Understanding Random SAT

Beyond the Clauses-to-Variables Ratio

Eugene Nudelman
Stanford University

joint work with...
Kevin Leyton-Brown
Holger Hoos
University of British Columbia

Alex Devkar
Yoav Shoham
Stanford University

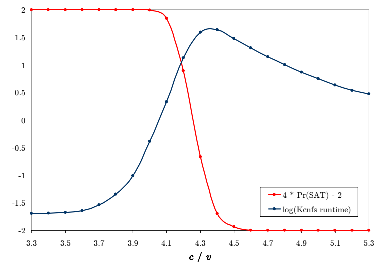
Introduction

- SAT is one of the most studied problems in CS
- Lots known about its worst-case complexity
  - But often, particular instances of $\mathcal{NP}$-hard problems like SAT are easy in practice
- "Drosophila" for average-case and empirical (typical-case) complexity studies

- (Uniformly) random SAT provides a way to bridge analytical and empirical work

Previously...

- **Easy-hard-less hard** transitions discovered in the behaviour of DPLL-type solvers [Selman, Mitchell, Levesque]
  - Strongly correlated with phase transition in solvability
  - Spawned a new enthusiasm for using empirical methods to study algorithm performance



- Follow up included study of:
  - Islands of tractability [Kolaitis et. al.]
  - SLS search space topologies [Frank et.al., Hoos]
  - Backbones [Monasson et.al., Walsh and Slaney]
  - Backdoors [Williams et. al.]
  - Random restarts [Gomes et. al.]
  - Restart policies [Horvitz et.al, Ruan et.al.]
  - ...

Empirical Hardness Models

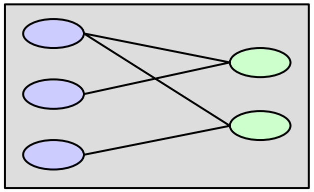
- We proposed building regression models as a disciplined way of predicting and studying algorithms' behaviour

  [Leyton-Brown, Nudelman, Shoham, CP-02]

- Applications of this machine learning approach:
  1) Predict running time
     - Useful to know how long an algorithm will run
  2) Gain theoretical understanding
     - Which variables are important to the hardness model?
  3) Build algorithm portfolios
     - Can select the right algorithm on a per-instance basis
  4) Tune distributions for hardness
     - Can generate harder benchmarks by rejecting easy instances

