How do Performance Indicator Parametrizations Influence the Assessment of Algorithm Portfolios?

Pascal Kerschke, Jakob Bossek, Heike Trautmann

September 17, 2018 in Paris, France

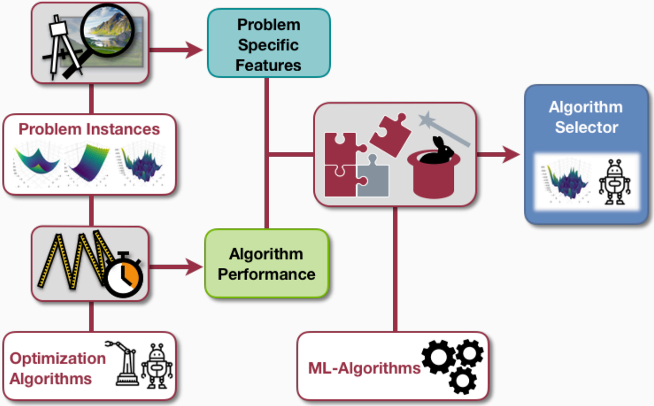
Informations Systems and Statistics,
University of Münster, Germany

Introduction

Algorithm Selection

Idea of Algorithm Selection:

- *Algorithm Selection Problem*[1]: find the individually best suited algorithm for an unseen optimization problem



1. Rice, J. (1976). *The Algorithm Selection Problem.* In Advances in Computers (pp. 65-118).

Algorithm Selection

Requirements:

- Comprehensive benchmark of portfolio solvers (as a foundation for algorithm selection)
- Suitable performance measure needed, e.g., PAR10[2], ERT[3].
- Performance measures often parameterized.
  $\rightsquigarrow$ How do parameters affect the benchmark results?

---

2. Bischl, B. et al. (2016). *ASlib: A Benchmark Library for Algorithm Selection.* In Artificial Intelligence Journal (pp. 41-58).
3. Hansen, N. et al. (2009). *Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental Setup.* In INRIA Research Report RR-6828.

Algorithm Selection

Requirements:

- Comprehensive benchmark of portfolio solvers (as a foundation for algorithm selection)
- Suitable performance measure needed, e.g., PAR10[2], ERT[3].
- Performance measures often parameterized.
  $\rightsquigarrow$ How do parameters affect the benchmark results?

Our contribution:

- Systematic analysis of parameterizations on a comprehensive benchmark study of inexact TSP solvers.

---

2. Bischl, B. et al. (2016). *ASlib: A Benchmark Library for Algorithm Selection.* In Artificial Intelligence Journal (pp. 41-58).
3. Hansen, N. et al. (2009). *Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental Setup.* In INRIA Research Report RR-6828.

Notation

Notation of considered input parameters:

- Set of problem instances $\mathcal{I} = \{I_1, \ldots, I_{n_{\mathcal{I}}}\}$,

Notation

Notation of considered input parameters:

- Set of problem instances $\mathcal{I} = \{I_1, \ldots, I_{n_\mathcal{I}}\}$,
- Set of algorithms/solvers $\mathcal{A} = \{A_1, \ldots, A_{n_\mathcal{A}}\}$,

Notation

Notation of considered input parameters:

- Set of problem instances $\mathcal{I} = \{I_1, \ldots, I_{n_{\mathcal{I}}}\}$,
- Set of algorithms/solvers $\mathcal{A} = \{A_1, \ldots, A_{n_{\mathcal{A}}}\}$,
- $m > 1$ independent runs of each $A \in \mathcal{A}$ on $I \in \mathcal{I}$

Notation

Notation of considered input parameters:

- Set of problem instances $\mathcal{I} = \{I_1, \ldots, I_{n_\mathcal{I}}\}$,
- Set of algorithms/solvers $\mathcal{A} = \{A_1, \ldots, A_{n_\mathcal{A}}\}$,
- $m > 1$ independent runs of each $A \in \mathcal{A}$ on $I \in \mathcal{I}$
- $\rightsquigarrow$ empirical runtimes $r_1^{A,I}, \ldots, r_m^{A,I}$.

