



On the Effect of Training Data Selection in Automated Algorithm Selection CP2026

Erdem Kuş, Özgür Akgün, Nguyen Dang, Lars Kotthoff, Ian Miguel

University of St Andrews, School of Computer Science
ek232@st-andrews.ac.uk

No Free Lunch

Images and text

- Deep learning dominates, by a wide margin
- Scale and representation learning pay off

Winner: neural networks

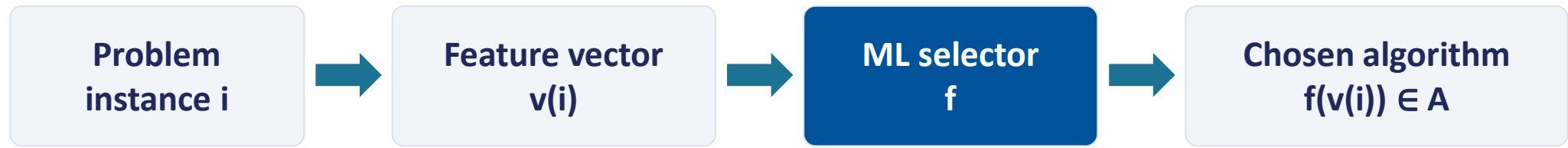
Tabular data

- Gradient-boosted trees still outperform deep learning on medium-sized tabular benchmarks
- Grinsztajn et al., NeurIPS 2022: 45 datasets, systematic study

Winner: tree ensembles

The same pattern in combinatorial solvers: on instances where one algorithm fails, another shines. Performance complementarity is the empirical basis of algorithm selection (Rice 1976).

The Algorithm Selection Problem



Goal: minimise expected solving runtime $\mathbb{E}_{i \sim \mathbb{D}_I} [r(i, f(v(i)))]$ over the instance distribution

VBS - Virtual Best Solver Oracle picking the best algorithm per instance; upper bound on any selector.

SBS - Single Best Solver: Best single algorithm on average; the baseline a selector must beat.