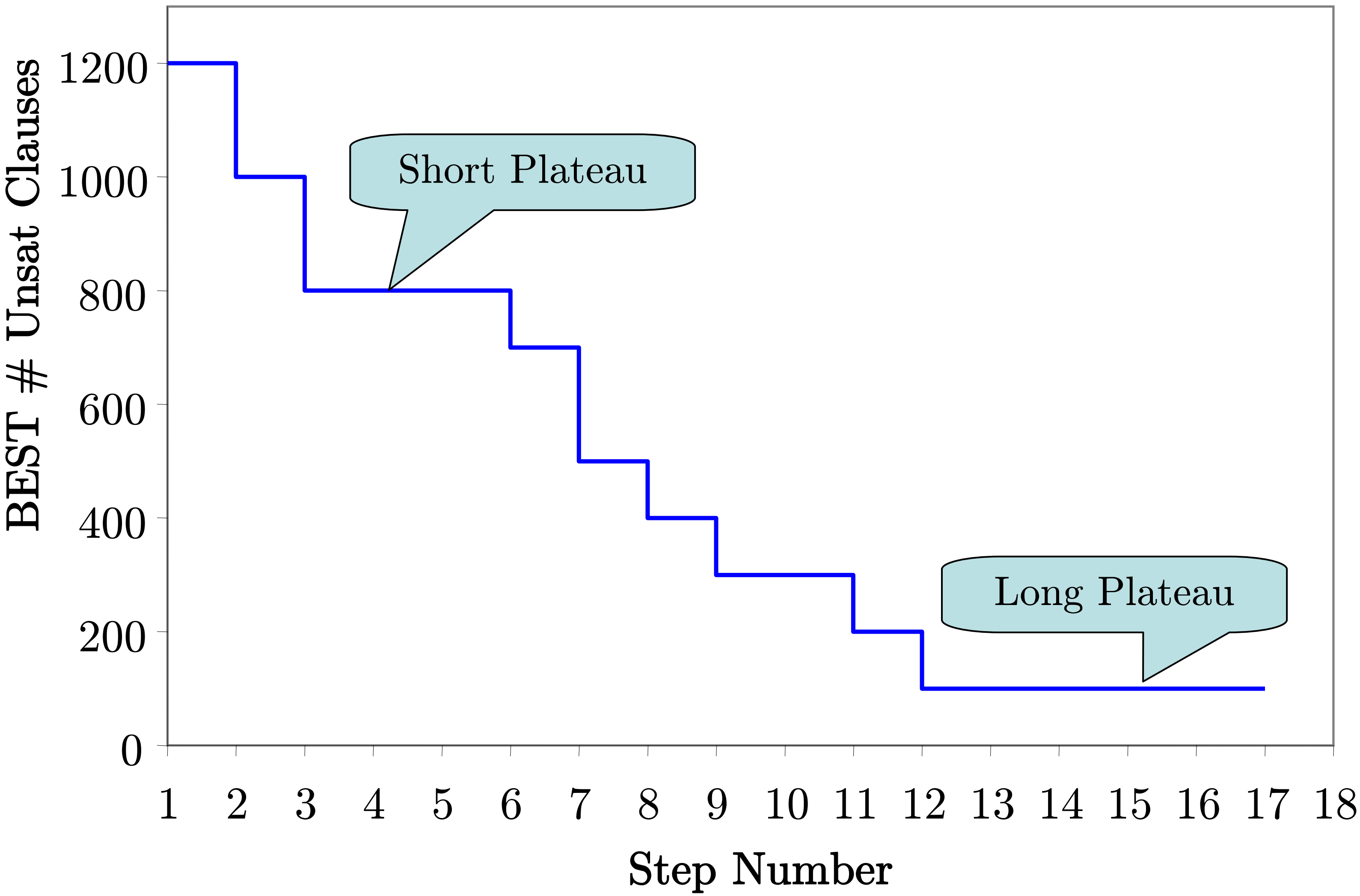
Outline

- Features

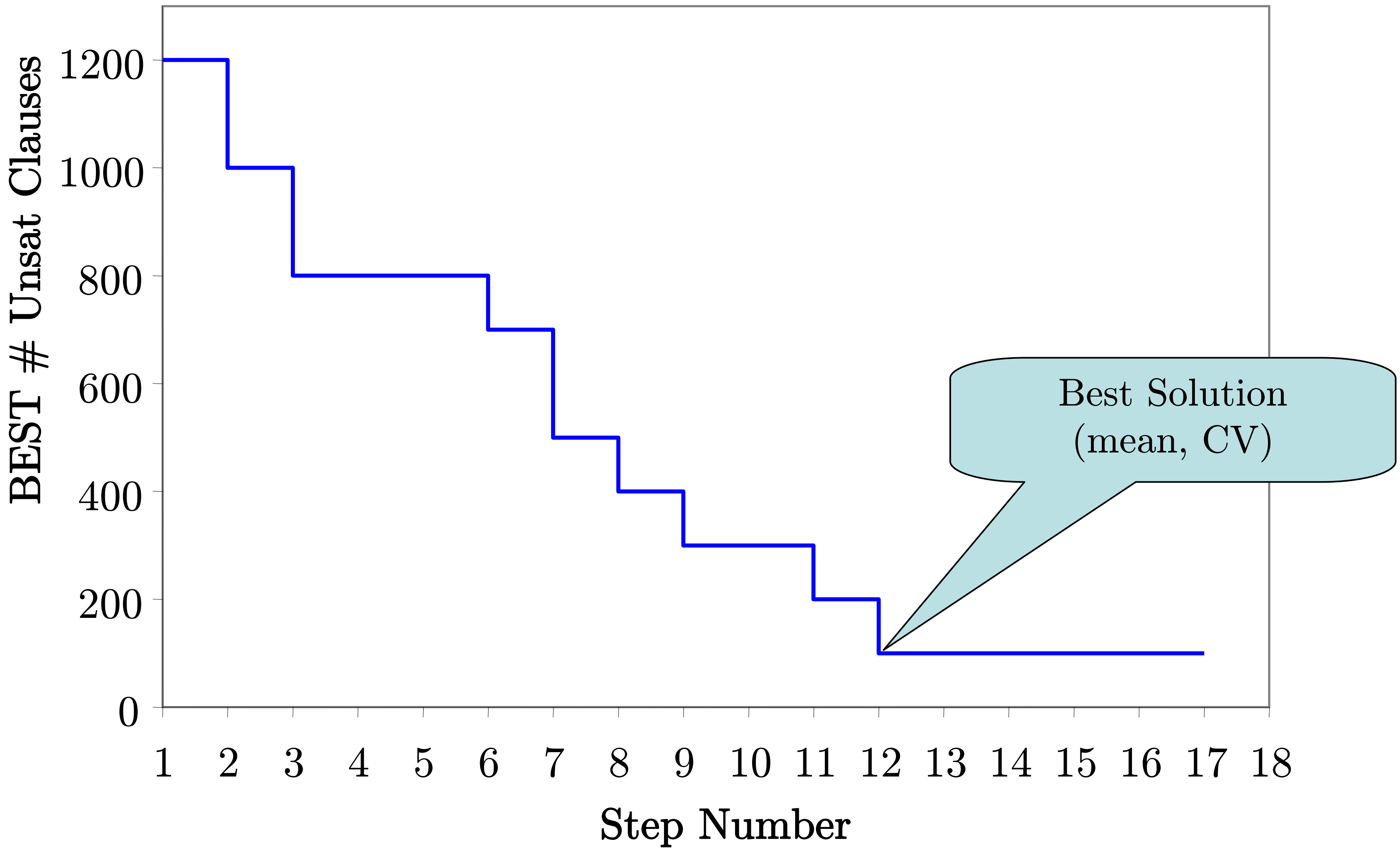


- Experimental Results
  - Variable Size Data
  - Fixed Size Data

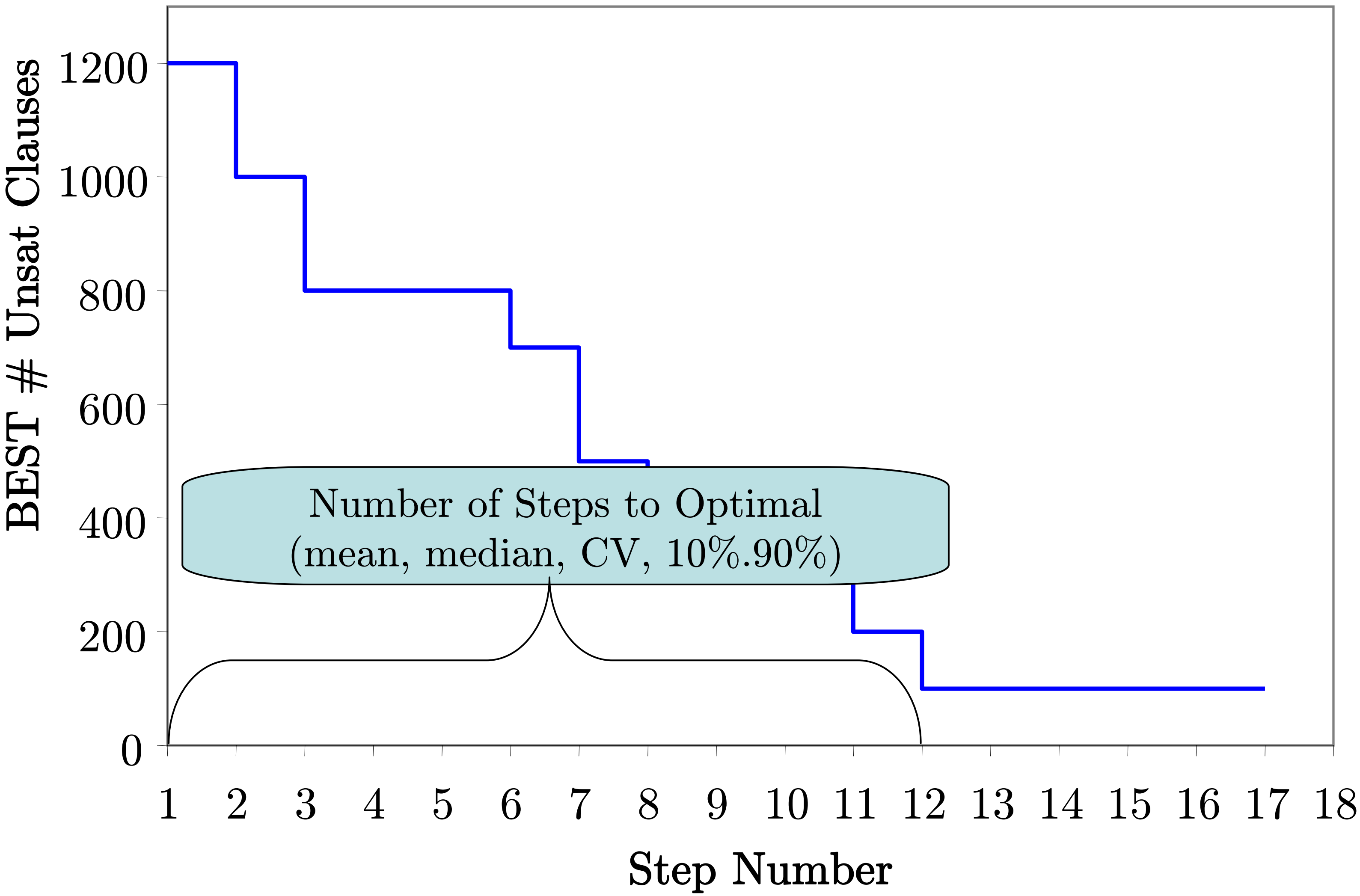
Features: Local Search Probing



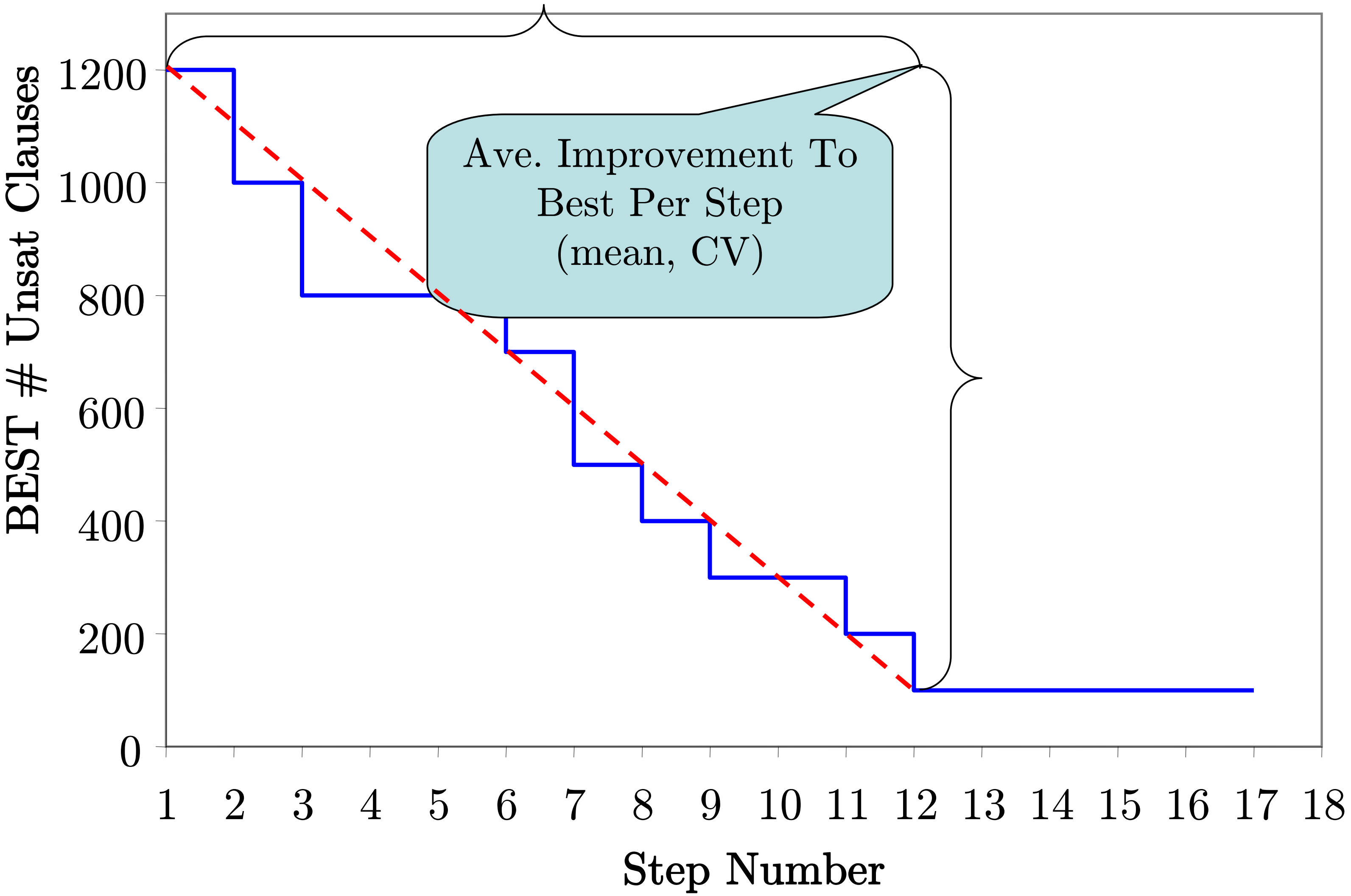
Features: Local Search Probing



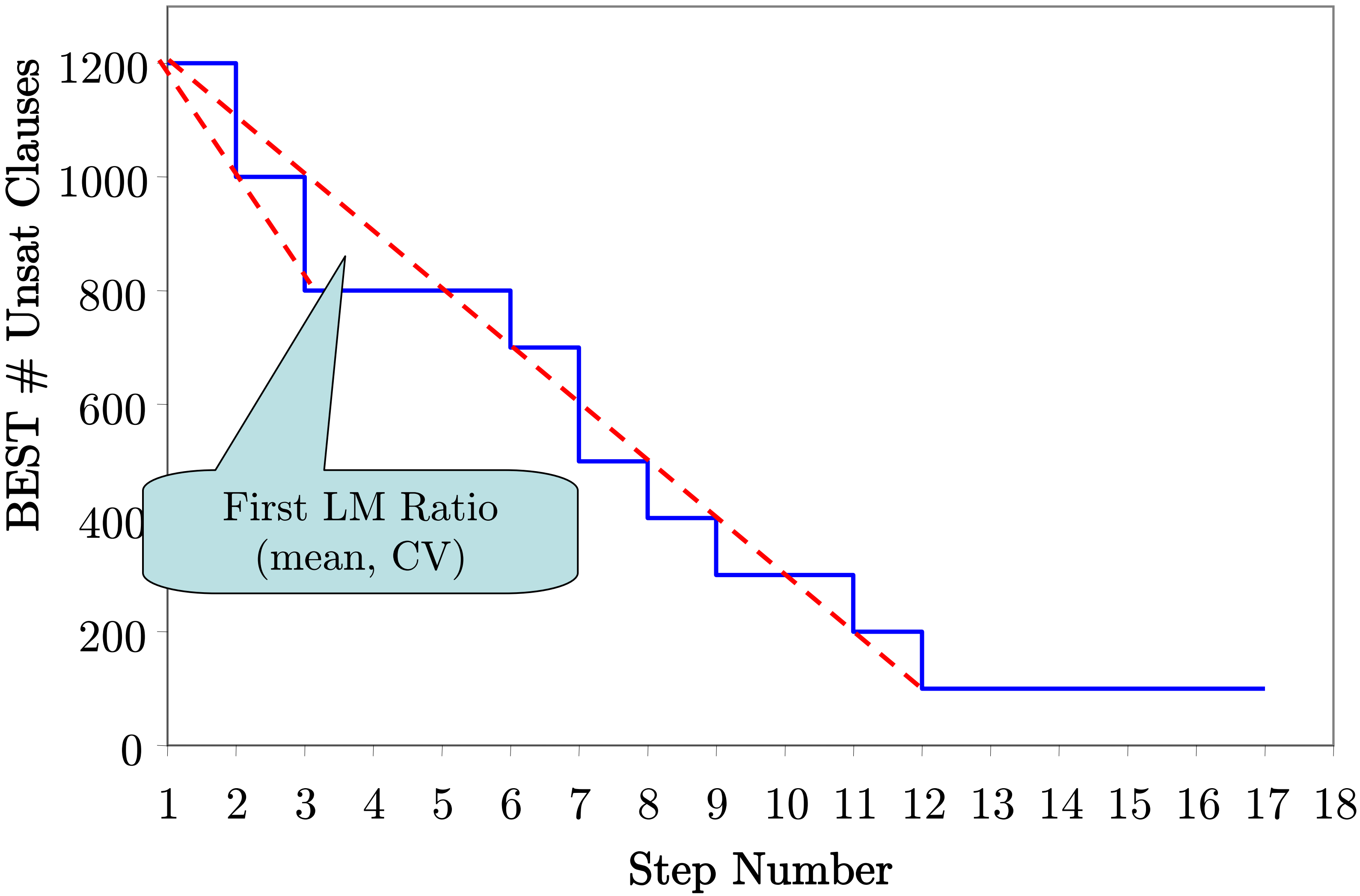
Features: Local Search Probing



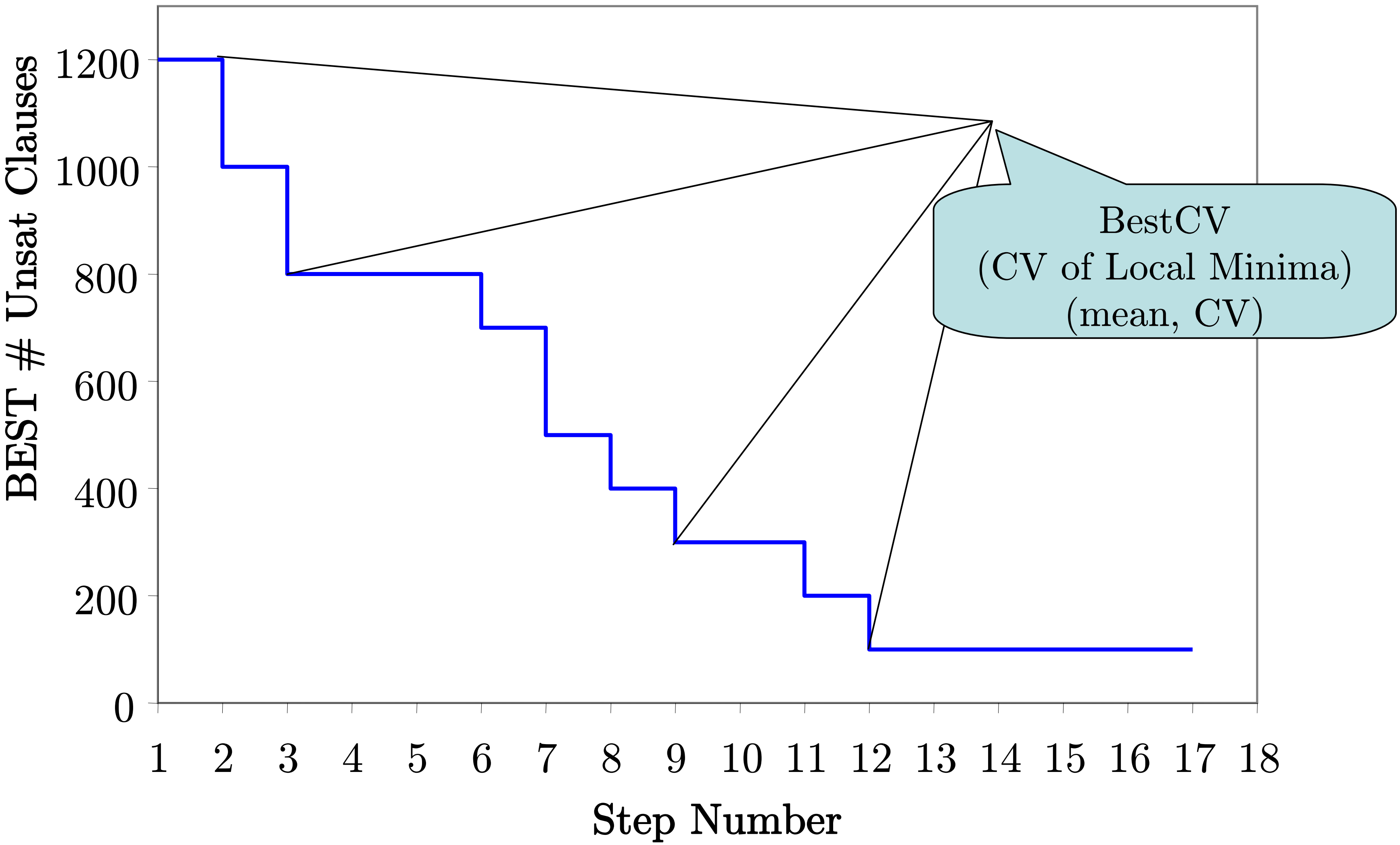
Features: Local Search Probing



Features: Local Search Probing

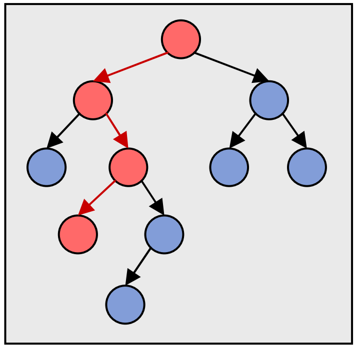


Features: Local Search Probing



Features: DPLL, LP

- DPLL search space size estimate
  - Random probing with unit propagation
  - Compute mean depth till contradiction
  - Estimate log(#nodes)



- Cumulative number of unit propagations at different depths (DPLL with Satz heuristic)

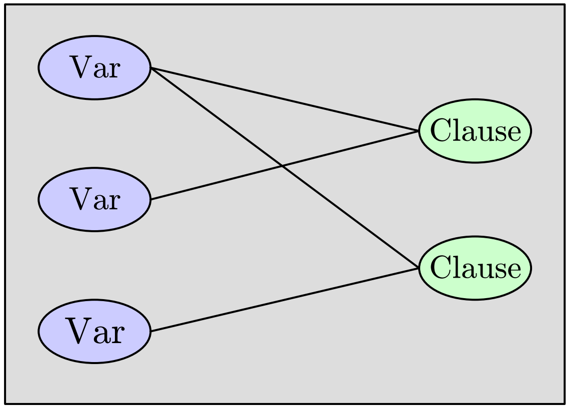
- LP relaxation
  - Objective value
