Automatic Design of Multi-Objective Local Search Algorithms

Case Study on a bi-objective Permutation Flowshop Scheduling Problem

Aymeric Blot[1] Marie-Éléonore Kessaci[1] Laetitia Jourdan[1]
Holger H. Hoos[2]

[1]Université de Lille, Inria, CNRS, UMR 9189 CRIStAL, France
[2]Universiteit Leiden, The Netherlands & University of British Columbia, Canada

COSEAL – September, 2017

Motivation

Goal

- Use MO-AAC to design multi-objective metaheuristics

Case Study

- ParamILS & MO-ParamILS
- Multi-objective Local Search algorithms
- Bi-objective Permutation Flowshop Scheduling Problem

Context

Automatic Algorithm Configuration (AAC)

Single-Objective AAC

Single performance indicator

- irace [López-Ibáñez *et al.*, 2016]
- ParamILS [Hutter *et al.*, 2009]
- SMAC [Hutter *et al.*, 2010]
- GGA++ [Ansótegui *et al.*, 2015], . . .

Multi-Objective AAC

Multiple performance indicators

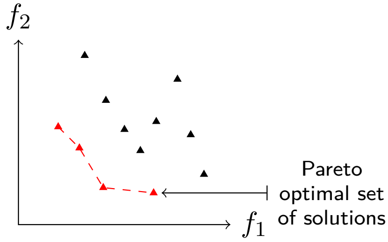
- MO-ParamILS [Blot *et al.*, 2016]
- SPRINT-Race [Zhang *et al.*, 2015]

Context

Multi-Objective algorithms

Performance Indicators

- Convergence (HV)
- Distribution (Δ)
- Diversity
- Size



Our Approach

Small-Scale Analysis

Blot, Jourdan, Kessaci-Marmion

Automatic Design of Multi-objective Local Search Algorithms

GECCO'17, Berlin (Germany), July 2017

Large-Scale Analysis

Blot, Pernet, Jourdan, Kessaci-Marmion and Hoos

Automatically Configuring Multi-objective Local Search using Multi-objective Optimisation

EMO'17, LNCS 10173: 61–73, Springer, Münster (Germany), March 2017

MO-ParamILS

Roots

ParamILS

Hutter, Hoos, Leyton-Brown, Stützle

ParamILS: An Automatic Algorithm Configuration Framework

Journal of Artificial Intelligence Research (36), 2009

MO-ParamILS

Blot, Hoos, Jourdan, Kessaci-Marmion and Trautmann

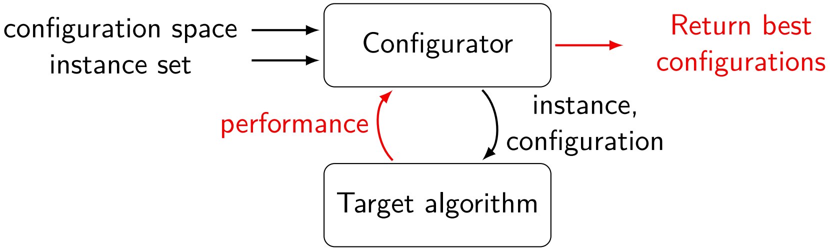
MO-ParamILS: A Multi-objective Automatic Algorithm Configuration Framework

LION'16, LNCS 10079: 32–47, Springer, Ischia Island (Napoli, Italy), May 2016

MO-ParamILS

MO-ParamILS

- Extension of ParamILS for multiple performance indicators
- Iterated MOLS on the configuration space
- Outputs a Pareto set of configurations



MO-ParamILS

Machine learning process

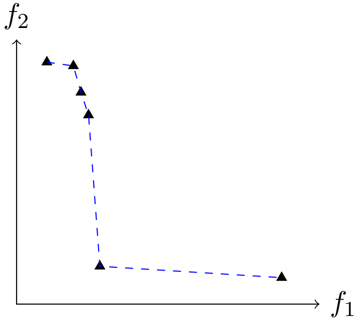
Experimental protocol

- Training (multiple times)
  - Training instances
  - Randomised seed and instance order
- Validation
  - Training instances
  - Every final training configuration
- Test
  - Testing instances
  - Every non-dominated validated configuration

MO-ParamILS

Training

Problem space

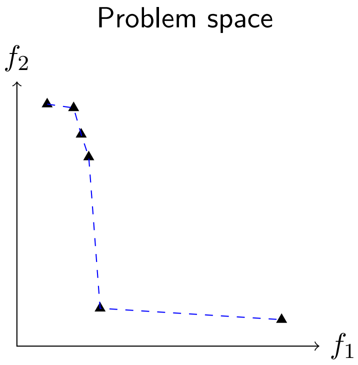


(execution on single instance)