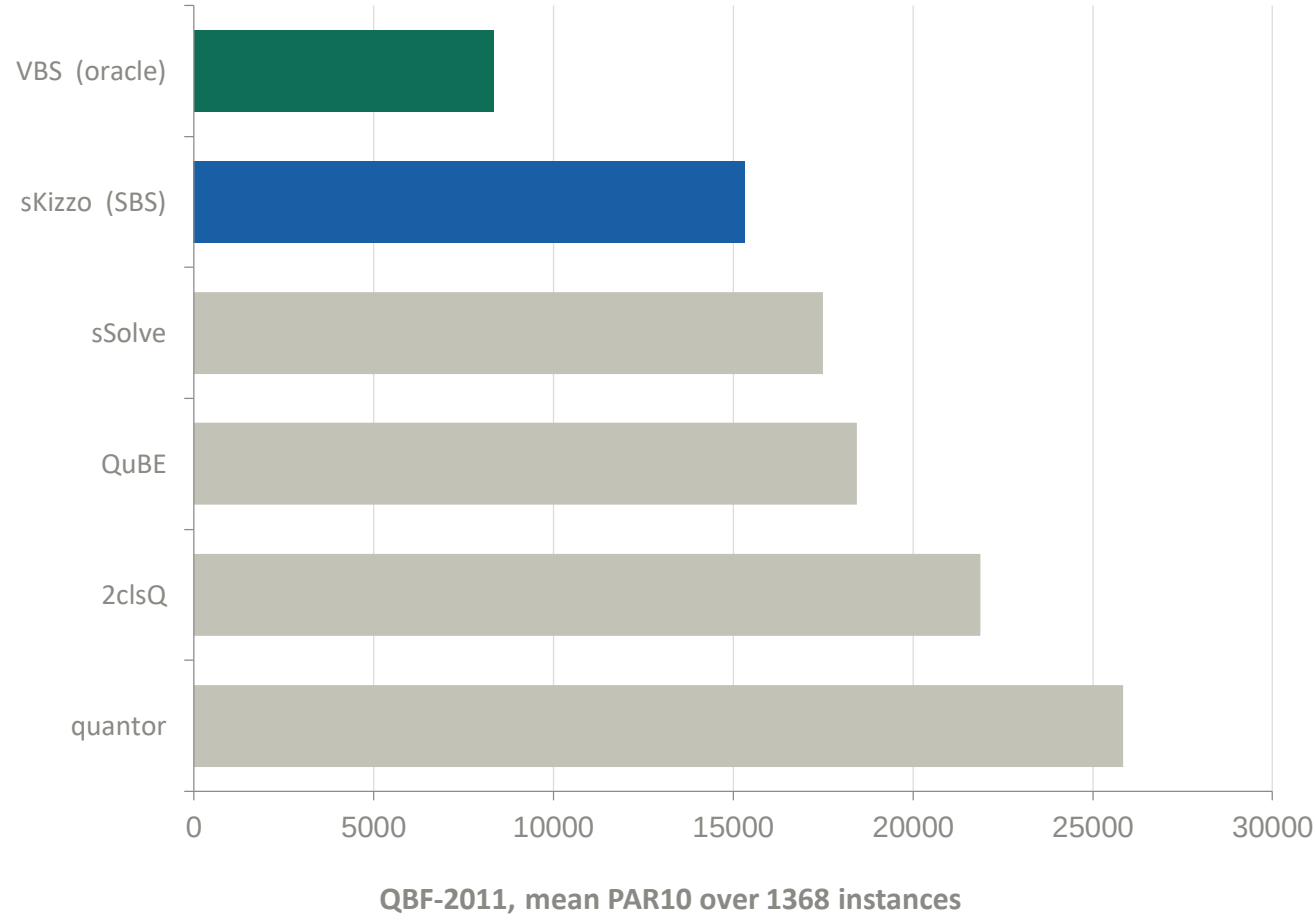
Algorithm Selection Problem

Instance	brareqs	hiqqr3	xbdepqbf	rareqs	sqube
complexity/query03	0.42	timeout	timeout	0.82	timeout
hardness/arbiter-07	timeout	24.97	129.75	timeout	timeout
testing/b17_PR_2_20	timeout	199.80	28.15	timeout	timeout
bomb/p10-1 len=17	307.70	timeout	timeout	0.98	timeout
eval2012r2/ethernet	timeout	timeout	110.70	timeout	24.82

Illustrative examples selected from ASLib QBF-2014

If we could pick the right algorithm per instance, we would get all the **green** cells.

Algorithm Selection Problem



15,330 s

SBS sKizzo
789 of 1368 solved

8,337 s

VBS oracle
1054 of 1368 solved

1.8x

the gap selection can work in

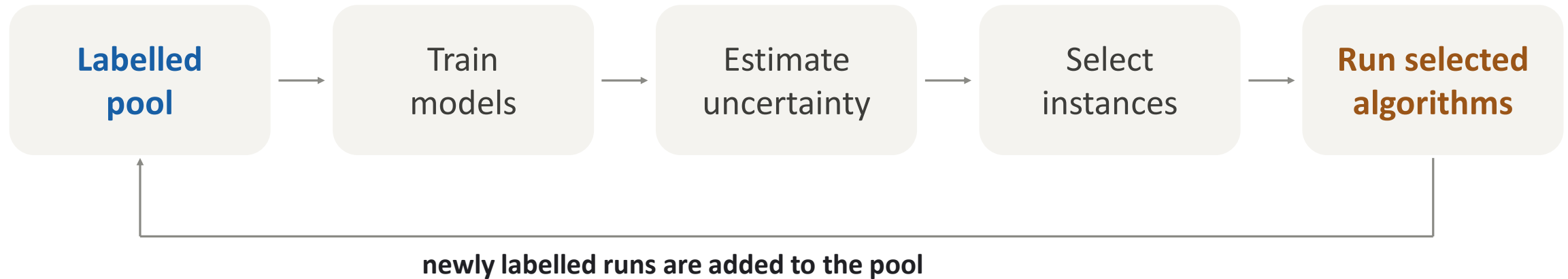
Labelling Cost is Bottleneck

Every algorithm must be run on every training instance to obtain the labels

Scenario	Instances	Algorithms	Runs needed	Labelling cost
CSP-2010	2,024	2	4,048	435 h
QBF-2014	1,254	14	17,556	6,510 h
SAT12-INDU	1,167	31	36,177	12,272 h
CSP-MZN-2013	4,642	11	51,062	26,263 h
All 16 scenarios			232,903	87,164 h