  - stats of integer slacks
  - #vars set to an integer

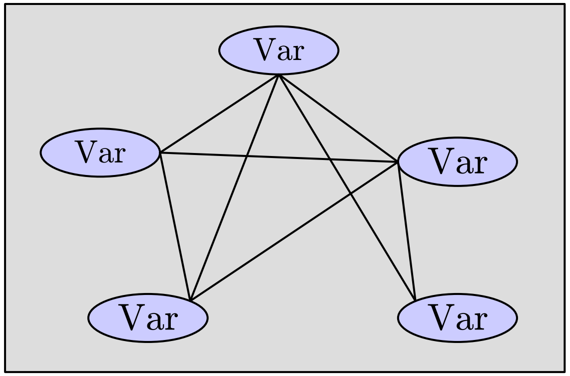
$$\begin{array}{lll} \text{maximize:} & \displaystyle\sum_{k\in C}\left(\sum_{i\in L,\, i\in k} v_i + \sum_{j\in \overline{L},\, i\in k}(1-v_j)\right) & \\ \text{subject to:} & \displaystyle\sum_{i\in k,\, i\in L} v_i + \sum_{j\in k,\, j\in \overline{L}}(1-v_j) \geq 1 & \forall k\in C \\ & v_i \in \{0,1\} & \forall i \end{array}$$

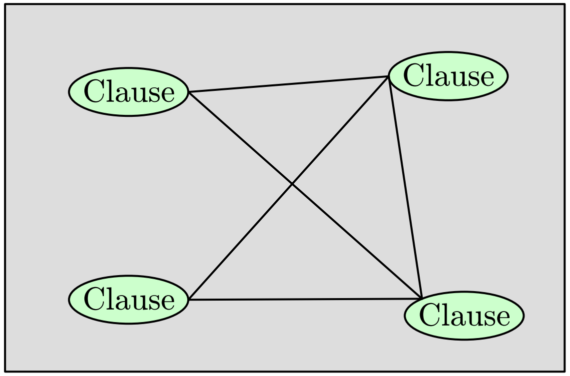
Other Features

- **Problem Size**:
  - v (#vars)
  - c (#clauses)

  } used for normalizing many other features
  - Powers of c/v, v/c, $|c/v - 4.26|$
- **Graphs**:
  - Variable-Clause (VCG, bipartite)
  - Variable (VG, edge whenever two variables occur in the same clause)
  - Clause (CG, edge iff two clauses share a variable with opposite sign)
- **Balance**
  - #pos vs. #neg literals
  - unary, binary, ternary clauses
- Proximity to **Horn formula**



