Performance Testing of Combinatorial Solvers With Isomorph Class Instances

Franc Brglez  
Computer Science  
NC STATE UNIVERSITY  
Raleigh, NC, USA

Jason Osborne  
Statistics  
NC STATE UNIVERSITY  
Raleigh, NC, USA


Highlights from prior peer-reviewed articles

1994 ... Needed: An Empirical Science of Algorithms
1995 ... Designing and reporting on computational experiments with heuristic methods
1996 ... Testing heuristics: We have it all wrong
1998 ... Design of Experiments to Evaluate Algorithms: Which Improvements Are Due to Improved Heuristic and Which Are Due to Chance?
2002 ... A Theoretician's Guide to the Experimental Analysis of Algorithms
2005 ... Experimental Computer Science: The Need for a Cultural Change
Also, a question from a sponsor ...

paraphrased from American Scientist, July 2002

Can one replicate these experiments?

{ Wow, this guy must have dropped from CS ... }

Outline (the scientific method)
... in two major sections as per PNAS (6-pages):

(1) Materials and Methods
• Instance isomorphs (what, why, and how)
• Computational procedure (top view)

(2) Discussion
• Three designed experiments, each presented in three parts:
  (a) hypothesis and design
  (b) summary of observations
  (c) resolution of hypothesis
### About Instance Isomorphs

- Introduced for learning experiments by H. Simon (1969), and Tower of Hanoi *isomorphs* still being published (ACCSS’2000)


*Instance generation rules* for performance evaluation of algorithms depend on the problem domain:

- Almost trivial for many graph problems
- Need to be “invented” for some problem domains
- May be initiated within the scope of five generic rules

### Rules on Constructing Isomorphs

**Define, for a problem domain:**

- **R1: syntax**
  - *Calculator Dexterity* Test
  - *Example: find 2 roots*
  - **E1:** $ax^2 + bx + c$

- **R2: reference instance parameters**
  - **E2:** $a=2, b=-10, c=-100$

- **R3: instance invariants**
  - **E3:** $a$ digit, $b$ 2 digits, $c$ 3 digits

- **R4: parameter sampling range**
  - **E4:** $1 < a < 10$, $-100 < b <+9$, $-1000 < c < +99$

- **R5: sampling process to generate each instance**
  - **E5:**
    | I | a  | b  | c  |
    |---|----|----|----|
    | 0 | 2  | -10| -100|
    | 1 | 5  | -49| -631|
    | 2 | 3  | -71| -239|
    | ...| ...| ...|     |

*Dexterity is a term referring primarily to the ability of a person to "gracefully" coordinate their movements*
A Textbook Example: Two Sorters

Materials and Methods
- two isomorph classes of lists (5%, 100% items unsorted)
- list lengths are 32, 64, 128, … , 1024
- two QuickSorts: (1) list-based, (2) pivot-based

Discussion
Hypothesis: both sorters are equivalent
Design: run both sorters on all class instances and observe $\text{NumComparisons}$
Observation: solvers report different $\text{NumComparisons}$ means for some of the same-size isomorph classes
Resolution: Pivot-based QuickSort significantly dominates list-based QuickSort, but on 5% class only!

A list isomorph class

<table>
<thead>
<tr>
<th>reference</th>
<th>class A005</th>
</tr>
</thead>
<tbody>
<tr>
<td>in0</td>
<td>...</td>
</tr>
<tr>
<td>in1</td>
<td>...</td>
</tr>
<tr>
<td>in2</td>
<td>...</td>
</tr>
<tr>
<td>in3</td>
<td>...</td>
</tr>
</tbody>
</table>

class Apc
pairs = int(size*pc*100)
(64, 5%) => 3 (pairs permuted)
For the list class A005, the sorter RandomPivotQS performs significantly better than the sorter ListBasedQS.

Punchline: if lists are in 100% random order, the two solvers are equivalent!
Isomorph Examples: LA Problem

Where to place 2 warehouses to minimize travel to 7 locations?

ref problem

ref problem rot. -15 deg

ref problem rot. -30 deg

LA7x2

LA20x3

LA35x2

Just the “rotation of x-y” coordinates

............

... induces RunTime cost variability exceeding two orders of magnitude!