9.9 CPU years to produce the training data for these 16 scenarios alone

Frugal Algorithm Selection

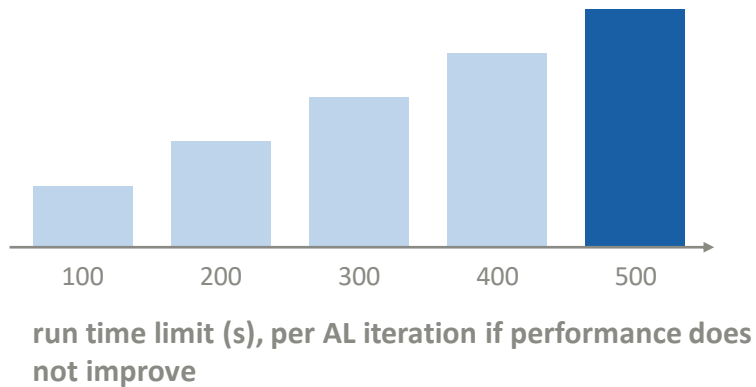


$$\arg \min_{(\mathcal{L} \subseteq \mathcal{I} \times \mathcal{A})} C(\mathcal{L}) \quad \text{s.t.} \quad \mathbb{E}_{\mathbf{i} \sim \mathbb{D}_{\mathcal{I}}} \left[r \left(i, f_{\mathcal{L}}(v(i)) \right) \right] \leq \mathbb{E}_{\mathbf{i} \sim \mathbb{D}_{\mathcal{I}}} \left[r \left(i, f_{\mathcal{I} \times \mathcal{A}}(v(i)) \right) \right]$$

Prior work (Frugal Algorithm Selection, CP 2024) cut labelling cost by around 40 % on average, up to 90 % in some scenarios (pairwise classification)

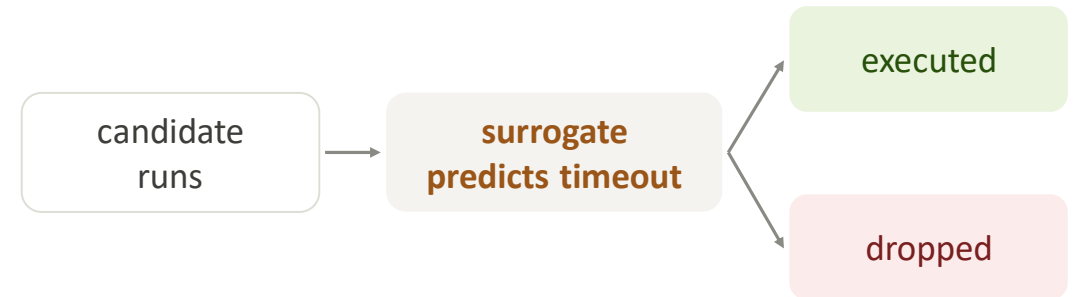
Two frugality mechanisms

DTL - Dynamic time limit



Runs are capped during AL. The limit starts at 100 s and grows by 100 s whenever validation performance stops improving.

TO - Timeout prediction



Runs predicted to time out are dropped before execution, so budget is not spent on uninformative runs.

DTL + TO both mechanisms applied together. All four settings are evaluated: none, DTL, TO, DTL + TO.

What is not yet studied

Frugal algorithm selection has only ever been studied with one learning formulation: pairwise classification.

