White, Gray and Black Box Tuning

Tommy Messelis, Nguyen Dang, Patrick De Causmaecker
KU Leuven
Department of Computer Science
CODeS research group

April 4, 2014

Outline

- **Predictors, Algorithm Selection, Portfolio**
- Tuning
- Opening the box

Predictors, Algorithm Selection, Portfolio for MRCPSP

- Problem:
  - Combine two algorithms into one algorithm by selecting the most promising one
- Datasets
  - PSPLib (Kolisch, R., Sprecher, A., EJOR 1996),
  - MMLib (Van Peteghem, V., PhD Thesis Ghent 2010).
- Algorithms
  - Algorithm A: Nonobe and Ibaraki, Tabu Search (2002)
  - Algorithm B: Lova et al., Hybrid GA (2009) t

Predictors and Algorithm Selection, MRCPSP

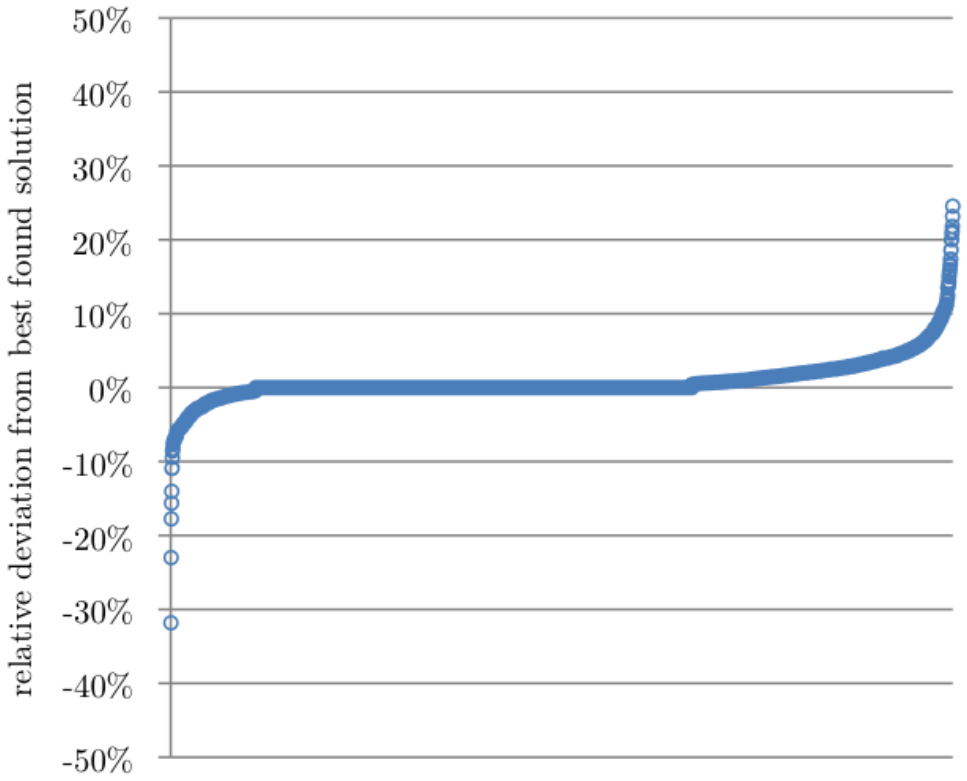
- RCPSP
  - Jobs, Renewable Resources, Precedence relations, Minimal Makespan
  - NP-hard (Blazewicz et al. (1983)).
- MRCPSP
  - Jobs with multiple modes, Resources, Precedence relations, Minimal Makespan
  - Herroelen et al. (1998) m, 1T |cpm, disc, mu|Cmax, Brucker et al. (1999) MPS|prec|Cmax

Predictors and Algorithm Selection, related work

- Smith-Miles, K., 2009. Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys 41 (1).
- Xu, L., Hutter, F., Hoos, H. H., Leyton-Brown, K., 2008. Satzilla: Portfolio-based algorithm selection for SAT. Journal of Artificial Intelligence Research 32.
- Xu, L., Hutter, F., Hoos, H. H., Leyton-Brown, K., 2011. Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming (IJCAI).