Isomorph Examples: cn Problem

Here, isomorphs are induced by permutation of vertices

ref instance (crossing no = 80)

I-class instance (crossing no = 101)

Optimum placement (crossing no = 5) for all instances in this class

Again, even with state-of-the-art cn-minimization algorithms, isomorphs may induce RunTime variability that spans orders of magnitude ....
Isomorph Examples: Boolean Functions

Here, isomorphs are induced by permutation and, when permissible, also complementation of variables ...

- reference instance
- iso-class instance

Such isomorphs may induce RunTime cost variability exceeding several orders of magnitude:

- logic optimization problems (IWLS'1999)
- BDD variable ordering problems (ICCAD'98, SSST'2001)
- SAT problems (SAT'2002, SAT'2003, AMAI'2005, ...)
- BCSP problems (DAC'2005, ExpCS'2007, ...)

Same Coin: Reliability vs Solvability

component (A/C unit of Boeing 720)

What lifetime?

component replicas

213 total

MTTF = 93.3 hours
Std = 107 hours

(component replicas)

(Prochan, 1963)

median = 2
mean = 14.6
std = 25.1

a case of heavy-tail ECDF --- median statistics is far from revealing ...

LA-x*y instance

What runtime to optimally place warehouses that minimize travel toy locations?

(instance replicas)

(rot. -15 deg)

(rot. -30 deg)

(24 total)
Method: A Generic Procedure

```vhdl
proc GenericExperiment {RefFile, ClassSize, ClassID, SolverIDs} {

    set ClassFiles [ClassGen $RefFile $ClassSize $ClassID ]

    foreach SolverID $SolverIDs {
        foreach file $ClassFiles {
            set Results($ClassID,$SolverID,$file) \ 
                [Encap $SolverID $file]
        }
    }
    set SummaryTables [GenTables [array get Results]]
    return $SummaryTables
}
```

In this paper: RefFile = {in401_sp, f51mb_0350, ...}

ClassSize = 32, ClassID = {LR, CLR}

SolverIDs = {cplex090, cplex090-dfs, ...}

Reference Files (46 total)

Instance directories:
  - min vertex cover, max independent set, max clique,
  - min unate cover, min binate cover, max set packing

Instance parameters:
  - variables: 125 -- 2000, constraints: 187 -- 74959,
  - sparsity: 0.1% -- 7.8%, completeness: 1.18% -- 90.88%
  - block-structured with hidden solutions: 2 -- 16

RunTimes (cplex090, branch-and-bound):
  - 3.56 -- 2132.1 seconds
  - RunTimes >= 2112 seconds represent TimeOut,
  - ObjectiveBest proven as optimum only on
  - 13 out of 46 instances

See Table 1 for more details ....
Examples of ClassIDs \{LR, CLR\}

reference instance

\[ \text{Max} \]
\[
+21x_1 + 22x_2 + 23x_3 + 25x_4 + 26x_5 + 27x_6 + 34x_7 \\
\text{st} \\
\begin{align*}
e_1: & \quad +x_3 + x_4 & \geq & & +1 \\
e_2: & \quad -x_2 & \geq & -x_4 & = & -2 \\
e_3: & \quad +x_3 + x_5 - x_7 & \geq & 0 \\
e_4: & \quad -x_3 + x_7 & \geq & 0 \\
e_5: & \quad -x_1 - x_4 & = & -1 \\
e_6: & \quad -x_1 - x_3 - x_7 & \geq & -1 
\end{align*}
\]

class LR isomorph

\[ \text{Max} \]
\[
+21x_1 + 22x_2 + 33x_3 + 25x_4 + 26x_5 + 27x_6 + 34x_7 \\
\text{st} \\
\begin{align*}
e_3: & \quad -x_1 - x_4 & \geq & -1 \\
e_1: & \quad -x_4 - x_7 & \geq & -1 \\
e_5: & \quad -x_5 - x_6 & \geq & -2 \\
\text{opt} & \quad +x_3 + x_4 + x_7 & \geq & +1 \\
\text{opt} & \quad -x_2 - x_3 - x_4 & \geq & 0 \\
\text{opt} & \quad -x_3 - x_4 & \geq & -1 \\
\text{opt} & \quad +x_3 + x_6 + x_8 & \geq & 0 
\end{align*}
\]