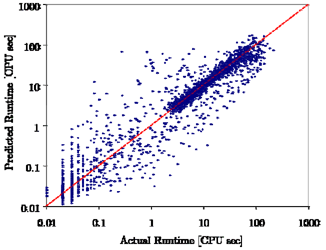




Outline

- Features

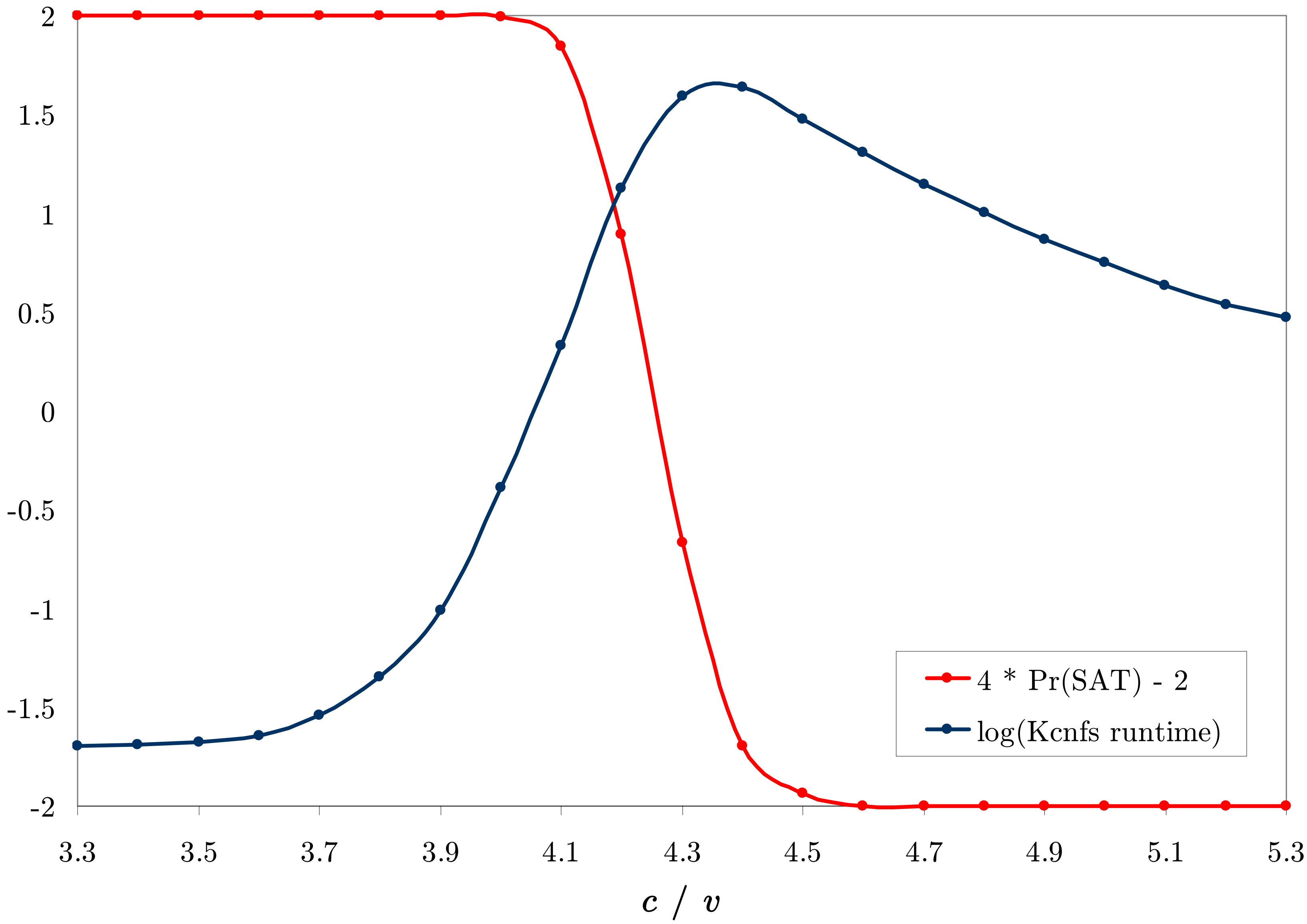
- **Experimental Results**
  - –Variable Size Data
  - –Fixed Size Data



Experimental Setup

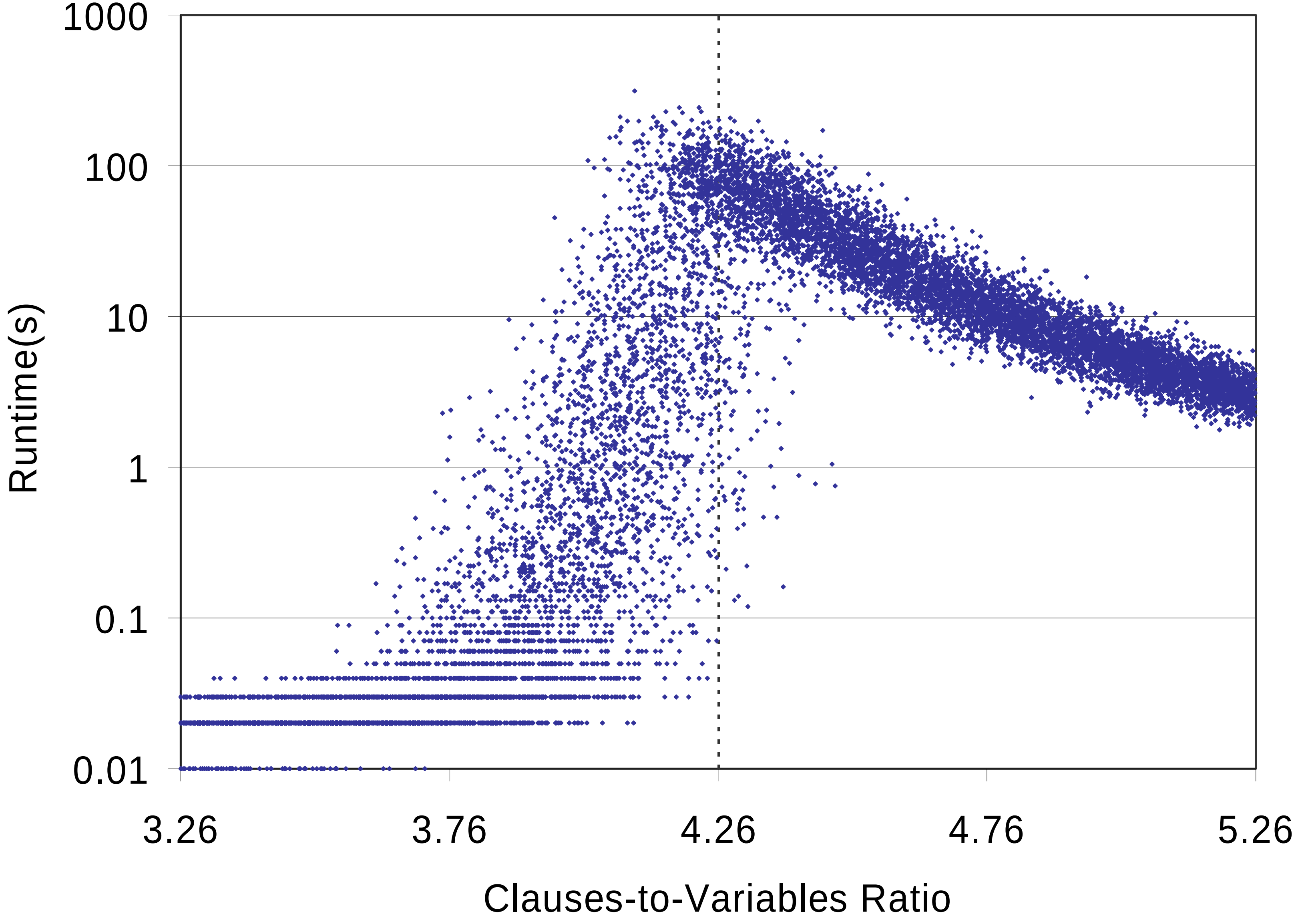
- Uniform random 3-SAT, 400 vars
- Datasets (20000 instances each)
  - Variable-ratio dataset (1 CPU-month)
    - c/v uniform in [3.26, 5.26] ($\therefore c \in$[1304,2104])
  - Fixed-ratio dataset (4 CPU-months)
    - c/v=4.26 ($\therefore v$=400, c=1704)
- Solvers
  - Kcnfs [Dubois and Dequen]
  - OKsolver [Kullmann]
  - Satz [Chu Min Li]
- Quadratic regression with logistic response function
- Training : test : validation split – 70 : 15 : 15

Kcnfs Data



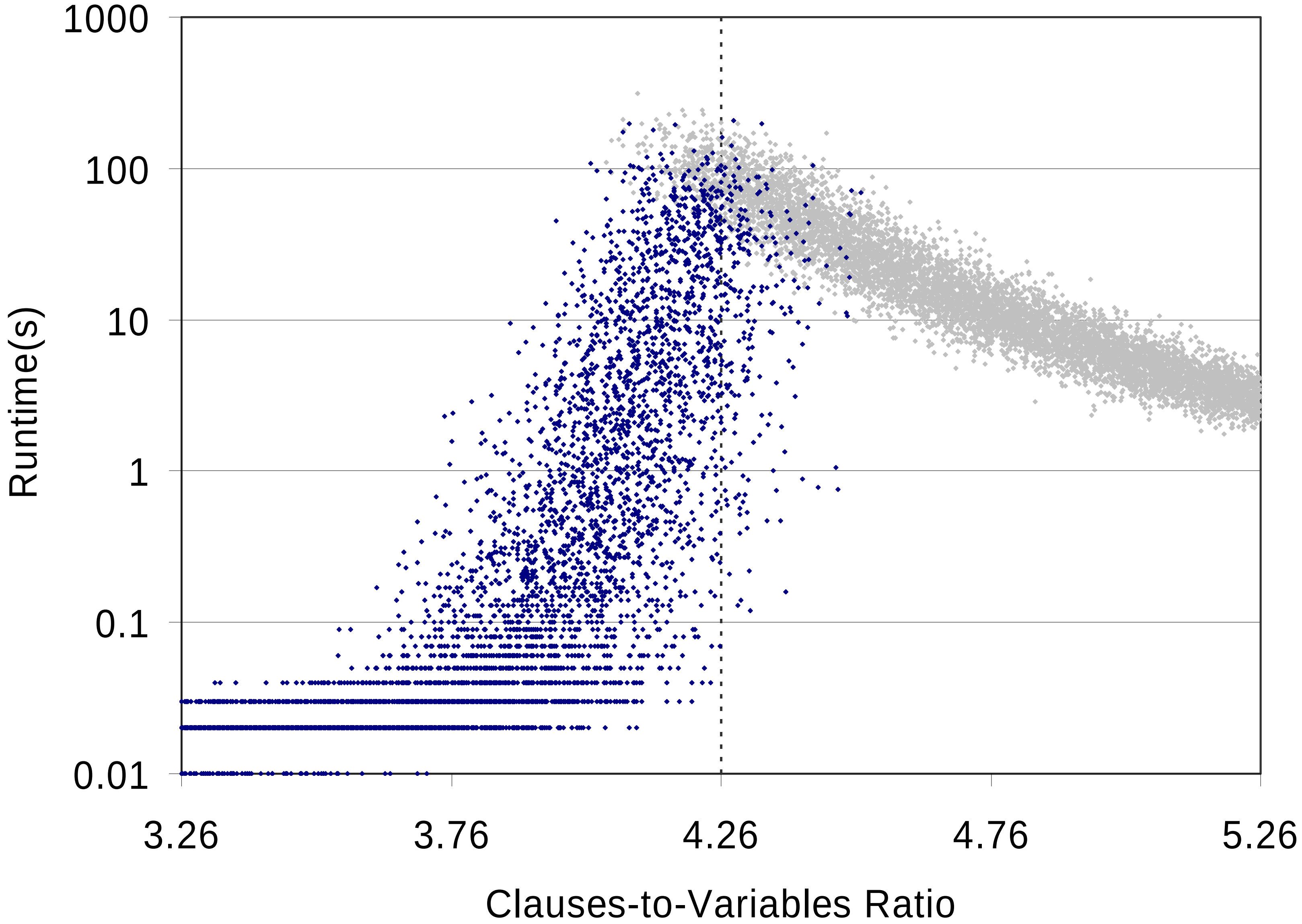
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
3.3
3.5
3.7
3.9
4.1
4.3
4.5
4.7
4.9
5.1
5.3
4 * Pr(SAT) - 2
log(Kcnfs runtime)
c / v