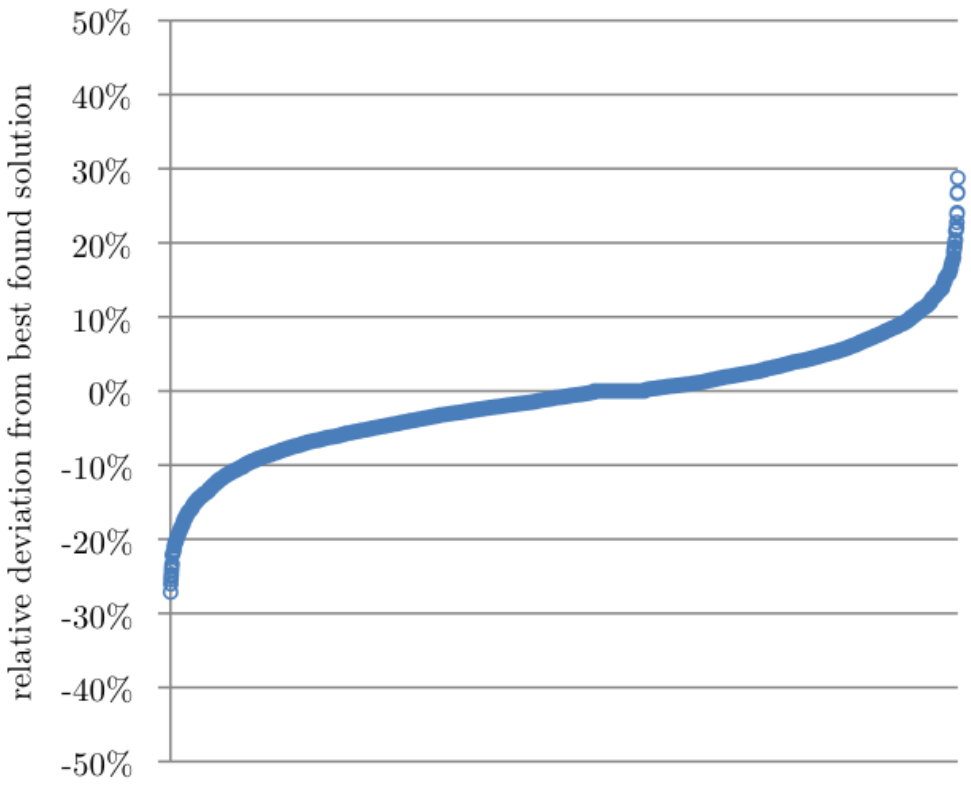
Messelis, T., De Causmaecker, P. (2013). An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem. (EJOR).

Relative difference Algorithm A / Algorithm B PSPLib, 5000 evaluations

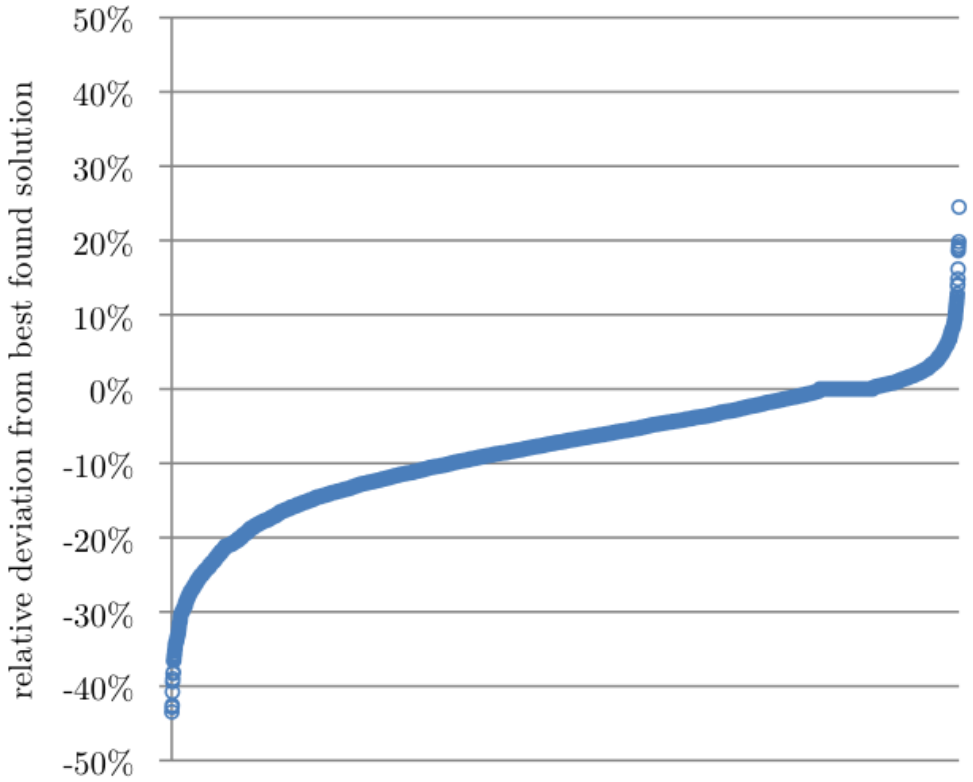


(b) relative difference

Relative difference Algorithm A / Algorithm B MMLib, 5000 evaluations



Relative difference Algorithm A / Algorithm B MMLib, 25000 evaluations



(b) relative difference

Predictors and Algorithm Selection, MRCPSP

Given a new instance of MRCPSP, which is the algorithm will we use?