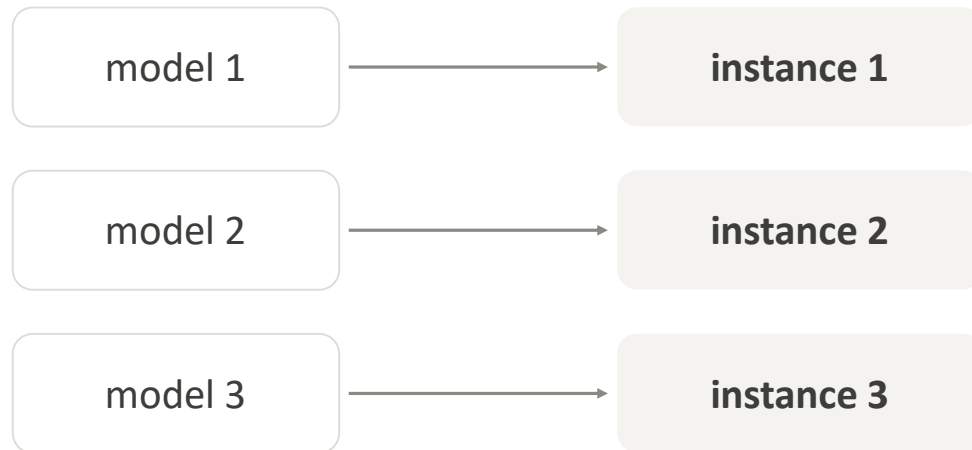
- 1 How do different AS learning formulations behave under a limited labelling budget?
- 2 Does the query granularity implied by a formulation affect frugality on its own?
- 3 Do the frugality mechanisms interact differently with different formulations?

Learning Formulations

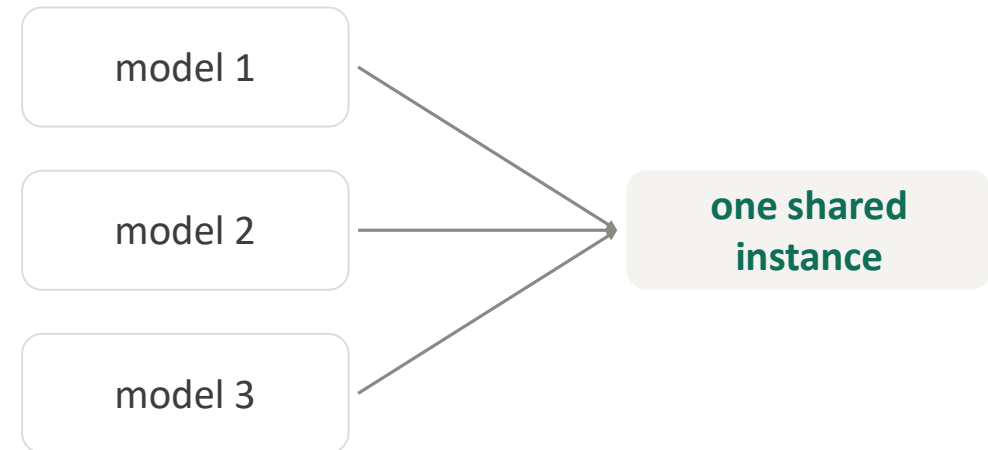
Formulation	What it predicts	ML models	Runs per data point
PC pairwise classification	which of two algorithms is better	$n(n-1)/2$	2
PR pairwise regression	the performance difference in a pair	$n(n-1)/2$	2
SR single regression	the cost of one algorithm	n	1
MC multiclass classification	the single best algorithm	1	n

Query Granularity

Fine-grained



Coarse-grained



- Each model independently selects its own most informative instance.
- Native to PC, PR and SR.

- All models agree on a single instance by mean uncertainty, and the whole portfolio is run on it.
- Native to MC, applicable to the others.

