Automatic (Offline) Configuration & Design of Optimisation Algorithms

Manuel López-Ibáñez
manuel.lopez-ibanez@manchester.ac.uk

University of Manchester

October 27, 2015
CREST Open Workshop, London

MANCHESTER 1824
The University of Manchester
Alliance Manchester Business School

What do we talk when we talk about hyper-heuristics?

- Online tuning / Parameter control, (self-)adaptation / Reactive search / Adaptive Selection
- Algorithm selection / Algorithm portfolios
- Offline tuning / Parameter configuration

Design choices and parameters everywhere

Modern high-performance optimisers involve a large number of design choices and parameter settings

- *categorical* parameters

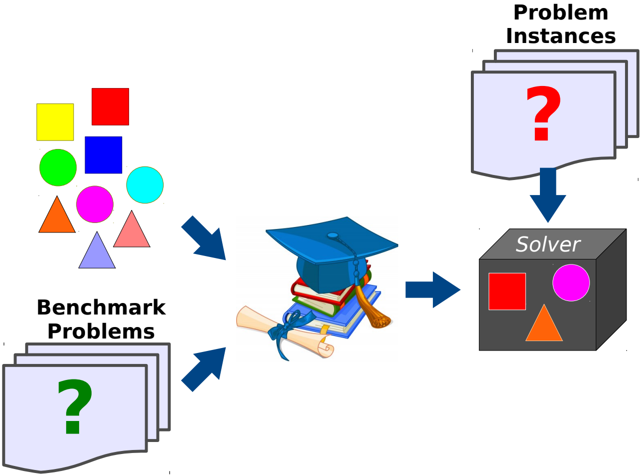
  recombination $\in$ { uniform, one-point, two-point }
  localsearch $\in$ { tabu search, SA, ILS }
- *ordinal* parameters

  neighborhoods $\in$ { small, medium, large }
- *numerical* parameters
  - weighting factors, population sizes, temperature, hidden constants, ...
- Parameters may be *conditional* to specific values of other parameters: `--temperature` if `LS == "SA"`

Configuring algorithms involves setting categorical, ordinal and numerical parameters

Manual tuning

Human expert + trial-and-error/statistics



Towards more systematic approaches

Traditional approaches

- Trial–and–error design guided by expertise/intuition
  - ✘ prone to over-generalizations,
  - ✘ limited exploration of design alternatives,
  - ✘ human biases
- Guided by theoretical studies
  - ✘ often based on over-simplifications,
  - ✘ specific assumptions,
  - ✘ few parameters

Can we make this approach more principled and automatic?

Automatic Algorithm Configuration

1. Find the best parameter configuration of a solver from a set of *training problem instances*
2. Repeatedly use this configuration to solve *unseen problem instances* of the same problem
3. Performance measured over test ($\neq$ training) instances

A problem with many names:

offline parameter tuning, automatic algorithm configuration, automatic algorithm design, hyper-parameter tuning, hyper-heuristics, meta-optimisation, programming by optimisation, ...

Offline configuration and online parameter control

Offline tuning / Algorithm configuration

- Learn best parameters before *solving* an instance
- Configuration done on training instances
- Performance measured over test ($\neq$ training) instances

Online tuning / Parameter control / Reactive search

- Learn parameters *while* solving an instance
- No training phase
- Limited to very few crucial parameters

All online methods have parameters that are configured offline

Automatic Algorithm Configuration: How?

A stochastic black-box optimisation problem

- Mixed decision variables:

  discrete (categorical, ordinal, integer) and continuous

- Stochasticity from algorithm and problem instances
- Black-box: evaluation requires running the algorithm

Methods for Automatic Algorithm Configuration

- **SPO** [Bartz-Beielstein, Lasarczyk & Preuss, 2005]
- **ParamILS** [Hutter, Hoos & Stützle, 2007]
- **GGA** [Ansótegui, Sellmann & Tierney, 2009]
- **SMAC** [Hutter, Hoos & Leyton-Brown, 2011]
- **IRACE** [López-Ibáñez, Dubois-Lacoste, Stützle & Birattari, 2011]

Automatic Algorithm Configuration: How?

- Complex parameter spaces:
  numerical, categorical, ordinal, subordinate (conditional)
- Large parameter spaces (hundreds of parameters)
- Heterogeneous instances
- Medium to large tuning budgets
  (few hundred to thousands of runs)
- Individual runs require from seconds to hours
- Multi-core CPUs, MPI, Grid-Engine clusters

☞ Modern automatic configuration tools are general, flexible, powerful and easy to use

Automatic Algorithm Configuration: Applications

- *Parameter tuning*
  - Exact MIP solvers (CPLEX, SCIP)
  - single-objective optimisation metaheuristics
  - multi-objective optimisation metaheuristics
  - anytime algorithms (improve time-quality trade-offs)
- *Automatic algorithm design*
  - *From a flexible framework of algorithm components*
  - From a grammar description
- Machine learning
  - Automatic model selection for high-dimensional survival analysis [Lang et al., 2014]
  - Hyperparameter tuning in **mlr** R package [Bischl et al., 2014]
- Automatic design of control software for robots [Francesca et al., 2015]