Kcnfs Data



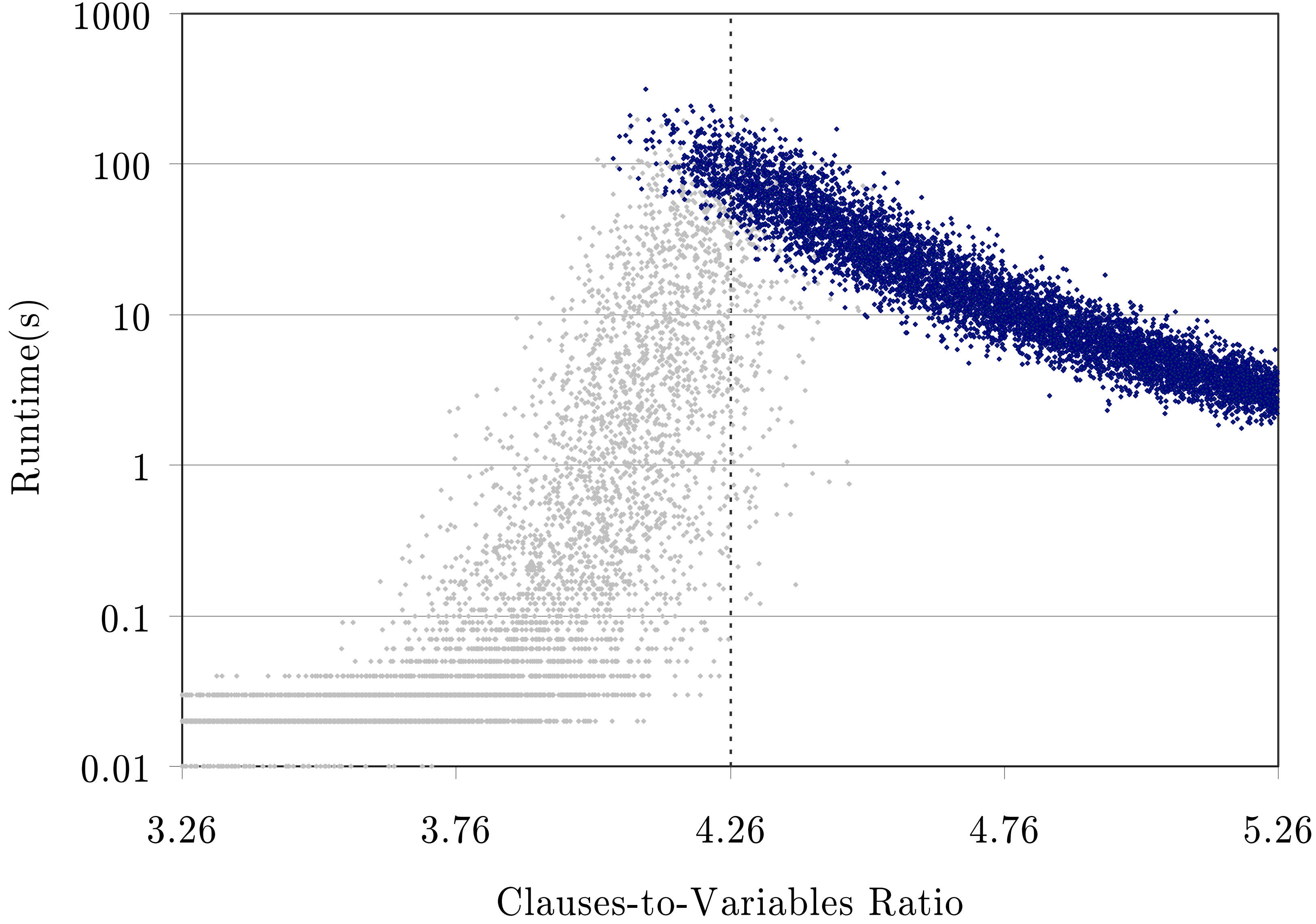
1000
100
10
1
0.1
0.01
Runtime(s)
3.26
3.76
4.26
4.76
5.26
Clauses-to-Variables Ratio

Kcnfs Data



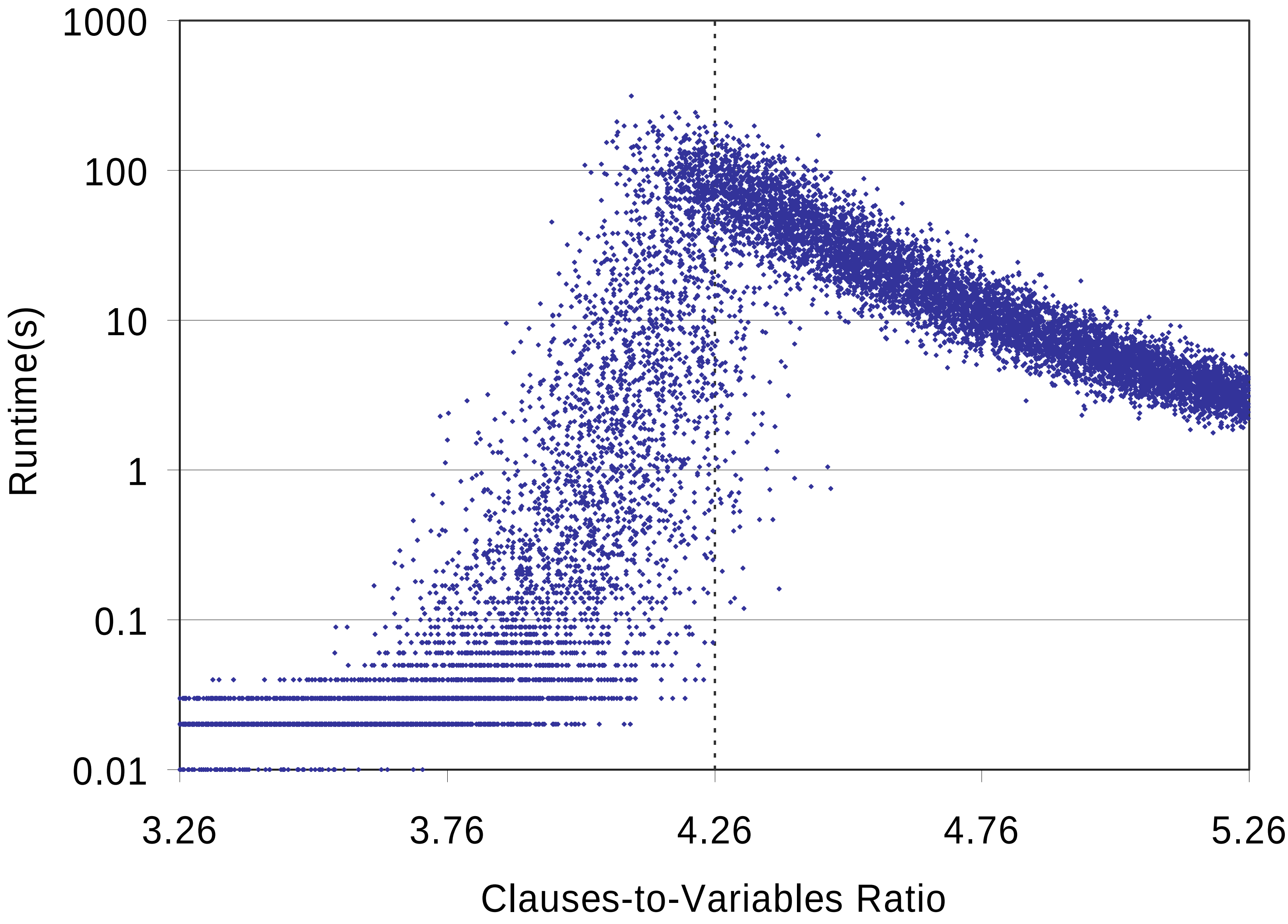
1000
100
10
1
0.1
0.01
Runtime(s)
3.26
3.76
4.26
4.76
5.26
Clauses-to-Variables Ratio

Kcnfs Data



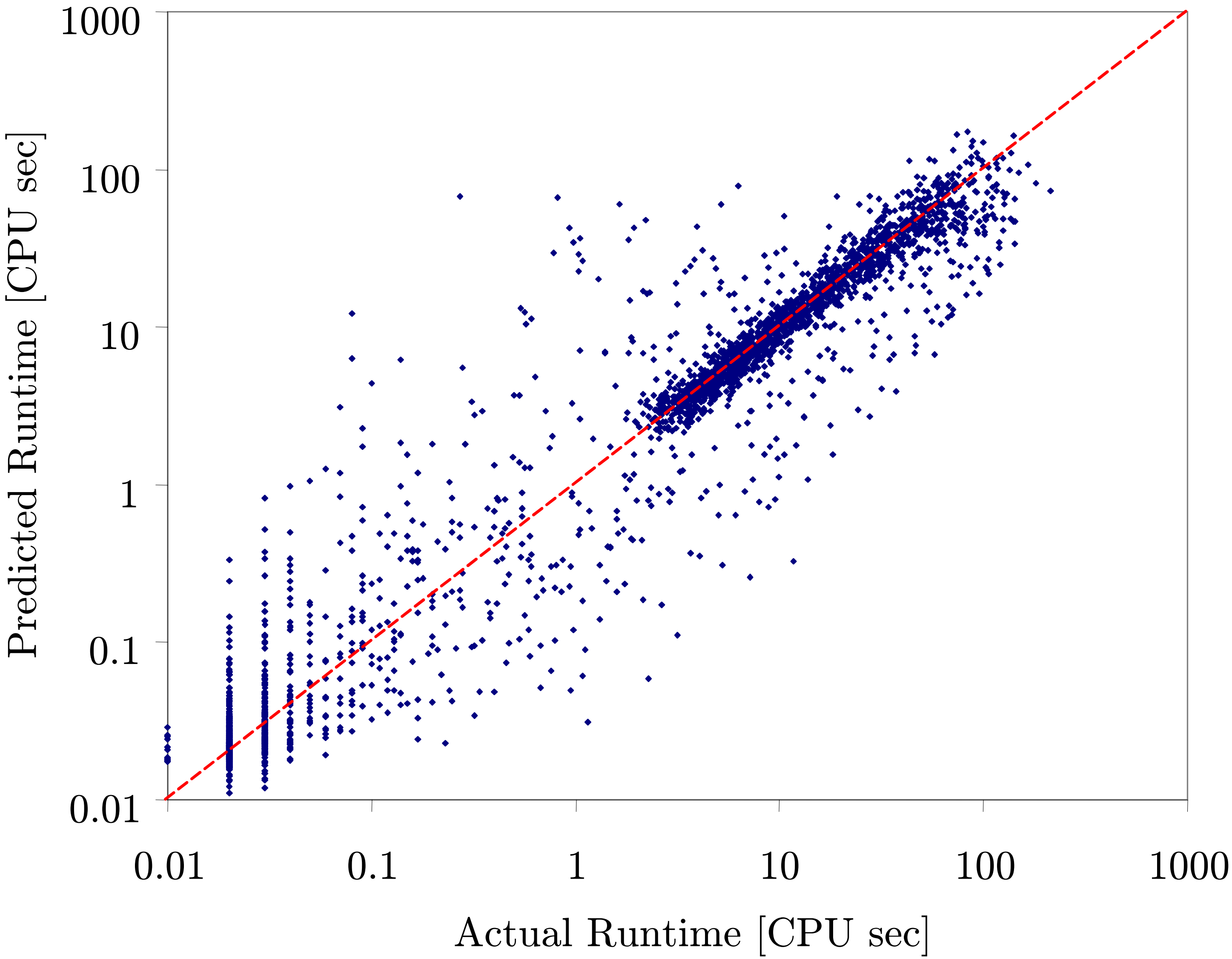
1000
100
10
1
0.1
0.01
Runtime(s)
3.26
3.76
4.26
4.76
5.26
Clauses-to-Variables Ratio