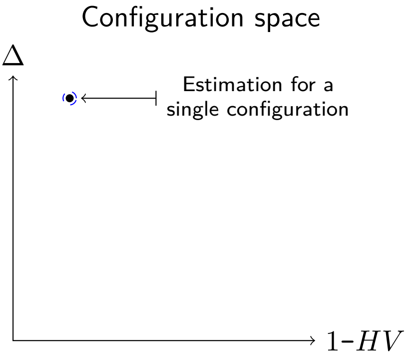
- For every configuration
  - multiple runs
  - multiple instances
- Average HV and Δ over multiple runs

MO-ParamILS

Training



(execution on single instance)



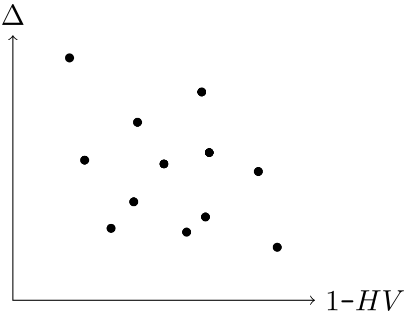
(average over training instances)

MO-ParamILS

Training

- iteratively investigates configurations
- refining quality estimations
- returns non-dominated
- shuffles training instances

Configuration space



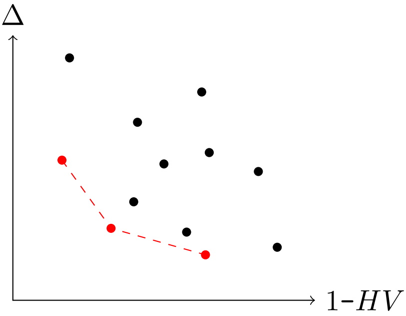
(average over training instances)

MO-ParamILS

Training

- iteratively investigates configurations
- refining quality estimations
- returns non-dominated
- shuffles training instances

Configuration space



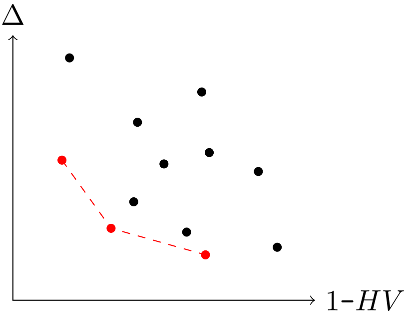
(average over training instances)

MO-ParamILS

Training

- iteratively investigates configurations
- refining quality estimations
- returns non-dominated
- shuffles training instances

Configuration space

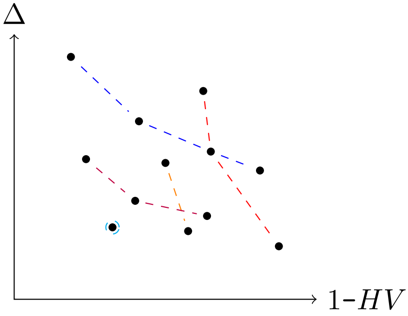


(average over training instances)

MO-ParamILS

Validation, test

Training



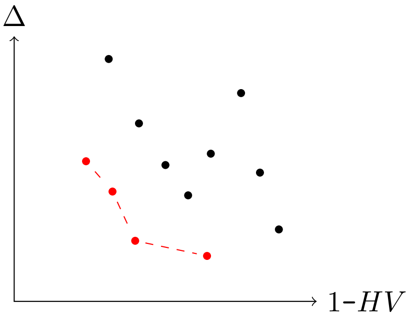
(average over training instances)

- Training
  - multiple training runs
  - different instance subsets
  - incomparable quality estimation
- Validation
  - all training configurations
  - all training instances

MO-ParamILS

Validation, test

Validation

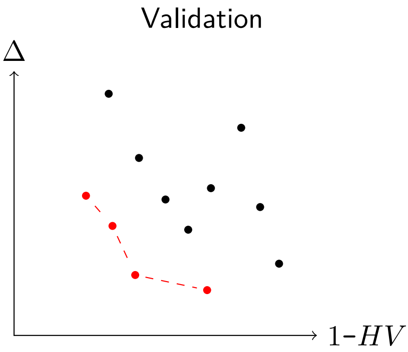


(average over training instances)