class CLR isomorph

\[ \text{Max} \]
\[
+21x_1 + 22x_2 + 23x_3 + 25x_4 + 26x_5 + 27x_6 + 34x_7 \\
\text{st} \\
\begin{align*}
e_3: & \quad -x_1 - x_4 & \geq & -1 \\
e_1: & \quad -x_4 - x_7 & \geq & -1 \\
e_5: & \quad -x_5 - x_6 & \geq & -2 \\
\text{opt} & \quad +x_3 + x_4 + x_7 & \geq & +1 \\
\text{opt} & \quad -x_2 - x_3 - x_4 & \geq & 0 \\
\text{opt} & \quad -x_3 - x_4 & \geq & -1 \\
\text{opt} & \quad +x_3 + x_6 + x_8 & \geq & 0 
\end{align*}
\]

SolverIDs and ClassIDs

Six versions of CPLEX as branch-and-bound solver under the timeout constraint of 2112 seconds (on each instance)

\begin{itemize}
\item cplex090
\item cplex090-dfs
\item cplex090-feas2
\item cplex101
\item cplex101-dfs
\item cplex101-feas2
\end{itemize}

Instance classes of most interest (timeout is rare):

\begin{itemize}
\item in401_sp_LR
\item in401_sp_CLR
\item f51mb_350_LR
\item f51mb_350_CLR
\end{itemize}
Outline (the scientific method)

... in two major sections as per PNAS (6-pages):

(1) Materials and Methods
• Instance isomorphs (what, why, and how)
• Computational procedure (top view)

(2) Discussion
• Three designed experiments, each presented in three parts:
  (a) hypothesis and design
  (b) summary of observations
  (c) resolution of hypothesis

Hypothesis and Design 1

Hypothesis:
For the same reference instance, the isomorph class CLR is equivalent to the isomorph class LR.

Design:

<table>
<thead>
<tr>
<th></th>
<th>cplex090</th>
<th>cplex101</th>
</tr>
</thead>
<tbody>
<tr>
<td>*_LR</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>* _CLR</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

class size N = 32

<table>
<thead>
<tr>
<th>references</th>
<th>variables</th>
<th>constraints</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>in401_sp</td>
<td>500</td>
<td>1000</td>
<td>max set packing</td>
</tr>
<tr>
<td>f51mb_350</td>
<td>350</td>
<td>374</td>
<td>min binate cover</td>
</tr>
</tbody>
</table>
Design 1: Results (in401_sp_LR/CLR)

Hypothesis resolution (ANOVA test):
There is no evidence of any difference in RunTime due to the rule that generates isomorphs in LR class and the rule that generates isomorphs in CLR class.

Design 1: Results (f51mb.._LR/CLR)

Hypothesis resolution (parametric test):
There is no evidence of any difference in RunTime due to the rule that generates isomorphs in LR class and the rule that generates isomorphs in CLR class.
Hypothesis and Design 2

Hypothesis:
The branch-and-bound performance of solvers cplex090 and cplex101, w/o options, are equivalent.

Design:

<table>
<thead>
<tr>
<th>references</th>
<th>variables</th>
<th>constraints</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>in401_sp</td>
<td>500</td>
<td>1000</td>
<td>max set packing</td>
</tr>
<tr>
<td>f51mb_350</td>
<td>350</td>
<td>374</td>
<td>min binate cover</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>cplex090</th>
<th>cplex101</th>
</tr>
</thead>
<tbody>
<tr>
<td>in401_sp_CLR</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>f51mb_350_CLR</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

class size
N = 32

Design 2: Results (in401_sp_CLR)

RunTime statistics

<table>
<thead>
<tr>
<th>Solver</th>
<th>Class</th>
<th>RefV</th>
<th>MinV</th>
<th>MaxV</th>
<th>MedV</th>
<th>MeanV</th>
<th>StdV</th>
<th>N</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>cplex090</td>
<td>in401_sp_CLR</td>
<td>885</td>
<td>407</td>
<td>957</td>
<td>638</td>
<td>666</td>
<td>153</td>
<td>32</td>
<td>uniform</td>
</tr>
<tr>
<td>cplex101</td>
<td>in401_sp_CLR</td>
<td>841</td>
<td>525</td>
<td>1316</td>
<td>816</td>
<td>848</td>
<td>142</td>
<td>32</td>
<td>normal</td>
</tr>
</tbody>
</table>

