The Algorithm Selection Problem

for Solving Sudoku with Metaheuristics

Danielle Notice
Ahmed Kheiri
Nicos G. Pavlidis

Lancaster University

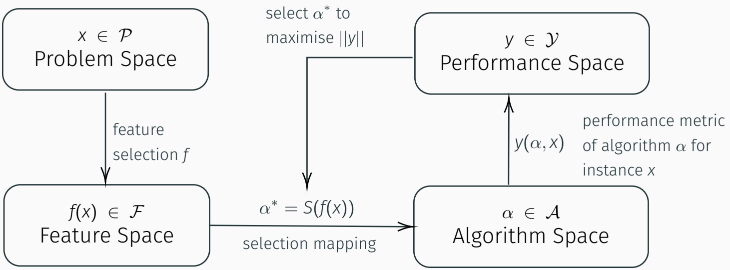
Introduction

- Which algorithm performs best on a particular problem instance?

- No Free Lunch Theorem [1] - there is no single algorithm that will be guaranteed to perform well across all instances.

- Relationship between instances and algorithm performance for an automated algorithm selection model.

The Algorithm Selection Problem

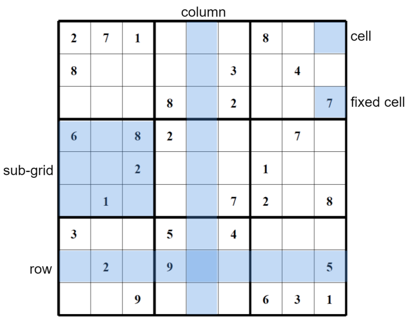


Instance space - problem instances, their features and algorithm performance in a shared space [2].

Sudoku Meta-data

Problem Description

Sudoku is a puzzle which consists of an $n^2 \times n^2$ grid divided into n^2 sub-grids each of size $n \times n$.



Objective: to fill each cell in a way that every row, column and sub-grid contains each integer between 1 and n^2 exactly once.

The problem space included 1000 instances from [3], all with $n = 3$.

Feature Space

A set of 54 features from the following groups:

- Puzzle mask
- Puzzle Rating Systems

Feature Space

A set of 54 features from the following groups:

- Puzzle mask
- Puzzle Rating Systems
- Graph Colouring Problem
- SAT Problem

Algorithm Space

We considered four local-search metaheuristic solvers:

- simulated annealing (SA)
- record-to-record travel (RR)
- reduced variable neighbourhood search (RVNS)
- steepest descent algorithm (SD)

Performance Space

- The cost function represents the number of values from one to nine that are not present in each row, each column and each sub-grid.
- A problem instance is **solved** when the cost is zero.
- 20 runs with fixed budget for each problem instance and algorithm.

Performance Space

We considered 2 performance metrics:

- Success Rate - the proportion of runs in which a solution with cost zero is found within the fixed budget.

- Mean cost-time - where for a given instance and run, **cost-time** is defined as:

$$c_{\text{best}} + \frac{i_{\text{best}}}{maxIts}$$

MATILDA

Melbourne Algorithm Test Instance Library with Data Analytics [2, 4]

MATILDA - Constructing the Instance Space

1. Preparation for Learning of Instance Meta-data (PRELIM)
   Pre-processes the meta-data by:
   - bounding and scaling the feature matrix;
   - calculating a binary performance metric.

MATILDA - Constructing the Instance Space

1. Preparation for Learning of Instance Meta-data (PRELIM)
   Pre-processes the meta-data by:
   - bounding and scaling the feature matrix;
   - calculating a binary performance metric.

2. Selection of Instance Features to Explain Difficulty (SIFTED)
   Identifies a subset of features which are most correlated with algorithm performance and are uncorrelated with each other.

MATILDA - Constructing the Instance Space

1. Preparation for Learning of Instance Meta-data (PRELIM)
   Pre-processes the meta-data by:
   - bounding and scaling the feature matrix;
   - calculating a binary performance metric.

2. Selection of Instance Features to Explain Difficulty (SIFTED)
   Identifies a subset of features which are most correlated with algorithm performance and are uncorrelated with each other.

3. Projecting Instances with Linearly Observable Trends (PILOT)
   Aims to find a lower-dimensional projection of the features which has a linear relationship with these features and the performance of each of the algorithms.

Performance Prediction

MATILDA trains a support vector machine (SVM) for each algorithm to predict the binary performance measure.

The Selector:

- If only one algorithm with good performance, it is selected as the best.

Performance Prediction

MATILDA trains a support vector machine (SVM) for each algorithm to predict the binary performance measure.

The Selector:

- If only one algorithm with good performance, it is selected as the best.
- If multiple algorithms, then the algorithm whose model has the highest precision is selected.

Performance Prediction

MATILDA trains a support vector machine (SVM) for each algorithm to predict the binary performance measure.

The Selector:

- If only one algorithm with good performance, it is selected as the best.
- If multiple algorithms, then the algorithm whose model has the highest precision is selected.
- If none of the algorithms, then the algorithm with the highest average performance is selected.

