Performance Prediction and Preselection for Optimization Procedures

Nicholas G. Hall
Marc E. Posner

The Ohio State University

Presented at ISE Department
National University of Singapore

July 2009

Outline

- Research program
- Introduction to algorithm preselection
- Branch and search vs. dynamic programming
- Heuristic decision rules
- Computational study
- Conclusions and future research

Perspectives on Empirical Research

- Empirical research uses data to draw scientific conclusions
- Methodologies vary widely and are frequently *ad hoc*
- Leads to inconsistent and misleading results
- Compromises future research
- No operational guidelines exist for conducting empirical research

Research Program – Topic 1

- Real world data is expensive to collect, and may contain local anomalies
- Library data is limited and often biased
- In empirical testing, many studies use inconsistent methods for generating synthetic data
- Lack of general principles for synthetic data generation
- Need for guidelines for specific applications

Research Program – Topic 2

- Computational studies of optimization procedures fail to identify when a procedure works well or badly
- Multiple optimization procedures are often available
- It may not be possible to run all procedures within the available time and budget
- A systematic procedure for algorithm preselection is needed

Introduction to this Talk (Topic 2)

- Several competing optimization, or heuristic, procedures are available
- Standard approach to decide among them is to test against each other
- Procedure is computationally burdensome
- Usually, some data sets support one procedure, but other data sets support another
- Often unclear whether results can be extrapolated to other data sets

Objectives of Research

- Develop a method to predict the performance of different procedures, based on easily found characteristics of each data set
- Preselect a procedure from the characteristics of the data
- Validate the robustness of this choice
- Use knowledge of procedure performance to make other predictions

Related Literature

Garey and Johnson (1979)

- Complexity does not describe average performance

Cheesman, Kanefsky and Taylor (1991)

- Typical instances of many intractable problems are easy to solve
- Intractable instances occur when the data falls within ***critical ranges***

Angel and Zissimopoulos (1998)

- Success of local search can be predicted from ***autocorrelation coefficient***, the maximum distance between any two solutions, and the neighborhood size

Angel and Zissimopoulos (2000)

- Obtain a hierarchy of combinatorial optimization problems, based on their autocorrelation coefficients

Dolan and Moré (2001)

- Introduce the ***performance profiling*** method which compares performance across whole range of values