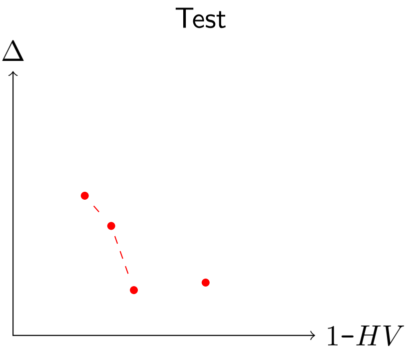
- Training
  - multiple training runs
  - different instance subsets
  - incomparable quality estimation
- Validation
  - all training configurations
  - all training instances

MO-ParamILS

Validation, test



(average over training instances)



(average over test instances)

Case Study

Multi-objective Local Search Algorithms (MOLS)

Key Points

- Efficient metaheuristics
- Used on many problems (*e.g.*, scheduling, routing, assignment)
- Many strategies and parameters

Instantiations

- Methods
  - Pareto Archived Evolution Strategy (PAES, 1999, 2000)
  - Pareto Local Search (PLS, 2001, 2004, 2011, 2012, 2015)
- Unifications
  - Stochastic Pareto Local Search (SPLS, 2012)
  - Dominance-based Multi-objective Local Search (DMLS, 2012)

Case Study

Permutation Flowshop Scheduling Problem (PFSP)

Bi-objective PFSP

- 2 objectives
  - Makespan (max completion time)
  - Flowtime (sum completion time)
- Classical Taillard instances (20-100 jobs; 5-20 machines)

Small-Scale Analysis

Does MO-AAC actually finds the best configurations?

Exhaustive Analysis

- Every possible configuration
- On the test set

MO-ParamILS

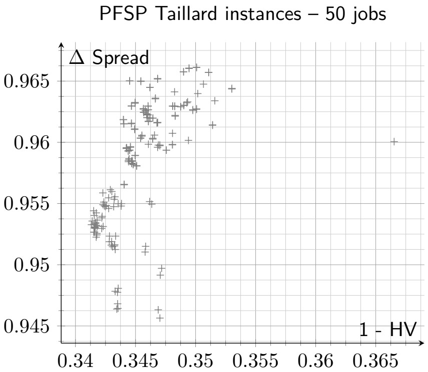
- Training (30 times)
- Validation
- Test

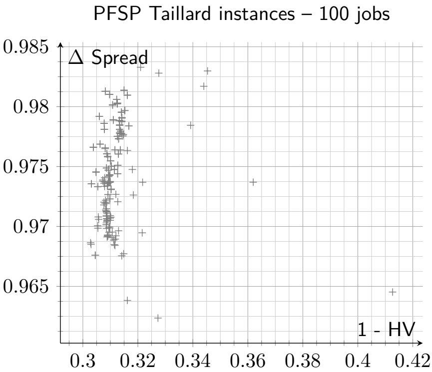
Target Algorithm

- MOLS (189 configurations)
  - 4 categorical parameters
  - 3 integer parameters

Exhaustive Analysis

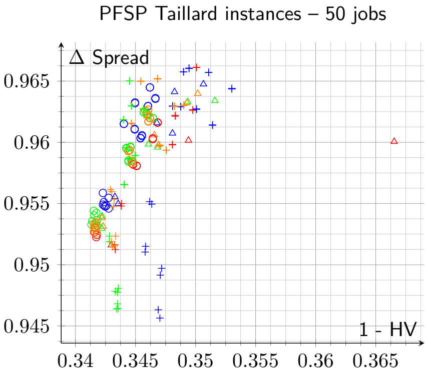
+: Configuration

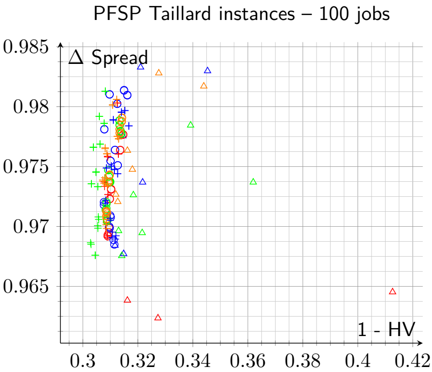




Exhaustive Analysis

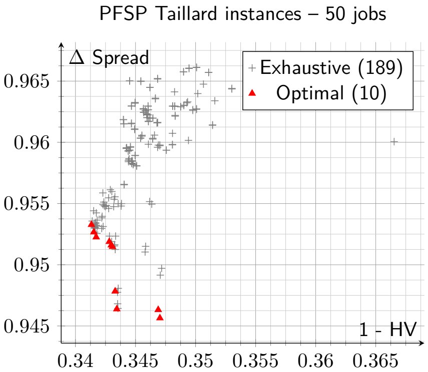
Exploration strategy: Δ o + ; Selection strategy: ■ ■ ■ ■