Hypothesis resolution (parametric test):

<table>
<thead>
<tr>
<th>class</th>
<th>cplex</th>
<th>median</th>
<th>95% confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>in401_sp_CLR</td>
<td>090</td>
<td>638.4</td>
<td>605.3</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>816.3</td>
<td>790.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>857.7</td>
</tr>
</tbody>
</table>
Design 2: Results (f51mb.._CLR)

RunTime statistics

<table>
<thead>
<tr>
<th>Solver</th>
<th>Class</th>
<th>RefV</th>
<th>MinV</th>
<th>MaxV</th>
<th>MedV</th>
<th>MeanV</th>
<th>StdV</th>
<th>N</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>cplex090</td>
<td>f51mb_350_CLR</td>
<td>115</td>
<td>71.3</td>
<td>2118</td>
<td>127</td>
<td>232</td>
<td>393</td>
<td>32</td>
<td>heavy-tail</td>
</tr>
<tr>
<td>cplex101</td>
<td>f51mb_350_CLR</td>
<td>87.3</td>
<td>53.5</td>
<td>2117</td>
<td>159</td>
<td>408</td>
<td>619</td>
<td>32</td>
<td>heavy-tail</td>
</tr>
</tbody>
</table>

Hypothesis resolution (parametric test):

<table>
<thead>
<tr>
<th>class</th>
<th>cplex</th>
<th>median</th>
<th>95% confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>f51mb_350_CLR</td>
<td>090</td>
<td>127.4</td>
<td>105.6 - 148.8</td>
</tr>
<tr>
<td>f51mb_350_CLR</td>
<td>101</td>
<td>159.4</td>
<td>110.4 - 237.3</td>
</tr>
</tbody>
</table>

Hypothesis and Design 3

Hypothesis:

*The branch-and-bound performance of any pair of solvers, formed from the table below, are equivalent.*

Design:

<table>
<thead>
<tr>
<th>solver</th>
<th>solver options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>none</td>
</tr>
<tr>
<td>cplex090</td>
<td>✓</td>
</tr>
<tr>
<td>cplex101</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>classes</th>
<th>class size</th>
</tr>
</thead>
<tbody>
<tr>
<td>in401_sp_CLR</td>
<td>N = 32</td>
</tr>
<tr>
<td>f51mb_350_CLR</td>
<td></td>
</tr>
</tbody>
</table>
Design 3: Results (in401_sp_CLR)

RunTime statistics

<table>
<thead>
<tr>
<th>Solver</th>
<th>Class</th>
<th>RefV</th>
<th>MinV</th>
<th>MaxV</th>
<th>ModV</th>
<th>MeanV</th>
<th>StdV</th>
<th>N</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>cplex000</td>
<td>in401_sp_CLR</td>
<td>865</td>
<td>407</td>
<td>957</td>
<td>638</td>
<td>666</td>
<td>133</td>
<td>32</td>
<td>uniform</td>
</tr>
<tr>
<td>cplex101</td>
<td>in401_sp_CLR</td>
<td>841</td>
<td>625</td>
<td>1316</td>
<td>816</td>
<td>843</td>
<td>142</td>
<td>32</td>
<td>normal</td>
</tr>
<tr>
<td>cplex099-dfs</td>
<td>in401_sp_CLR</td>
<td>798</td>
<td>411</td>
<td>748</td>
<td>574</td>
<td>576</td>
<td>85.7</td>
<td>32</td>
<td>uniform</td>
</tr>
<tr>
<td>cplex101-dfs</td>
<td>in401_sp_CLR</td>
<td>678</td>
<td>592</td>
<td>1200</td>
<td>904</td>
<td>925</td>
<td>149</td>
<td>32</td>
<td>normal</td>
</tr>
<tr>
<td>cplex099-feas2</td>
<td>in401_sp_CLR</td>
<td>451</td>
<td>321</td>
<td>493</td>
<td>413</td>
<td>416</td>
<td>38.9</td>
<td>32</td>
<td>uniform</td>
</tr>
<tr>
<td>cplex101-feas2</td>
<td>in401_sp_CLR</td>
<td>1491</td>
<td>950</td>
<td>1987</td>
<td>1496</td>
<td>1510</td>
<td>247</td>
<td>22</td>
<td>normal</td>
</tr>
</tbody>
</table>