Notation

Notation of considered input parameters:

- Set of problem instances $\mathcal{I} = \{I_1, \dots, I_{n_\mathcal{I}}\}$,
- Set of algorithms/solvers $\mathcal{A} = \{A_1, \dots, A_{n_\mathcal{A}}\}$,
- $m > 1$ independent runs of each $A \in \mathcal{A}$ on $I \in \mathcal{I}$
- $\leadsto$ empirical runtimes $r_1^{A,I}, \dots, r_m^{A,I}$.
- Time limit / cutoff time $T \in \mathbb{R}_{>0}$.

Performance Measures

Numeric Example:

- 10 runs of solvers X and Y
- budget for runtime r_s of successful runs is set to $T = 1$
- solver X: 8 successful runs ($p_f = 0.2$) with $r_s = 0.8$
- solver Y: 2 successful runs ($p_f = 0.8$) with $r_s = 0.2$

Numeric Example:

- 10 runs of solvers X and Y
- budget for runtime r_s of successful runs is set to $T = 1$
- solver X: 8 successful runs ($p_f = 0.2$) with $r_s = 0.8$
- solver Y: 2 successful runs ($p_f = 0.8$) with $r_s = 0.2$

$\leadsto$ How do we aggregate the runs (meaningfully)?

Performance Measures (cont.)

Penalized Average Runtime (PAR)[4]:

- Arithmetic mean of running times, $r_i^{A,I}, i \in [m]$
- Expired runs are penalized by $f \cdot T$, where f is the **penalty factor**

4. Bischl, B. et al. (2016). *ASlib: A Benchmark Library for Algorithm Selection.* In Artificial Intelligence Journal (pp. 41-58).

Performance Measures (cont.)

Penalized Average Runtime (PAR)[4]:

- Arithmetic mean of running times, $r_i^{A,I}, i \in [m]$
- Expired runs are penalized by $f \cdot T$, where f is the **penalty factor**

$$\mathrm{PAR}_{A,I}(f) := \frac{1}{m}\sum_{i=1}^{m}\tilde{r}_i^{A,I} \quad \text{with} \quad \tilde{r}_i^{A,I} = \begin{cases} f \cdot T, & \text{if } r_i^{A,I} > T \\ r_i^{A,I}, & \text{otherwise.} \end{cases}$$

4. Bischl, B. et al. (2016). *ASlib: A Benchmark Library for Algorithm Selection.* In Artificial Intelligence Journal (pp. 41-58).

Performance Measures (cont.)

Penalized Average Runtime (PAR)[4]:

- Arithmetic mean of running times, $r_i^{A,I}, i \in [m]$
- Expired runs are penalized by $f \cdot T$, where f is the **penalty factor**

$$\text{PAR}_{A,I}(f) := \frac{1}{m} \sum_{i=1}^{m} \tilde{r}_i^{A,I} \quad \text{with} \quad \tilde{r}_i^{A,I} = \begin{cases} f \cdot T, & \text{if } r_i^{A,I} > T \\ r_i^{A,I}, & \text{otherwise.} \end{cases}$$

- Usually, the rather heuristic value $f = 10$ is employed.
  $\rightsquigarrow PAR_{A,I}(10)$

4. Bischl, B. et al. (2016). *ASlib: A Benchmark Library for Algorithm Selection.* In Artificial Intelligence Journal (pp. 41-58).

Performance Measures (cont.)

Penalized Quantile Runtime (PQR)[5]:

- Replace outlier-sensitive mean by more robust p-**quantile**, $p \in (0, 1]$.

5. Kerschke, P. et al. (2018). *Parameterization of State-of-the-Art Performance Indicators: A Robustness Study Based on Inexact TSP Solvers.* In Proceedings of GECCO 2018 Companion (pp. 1737-1744).

Performance Measures (cont.)

Penalized Quantile Runtime (PQR)[5]:

- Replace outlier-sensitive mean by more robust p-**quantile**, $p \in (0, 1]$.

$$\text{PQR}_{A,I}(p, f) := \begin{cases} f \cdot T, & \text{if } \sum_{i=1}^{m} \mathbb{1}\{r_i^{A,I} < T\} < \lfloor mp + 1 \rfloor \\ q_p(r_1^{A,I}, \ldots, r_m^{A,I}), & \text{otherwise.} \end{cases}$$

5. Kerschke, P. et al. (2018). *Parameterization of State-of-the-Art Performance Indicators: A Robustness Study Based on Inexact TSP Solvers.* In Proceedings of GECCO 2018 Companion (pp. 1737-1744).

Performance Measures (cont.)

