For example, all adult-onset diabetic patients in the world.
- **sample:** a subset of population
  - **R:** random selection
  - **E:** experimental treatment (applied to experimental group)
  - **C:** alternative (placebo) treatment (applied to control group)
  - **O:** observation (measurement) of the effect of the treatment
A Message on DoE from JAMA ....

Experimental design methods, first formalize in 1920’s, have been adopted widely in many fields of science. Guidelines on reporting results of experiments, adopted by over 500 journals in biomedicine, have been in existence since 1978, e.g.

JAMA. 1997;277:927-934
(http://www.ama-assn.org/public/journals/jama/sc6336.htm)

.... Describe statistical methods with enough detail to enable a knowledgeable reader with access to the original data to verify the reported results. When possible, quantify findings and present them with appropriate indicators of measurement error or uncertainty (such as confidence intervals).

.... Discuss the eligibility of experimental subjects. Give details about randomization.

Status for algorithms:
• only one recent textbook in computational biology addresses sample size determination and confidence interval estimation in analyzing results of experiments (B. H. Liu. Statistical Genomics: Linkage, Mapping, and QTL Analysis. CRC Press’97)

Contrast: Conventional Method of Benchmarking

• Performance comparisons based on a single graph instance measurement
• Graphs are mostly unrelated and of varying complexity

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Min. cost via Alg1</th>
<th>Min. cost via Alg2</th>
<th>‘Improvement’</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>a1</td>
<td>a2</td>
<td>(a2/a1 - 1)*100</td>
</tr>
<tr>
<td>c880</td>
<td>b1</td>
<td>b2</td>
<td>(b2/b1 - 1)*100</td>
</tr>
<tr>
<td>c1355</td>
<td>c1</td>
<td>c2</td>
<td>(c2/c1 - 1)*100</td>
</tr>
<tr>
<td>•</td>
<td>•</td>
<td>•</td>
<td>(••••• - 1)*100</td>
</tr>
</tbody>
</table>

‘Average Improvement’ •••••

We can show that
100 instances of a randomly re-ordered graph, even a small one , may induce variations, either with alg1, alg2, or both, that exceed the best of ‘improvements’ reported above.
Isomorphism Class: Destroying Myths about Benchmarking

- The class consists of a number of graphs that are isomorphic w.r.t. each other
- Graph nodes are represented in a random order
- Each node may be assigned a new random name
  (essential to bring out differences in some algorithms that hash input data)
- The algorithms are applied to each graph in the class
- Algorithms:
  - wire crossing minimization,
  - logic and layout minimization,
  - BDD minimization technology
  - TSP, etc.

*No algorithm reports costs with 0 variance for all graphs*

---

Isomorphism Class: Layout Optimization Performance

100 identical netlists, in random order, of a planar 11-node 3-level acyclic graph are being optimized by two algorithms.....

- only 22 circuits out of 100 are reported as having 0 wire crossings (by DOT)
- placed&routed layout reported by OASIS varies from 23 to 31 area units
- results are not unexpected since both the problems are NP-hard
Isomorphism Class: Logic Optimization Performance

100 identical netlists of rd53 - 5 PIs, 3 POs, 58 nodes, 10 levels

Summary of such and other postings under http://www.cbl.ncsu.edu/experiments/......

<table>
<thead>
<tr>
<th>Column</th>
<th>Population Mean (95% conf. interval)</th>
<th>Sample Mean</th>
<th>Sample Std. Dev.</th>
<th>Sample Min.</th>
<th>Sample Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>iso-node</td>
<td>[4.0134e+01, 4.0706e+01]</td>
<td>4.0420e+01</td>
<td>1.4297e+00</td>
<td>3.8000e+01</td>
<td>4.2000e+01</td>
</tr>
<tr>
<td>bool</td>
<td>[3.4292e+01, 3.6648e+01]</td>
<td>3.5470e+01</td>
<td>5.8921e+00</td>
<td>2.9000e+01</td>
<td>4.7000e+01</td>
</tr>
<tr>
<td>rug</td>
<td>[2.8134e+01, 2.9446e+01]</td>
<td>2.8790e+01</td>
<td>3.2792e+00</td>
<td>2.5000e+01</td>
<td>3.7000e+01</td>
</tr>
</tbody>
</table>

Formalized Models for DoE

Views from the Web:

---------|---------------------|---------------------|-----------------|

design of experiments | 10,426 URLs | 14,594 URLs | 2,810,431 matches |

experimental design | 32,566 URLs | 50,984 URLs | 2,726,233 matches |

Clearly, DoE is many things to many people ....