Design 3: Results (f51mb.._CLR)

RunTime statistics

<table>
<thead>
<tr>
<th>Solver</th>
<th>Class</th>
<th>RefV</th>
<th>MinV</th>
<th>MaxV</th>
<th>MedV</th>
<th>MeanV</th>
<th>StdV</th>
<th>N</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>cplex060</td>
<td>f51mb_350_CLR</td>
<td>115</td>
<td>71.3</td>
<td>2118</td>
<td>127</td>
<td>292</td>
<td>393</td>
<td>32</td>
<td>heavy-tail</td>
</tr>
<tr>
<td>cplex101</td>
<td>f51mb_350_CLR</td>
<td>87.3</td>
<td>35.5</td>
<td>2117</td>
<td>159</td>
<td>408</td>
<td>619</td>
<td>32</td>
<td>heavy-tail</td>
</tr>
<tr>
<td>cplex060-dfs</td>
<td>f51mb_350_CLR</td>
<td>102</td>
<td>60.1</td>
<td>2117</td>
<td>225</td>
<td>388</td>
<td>526</td>
<td>32</td>
<td>near-exponential</td>
</tr>
<tr>
<td>cplex101-dfs</td>
<td>f51mb_350_CLR</td>
<td>179</td>
<td>89.2</td>
<td>2116</td>
<td>227</td>
<td>441</td>
<td>585</td>
<td>32</td>
<td>exponential</td>
</tr>
<tr>
<td>cplex060-feas2</td>
<td>f51mb_350_CLR</td>
<td>115</td>
<td>49.2</td>
<td>446</td>
<td>94.2</td>
<td>113</td>
<td>69.8</td>
<td>32</td>
<td>near-exponential</td>
</tr>
<tr>
<td>cplex101-feas2</td>
<td>f51mb_350_CLR</td>
<td>99.1</td>
<td>58.1</td>
<td>2118</td>
<td>127</td>
<td>316</td>
<td>501</td>
<td>32</td>
<td>heavy-tail</td>
</tr>
</tbody>
</table>
Design 3: Hypothesis Resolution

RunTime means

<table>
<thead>
<tr>
<th>Instance class</th>
<th>Solver Options</th>
<th>Instance class</th>
<th>Solver Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cplex</td>
<td></td>
<td>cplex</td>
</tr>
<tr>
<td></td>
<td>none</td>
<td></td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>dfs</td>
<td></td>
<td>dfs</td>
</tr>
<tr>
<td></td>
<td>feas2</td>
<td></td>
<td>feas2</td>
</tr>
<tr>
<td>090</td>
<td>666.1</td>
<td>090</td>
<td>231.6</td>
</tr>
<tr>
<td>101</td>
<td>843.3</td>
<td>101</td>
<td>408.3</td>
</tr>
<tr>
<td>Diff</td>
<td>177.2</td>
<td>Diff</td>
<td>176.7</td>
</tr>
<tr>
<td></td>
<td>349.0</td>
<td></td>
<td>52.6</td>
</tr>
<tr>
<td></td>
<td>1094.2</td>
<td></td>
<td>202.6</td>
</tr>
</tbody>
</table>

Using Tukey-Kramer adjustments to control experimental error rate at 5% in pairwise comparisons among the means, all 6*5 = 15 pairs differ significantly ... there are effects from both Solvers and Solver Options: the effects of one factor depend on the level of the other factor ....

The solver cplex090-feas2 dominates all other solvers..

Conclusions

- not about whether cplex090 beats cplex101 on BCSP instances (it appears likely ...
- but about application of the scientific method to assessing the performance of combinatorial solvers

(1) by designing instance isomorph classes
(2) by reporting on reproducible solver experiments on such classes

• and about making more reliable decisions whether improvements of heuristics are due to the design or due to chance!
Future Work

• update home page of xBed: http://www.cbl.ncsu.edu/xBed/ (Datasets, Open Experiments, and Utilities)

• complete the work on block-dominant compositions of BCSP datasets with hidden solutions, including the refinement for finer control of variable size increments ....

• create new data sets for robust designs of new and scalable heuristics .... (an invitation for a collaborative project)