Results

Prediction Model Evaluation - Success Rates

Two **absolute thresholds** for good performance: SR $>$ 0 and SR $>$ 0.5.

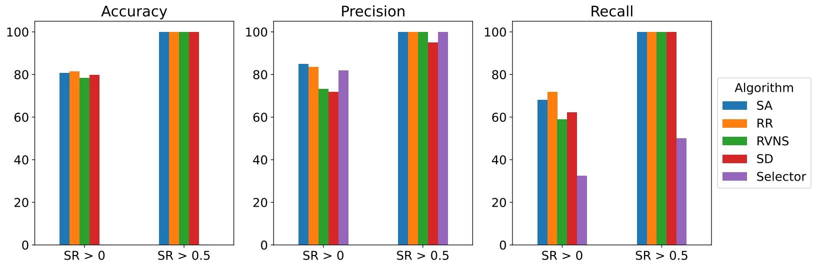


Figure 1: Average model evaluation metrics for SVMs and Selector

	SA	RR	RVNS	SD
Average SR	0.088	0.086	0.062	0.058
Pr(SR $>$ 0)	0.438	0.442	0.345	0.326
Pr(SR $>$ 0.5)	0.041	0.041	0.041	0.039

Table 1: Probability distribution of Success Rates

Instance Space Projections - Success Rates

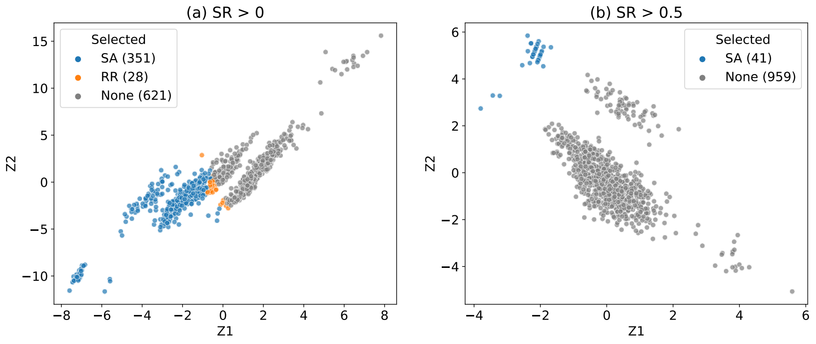


Figure 2: Projection of instance space showing the selected algorithm

	SA	RR	RVNS	SD
Average SR	0.088	0.086	0.062	0.058
$\Pr(SR > 0)$	0.438	0.442	0.345	0.326
$\Pr(SR > 0.5)$	0.041	0.041	0.041	0.039

Table 1: Probability distribution of Success Rates

Prediction Model Evaluation - Mean Cost-Time

Two **absolute thresholds** for good performance: CT < 2.5 and CT < 2.

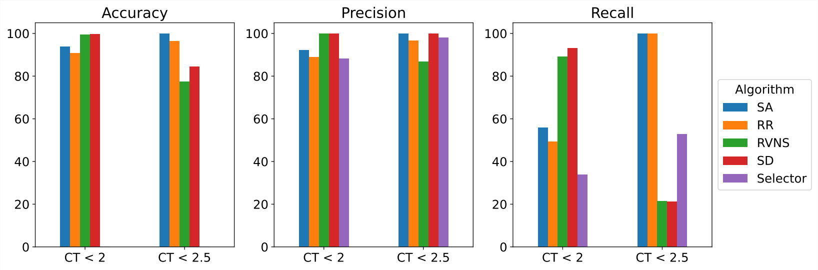


Figure 3: Average model evaluation metrics for SVMs and Selector

	SA	RR	RVNS	SD
Average CT	2.202	2.085	2.679	2.775
Pr(CT < 2.5)	0.873	0.962	0.275	0.197
Pr(CT < 2.0)	0.127	0.162	0.046	0.044

Table 3: Probability distribution of Mean Cost-Time

Instance Space Projections - Mean Cost-Time

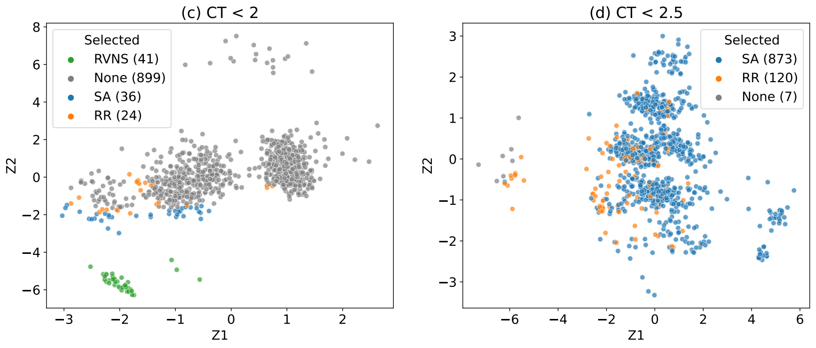


Figure 4: Projection of instance space showing the selected algorithm

	SA	RR	RVNS	SD
Average CT	2.202	2.085	2.679	2.775
Pr(CT $<$ 2.5)	0.873	0.962	0.275	0.197
Pr(CT $<$ 2.0)	0.127	0.162	0.046	0.044

