# AUTOMATIC ALGORITHM SELECTION FOR PROJECT SCHEDULING

TOMMY MESSELIS
PATRICK DE CAUSMAECKER
KU LEUVEN

PPSN 2012 – SEPTEMBER 1

# OVERVIEW

ALGORITHM SELECTION

- why?

PROJECT SCHEDULING

- what?
- state-of-the-art?

PERFORMANCE PREDICTION

- accurate?

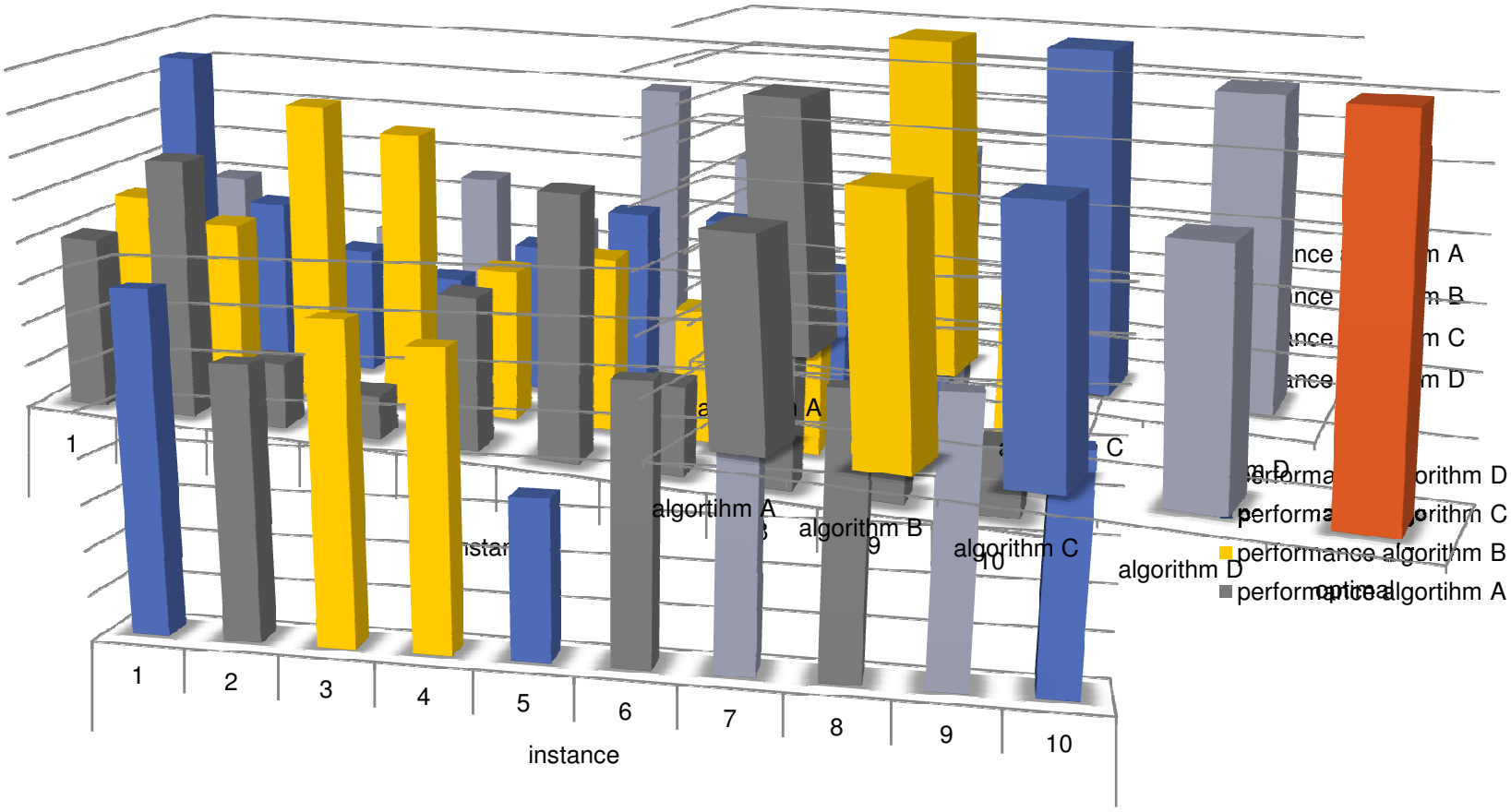
AUTOMATIC ALGORITHM SELECTION FOR PROJECT SCHEDULING

- better?