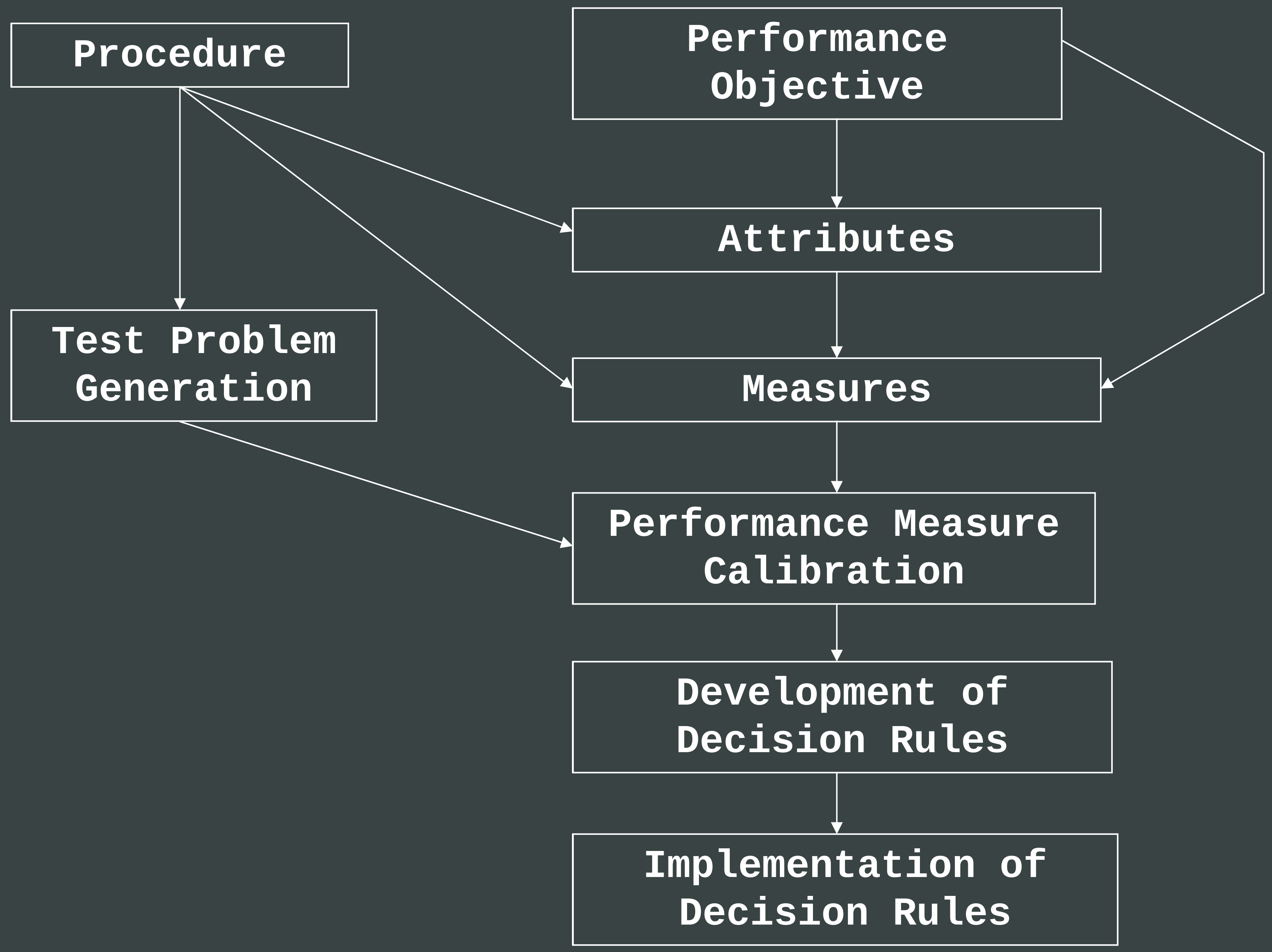
Research Strategy

Step 1

- Develop a methodology to predict performance of competing solution procedures, based on the characteristics of each data set

Step 2

- Select the procedure with the best predicted performance


Procedure
Performance Objective
Attributes
Test Problem Generation
Measures
Performance Measure Calibration
Development of Decision Rules
Implementation of Decision Rules

Performance Objectives

Most common performance objectives:

- For optimization procedures, computation time
- For heuristics, accuracy of the solutions delivered

Other possibilities include:

- Computer storage required
- Sensitivity of solutions to parameter changes

Attributes

Choose attributes (relevant data characteristics), depending on the solution procedures and the performance objectives

Example: for the 0-1 knapsack problem, as the ratio of value to item size becomes more uniform, the L.P. relaxation bound becomes weaker

Affects performance of solution procedures that use L.P. for bounding, but does not affect performance of dynamic programming

Measures

Measures, which quantify attributes, also depend on performance objectives

To evaluate a solution procedure that has minimum average computation time, choose a *continuous scale measure*

To evaluate whether termination will occur within a given time, choose a *categorical measure*, such as “short” *vs.* “long”

Performance Measure Calibration

Regression analysis

Discriminant analysis

ANOVA

Neural networks

Surrogate Measures

- Estimate expected value of measure
- Provide lower bound for measure
- Provide upper bound for measure

Development of Decision Rules

Method used depends on statistical results available:

- Reliable regression results are excellent when they exist
- Neural networks may define decision rules directly

If predictive power is weak, then an experimental process is useful:

- Find combinations of measure values where one procedure performs better than others

Validation of Decision Rules

Test computationally on a wide range of problems

Do not use the same data that was used for calibration

For example, allow interpolation between the original test values

Implementation of Decision Rules

Use the methodology to select a particular procedure for every given data set, based on the characteristics of that data set

Example: 0-1 Knapsack Problem

$$\text{Maximize} \quad cx$$

$$\text{subject to} \quad ax \leq b$$

$$x \in \{0, 1\}$$

- Both dynamic programming (DP) and branch and search (BAS) procedures work successfully
- Balas and Zemel (1980) show problems are harder when value and size are positively correlated
- Martello and Toth (1984) describe a hybrid approach that incorporates both procedures

Attributes and Measures

Problem size

- Number of jobs

Knapsack capacity

- Size of knapsack for DP
- Maximum, minimum or average number of items which the knapsack can hold for BAS

Characteristics of the item values

- Coefficient of variation of the item values

Characteristics of the item sizes

- Knapsack size / total item size
- Coefficient of variation of the item sizes

Relationship between item value and item size

- Proportion of pairs of items which are “dominant”, i.e. larger value and smaller size
- Coefficient of correlation of value and size
- Coefficient of variation of value / size ratio

Characteristics of the linear relaxation

- Gap between the value of the first integer solution and the lower bound
- Gap between the value of the first integer solution and the lower bound, normalized by the average item value
- Size of the linear programming core
- Balas-Zemel measure