Kcnfs Data

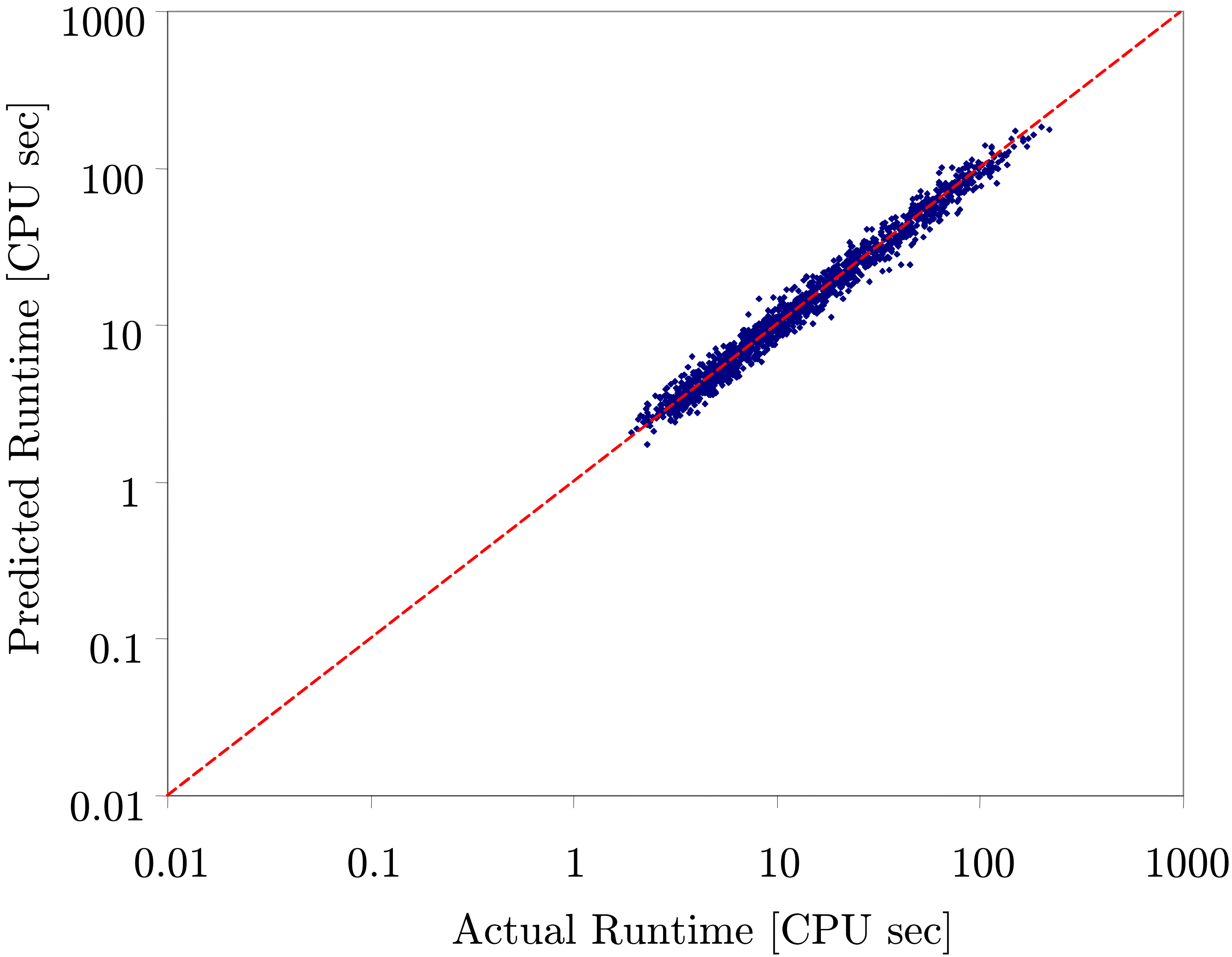


1000
100
10
1
0.1
0.01
Runtime(s)
3.26
3.76
4.26
4.76
5.26
Clauses-to-Variables Ratio

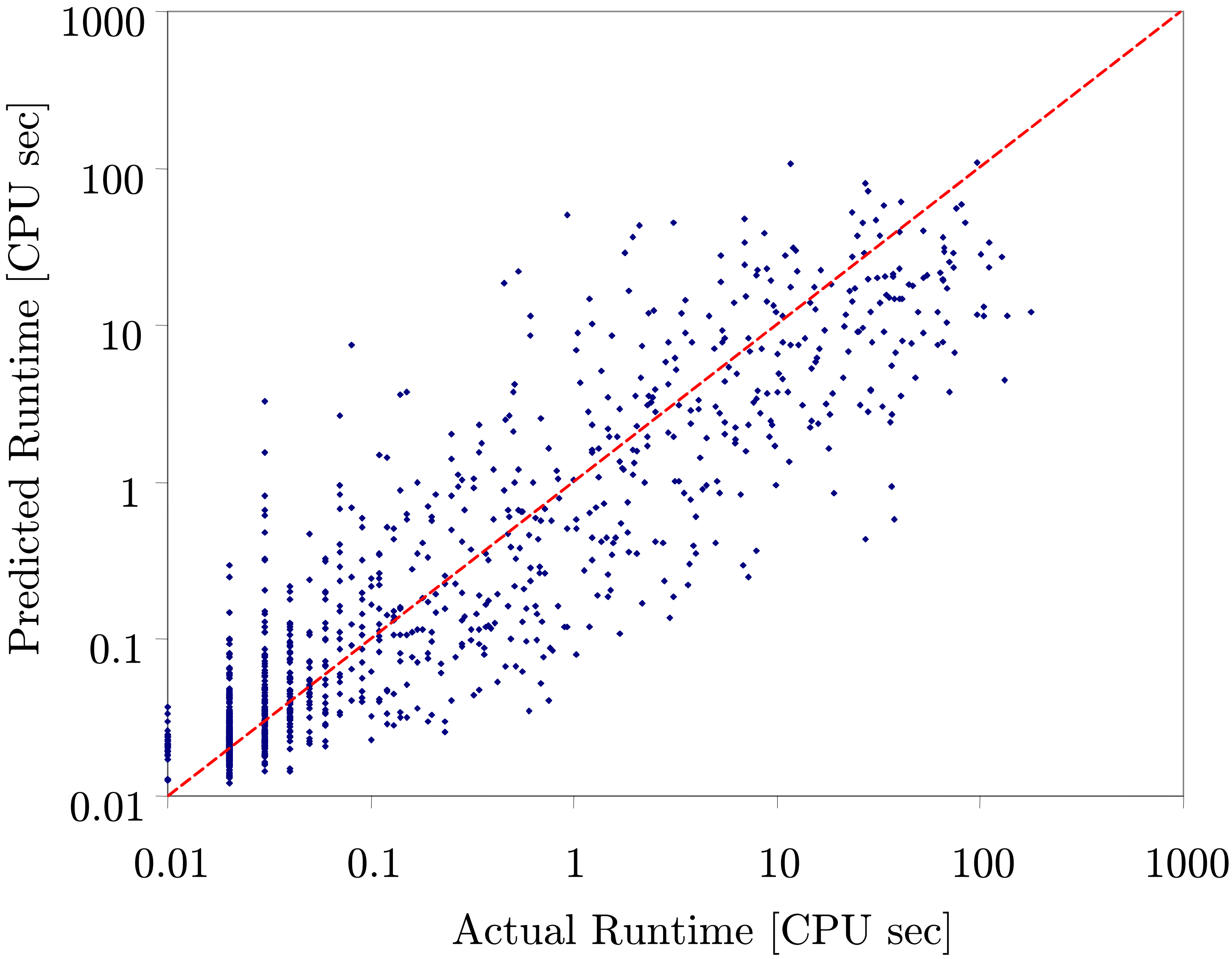
Variable Ratio Prediction (Kcnfs)