Expected Runtime (ERT)[6]:

- Popular / most common measure in continuous optimization.

6. Hansen, N. et al. (2009). *Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental Setup.* In INRIA Research Report RR-6828.

Performance Measures (cont.)

Expected Runtime (ERT)[6]:

- Popular / most common measure in continuous optimization.

$$\mathrm{ERT}_{A,I} = \frac{1}{s}\sum_{j=1}^{s} r_{i_j}^{A,I} + \left(\frac{1-p_s}{p_s}\right)\cdot T$$

$$= \frac{1}{s}\left(\sum_{j=1}^{s} r_{i_j}^{A,I} + (m-s)\cdot T\right).$$

6. Hansen, N. et al. (2009). *Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental Setup*. In INRIA Research Report RR-6828.

Performance Measures (cont.)

Expected Runtime (ERT)[6]:

- Popular / most common measure in continuous optimization.

$$\mathrm{ERT}_{A,I} = \frac{1}{s}\sum_{j=1}^{s} r_{i_j}^{A,I} + \left(\frac{1-p_s}{p_s}\right)\cdot T$$
$$= \frac{1}{s}\left(\sum_{j=1}^{s} r_{i_j}^{A,I} + (m-s)\cdot T\right).$$

- Corresponds to average runtime for observing one successful run.

6. Hansen, N. et al. (2009). *Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental Setup.* In INRIA Research Report RR-6828.

Performance Measures (cont.)

Penalized Expected Runtime (PERT):

- Introducing penalty factor into ERT.

Penalized Expected Runtime (PERT):

- Introducing penalty factor into ERT.

$$\mathrm{PERT}_{A,I}(f) = \frac{1}{s}\sum_{j=1}^{s} r_{i_j}^{A,I} + \left(\frac{1-p_s}{p_s}\right)\cdot f \cdot T$$
$$= \frac{1}{s}\left(\sum_{j=1}^{s} r_{i_j}^{A,I} + (m-s)\cdot f \cdot T\right).$$

Performance Measures (cont.)

Numeric Example:

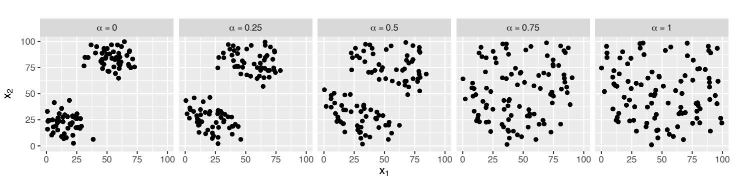
- 10 runs of solvers X and Y
- budget for runtime r_s of successful runs is set to $T = 1$
- solver X: 8 successful runs ($p_f = 0.2$) with $r_s = 0.8$
- solver Y: 2 successful runs ($p_f = 0.8$) with $r_s = 0.2$

Performance Indicator	$f = 10$		$f = 100$	
	X	Y	X	Y
$PAR_{A,I}(f)$	2.64	8.04	20.64	80.04
$PQR_{A,I}(0.5, f)$	0.80	10.00	0.80	100.00
$ERT_{A,I}$	1.05	4.20	1.05	4.20
$PERT_{A,I}(f)$	3.30	40.20	25.80	400.20

Case Study

Case Study

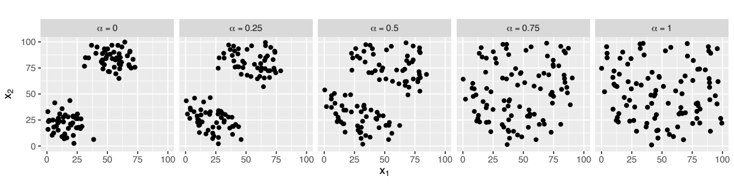
- Based on performance data from our previous TSP algorithm selection study[7]:
- Five state-of-the-art inexact TSP solvers (**Algorithms** $\mathcal{A}$):
  - MAOS [4], EAX [3], LKH [2], EAX+restart and LKH+restart [1].
- Six sets of TSP instances (**Problems** $\mathcal{I}$):
  - VLSI, TSPLIB, National, RUE, clustered (netgen) and morphed.