FUTURE WORK

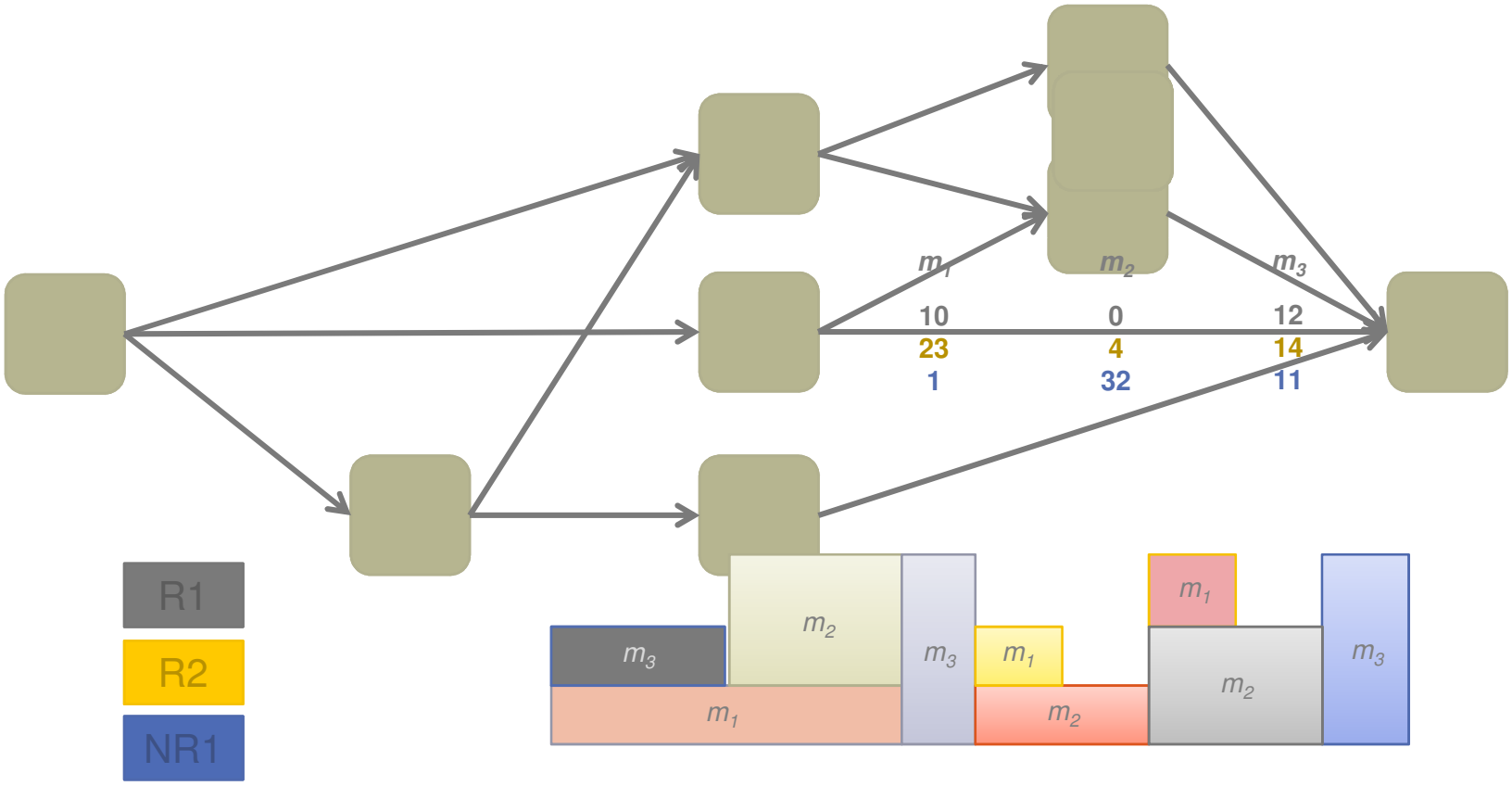
- why?

# ALGORITHM SELECTION



# PROJECT SCHEDULING

**multi-mode resource-constrained project scheduling**



# STATE OF THE ART

**Many different solution methods exist, mainly (hybrid) metaheuristics**

**Evaluation on PSPlib benchmark instances**

Kolish R., Sprecher, A., 1996. PSPlib – a project scheduling problem library. *European Journal of Operational Research.*

**> differences are only marginal...**

**Larger and more diverse instances: MMlib**

Van Peteghem V., 2010. Multi-mode resource-constrained project scheduling problem: Solution procedures and extensions. *PhD thesis, University of Ghent.*

**> differences are much larger**

# STATE OF THE ART

**hybrid genetic algorithm** (`algorithm A`)

**> based on Lova et al. (2009)**

Lova, A., Tormos, P., Cervantes, M., Barber, F., 2009. An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics.*