Data Generation

- Number of jobs: 100, 500, 1000 (500 instances each)
- Knapsack size = total item size x {.1, .3, .5}
- Item sizes from U[LB, UB]
  - UB $\in$ {10, 100, 1000}
  - LB $\in$ {.1UB, .4UB, .8UB}
- $c_i \sim$ U[LB, 1000]
  - LB $\in$ {.1UB, .4UB, .8UB}
- Proportion of dominant pairs: .1, .3, .5, .7, .9

Experimental Design

- Full factorial design
- 10 random instances for each combination of parameters
- Total of 12,150 instances

Dynamic Programming

Value Function:

$f(i, q)$ = maximum value of a selection of items 1, 2, . . . , i with total size q

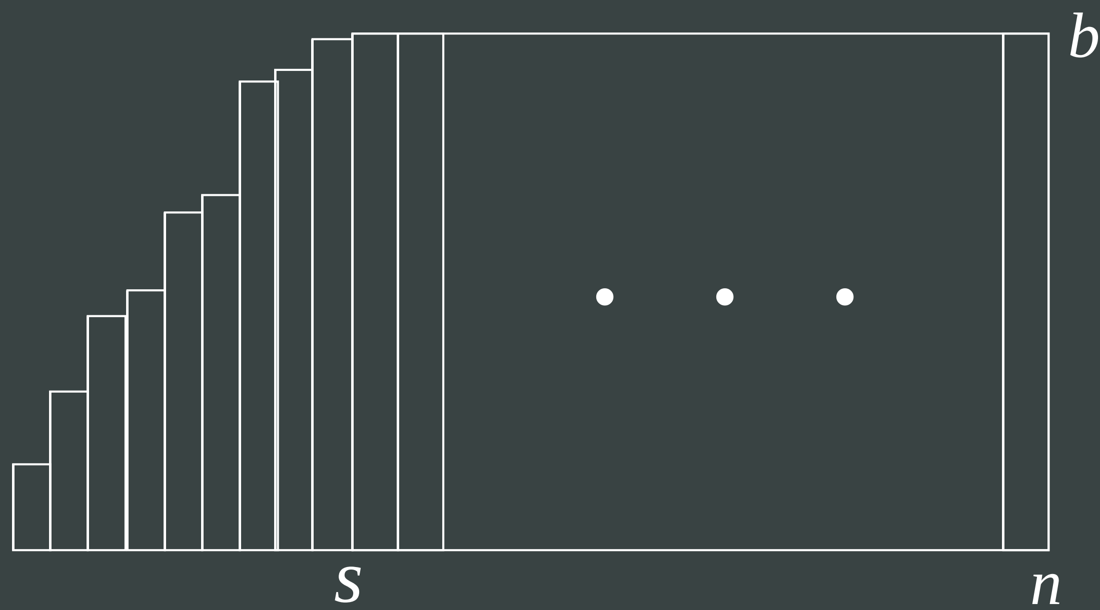
$$= max\{ f(i-1, q),\ c_i + f(i-1, q- a_i) \},\quad i = 1, 2, . . . , n,$$