7. Kerschke, P. et al. (2017). *Leveraging TSP Solver Complementarity through Machine Learning.* In ECJ.

Case Study

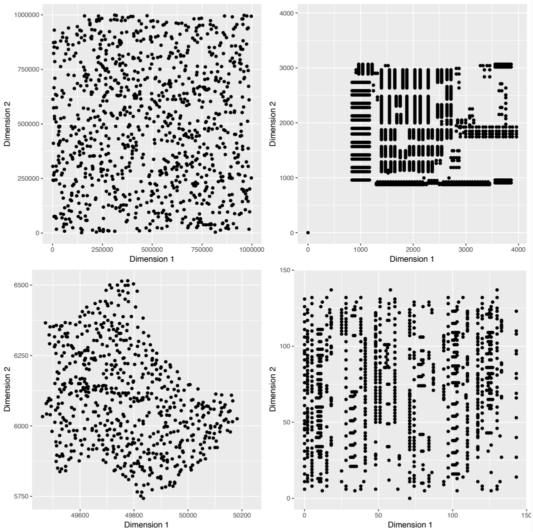
- Based on performance data from our previous TSP algorithm selection study[7]:
- Five state-of-the-art inexact TSP solvers (**Algorithms** $\mathcal{A}$):
  - MAOS [4], EAX [3], LKH [2], EAX+restart and LKH+restart [1].
- Six sets of TSP instances (**Problems** $\mathcal{I}$):
  - VLSI, TSPLIB, National, RUE, clustered (netgen) and morphed.



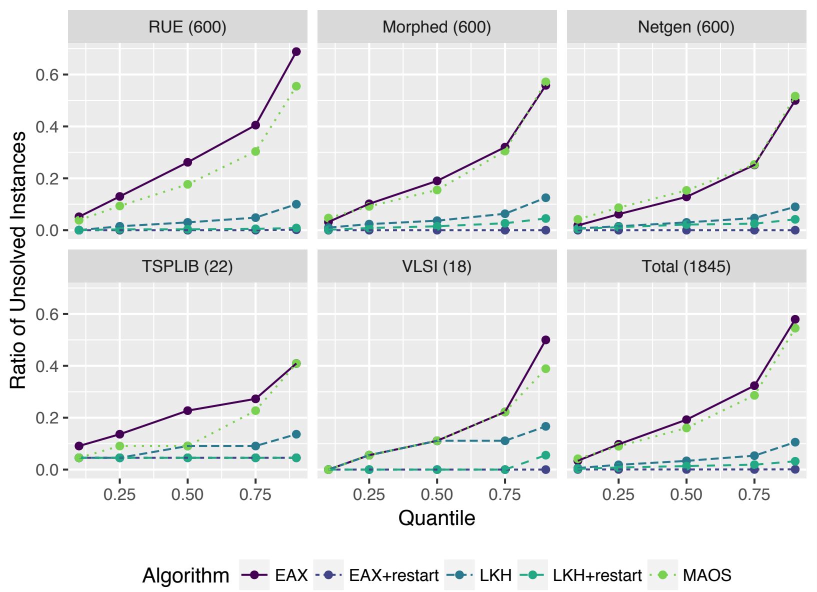
- EAX+restart was single-best-solver (SBS) regarding $PAR_{A,I}(10)$.

