MO-ParamILS

A Multi-objective Framework for Automatic Algorithm Configuration

Aymeric BLOT Holger H. HOOS Laetitia JOURDAN
Marie-Éléonore MARMION Heike TRAUTMANN

Université de Lille – CRIStAL – Inria Dolphin team, France
University of British Columbia, Canada
University of Münster, Germany

LION 10 – 31 May 2016

Context

Problematic: Parameter Setting

- Algorithms with many parameters
- Default configuration is not necessarily best!

IBM ILOG CPLEX Optimization Studio

- Commercial solver for mixed integer programming problems
- More than 70 performance parameters, $\approx 10^{46}$ configurations!

Automatic Algorithm Configuration

- How to deal with those parameters?
- How to find the best configuration?

Offline Configuration

Algorithm Configuration – Parameter Tuning

Given:

- Problem (*e.g.*, MIP, Knapsack, SAT)
- Set of training instances
- Performance objective
- Parameterised target algorithm (*e.g.*, CPLEX, GA)

Find best configuration, *i.e.*, most adequate set of parameters.

Statistical methods

- F-Race [Birattari *et al.*, 2009]
- irace [López-Ibáñez *et al.*, 2011]

Optimisation methods

- ParamILS [Hutter *et al.*, 2009]
- SMAC [Hutter *et al.*, 2011]
- GGA [Antosegui *et al.*, 2009]

Motivation

Target Algorithm Performance Assessment

Generally with regard to a single performance objective:

- Solution quality
- CPU time

Motivation

May want to use multiple performance objectives for comparing different configurations of the target algorithm.

Outline

ParamILS

Reference

Hutter, Hoos, Leyton-Brown, Stützle (2009)

Why ParamILS?

- Prominent, state-of-the-art, general-purpose automated algorithm configurator
- Many successful applications
- Deals with very large configuration spaces
- Part of ACLib [Hutter *et al.*, 2014]

ParamILS

Principles

- Model-free search procedure
- Iterated local search (ILS) [Louranço *et al.*, 2003]

ParamILS

- Single-objective optimisation
- Input
  - Set of problem instances
  - Target algorithm
  - Configuration space
- Output: best configuration found

ParamILS

General framework

```
best_config ← init();
until termination criterion met do
    config ← perturb(best_config);
    config ← local_search(config);
    best_config ← accept(config, best_config);
return best_config;
```

ParamILS

Initialisation

Best of:

- Default or hand-picked configurations
- $r = 10$ random configurations

Perturbation

- After the first local search descent
- $s = 3$ random one-exchange moves

Neighbourhood: One-exchange

Two configurations are neighbours if and only if they differ by a single parameter value.

ParamILS

Local Search

- Exploration
  - Neutrality-based Hillclimbing
  - Stops on better or equal neighbours
- Tabu list
  - Unbounded
  - All visited configurations
- Stops if all neighbours are worse or tabu

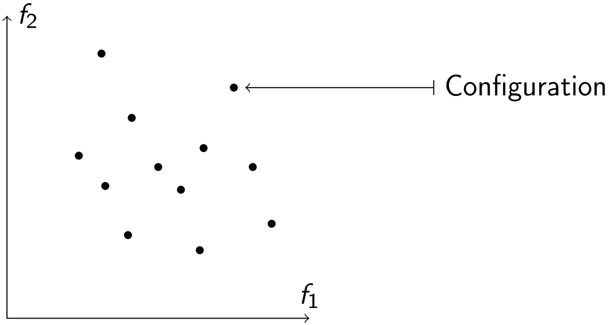
Acceptance Criterion

- Accept better of two given configurations

Multi-objective Optimisation

Pareto Dominance – Minimisation

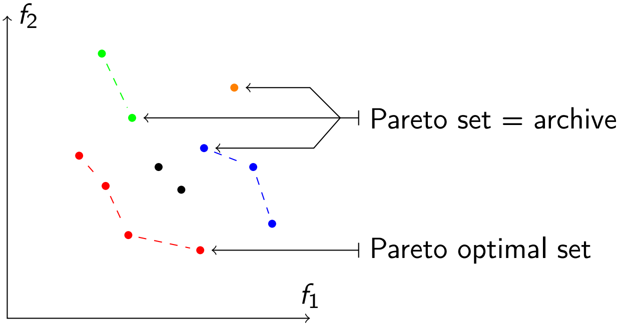
$$x \prec y \iff \begin{cases} \forall i \in \{1, \ldots, n\} : & c_i(x) \leq c_i(y) \\ \exists i \in \{1, \ldots, n\} : & c_i(x) < c_i(y) \end{cases}$$