$$q = 1, 2, . . . , min\{\ ,\ b \}.$$

Initial regression model

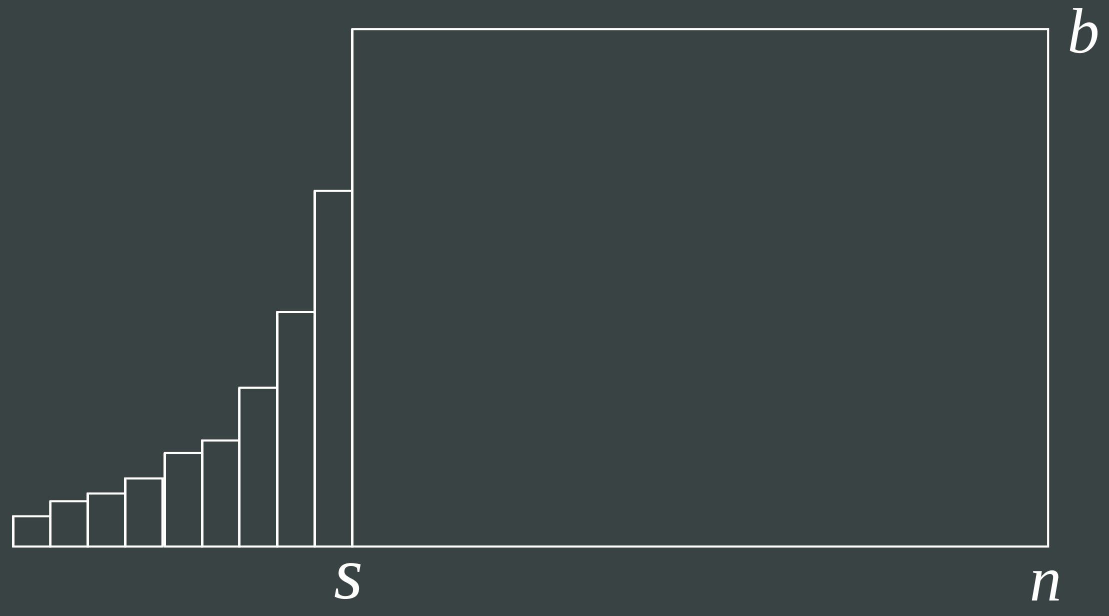
$$t = 0.0519 + 0.00000003nb$$

Systematic bias: When $j < S = max\{ j \mid \ \leq b \}$, because all b knapsack sizes are not considered for small n



Prediction improves when we include *S* in the model

⇨ R^2 of 97.7%



Advantage to sorting items in nondecreasing a_i order

Branch and Search

Formal statistical models do not provide reliably accurate point estimates of computation time for BAS

BAS works very fast most of the time, but has occasional very long computation times

Methodology to determine when poor performance will occur is needed

BAS does not have long computation times for certain combinations of

- Dominance proportion
- Coefficient of variation of item values
- Coefficient of variation of item sizes
- Average number of items the knapsack can hold

Numerous additional tests were performed at the boundaries of these ranges

Within the ranges, it is “safe” to use BAS

Heuristic to provide an estimate on the number of nodes that BAS has to evaluate

- Assume all items have equal size
- Estimate the maximum number of possible solutions that can fit in the knapsack
- Use a heuristic lower bound on optimal solution value for pruning

Estimation heuristic runs in $O(n^3)$ time

Provides an upper bound for 86% of instances

For remaining 14% of instances where heuristic underestimates the actual number of nodes, our computational results show that the actual number of nodes is less than 1,200 with probability .9975

⇒ When BAS is large, heuristic reports large

Procedure KPCHOICE

1. Preprocess data

2. Use BAS if
   - ⇨ dominance proportion, coefficient of variation of item values, coefficient of variation of item sizes, average number of items knapsack can hold, are in appropriate ranges, OR
   - ⇨ estimated time to evaluate heuristic number of nodes is less than predicted DP time

3. Otherwise, use DP

Testing Model Performance

Tested on 5,000 randomly generated problems

- $n \sim UI[100, 1000]$
- $a_i \sim UI[a_{min}, a_{max}]$, where
  - $a_{max} \sim UI[10, 1000]$, $a_{min} \sim UI[1, .8\, a_{max}]$
- $c_i \sim UI[c_{min}, 1000]$, where
  - $c_{min} \sim UI[1, 750]$
- Proportion of dominant pairs ~ U[.1, .9]
  - c_i's are adjusted using random interchange
- $b = size \sum a_i$, where $size \sim U[.1, .5]$

Performance of Procedures

Algorithm	BAS	DP	KPCHOICE	Best
Mean	35.73	1.35	0.53	0.25
Median	0.10	0.50	0.10	0.10
Maximum	1000.00	18.00	28.40	12.80

Improvement in *mean computation time* provided by KPCHOICE, compared to DP, is 74.5% of the best possible improvement

Summary of Results

- Methodology evaluates the relative effectiveness of different procedures
- 97.7% *R*-squared value for dynamic programming
- Using easily calculated measures of any given data set, the expected computation time can be significantly reduced
- Effective solution procedure preselection is possible

Conclusions

- An efficient method for the preselection of solution procedures for optimization problems
- Method incorporates various statistical alternatives, depending upon optimization problem and solution procedures being compared
- For the 0-1 knapsack problem, method performs significantly better than existing alternatives

Future Research (Topic 2)

- Extend preselection approach to other optimization problems
- Identify problems where preselection is valuable
- Use problem similarity to enlarge the predictive range of our approach
- Develop methods to obtain robust selections of two or more solution procedures from a larger set

Research Papers

Hall, N.G., M.E. Posner. 2001. Generating experimental data for computational testing with machine scheduling applications. *Operations Research* **49** 854-865.

Hall, N.G., M.E. Posner. 2007. Performance prediction and preselection for optimization and heuristic solution procedures. *Operations Research* **55** 703-716.

This work is supported by two grants from The National Science Foundation

Thank you for your attention!

Are there any questions?

Are there any questions?

Are there any questions?

Are there any questions?

Are there any questions?

Are there any questions?

Are there any questions?