We evaluate every combination of formulation, granularity and frugality mechanism

Experimental Setup

4

learning formulations
MC, PC, PR, SR

2

query granularities
fine, coarse

4

mechanism settings
none, DTL, TO, DTL+TO

16

ASLib scenarios
ASP, BNSL, CP, SAT, QBF,
MaxSAT, CPMP

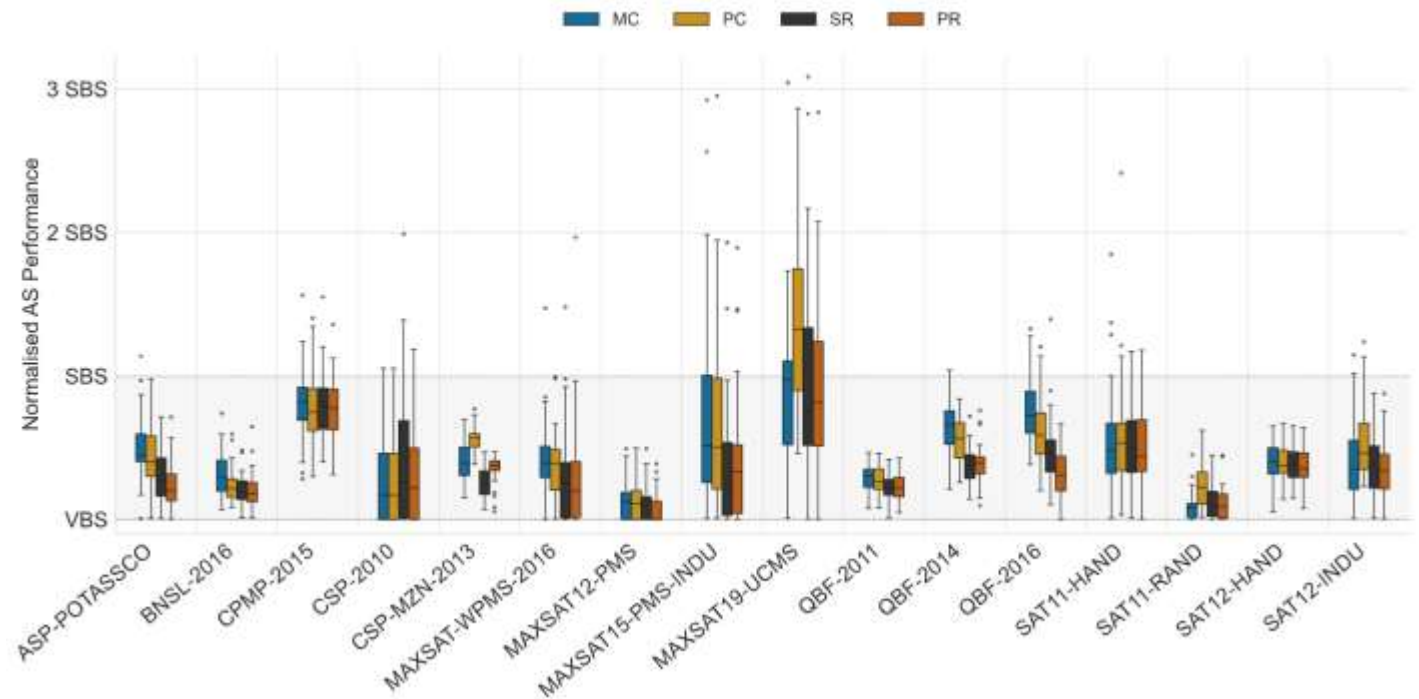
Every combination is evaluated

- Portfolios of 2 to 31 algorithms, 22 to 155 features, 296 to 4642 instances
- Random forests for all formulations, uncertainty taken from the ensemble
- 10-fold cross-validation, averaged over five random seeds
- AL starts from 20 random instances; each iteration labels 1 % of the remaining runs
- Friedman test with Nemenyi post-hoc, reported as CD diagrams