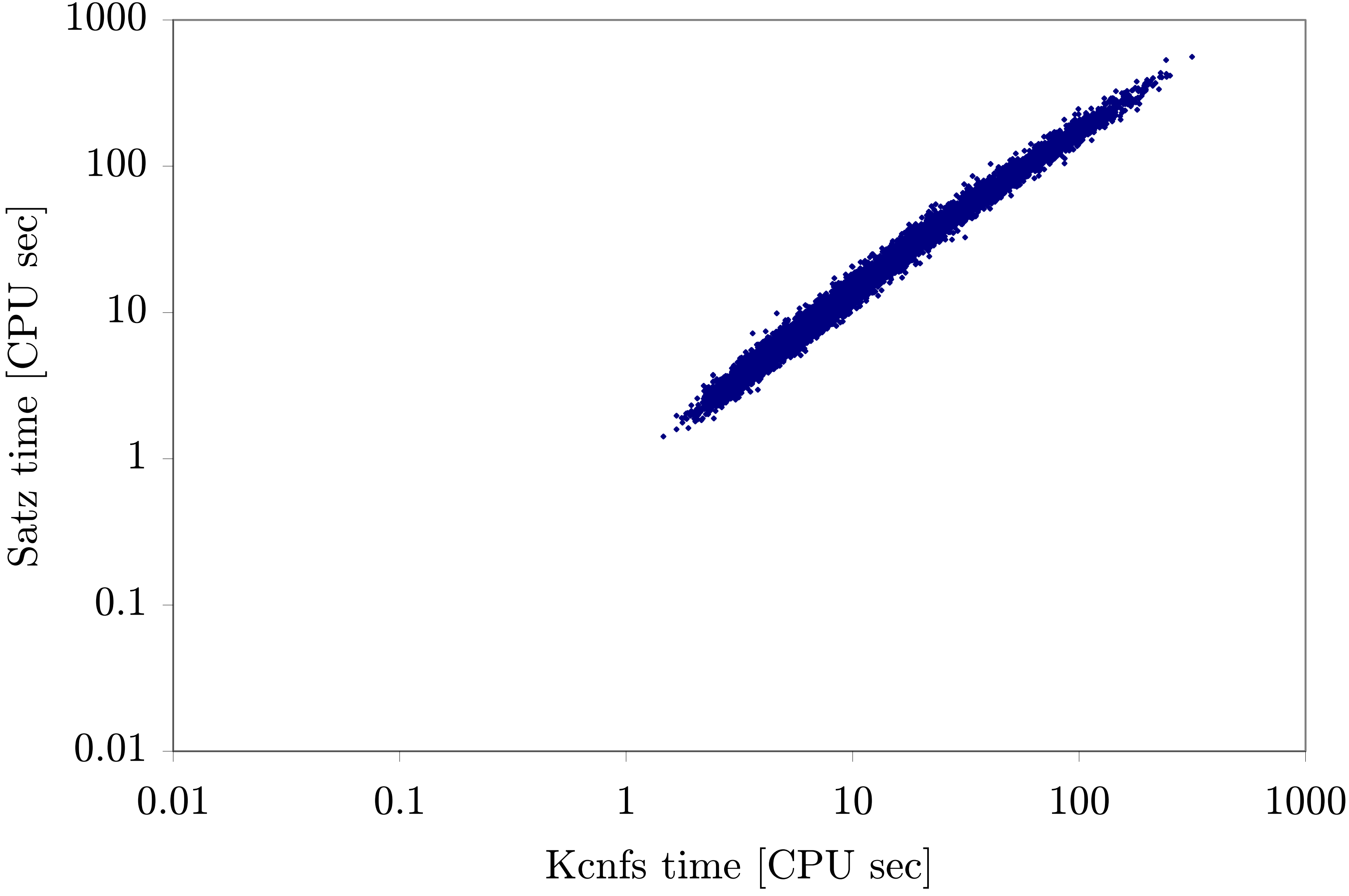
Variable Ratio - UNSAT



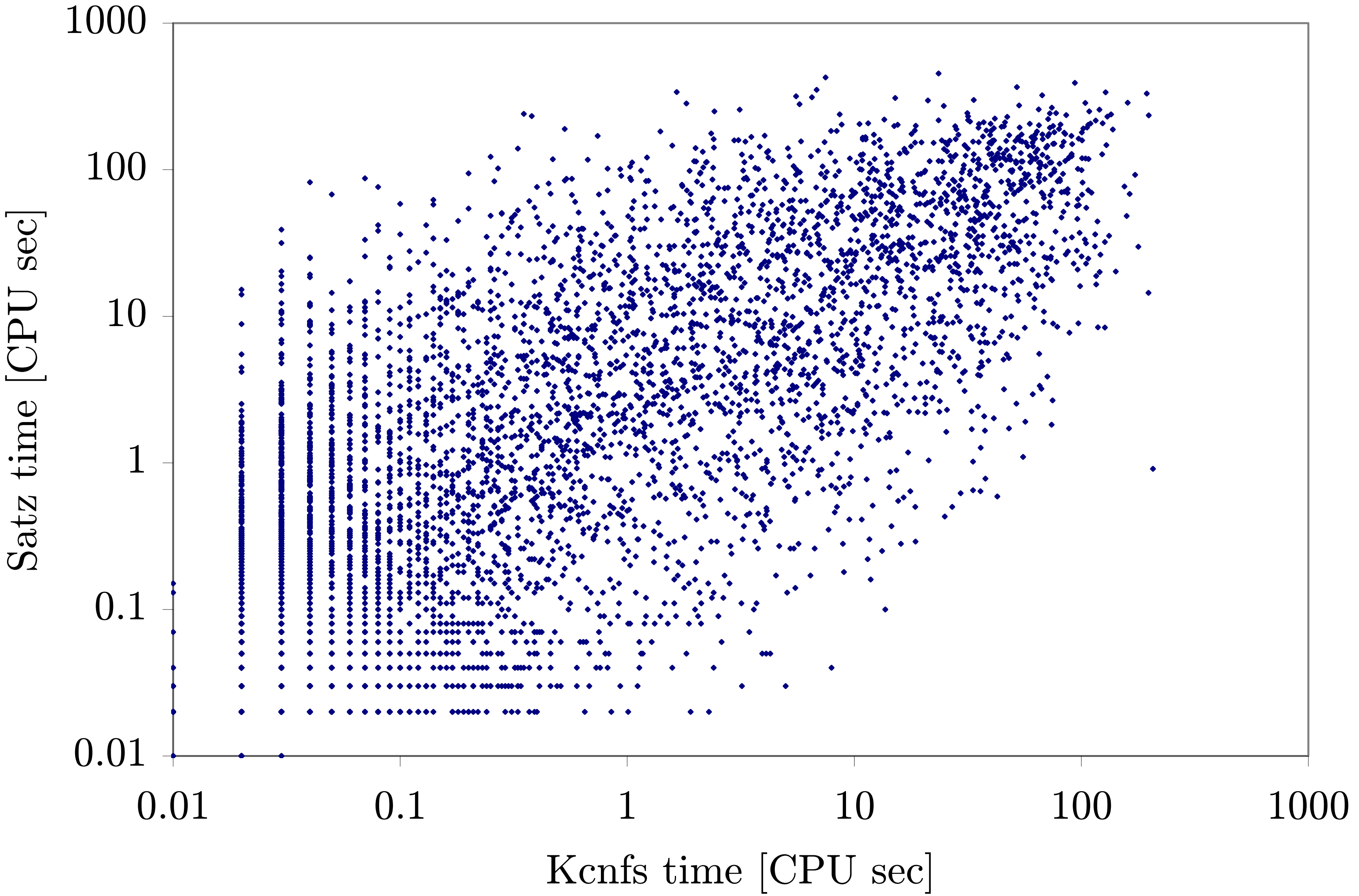
Variable Ratio - SAT



Kcnfs vs. Satz (UNSAT)



Kcnfs vs. Satz (SAT)



Feature Importance – Variable Ratio

- Subset selection can be used to identify features sufficient for approximating full model performance
- Other (correlated) sets could potentially achieve similar performance

Variable	Cost of Omission
$\lvert c/v\text{-}4.26\rvert$	100
$\lvert c/v\text{-}4.26\rvert^2$	69
$(v/c)^2 \times SapsBestCVMean$	53
$\lvert c/v\text{-}4.26\rvert \times SapsBestCVMean$	33

Feature Importance – Variable Ratio

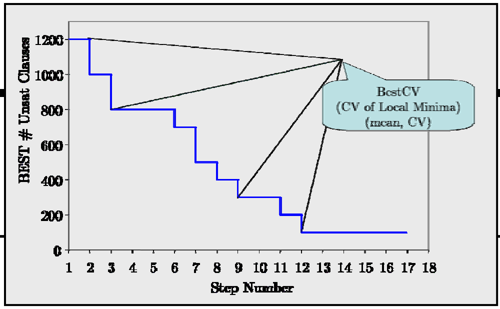
- Subset selection can be used to identify features sufficient for approximating full model performance
- Other (correlated) sets could potentially achieve similar performance

Variable	Cost of Omission
$\lvert c/v\text{-}4.26\rvert$	100
$\lvert c/v\text{-}4.26\rvert^2$	69
$(v/c)^2 \times SapsBestCVMean$	53
$\lvert c/v\text{-}4.26\rvert \times SapsBestCVMean$	33