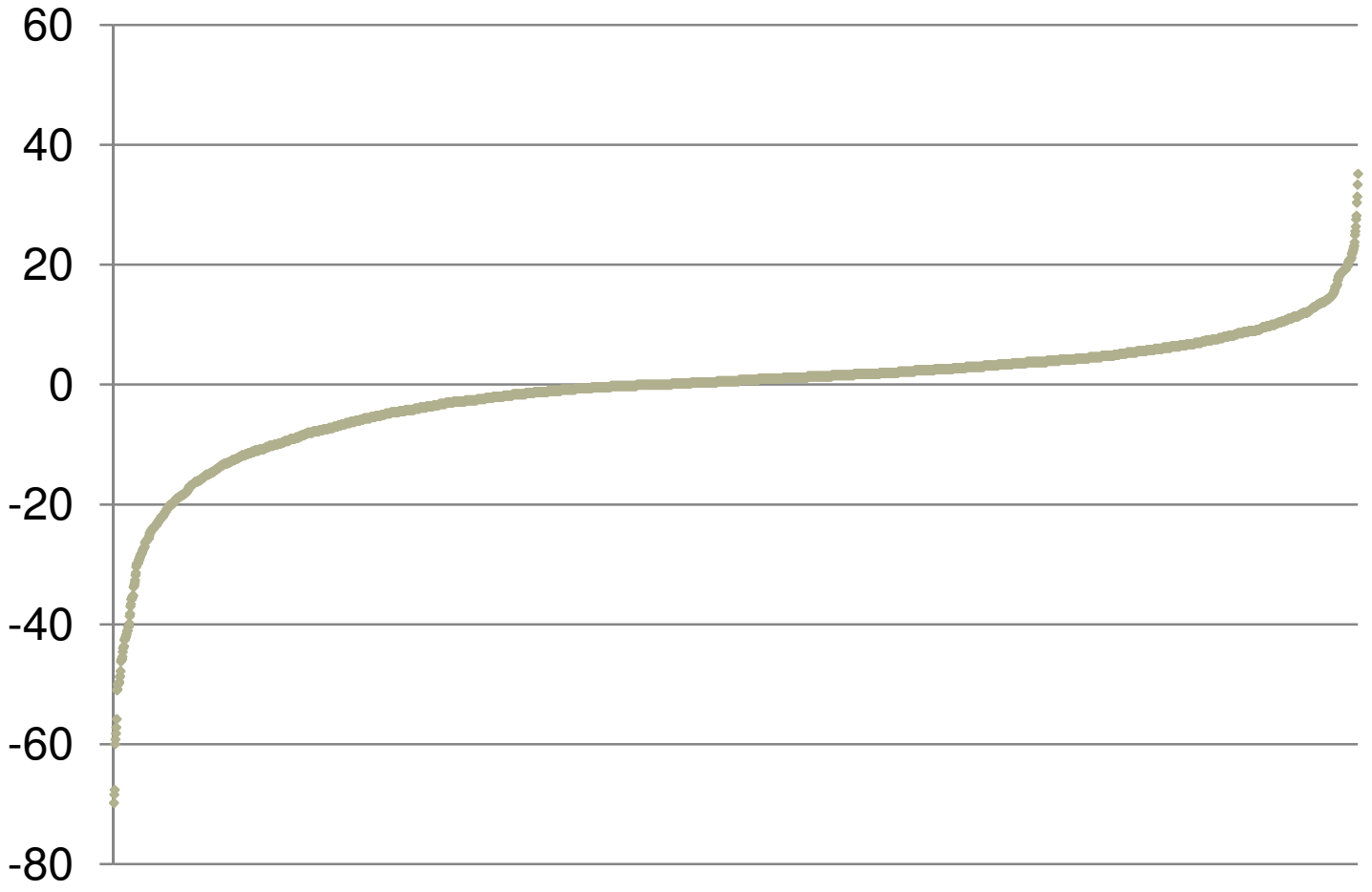
**tabu search algorithm** (`algorithm B`)

**> based on Nonobe and Ibaraki (2002)**

Nonobe, K., Ibaraki, T., 2002. Formulation and tabu search algorithm for the resource constrained project scheduling problem. *Essays and Surveys in Metaheuristics.*

**Similar behavior on PSPlib instances, but very competitive on MMlib instances!**

# STATE OF THE ART




difference in performance on MMlib instances

# PERFORMANCE PREDICTION

**through empirical hardness models**

***empirical***
**> we need to run an algorithm to measure performance**

***hardness***
**> measured as a quality of the obtained solution**

**strategy to build accurate empirical hardness models, based on Leyton-Brown et al. (2002)**

Leyton-Brown, K., Nudelman, E., Shoham, Y., 2002. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. *Constraint Programming.*