Our major objective of is to formalize a test of hypothesis as it applies to performance evaluation of algorithms:

Are the measured improvements due to the improved design of algorithms or due to chance?

We propose a unified model by drawing parallels with bio-medicine ......
**DoE Model for Treatments in Bio-Medicine**

Class $AB = \text{SAME\{species, sex, age, weight, diet, activity, .....\}}$

1. draw subjects randomly from a well-controlled class of eligible population;
2. apply the prescribed treatment to each subject and measure symptom index(es);
3. apply well-defined statistical tests to establish (relative) treatment effectiveness.

---

**DoE Model for Treatments as Algorithms**

Advantage over bio-medicine:
- class-instances are preserved regardless of number of treatments!!

1. create a number of well-defined equivalence classes;
2. `treat` each class instance by optimizing well-defined cost index(es);
3. apply significance tests to establish (relative) treatment effectiveness.
Algorithms as Treatments: Illustrating DoE Concepts

Input data set = graph isomorphism class
(64--128 randomly relabeled/reordered instances)

For this input class, variance should be 0 -- in general, not achievable for all instances!!

Algorithms as Treatments on C432_iso: Note the Variance

rare case of 0 variance:
bi-partitioning by hmetis

non-trivial variance observed
bi-partitioning by prop

crossing number, and later, hyperedge length, are two of distinguishing characteristics
of a graph we use to define existing equiv. classes and to generate new ones.
The Crossing Number and Hyperedge Length in Bigraph

This remarkable correlation would not be discovered without designing a series of experiments with graph equivalence classes.

Equivalence Classes to Evaluate Graph Algorithms

Desirable but far from sufficient/complete characterizations:

1. the same number of nodes
2. the same number of edges
3. the same node pin distributions (hyperedges)
4. the same number of levels for given ranked order of nodes
5. the same number of connected components

Major graph classes other than isomorphism:

- random class:
  - strictly maintains (1) only, approximately (2), none others
- clone class (Hutton et al. '97):
  - strictly maintains (1) only, approximately (2) and (4), none others
- mutant class (Ghosh et al. '98):
  - strictly maintains (1), (2), (3), (4), (5)

However, as the experiments demonstrate next, even (1--5) are not sufficient/complete characterizations of a class:

- clone versus mutant class
- random versus mutant class
- clone, mutant, and sibling classes versus isomorphism class
**c432 Instances from Isomorphism, Clone, Mutant Classes**

- all graphs have 179 nodes, 355 edges, and 24 rank-order levels
- histograms based on 64 instances in each class and 3 algorithms show:
  - different means with variance $>> 0$ in general
  - both clone and mutant class differ significantly from isomorphism class
  - shifts in distributions for clone and mutant class depend on the choice of algorithm!!

---

**c432 histograms for Isomorphism, Clone, Mutant Classes**
Which, if any, Class is Best Suited for Comparative Performance Evaluation?

**Background:**
When tested on a circuit isomorphism class, an ideal algorithm will return an optimized cost with variance = 0!!

**Our objectives:**
- use a circuit from the isomorphism class as a reference circuit
- find an equivalence class of circuit mutants related to the reference circuit (i.e. its isomorphism equivalence class)

**Key criteria:**
- mean(cost) is nearly the same for both classes
- variance(cost) is larger, but not much larger for the mutant class

---

**Graph Equivalence Classes and the Magic Bullet**

Since the following characterization is necessary to synthesize a mutant class:

1. the same number of nodes
2. the same number of edges
3. the same node pin distributions (hyperedges)
4. the same number of levels for given ranked order of nodes
5. the same number of connected components

but not sufficient for to consistently distinguish between two algorithm w.r.t. the isomorphism class, we look further ..... and define and demonstrate merits of a sibling class as an extension of mutant class:

6. the same average crossing number (hyperedge length) of the underlying spanning tree of each bipartite connected component of the reference isomorphism class
7. the order of net nodes and cell nodes in the “average” spanning tree

**Note:** finding the minimum crossing number in a bipartite embedding of a tree is O(n^2), hence the characterization is feasible.
G(17,10,50) Instances from Random and Sibling Classes

Each instance has (n0=17, n1=10, e=50), each spanning tree has 26 edges

Random class: avg(CN for spanning tree) = 8.61

Sibling class1: avg(CN for spanning tree) = 6.69

Sibling class2: avg(CN for spanning tree) = 4.00

Lessons Learned: Random/Sibling Classes of G(17,10,50)

Given that

Random class: avg(CN for spanning tree) = 8.61, avg(CN) ----------------- TR12 205.6 TR17 209.0

Sibling class1: avg(CN for spanning tree) = 6.69, avg(CN) ----------------- TR12 132.4 TR17 129.5

Sibling class2: avg(CN for spanning tree) = 4.00, avg(CN) ----------------- TR12 101.6 TR17 66.0

• random class does not differentiate between algorithms (TR12, TR17)

• sibling class1 does not differentiate between algorithms (TR12, TR17), but may or may not for some new algorithm

• sibling class2 strongly differentiates between algorithms (TR12, TR17)

• correlation between avg(CN for spanning tree) and avg(CN for graph) is present and STRONGLY CONFIRMED by a series of experiments on well-defined equivalence classes of siblings!!
Lessons Learned: CN(ST) and CN(graph) Correlate!!

![Graph showing correlation between average crossing numbers (avg(CN)) for full graphs and spanning trees for different classes.](image)

On Synthesis of Mutant/Sibling Equivalence Classes

Major steps are (for simplicity, only acyclic case is considered here):

1. identify a reference hypergraph to form a corresponding isomorphism class
2. transform the reference hypergraph into a canonical 2k-partite graph form
3. extract and save characteristic parameters from the canonical form
   (nodes, edge distributions, ..., bounds on connectivity,
   **target hyperedge lengths** for each connected component in the
   bipartite slice after crossing number minimization)
4. synthesize, bottom up, $i=0 .. 2k$, the connected component in each
   bipartite slice, subject to bounds on connectivity and target hyperedge length
   within each component.
5. verify each synthesized graph instance for
   -- NOT being a member of the isomorphism class
   -- pre-specified “distance” from the isomorphism class in terms of
     • global and CC/spanning tree crossing numbers
     • global and CC/spanning tree hyperedge lengths
     • one or more algorithms that minimize cost functions
       considered critical for the equivalence class
A Simple Bipartite Slice: a Naive Synthesis Algorithm

0 1 1 0 1 0 = 3

reference graph
HL(net) = 3

1 1 2 2 = 6

integer repartition on net nodes
HL(cell) = 6

0 0 1 0 0 2 = 3

permutation on cell nodes
feasible sibling

2 1 2 1 = 6

integer repartition on net nodes
permutation on cell nodes
sibling is NOT feasible

violations

A Pragmatic Synthesis Algorithm: Few Results

Here, the search for feasible solutions does not involve explicit
- integer partitions on net nodes
- permutations on cell nodes

Rather, the search finds the subsets of partial solutions quickly while incorporating constraints towards global feasible solution dynamically.

As shown below, results are very encouraging.

<table>
<thead>
<tr>
<th>Size</th>
<th>ISO Class</th>
<th>Sibling Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CN</td>
<td>HL</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>180</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>372</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>756</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1524</td>
</tr>
</tbody>
</table>

>------------------------------------------
>Size                  |
>ISO Class  | Sibling Class
>------------------------------------------
>| CN | HL | CN | HL |
>|-----|-----|-----|-----|
>| 0   | 12  | 0.08 | 12.1 |
>| 0   | 36  | 1.2 | 38 |
>| 0   | 180 | 1.43 | 182 |
>| 0   | 372 | 3.60 | 375 |
>| 0   | 756 | 5.0 | 760 |
>| 0   | 1524 | 21.0 | 1526 |
>------------------------------------------
Experimental Results

Quality of the mutants/siblings is tested through a number of experiments with various physical design tools.

