# Instance Space Analysis and Complexity Estimation for Scheduling Problems


**Isabel Catalá**
Universitat Politècnica de València, Spain

**Christian Pérez**
Universitat Politècnica de València, Spain

**Unai López**
Universitat Politècnica de València, Spain

**Miguel A. Salido**
Universitat Politècnica de València, Spain



## Abstract

Job-Shop Scheduling Problem (JSP) instances with identical size parameters often exhibit radically different empirical hardness. This extended abstract summarises a supervised framework for *solver-aligned complexity estimation* that predicts instance difficulty from static structural features. The method constructs a normalised hardness target from multi-solver traces, learns a bounded difficulty score, and induces balanced hardness categories. The resulting estimator aligns with solver effort, provides interpretable explanations, and exhibits partial transfer to classical benchmarks. The approach supports difficulty-aware benchmarking and solver-behaviour analysis.



**2012 ACM Subject Classification** Theory of computation → Constraint and logic programming; Mathematics of computing → Discrete optimization; Computing methodologies → Machine learning

**Keywords and phrases** Job-shop Scheduling, Instance Space Analysis, Solver Hardness, Multi-solver supervision, Supervised Learning

**Digital Object Identifier** 10.4230/LIPIcs.CVIT.2016.23


## 1 Introduction

Constraint Programming (CP) solvers exhibit substantial performance variability across instances that share the same modelling template. In JSP, two instances of identical size may require radically different search efforts due to structural properties that influence propagation strength, branching behaviour, and proof complexity. Traditional benchmarking practices that rely on nominal size, therefore, fail to capture the true landscape of solver difficulty. Existing approaches such as Instance Space Analysis (ISA) reveal structural patterns but do not provide a direct, solver-aligned hardness estimate.

This work addresses the gap by constructing a supervised estimator of instance hardness from static instance descriptors that is *aligned with solver behaviour*. Rather than predicting runtime directly, the method learns a bounded difficulty score derived from normalised multi-solver traces. The goal is an interpretable, solver-agnostic model that generalises across distributions and supports benchmark design and solver analysis.

## 2 Instance Representation and Feature Construction

Each JSP instance is represented through its disjunctive-graph formulation, where nodes correspond to operations and edges encode precedence and machine constraints. From this representation and from ISA-inspired summaries, a set of static features is extracted, including:



42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- degree statistics,
- machine-load imbalance,
- node mediation in precedence paths,
- higher-order conflict structures.

To capture heterogeneity, each raw feature vector is transformed using a suite of aggregation operators (minimum, maximum, mean, median, quartiles, range, and Gini dispersion). This produces a compact yet expressive feature space suitable for supervised modelling.

## 3 Solver-Aligned Hardness Target

Instead of relying on a single solver or raw runtime, hardness is defined through a normalised, multi-solver performance trace. For each instance, several CP solvers are executed under fixed configurations, and their search statistics (failures, propagations, runtime) are normalised and aggregated into a bounded score, $\mathcal{P}(x)$, where higher values indicate greater empirical difficulty.

This construction ensures:

- robustness to solver-specific artefacts,
- comparability across instances and solvers,
- interpretability of the resulting categories.

Balanced thresholds on $\mathcal{P}(x)$ induce three difficulty classes, `easy`/`medium`/`hard`, used for evaluation and subsequent analysis.

## 4 Learning and Evaluation

A Random Forest regressor is trained to predict $\mathcal{P}(x)$ from the static feature set. The model is chosen for its robustness, interpretability.

Experiments on a large corpus of synthetic JSP instances show that:

- the predicted score strongly correlates with solver effort indicators,
- SHAP explanations highlight meaningful structural drivers of hardness,
- the learned estimator exhibits partial ordinal transfer to classical JSPLIB instances.

These results demonstrate that solver-aligned hardness estimation is feasible, interpretable, and practically useful for benchmarking and instance selection.

## 5 Conclusion

This extended abstract summarises a framework for predicting solver-aligned hardness of JSP instances using static structural features and supervised learning. The approach provides a principled alternative to size-based benchmarking, enabling difficulty-aware evaluation, solver analysis, and dataset construction.