Automatic Design of Multi-Objective Local Search Algorithms

Case Study on a bi-objective Permutation Flowshop Scheduling Problem

Aymeric Blot   Marie-Éléonore Kessaci   Laetitia Jourdan

Université de Lille, Inria, UMR 9189 – CRIStAL

GECCO – July, 2017

Context

Metaheuristics: Highly Tunable Algorithms

Key Points

- Approximation algorithms for optimisation problems
- Few assumptions about the problem
- Many parameters and strategies

Performance

- Differs with the problem
- Differs with the instance
- Depends on its configuration (set of parameter values)

Context

Automatic Algorithm Configuration

Single-Objective Configuration

- Learn the optimal configuration on a training instance set
- irace [López-Ibáñez *et al.*, 2016], ParamILS [Hutter *et al.*, 2009], SMAC [Hutter *et al.*, 2010], GGA++ [Ansótegui *et al.*, 2015], ...

Multi-Objective Configuration

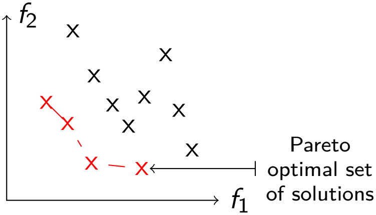
- MO-ParamILS [Blot *et al.*, LION 2016]
- Optimise multiple performance indicators

Context

Configuring Multi-Objective Algorithms

Performance Indicators

- Convergence
- Diversity
- Spread
- Size



Multi-Objective Configuration

- Configuring a MO algorithm is a MO problem [Blot *et al.*, EMO 2017]

Motivation

Designing Efficient Multi-Objective Metaheuristics

Question

- How efficient is MO-AAC to design MO metaheuristics?

Case Study

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

Case Study

MOLS: Multi-objective Local Search Algorithms

Key Points

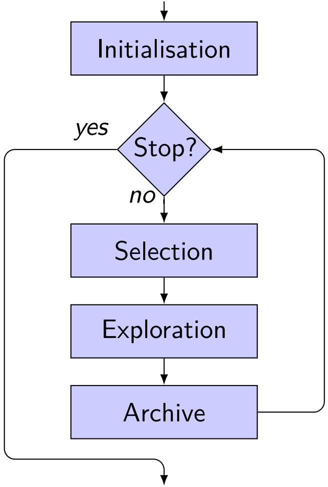
- Efficient metaheuristics
- Used on many problems (*e.g.*, scheduling, routing, assignment)
- Many strategies and parameters

In the Literature

- 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)

Multi-Objective Local Search Algorithms

Simple Workflow



Example Parameters

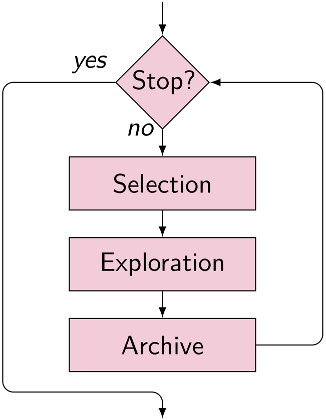
- Selection
  - Type and number of solutions
- Exploration
  - Neighbourhood
  - Reference point
  - Type and number of neighbours
- Archive
  - Archive size
  - Type of solutions

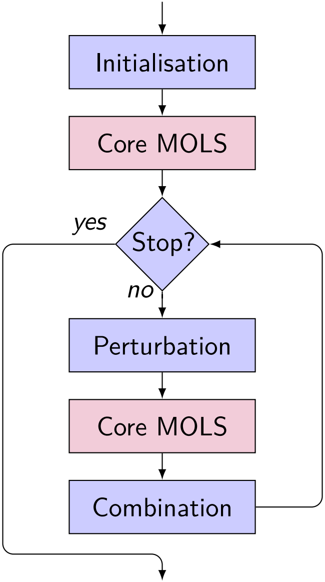
Multi-Objective Local Search Algorithms

Iterated Workflow

Iterated Local Search (ILS)

[Lourenço *et al.*, 2003]