Code and data: doi.org/10.5281/zenodo.19978961

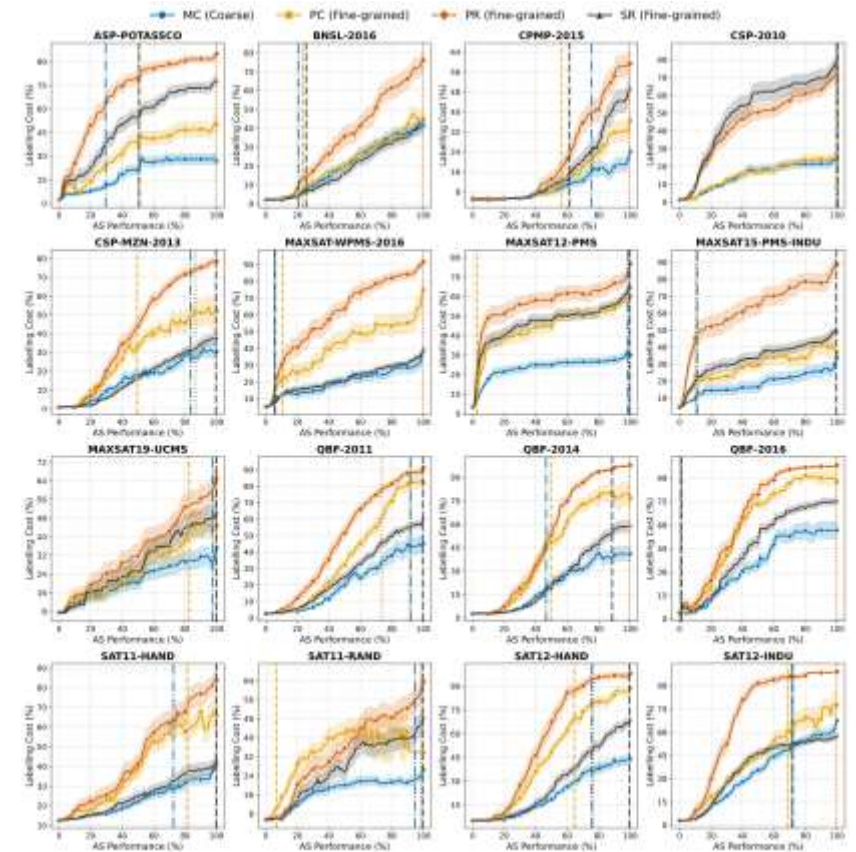
Under Passive Learning

- Lower is better 0=VBS and 1=SBS
- Regression-based formulations perform best. MC has the weakest median and the highest variance.



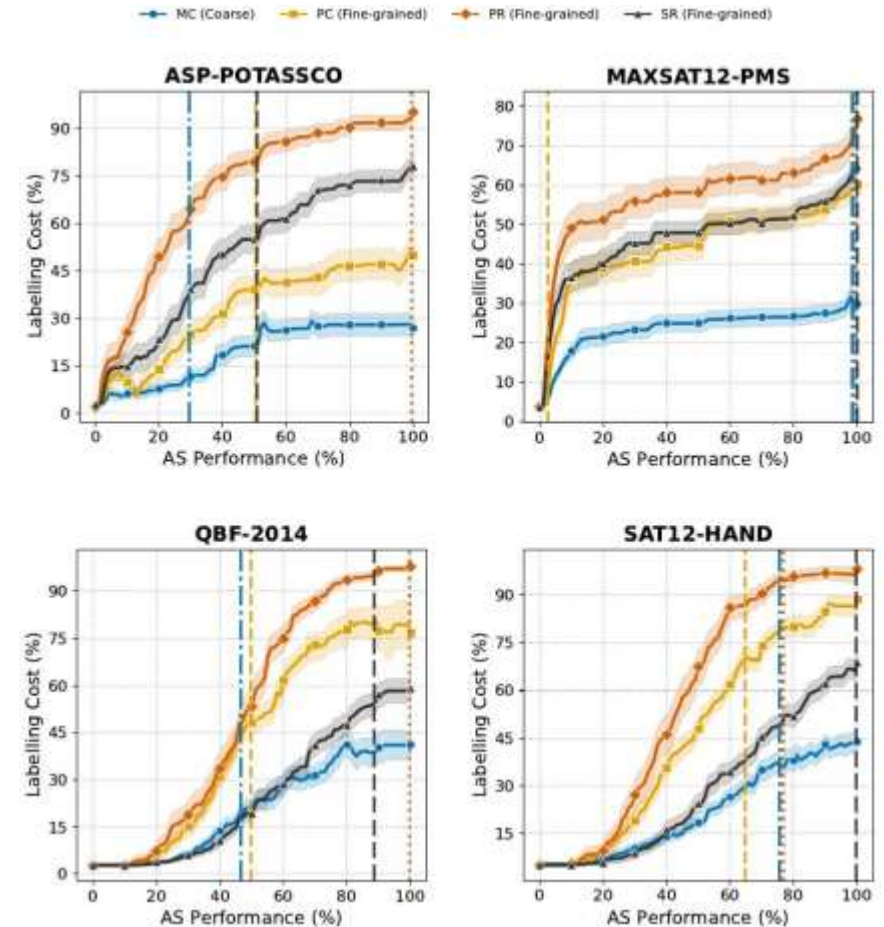
Under Active Learning on Default Granularity