Table 2: Probability distribution of Mean Cost-Time

Selected Features

Feature	Description	SR > 0	SR > 0.5	CT < 2	CT < 2.5
fixedDig_max	max number of times each value appears as a fixed cell				*
counts_CV	count of possible values each empty cell can take given fixed cells		*		
counts_min	minimum count as above	*		*	*
counts_naked1	number of empty cells that can take only 1 possible value given the fixed cells	*	*	*	*
counts_naked2	as above - 2 possible values	*		*	*
counts_naked3	as above - 3 possible values	*			
value_max	max number of empty cells that can take each value			*	*
value_mean	mean as above			*	*
value_min	minimum as above			*	*
GCP_avgPath	GCP - average length of the shortest paths for all possible vertex pairs	*			
GCP_clustcoef	GCP - average graph clustering coefficient	*			
GCP_density	GCP - density of the graph			*	*
GCP_nDeg_std	GCP - standard deviation of node degrees			*	
LP_fracInt	SAT - fraction of variables set to 0 or 1 in solution of LP relaxation	*	*	*	*
LPslack_CV	SAT - variable integer slack statistics of LP relaxation			*	*
LPslack_entropy	SAT - variable integer slack statistics of LP relaxation	*	*		
SAT_ratioLin	SAT - the linearised clause-to-variable ratio	*			
SAT_ratioRec	SAT - reciprocal of the clause-to-variable ratio	*	*		
VG_CV	SAT - node degree statistics for the variable graph		*		

Conclusion

- Informative Sudoku features
- Choice of performance metric
- Predicting “good” performance
- *To consider:*
  - More appropriate evaluation metrics
  - Higher order puzzles
  - Logic-based solvers

Thank You.

Any questions?

- D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- K. Smith-Miles and M. A. Muñoz, “Instance space analysis for algorithm testing: methodology and software tools,” *ACM Computing Surveys*, 2022.
- T. Dillon, “Tdoku: A fast Sudoku solver and generator,” 2019. **(Accessed: June 1, 2022).**
- M. Muñoz and K. Smith-Miles, “Instance space analysis: A toolkit for the assessment of algorithmic power,” 2021 2020.
- K. Smith-Miles and L. Lopes, “Measuring instance difficulty for combinatorial optimization problems,” *Computers & Operations Research*, vol. 39, no. 5, pp. 875–889, 2012.

J. R. Rice, "The algorithm selection problem," in *Advances in Computers* (M. Rubinoff and M. C. Yovits, eds.), vol. 15, pp. 65–118, Elsevier, 1976.

SIFTED Details

Selection of Instance Features to Explain Difficulty

- SIFTED first calculates the absolute value of Pearson's correlation coefficient between the features and algorithm performance.
- Select the feature most correlated to the performance metric for each algorithm and any other features moderately correlated to the performance of at least one algorithm.
- Apply k-means clustering to detect groups of similar features.
- Multiple subsets of k features are obtained by randomly selecting a single feature from each of the k clusters.
- For each such subset of k features SIFTED applies PCA to reduce the data to two dimensions.
- Each of the resulting datasets is used to train a Random Forest that predicts $\mathbf{Y}_{\mathbf{bin}}$. The optimal subset of features is the one which results in the lowest predictive error .

PILOT Details

Projecting Instances with Linearly Observable Trends

This algorithm aims to find a lower-dimensional projection of the features, $\mathsf{Z} = \mathsf{A_r F}$, such that Z has a linear relationship with the original features, and with the performance of the different algorithms. This is formulated as minimising the sum of squared approximation errors as:

$$
\begin{aligned}
\min_{\mathsf{A_r},\mathsf{B_r},\mathsf{C_r}} \quad & ||\mathsf{F} - \hat{\mathsf{F}}||_F^2 + ||\mathsf{Y} - \hat{\mathsf{Y}}||_F^2 && (1)\\
\text{s.t.} \quad & \mathsf{Z} = \mathsf{A_r F} && (2)\\
& \hat{\mathsf{F}} = \mathsf{B_r Z} && (3)\\
& \hat{\mathsf{Y}} = \mathsf{C_r Z}, && (4)
\end{aligned}
$$

where $\mathsf{A_r} \in \mathbb{R}^{2\times m}$, $\mathsf{B_r} \in \mathbb{R}^{m\times 2}$, $\mathsf{C_r} \in \mathbb{R}^{a\times 2}$.

Comparison to Logistic Regression

Multinomial logistic regression with l_1-penalty for feature selection.

- Five classes: one for each of the algorithms and **None** - if three or more algorithms tie for best performance.
- Model using success rate:
  - Similar features selected.
  - **Prediction**: **SA** for 338 instances, **RR** for 4 instance and **None** for 658 of the instances.
  - Relatively poor accuracy (51%) and precision (63%).
- Model using mean cost-time:
  - No non-zero coefficients (naïve model).
  - **Prediction**: **RR** for all instances.
  - Decent performance - accuracy (78%) and precision (83%). Better than the MATILDA selector.