Approach 1: prediction of the goal function

- Running time (exact, complete)
- **Obtained quality** (heuristic, fixed running time or number of evaluations)
- Predicion based on **features** [1]

[1] 686, size, resource constraints, precedence, duration, complexity

Prediction on quality (validation)

		always-A	always-B	AS1	AS*
5000 schedules	% correctly classified instances	58.1	49.9	69.0	100
	avg. relative deviation from best (%)	2.30	2.85	1.32	0
	avg. makespan	100.68	99.40	98.51	97.34
25000 schedules	% correctly classified instances	89.6	18.0	80.1	100
	avg. relative deviation from best (%)	0.36	8.01	1.04	0
	avg. makespan	92.25	99.98	92.66	91.90

Predictors and Algorithm Selection, MRCPSP

- Approach 2: prediction of the dominance
- Class A : algorithm A is best
- Class B : algorithm B is best
- Classification based on **features**

Prediction of dominance (validation)

		AS2a	AS2b	AS2c	AS*
5000 schedules	% correctly classified instances	80.2	77.1	83.2	100
	avg. relative deviation from best (%)	0.67	0.79	0.51	0
	avg. makespan	97.93	98.05	97.74	97.34
25000 schedules	% correctly classified instances	92.2	91.4	93.3	100
	avg. relative deviation from best (%)	0.24	0.27	0.18	0
	avg. makespan	92.05	92.08	92.02	91.90

Weka

- AS2a REPTree
- AS2b DecisionTable
- AS2c RandomForest

Prediction of dominance (reduced feature set)

		AS2a'	AS2b'	AS2c'	AS2d'	AS*
5000 schedules	% correctly classified instances	79.4	77.4	81.7	79.5	100
	avg. relative deviation from best (%)	0.69	0.84	0.57	0.67	0
	avg. makespan	97.92	98.10	97.85	97.87	97.34
25000 schedules	% correctly classified instances	91.7	91.6	93.4	91.2	100
	avg. relative deviation from best (%)	0.26	0.27	0.18	0.30	0
	avg. makespan	92.07	92.06	92.03	92.09	91.90

Weka

- AS2a REPTree
- AS2b DecisionTable
- AS2c RandomForest

Predictors and Algorithm Selection, MRCPSP

When is algorithm selection potentially fruitful?

- A = instances where alg A strictly dominates alg B
- B = instances where alg B strictly dominates alg A
-
  - competitiveness $c = 2min\{\frac{|A|}{|T|}, \frac{|B|}{|T|}\}$
  - equipotency $e = 2min(\frac{|A|}{|A|+|B|}, \frac{|B|}{|A|+|B|}\}$.
  - reach $r = \frac{|A|+|B|}{|T|}$
  - of course $c = e.r$

The fraction where prediction of dominance is correct $\geq 1 - \frac{c}{2}$.

Outline

- Predictors and Algorithm Selection
- **Tuning**
- Opening the box

Tuning: related work

- Stützle, T., López-Ibáñez, M. (2013). Automatic (offline) configuration of algorithms. In Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion. ACM.
- Hoos, H. H. (2012). Automated algorithm configuration and parameter tuning. In Autonomous Search. Springer
- i-Race, F-Racce, ParamILS, ...