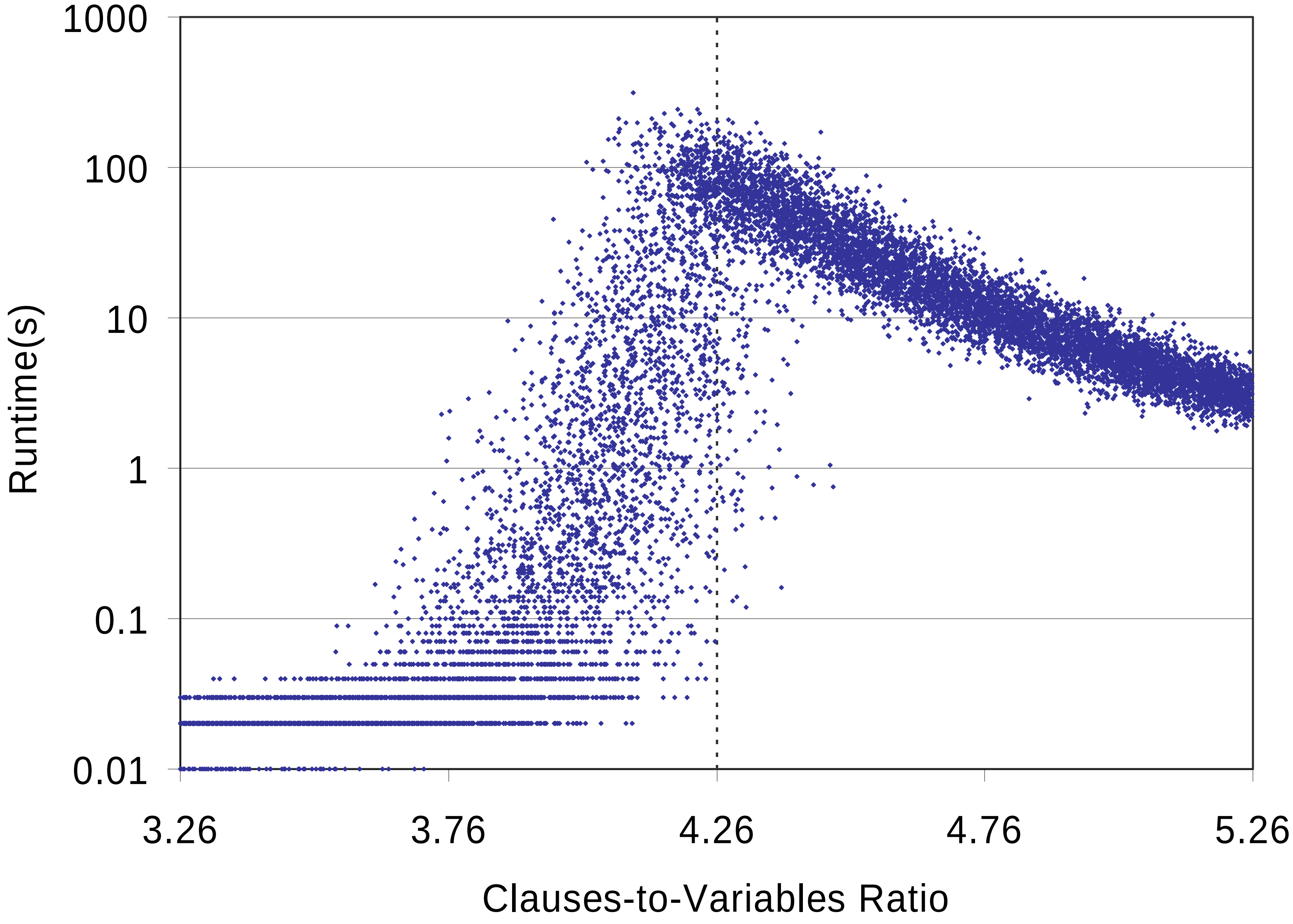
Feature Importance – Variable Ratio

- **Subset selection** can be used to identify features **sufficient** for approximating full model performance
- Other (correlated) sets could potentially achieve similar performance

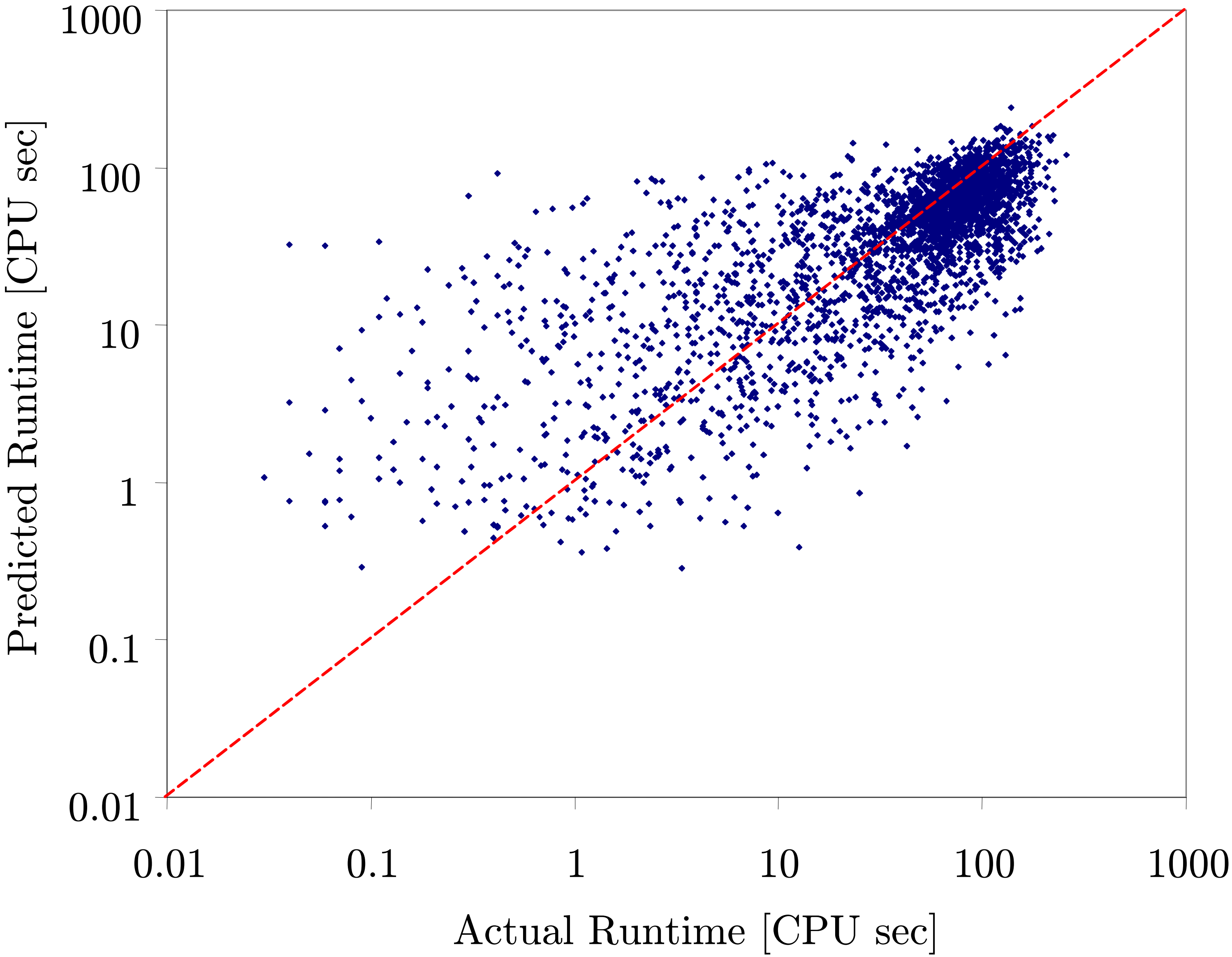
Variable	Cost of Omission
$\lvert c/v\text{-}4.26\rvert$	100
$\lvert c/v\text{-}4.26\rvert^2$	69
$(v/c)^2 \times$ *SapsBestCVMean*	53
$\lvert c/v\text{-}4.26\rvert \times$ *SapsBestCVMean*	33



Fixed Ratio Data



Fixed Ratio Prediction (Kcnfs)

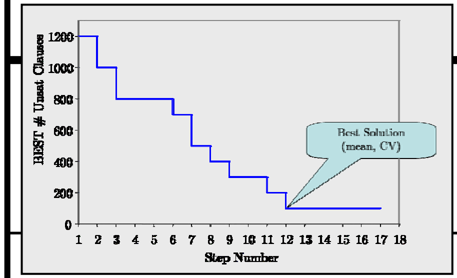


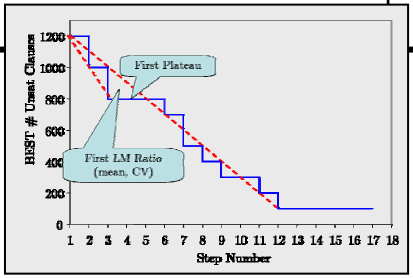
Feature Importance – Fixed Ratio

Variable	Cost of Omission
$SapsBestSolMean^2$	100
$SapsBestSolMean \times MeanDPLLDepth$	74
$GsatBestSolCV \times MeanDPLLDepth$	21
$VCGClauseMean \times GsatFirstLMRatioMean$	9

Feature Importance – Fixed Ratio

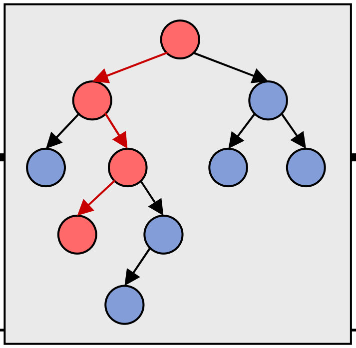
Variable	Cost of Omission
$SapsBestSolMean^2$	100
$SapsBestSolMean \times MeanDPLLDepth$	74
$GsatBestSolCV \times MeanDPLLDepth$	21
$VCGClauseMean \times GsatFirstLMRatioMean$	9

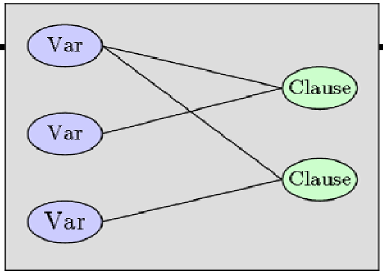




Feature Importance – Fixed Ratio

Variable	Cost of Omission
$SapsBestSolMean^2$	100
$SapsBestSolMean \times MeanDPLLDepth$	74
$GsatBestSolCV \times MeanDPLLDepth$	21
$VCGClauseMean \times GsatFirstLMRatioMean$	9



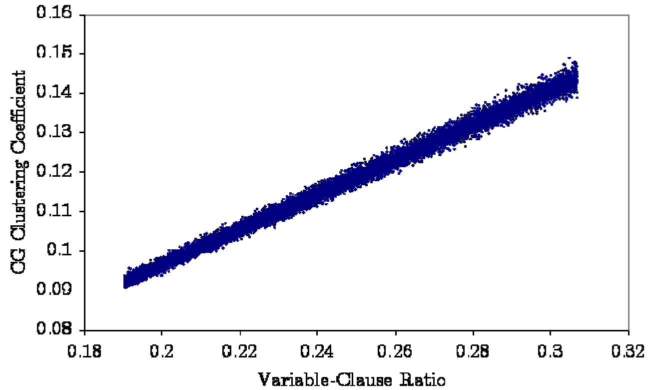


SAT vs. UNSAT

- Training models separately for SAT and UNSAT instances:
  - good models require fewer features
  - model accuracy improves
  - c/v no longer an important feature for VR data
  - Completely different features are useful for SAT than for UNSAT
- Feature importance on SAT instances:
  - Local Search features sufficient
    - 7 features for good VR model
    - 1 feature for good FR model (SAPSBestSolCV x SAPSAveImpMean)
  - If LS features omitted, LP + DPLL search space probing
- Feature importance on UNSAT instances:
  - DPLL search space probing
  - Clause graph features

Beyond Ratio: Weighted CG Clustering Coefficient

- Byproduct of our analysis: a very strong correlation between weighted CG clustering coefficient and v/c



- Clustering coefficient is a more fundamental concept than v/c, since it describes the structure of the constraints explicitly, not implicitly.
  - correlation between (unweighted) CC and hardness has been shown for other constraint problems (e.g., graph coloring, combinatorial auctions)
- We have a proof sketch of this correlation

Conclusions

- Can construct good models for DPLL solvers
- These models can be analyzed to gain understanding about what makes instances hard or easy for solvers
- Algorithm portfolios can be constructed (Satzilla)

- More specifically:
  - Strong relationship between LS and DPLL search spaces
  - Our approach automatically identified importance of c/v
  - SAT/UNSAT instances have very different performance characteristics; it helps to model them separately
  - Clustering Coefficient explains why c/v is important in terms of local properties of constraint graph