Multi-objective Optimisation

Pareto Dominance – Minimisation

$$x \prec y \iff \begin{cases} \forall i \in \{1, \ldots, n\} : \; c_i(x) \leq c_i(y) \\ \exists i \in \{1, \ldots, n\} : \; c_i(x) < c_i(y) \end{cases}$$



From ParamILS to MO-ParamILS

ParamILS

- Single-objective optimisation
- Input
  - Set of problem instances
  - Target algorithm
  - Configuration space
- Output: best configuration found

MO-ParamILS

- Multi-objective optimisation
- Input
- Output: Pareto set of the best configurations found

MO-ParamILS

General framework

```
best_arch ← init();
until termination criterion met do
    arch ← mo_perturb(best_arch);
    arch ← mo_local_search(arch);
    best_arch ← archive(arch, best_arch);
return best_arch;
```

MO-ParamILS

Initialisation

Best of:

- Default or hand-picked configurations
- $r = 10$ random configurations

Perturbation

- After the first local search descent
- Select a single configuration from the current archive
- $s = 3$ random one-exchange moves

Neighbourhood: One-exchange

Two configurations are neighbours if and only if they differ by a single parameter value.

MO-ParamILS

Multi-objective Local Search

- Selection
  - All current configurations are explored
- Exploration
  - Dominance-based Hillclimbing
  - Stops on dominating neighbours
  - Keeps non-dominated neighbours
- Tabu list
- Stops if all neighbours are worse or tabu

Acceptance Criterion

- Archive new configurations

MO-BasicILS, MO-FocusedILS

BasicILS – MO-BasicILS

- Evaluate on fixed subset of N random training instances

Issues of BasicILS

- Need to fix N
  - N too high: wasted time on poor configurations
  - N too low: imprecise evaluation on good configurations

FocusedILS – MO-FocusedILS

- Evaluate on increasingly large parts of training set
- Domination and intensification mechanisms

Experimental Protocol

Algorithms

- Default configuration
- FocusedILS (aggregation)
- MO-BasicILS
- MO-FocusedILS

Machine learning

- Training set
- Disjoint validation set

Scenarios

	Dataset	Algorithm	Training	Performance objectives
S1	Regions200	CPLEX (MIP)	1 day	[Quality, Cutoff]
S2	Regions200	CPLEX	1 day	[Quality, CPU time]
S3	CORLAT	CPLEX	1 day	[Quality, Cutoff]
S4	CORLAT	CPLEX	1 day	[Quality, CPU time]
S5	QUEENS	CLASP (SAT)	1 day	[CPU time, Memory usage]

Results

Minimisation of hypervolume (top) and ε-indicator values (bottom)

Approach	S1	S2	S3	S4	S5
Default	2.43e-01	3.57e-01	2.70e-01	5.30e-01	1.08e+00
FocusedILS	3.82e-02	5.82e-02	3.35e-01	1.72e-01	3.04e-02
MO-BasicILS	2.46e-03	5.41e-02	5.53e-02	1.02e-01	5.49e-02
MO-FocusedILS	9.02e-03	2.07e-03	2.37e-02	7.63e-04	1.57e-02
Default	2.22e-01	2.69e-01	2.33e-01	3.90e-01	1.00e+00
FocusedILS	5.77e-02	1.38e-02	3.33e-01	1.42e-01	6.52e-02
MO-BasicILS	1.80e-02	1.71e-01	1.11e-01	1.48e-01	8.35e-02
MO-FocusedILS	1.44e-02	9.05e-03	9.00e-02	8.06e-04	2.64e-02

MO-ParamILS > FocusedILS > Default

MO-FocusedILS > MO-BasicILS

Conclusion and Future Work

MO-ParamILS

- Efficient, general-purpose, *multi-objective* algorithm configurator

Future Work

- Compare to other multi-objective configurators
  - SPRINT-Race [Zhang *et al.*, 2015]
  - SMAC [Hutter *et al.*, 2011] $\rightarrow$ MO-SMAC
- Test MO-ParamILS on multi-objective target algorithms
- Distinguish symbolical and numerical parameters in ParamILS