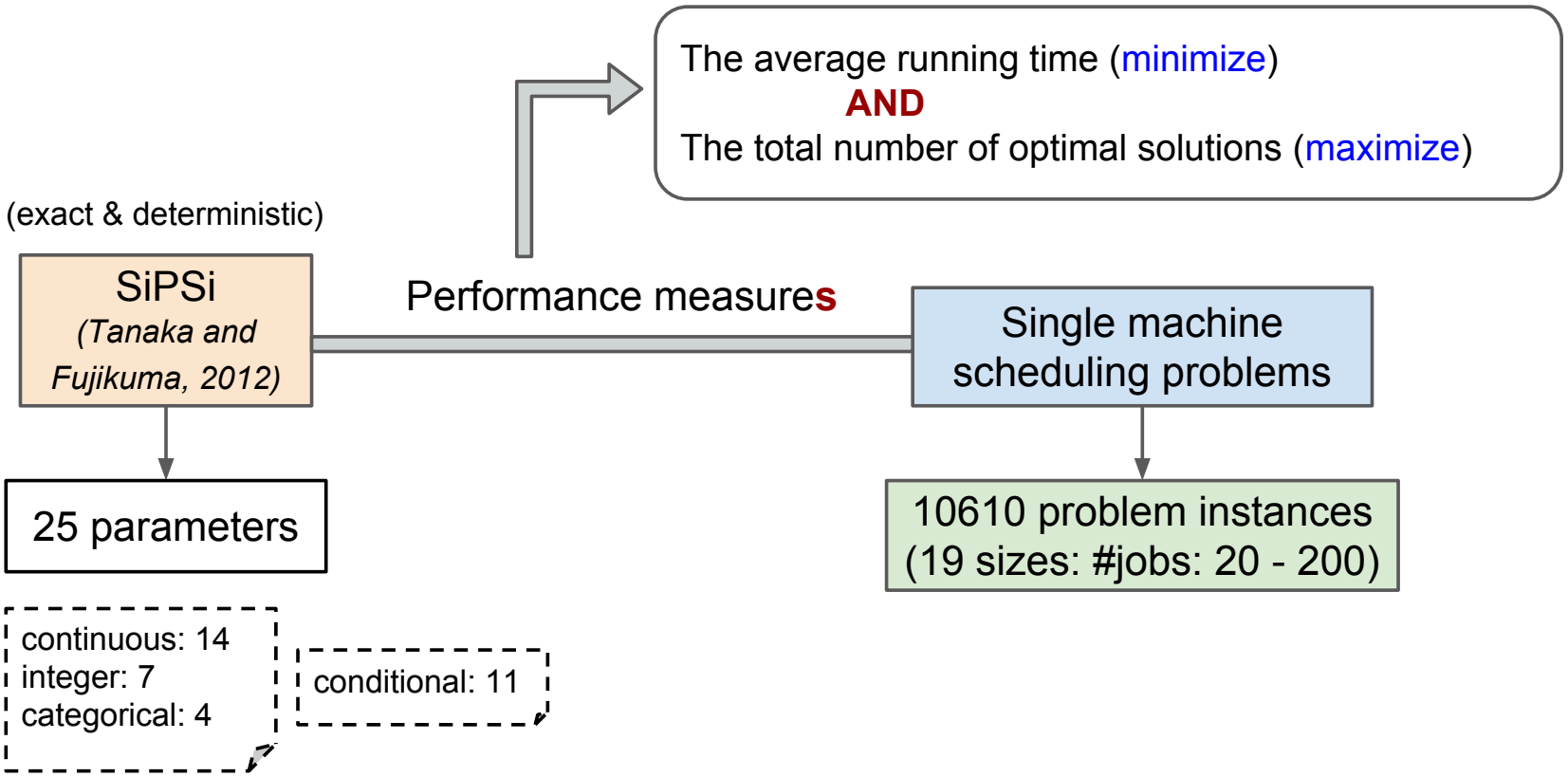
Nguyen Dang, Patrick De Causmaecker, (2014). Motivations for the Development of a Multi-objective Algorithm Configurator (ICORES 2014).

Tuning: multiple objectives

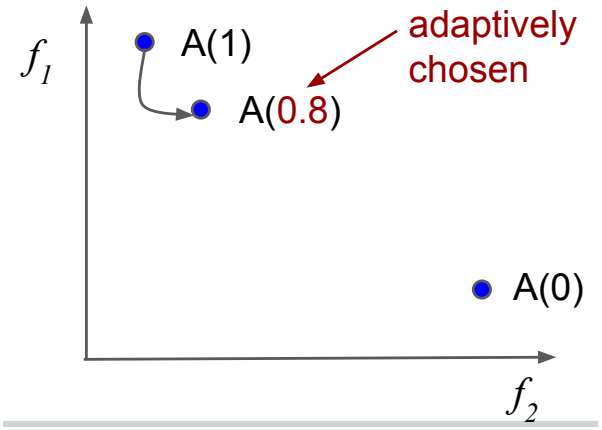
- Problem/Algorithm : Dynamic programming algorithm for single-machine scheduling [2]
- two objectives for tuning
  - running time
  - number of optimal solutions

	#optimal solutions	#total average runtime (s)
$c_{default}$	5963	**603.2**
c_{tuned}	**5970**	693.1

[2] Tanaka, S., Fujikuma, S. (2012). A dynamic-programming-based exact algorithm for general single-machine scheduling with machine idle time. Journal of Scheduling, 15(3).