Goals to be Achieved

The goal is to demonstrate that mutant classes we generate meet the desired criteria:

- \( \text{mean(cost)} \) is near-same for both classes
- \( \text{variance(cost)} \) is larger, but not much larger for the mutant class

Tools used:

- DOT for wire crossing minimization
- OASIS for standard cell placement and route
- VPR for FPGA place and route
- PROP and HMETIS for partitioning

Circuit classes used:

- isomorphism class (S1423)
- WSI/SCC class (S1423)
Wire Crossing Min. (DOT) and Bipartitioning (PROP)

Two different physical design problems -- similar results:

• DOT: means differ by 3.0%
  t-score = 1.3
  (more than 95/100 have same mean)

• PROP: means differ by 12.0%
  t-score = 3.1

Layout with Standard Cell (OASIS) and FPGA (VPR)

Two different layout algorithms -- similar results:

• OASIS: means differ by 2.2%
  t-score = 6.6

• VPR: means differ by 4.1%
  t-score = 2.3
  (~90/100 have the same mean)
Results of Partitioning (HMETIS)

- relatively small variance of ISO class appears to give HMETIS a significant edge for “testing” the key criteria on “quality” of circuit mutants ......
  - but
- best mincut(2-way) reported by PROP is 4, while METIS reports 12
  - thus
- low variance in ISO class gives NO guarantee of solution’s optimality

Comparison of the Proposed Approach with Existing Methods

- Significant contributions have been made by:

- Need to check how the ‘mutant equivalence classes’ compare with the ‘clones’ generated by Hutton et al.

- Need to compare the performance of mutants generated from a number of reference circuits.

- Circuits used: C499, C1908, C432, 9symml, C880, C1355, i5, C6288, des.

- Tools used: HMETIS, OASIS, VPR, PROP, DOT.

- Methodology: (1) Generate (a) isomorphic, (b) clone, (c) mutant equivalence classes for each reference circuit.
  (2) Generate a distribution of the cost reported by each tool.
  (3) Compare the mean and std. dev. of the distributions of iso., clone and mut. classes for each tool.
Mean of Hmetis Distribution (for Iso., Clone, Mutant)

- Mutants remain close to the Iso. Class under all of 2, 4, 8, 16-way partitioning
- Clones tend to diverge from the Iso. class as we increase the circuit complexity
- HMETIS returns a near-delta functions for small circuits, not so much for larger ones

Summary of Comparisons

- We report the % difference of the Clone and Mutant equivalence classes w.r.t. the isomorphic classes (average for all the classes used for experimentation).

<table>
<thead>
<tr>
<th>Tool</th>
<th>Difference in Mean Mutant</th>
<th>Difference in Mean Clone</th>
<th>Difference in Std. Dev. Mutant</th>
<th>Difference in Std. Dev. Clone</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMETIS</td>
<td>14.46%</td>
<td>221.7%</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>DOT</td>
<td>27.30%</td>
<td>79.66%</td>
<td>94%</td>
<td>330%</td>
</tr>
<tr>
<td>PROP</td>
<td>9.52%</td>
<td>236%</td>
<td>21%</td>
<td>47%</td>
</tr>
<tr>
<td>OASIS</td>
<td>5.26%</td>
<td>58.9%</td>
<td>16.6%</td>
<td>85%</td>
</tr>
<tr>
<td>VPR</td>
<td>7.47%</td>
<td>53.3%</td>
<td>32.8%</td>
<td>105%</td>
</tr>
</tbody>
</table>
What About Mutants vs Siblings?

Unlike clones and mutants, sibling class for c432 is, with high confidence, very close to the isomorphism class.

Conclusions

(1) The test of hypothesis, as applied to the performance evaluation of heuristics for NP-hard problems,

Are the measured improvements due to the improved design of heuristics or due to chance?

requires

(a) sufficient number and variety of (mutant) equivalence classes of data sets,

(b) rigorous application of the Design of Experiments methodology when evaluating the performance of the heuristics with these data sets.

(2) The emerging web-based technologies are essential to snowballing and sustaining this effort.