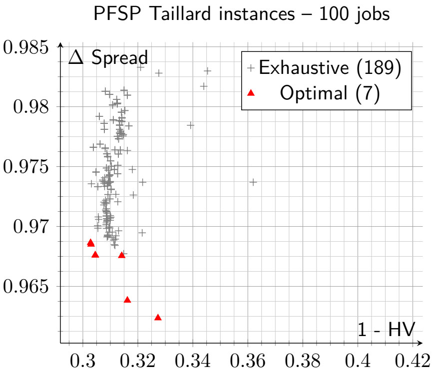




Exhaustive Analysis

Pareto optimal solutions

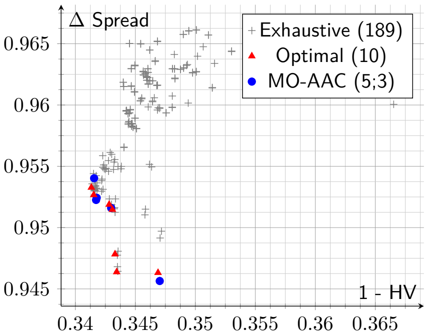




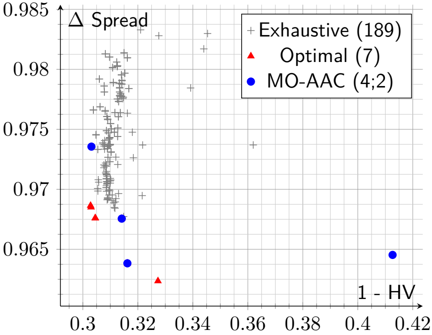
MO-ParamILS

Using exhaustive test analysis

PFSP Taillard instances – 50 jobs



PFSP Taillard instances – 100 jobs



Interpretations

Exhaustive Analysis

- Big differences wrt benchmark
- Parameter consistency

MO-ParamILS

- No loss of hypervolume
- Really close to the optimal configurations

Large-Scale Analysis

Should we use AAC or MO-AAC?

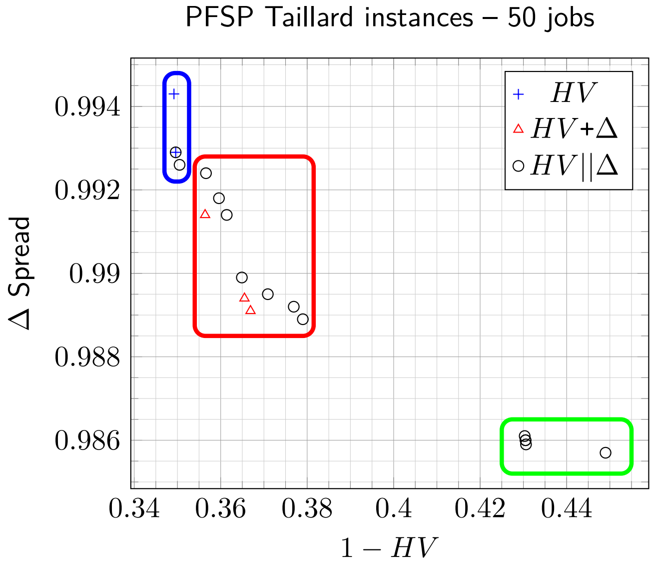
Compare 3 AAC Training Approaches

- MO-ParamILS ($HV \,||\, \Delta$)
- SO-ParamILS (HV)
- SO-ParamILS (HV+Δ) ($0.75HV + 0.25\Delta$)

Target Algorithm

- MOLS (2790 configurations)
  - 4 categorical parameters
  - 4 integer parameters

Test Performance



Interpretations

SO-ParamILS (HV)

- Disregards diversity entirely

SO-ParamILS (HV+Δ)

- Only in the aggregation direction
- Requires costly indicator normalisation

MO-ParamILS (HV || Δ)

- Wide, diverse covering

Wrap up

Perspectives

- Investigate other problems (*e.g.*, MO-TSP, MO-QAP)
- Extends to other algorithms (*e.g.*, EA)

Take-home Message

- Configuring a MO algorithm is a MO problem
- MO-ParamILS can design efficient MO metaheuristics
- Use multi-objective AAC!
  - No loss of performance
  - Way better diversity