MOLS Adaptability

Highly Configurable

- Initialisation
- Selection
- Exploration
- Archive
- Iteration
- Perturbation

Possible MOLS Parameters

Parameter	Type	Parameter values
initStrat	category	{rand, neh, ig}
selectStrat	category	{all, rand, newest, oldest}
selectSize	integer	1+
explorStrat	category	{all, imp, ndom, ...}
explorRef	category	{pick, arch}
explorSize	integer	1+
archiveStrat	category	{bounded, unbounded, ...}
archiveSize	integer	1+
iterationLength	integer	1+
iterationStagnation	integer	1+
perturbStrat	category	{restart, kick}
perturbSize	integer	1+
perturbStrength	integer	1+

Experimental Protocol

Goal: Compare MO-AAC Performance to Exhaustive Analysis

MO-ParamILS

- Training
- Validation
- Test

Exhaustive Analysis

- Every possible configuration
- Only on the test set

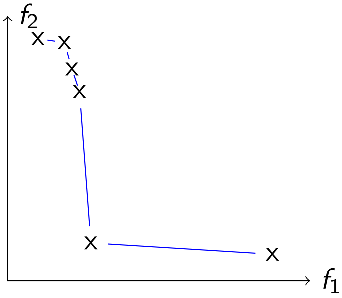
MOLS

- Optimise:
  - Convergence
  - Distribution
- Minimise:
  - 1 - Hypervolume
  - Δ Spread

MO-ParamILS

Training

Problem space



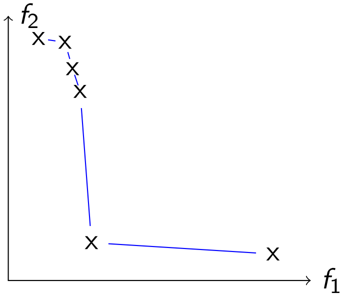
(execution on single instance)

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

MO-ParamILS

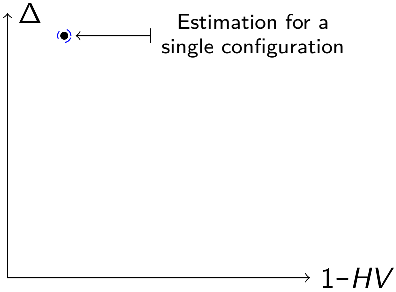
Training

Problem space



(execution on single instance)

Configuration space



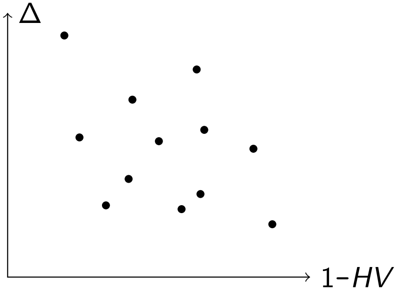
(average over training instances)

MO-ParamILS

Training

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

Configuration space



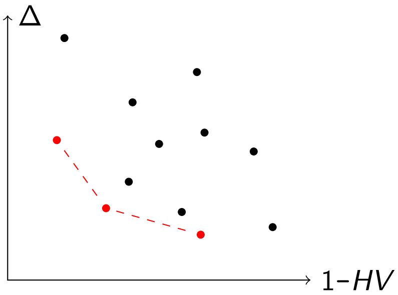
(average over training instances)

MO-ParamILS

Training

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

Configuration space



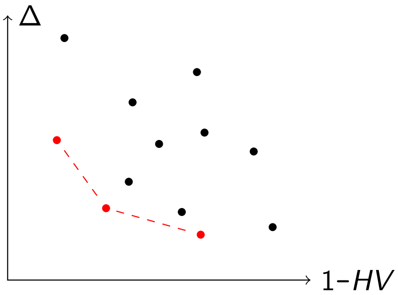
(average over training instances)

MO-ParamILS

Training

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

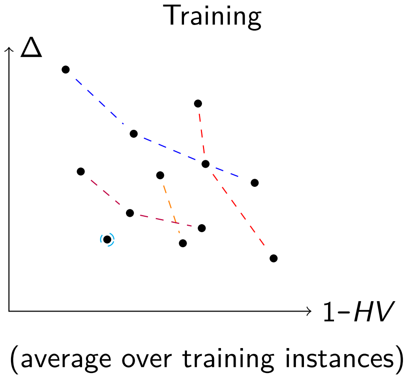
Configuration space



(average over training instances)