- x- axis: AS Performance
- y-axis: Labelling Cost
- Vertical lines: Passive Learning performance (100% best)
- **MC** reaches the shared reference at a substantially lower labelling cost than every other formulation.

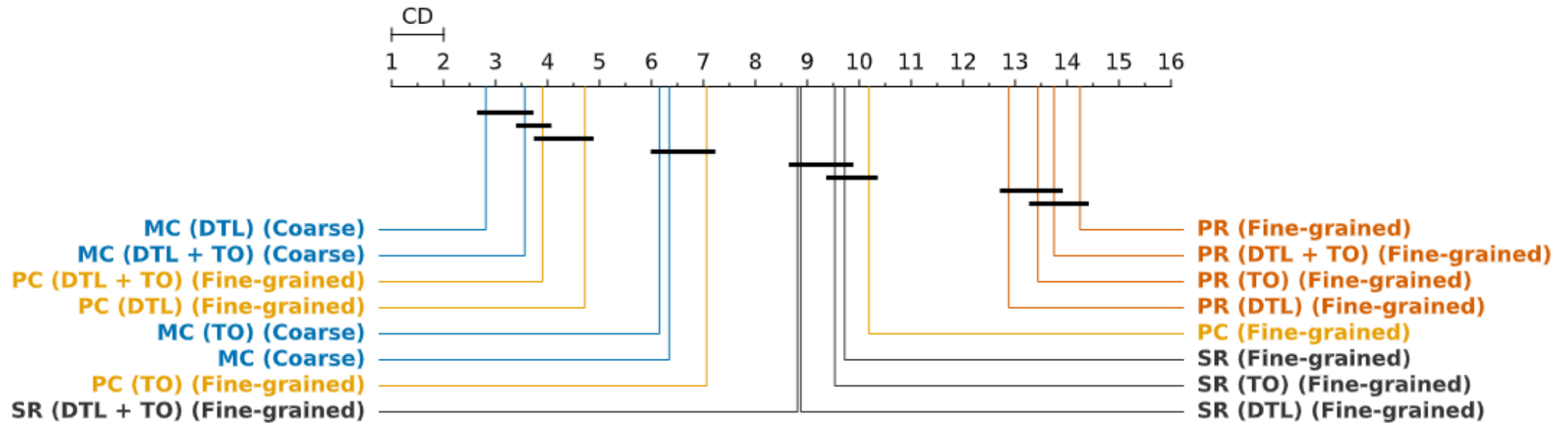


Under Active Learning on Default Granularity

- x- axis: AS Performance
- y-axis: Labelling Cost
- Vertical lines: Passive Learning performance (100% best)
- **MC** reaches the shared reference at a substantially lower labelling cost than every other formulation.