# PERFORMANCE PREDICTION

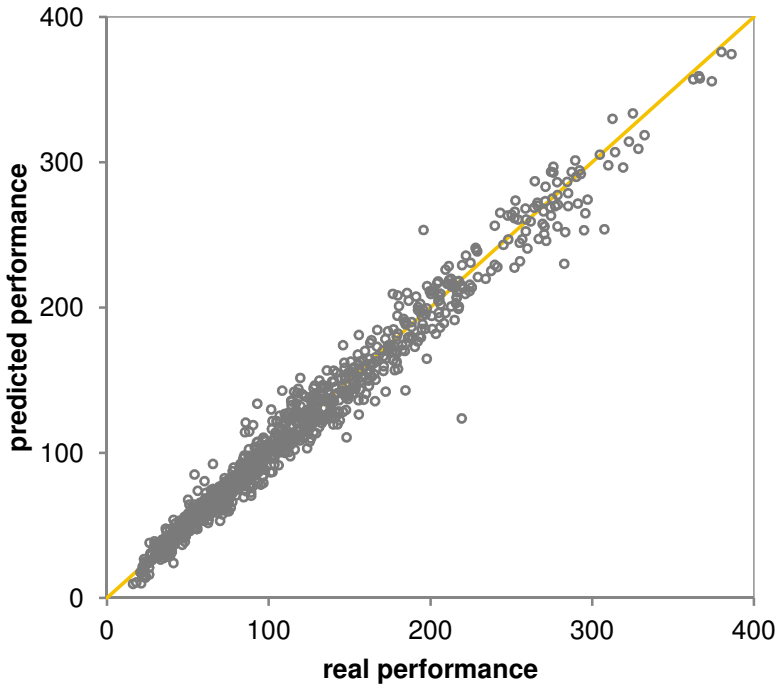
***Idea:***
**build mappings from instance features onto the performance measure**

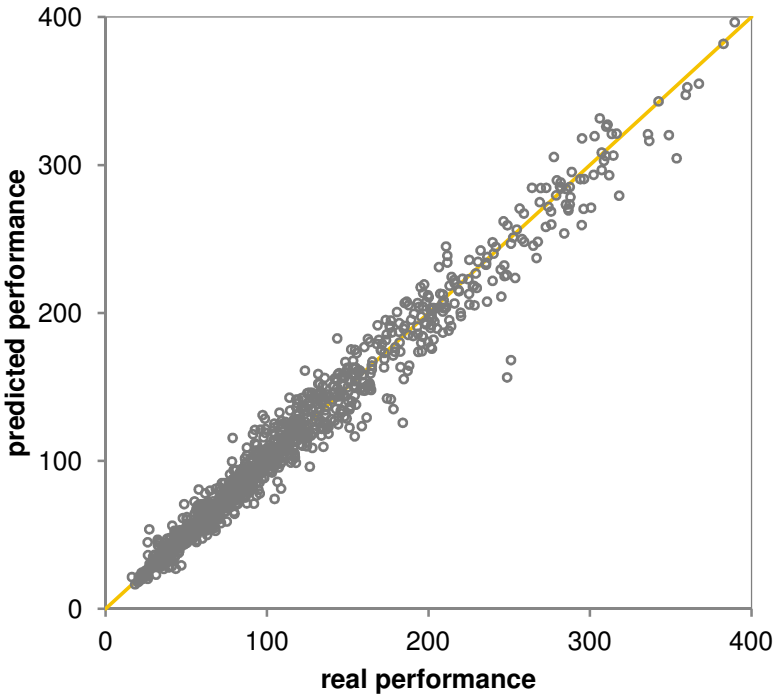
**> efficiently computable properties of the instances**

**> machine learning techniques**

**> a lot of data to learn from**

# PERFORMANCE PREDICTION




**algorithm A**

$R^2 = 0.98$
M5P tree models

**algorithm B**

# ALGORITHM SELECTION

**approach 1**

1. **predict the performance of both algorithms**
2. **select the algorithm with best prediction**

Table: results on a test set of unseen instances

| | always A | always B | approach 1 | optimal |
|---|---|---|---|---|
| % correct classified | 57.2 | 46.0 | 69.0 | 100 |
| sum performance | 115827 | 114847 | 113466 | 112065 |
| % difference from optimal | 3.36 | 2.48 | **1.25** | 0 |

# ALGORITHM SELECTION

**approach 2**

**build a model that predicts a class: `alg. A` or `alg. B`**

1. **select the algorithm that is predicted**

Table: results on a test set of unseen instances

| | always A | always B | approach 2 | optimal |
|---|---|---|---|---|
| % correct classified | 57.2 | 46.0 | 79.5 | 100 |
| sum performance | 115827 | 114847 | 112796 | 112065 |
| % difference from optimal | 3.36 | 2.48 | **0.65** | 0 |

# CONCLUSIONS – FUTURE

automatic algorithm selection works for project scheduling

but why?

and what can we learn from the relationship between the characteristics and the (type of) algorithm?

# QUESTIONS?

**Thank you.**