MO-ParamILS Protocol

Training, Validation, Test

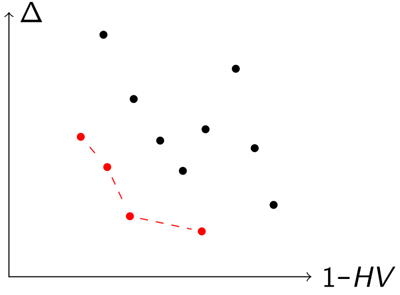


- Training
  - 30 training runs
  - different instance subsets
  - incomparable quality estimation
- Validation
  - all training configurations
  - all training instances

MO-ParamILS Protocol

Training, Validation, Test

Validation

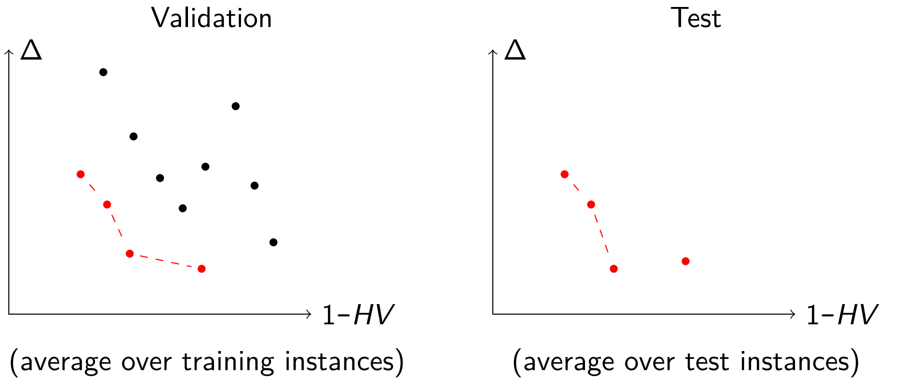


(average over training instances)

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

MO-ParamILS Protocol

Training, Validation, Test



Experimental Protocol

Summary

Permutation Flowshop Scheduling Problem

- Classical Taillard instances
- Bi-objective optimisation:
  - Makespan
  - Flowtime
- 3 scenarios:
  - 20-job instances
  - 50-job instances
  - 100-job instances

189 MOLS Configurations

Parameter	Parameter values
`initStrat`	{`rand`, `neh`, `ig`}
`selectStrat`	{`all`, `rand`, `new`, `old`}
`selectSize`	$\{1, 3\}$
`explorStrat`	{`all`, `imp`, `ndom`}
`explorRef`	{`pick`, `arch`}
`explorSize`	$\{1, 3\}$

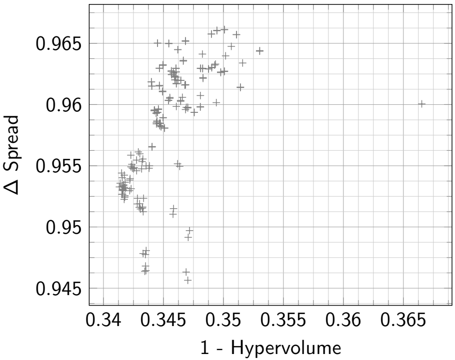
Total time

Approach	20-job	50-job	100-job
Training	36 min	90 min	3 hours
MO-AAC	2 days	7 days	15 days
Exhaustive	46 days	115 days	230 days