The average running time (minimize)
AND
The total number of optimal solutions (maximize)
(exact & deterministic)
SiPSi
(Tanaka and Fujikuma, 2012)
Performance measures
Single machine scheduling problems
25 parameters
10610 problem instances
(19 sizes: #jobs: 20 - 200)
continuous: 14
integer: 7
categorical: 4
conditional: 11

Use adaptive local search to determine the pareto surface [3]

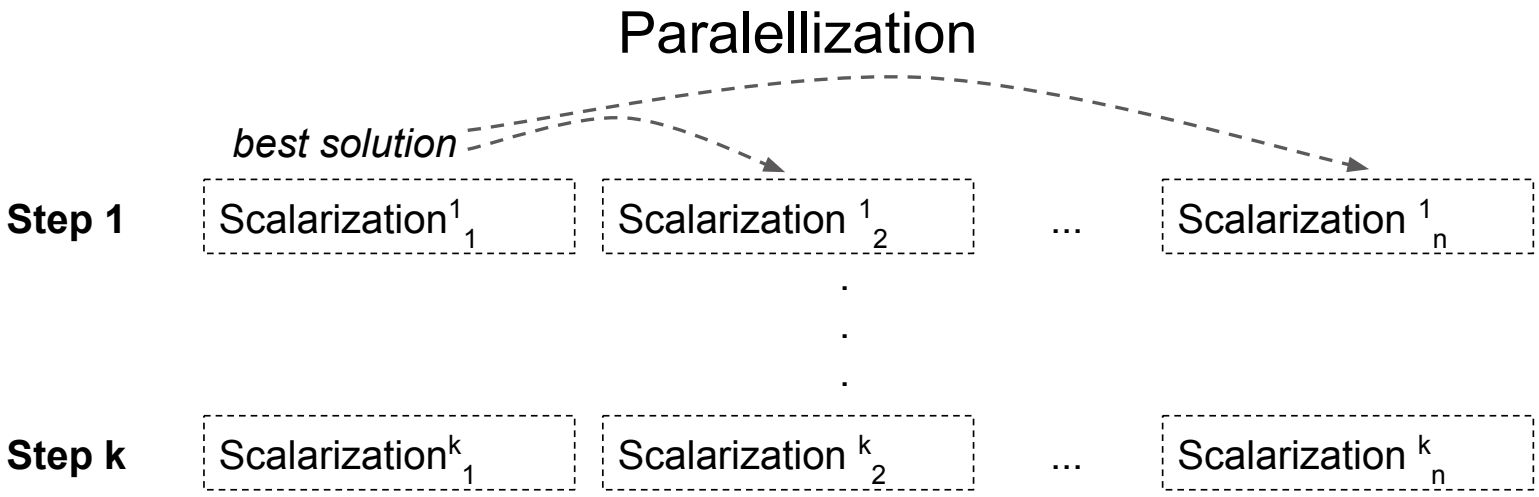


w: $f = w \times f_1 + (1-w) \times f_2$

Problem: each evaluation is a tuning exercise

[3] e.g. AA-TPLS, Dubois-Lacoste et a. 2011

Problem: each evaluation is a tuning exercise
Proposal:



Results to come!

Outline

- Predictors and Algorithm Selection
- Tuning
- **Opening the box**
  - Further work

Effort

- Datasets
- Running time
- Analysis

objectives reconsidered

- Running time and number of optimal solutions are only known at the end of the run
  - Black box

Opening the box: existing approaches, related work

- Hyperheuristics [4]
  - construction
  - selection
- Algorithm construction [5]
- Programming by optimization
  - http://www.prog-by-opt.net/

[4] Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R. (2013). Hyper- heuristics: A survey of the state of the art. Journal of the Operational Research Society, 64(12)

[5] Özcan, E., Parkes, A. J. (2011, July). Policy matrix evolution for generation of heuristics. In Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM.
Fukunaga, A. S. (2008). Automated discovery of local search heuristics for satisfiability testing. Evolutionary Computation, 16(1), 31-61.

Opening the box: existing approaches, related work

In particular:

- Halim, S., Yap, R. H., Lau, H. C. (2007). An integrated white+ black box approach for designing and tuning stochastic local search. In Principles and Practice of Constraint Programming-CP 2007. Springer.
- Visualization of the search trajectory.

Opening the box: potential

- Running time problem
- Observables
- How to obtain observables
  - standard (from metaheuristics)
  - standard (improvement, LLH runtimes,...)
  - user defined
    - detailed algorithmic
    - implementation bound (data structures, low level algorithms)

Come and see