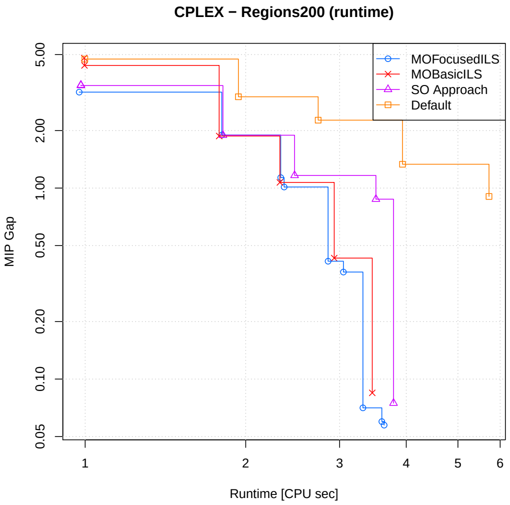
Example

CPLEX

- MIP solver
- 74 params

Regions200

- Actions
- 200 goods
- 1000 bids



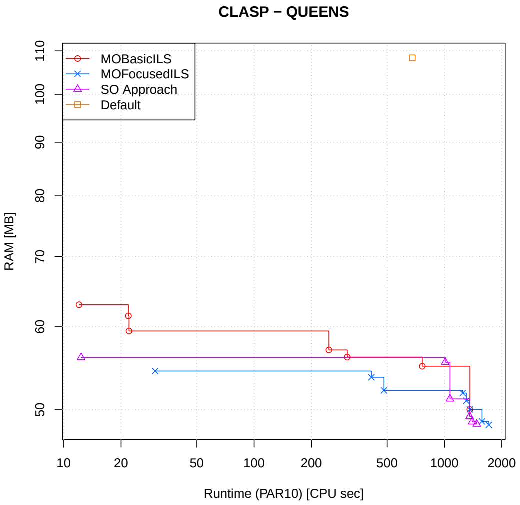
Example

CLASP

- ASP/SAT solver
- 73 params

QUEENS

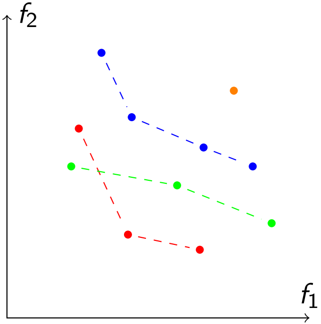
- n-queens
- $n \in \{10 \ldots 50\}$



Methodology

Suggested protocol

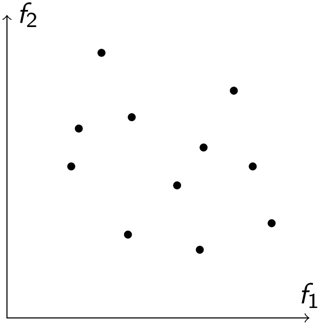
1. Train multiple times
2. Select everything
3. Validate on the training set
4. Select the Pareto set
5. Validate on the validation set



Methodology

Suggested protocol

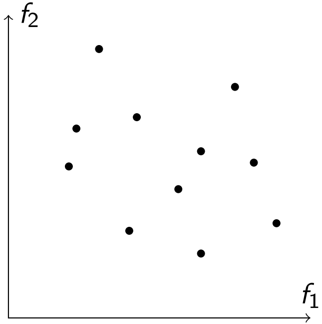
1. Train multiple times
2. Select everything
3. Validate on the training set
4. Select the Pareto set
5. Validate on the validation set



Methodology

Suggested protocol

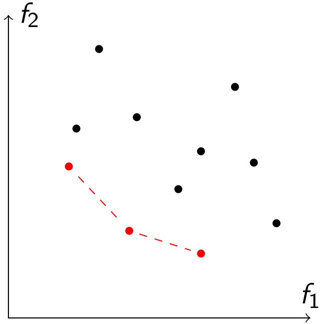
1. Train multiple times
2. Select everything
3. Validate on the training set
4. Select the Pareto set
5. Validate on the validation set



Methodology

Suggested protocol

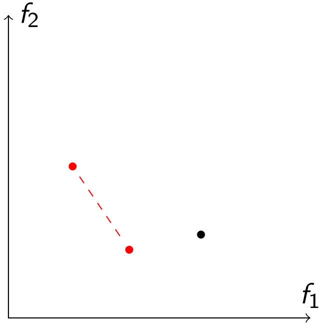
1. Train multiple times
2. Select everything
3. Validate on the training set
4. Select the Pareto set
5. Validate on the validation set



Methodology

Suggested protocol

1. Train multiple times
2. Select everything
3. Validate on the training set
4. Select the Pareto set
5. Validate on the validation set



ParamILS

General framework

```
best_config ← init();
until termination criterion met do
    config, best_config ← perturb(best_config);
    config ← local_search(config);
    best_config ← accept(config, best_config);
return incumbent;
```