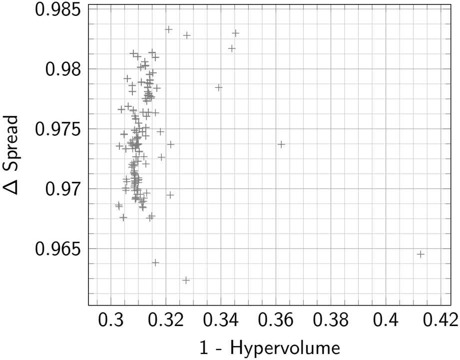
Results

Exhaustive analysis

PFSP Taillard instances – 50 jobs



PFSP Taillard instances – 100 jobs

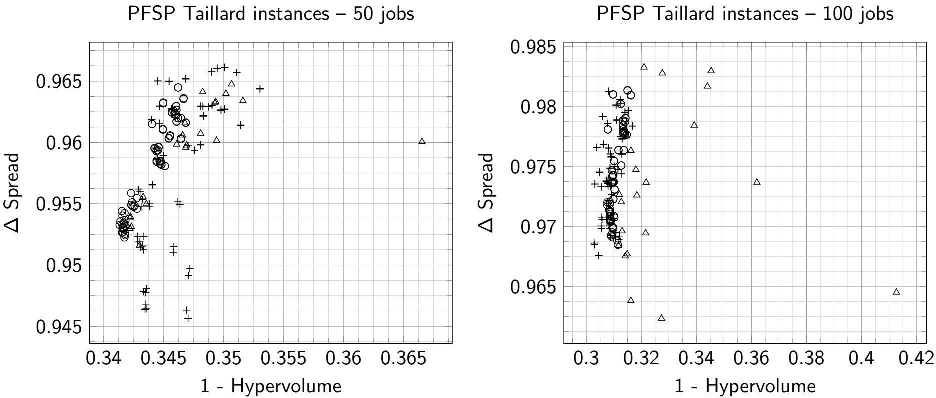


Exploration / Selection Strategies

- Δ all
- o imp
- + ndom
- all
- newest
- oldest
- rand

Results

Exhaustive analysis



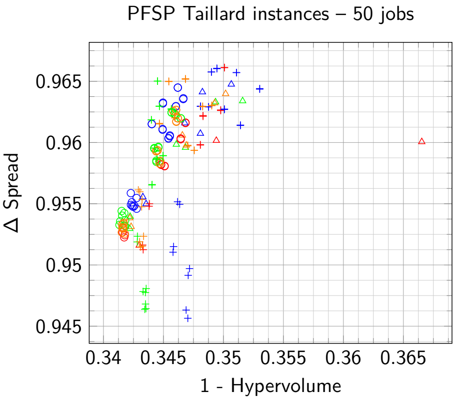
Exploration / Selection Strategies

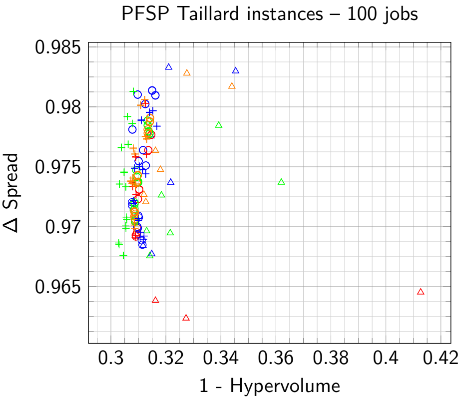
- Δ all
- o imp
- \+ ndom

- all
- newest
- oldest
- rand

Results

Exhaustive analysis



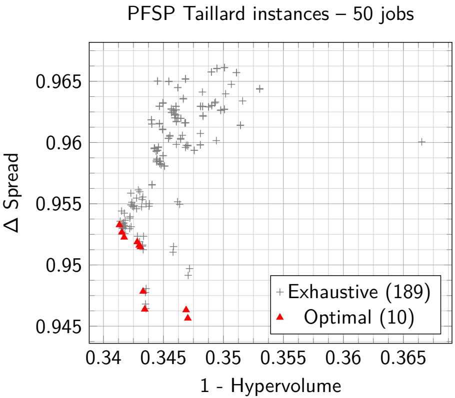


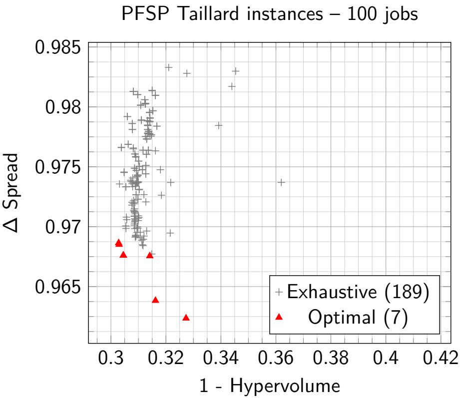
Exploration / Selection Strategies

- Δ all
- o imp
- + ndom
- ■ all
- ■ newest
- ■ oldest
- ■ rand

Results

Exhaustive analysis



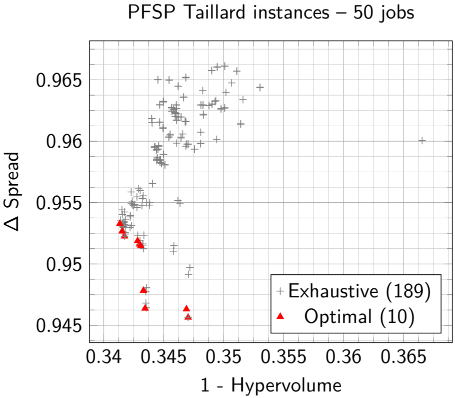


Exploration / Selection Strategies

- Δ all
- o imp
- + ndom
- all
- newest
- oldest
- rand

Results

Multi-Objective Automatic Design



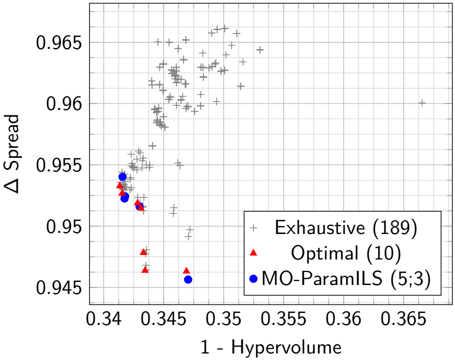
Optimal Configurations

Init	Selection		Exploration		
ig	oldest	3	imp	3	pick
ig	rand	3	imp	3	pick
ig	all	-	imp	1	arch
ig	newest	3	ndom	3	pick
ig	all	-	all	-	arch
ig	rand	1	ndom	1	arch