7. Kerschke, P. et al. (2017). *Leveraging TSP Solver Complementarity through Machine Learning.* In ECJ.

Case Study

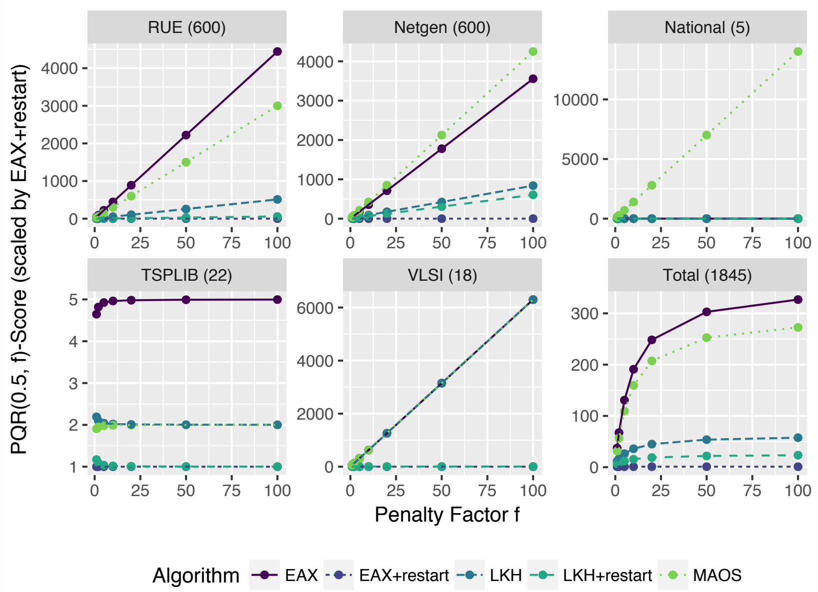


Case Study

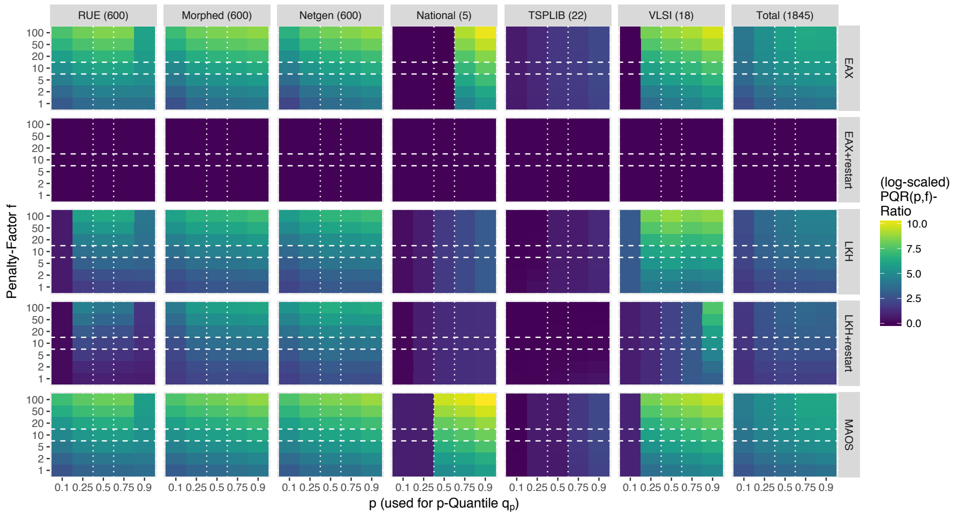


RUE (600)
Morphed (600)
Netgen (600)
TSPLIB (22)
VLSI (18)
Total (1845)
Ratio of Unsolved Instances
0.0
0.2
0.4
0.6
0.25
0.50
0.75
Quantile
Algorithm
EAX
EAX+restart
LKH
LKH+restart
MAOS

Case Study


RUE (600)
Netgen (600)
National (5)
TSPLIB (22)
VLSI (18)
Total (1845)
PQR(0.5, f)-Score (scaled by EAX+restart)
Penalty Factor f
Algorithm
EAX
EAX+restart
LKH
LKH+restart
MAOS

Case Study


RUE (600)
Morphed (600)
Netgen (600)
National (5)
TSPLIB (22)
VLSI (18)
Total (1845)
EAX
EAX+restart
LKH
LKH+restart
MAOS
Penalty-Factor f
100
50
20
10
5
2
1
0.1 0.25 0.5 0.75 0.9
p (used for p-Quantile q_p)
(log-scaled) PQR(p,f)-Ratio
10.0
7.5
5.0
2.5
0.0

Case Study

Observations:

- Finding a suitable pair of penalty factor f and quantile p quickly becomes very complex.
- (Visual) Comparison of solvers also becomes more difficult.

Case Study

Observations:

- Finding a suitable pair of penalty factor f and quantile p quickly becomes very complex.
- (Visual) Comparison of solvers also becomes more difficult.

Idea:

- One basically wants to optimize the runtime <u>and</u> success probability simultaneously.
- Why not use a multi-objective approach?
  $\rightsquigarrow$ consider for instance HV principle