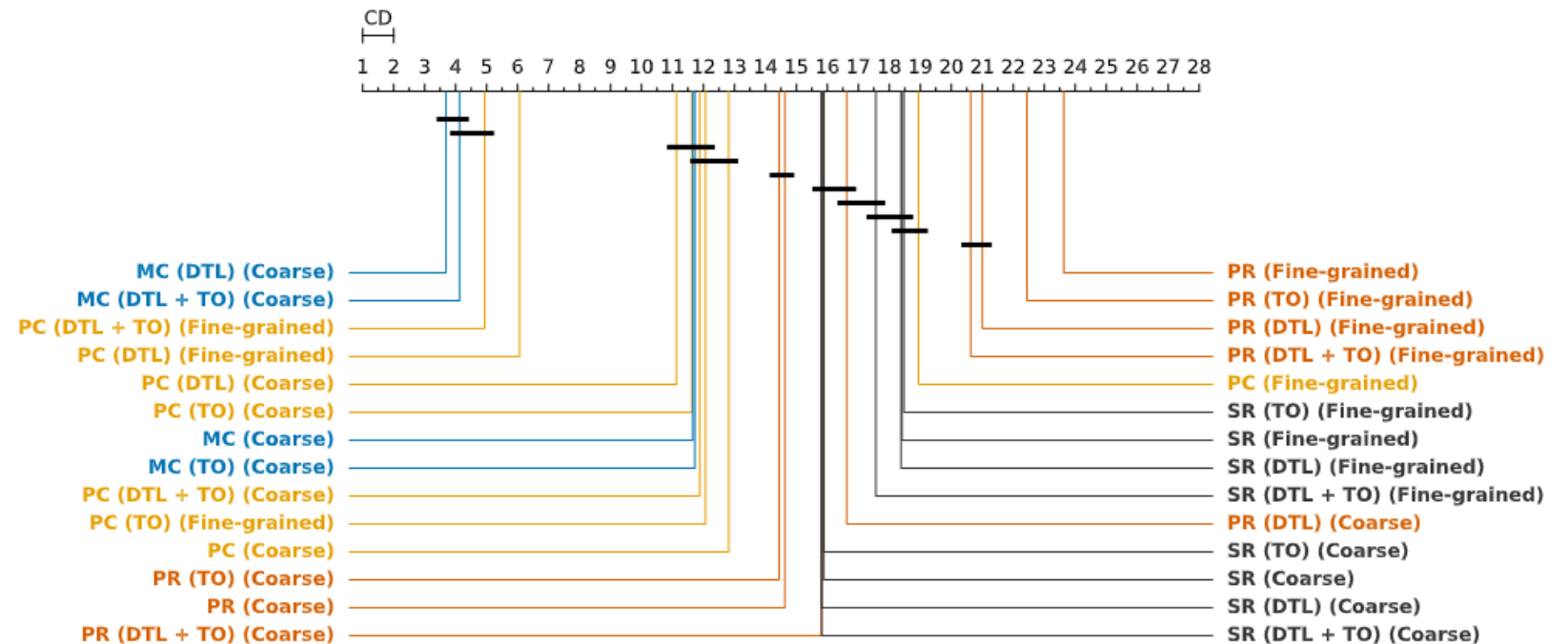
Frugality Ranking on Default Granularity



MC (DTL) is the most frugal formulation overall. Regression-based formulations rank lowest regardless of mechanism.

Granularity Effect

- Coarse-grained variants generally rank higher
- For PC, the gain vanishes with frugality mechanisms
- Regression still ranks lowest overall, so the formulation itself remains decisive



Findings

Weak passive, strong active MC has the weakest passive learning performance, yet is the most frugal formulation under active learning

Granularity matters: Coarse-grained querying improves frugality for every formulation, independently of the learner.

Practical recommendation: Use MC with a DTL when labelling budget is the constraint. When labels are already available, MC may not be feasible choice; its advantage is specific to the budget-constrained setting.

Paper: doi.org/10.4230/LIPIcs.CP.2026.38

Code and data: doi.org/10.5281/zenodo.19978961

We thank the University of St Andrews for computational support via its high-performance computing cluster Hypatia.

Q&A