ig	newest	3	ndom	3	arch
ig	newest	3	ndom	1	arch
ig	oldest	3	ndom	1	arch
ig	oldest	3	ndom	3	arch

Results

Multi-Objective Automatic Design

PFSP Taillard instances – 50 jobs

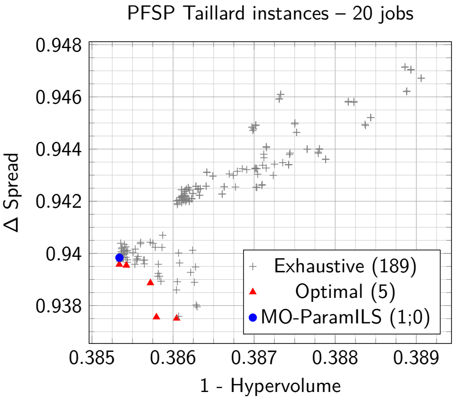


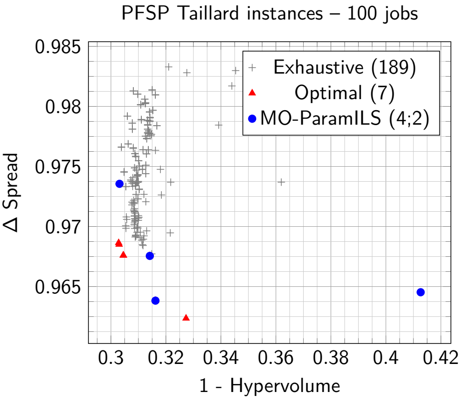
Optimal Configurations

Init	Selection		Exploration			●
ig	oldest	3	imp	3	pick	
ig	rand	3	imp	3	pick	
ig	all	-	imp	1	arch	✓
ig	newest	3	ndom	3	pick	
ig	all	-	all	-	arch	✓
ig	rand	1	ndom	1	arch	
ig	newest	3	ndom	3	arch	
ig	newest	3	ndom	1	arch	
ig	oldest	3	ndom	1	arch	
ig	oldest	3	ndom	3	arch	✓

Results

Multi-Objective Automatic Design





Conclusion

Main Contribution

- Efficient Automatic design of MOLS algorithms for bi-objective PFSP

Perspectives

- Investigate automatic design on other problems
  - *e.g.*, MO-TSP, MO-QAP
- Extends to other algorithms
  - *e.g.*, GA, EA

Additional Contribution

- New MOLS generalisation

Wrap up

Take-home Message 1

- Configuring a MO algorithm is a MO problem [Blot *et al.*, EMO 2017]

Take-home Message 2

- Use configurable algorithms
  - Or expose tunable parameters
- Propose new components and strategies
- Design the final algorithm automatically