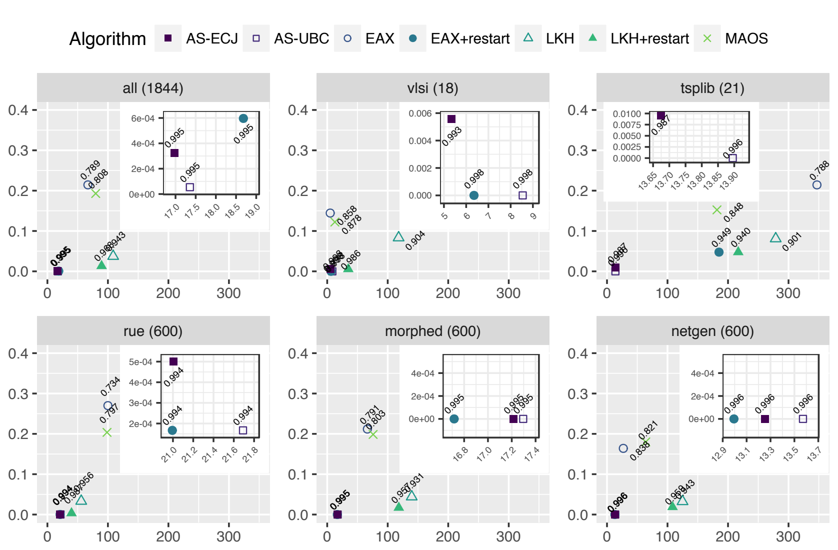
Numeric Example:

- solver X: 8 successful runs ($p_f = 0.2$) with $r_s = 0.8$
- solver Y: 2 successful runs ($p_f = 0.8$) with $r_s = 0.2$

$$\mathrm{HV}_{A,I} = \left(T - r_s\right) \cdot \left(1 - p_f\right).$$

Performance Indicator	$f = 10$		$f = 100$	
	X	Y	X	Y
$PAR_{A,I}(f)$	2.64	8.04	20.64	80.04
$PQR_{A,I}(0.5, f)$	0.80	10.00	0.80	100.00
$ERT_{A,I}$	1.05	4.20	1.05	4.20
$PERT_{A,I}(f)$	3.30	40.20	25.80	400.20
$HV_{A,I}$	0.16	0.16	0.16	0.16

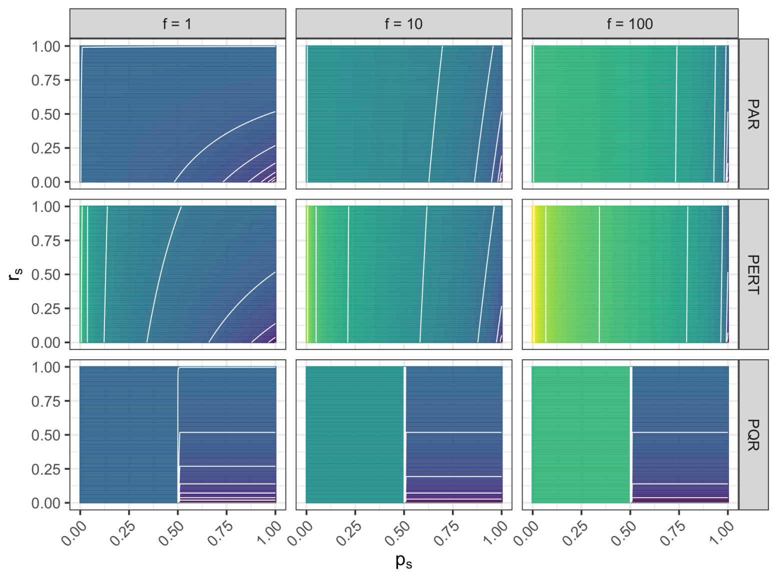
(Visual and Measure Independent) Comparison of TSP Solvers:



Further Comparison of Performance Measure Parameterizations

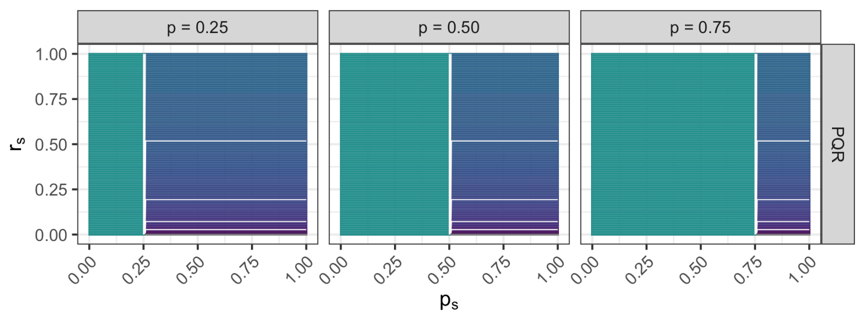
Further Comparison of Performance Measure Parameterizations

(Theoretical) Effect of Penalty Factor on Performance Measures:



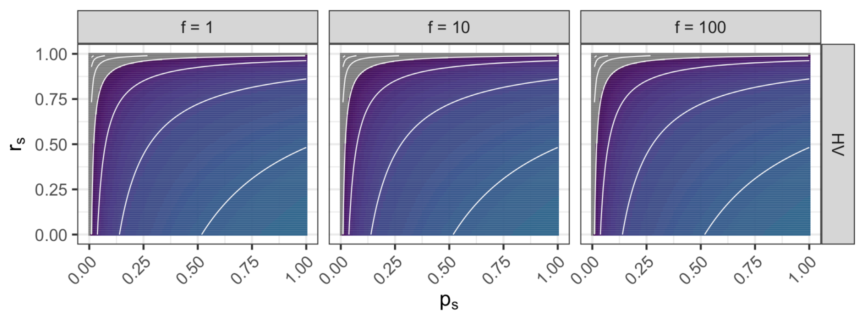
Further Comparison of Performance Measure Parameterizations

(Theoretical) Effect of Quantile on PQR(p, 10)-Score:



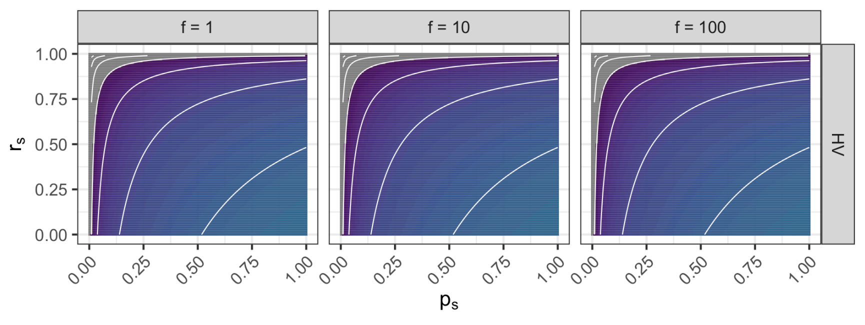
Further Comparison of Performance Measure Parameterizations

HV indicator as performance measure:



Further Comparison of Performance Measure Parameterizations

HV indicator as performance measure:



Note that HV is robust against alterations of the penalty score.

Conclusion & Outlook

Conclusion & Outlook

Conclusion:

- We systematically analyzed effects of different parameterizations of performance indicators.
- Varying penalty factor allows for altering leverage of failed runs.
- (P)ERT is much more prone to single runs
  $\rightsquigarrow$ huge impact of single failed runs.
- Choosing a suitable measure has a huge impact on the actual performance assessment (for solvers and selectors).
- HV might be a good alternative to common measures.

Outlook:

- Theoretical investigations of indicators.
- Introduction of alternative (multi-objective) indicators[8].
- Application in context of algorithm selection.

8. Bossek, J. & Trautmann, H. (2018). *Multi-Objective Performance Measurement: Alternatives to PAR10 and Expected Running Time.* In Proceedings of LION 2018.

Questions?

References

[1] Jérémie Dubois-Lacoste, Holger H. Hoos, and Thomas Stützle. On the Empirical Scaling Behaviour of State-of-the-art Local Search Algorithms for the Euclidean TSP. In **Proceedings of the 17th Annual Conference on Genetic and Evolutionary Computation (GECCO)**, pages 377 – 384. ACM, 2015. ISBN 978-1-4503-3472-3. doi: 10.1145/2739480.2754747. URL `http://dl.acm.org/citation.cfm?id=2754747`.

[2] Keld Helsgaun. General k-opt Submoves for the Lin-Kernighan TSP Heuristic. **Mathematical Programming Computation**, 1(2-3):119 – 163, 2009. doi: 10.1007/s12532-009-0004-6. URL `https://link.springer.com/article/10.1007/s12532-009-0004-6`.

[3] Yuichi Nagata and Shigenobu Kobayashi. A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. **INFORMS Journal on Computing**, 25(2):346 – 363, 2013. doi: 10.1287/ijoc.1120.0506. URL `http://dl.acm.org/citation.cfm?id=2466704`.

[4] Xiao-Feng Xie and Jiming Liu. Multiagent Optimization System for Solving the Traveling Salesman Problem (TSP). **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, 39(2):489 – 502, 2009. URL `http://ieeexplore.ieee.org/document/4717264/`.