Parameter tuning

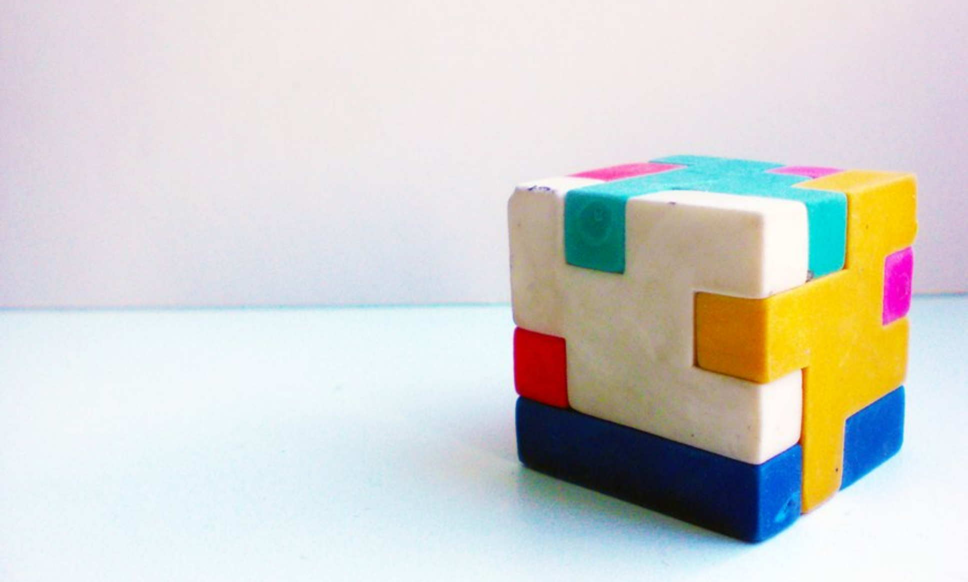
> “*For procedures that require parameter tuning, the available data must be partitioned into a training and a test set. Tuning should be performed in the training set only.*”
>
> [Journal of Heuristics: Policies on Heuristic Search Research]

- Essential tool when developing and comparing algorithms:
- First tune, then analyse
- Comparing with untuned algorithms is *always* unfair

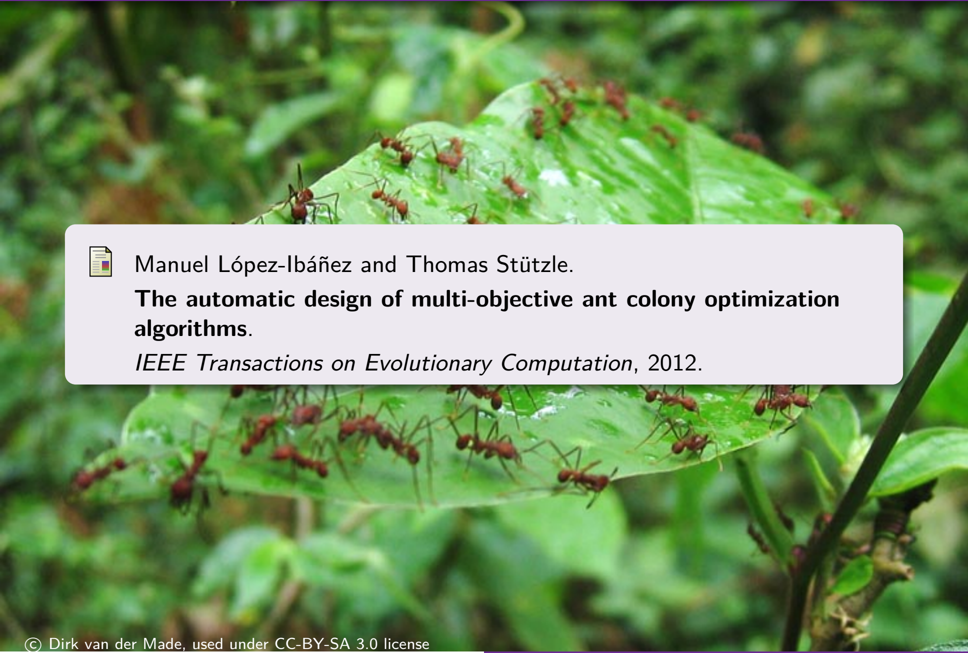
Automatic Design: Monolithic view

Normally, optimisation algorithms are viewed as this . . .

. . . but we prefer this view



Automatic Design of MOACO algorithms



Manuel López-Ibáñez and Thomas Stützle.
The automatic design of multi-objective ant colony optimization algorithms.
IEEE Transactions on Evolutionary Computation, 2012.

MOACO Algorithms

- Multiple objective Ant-Q (MOAQ) [Mariano & Morales, 1999] [García-Martínez et al., 2007]
- MACS-VRPTW [Gambardella et al., 1999]
- BicriterionAnt [Iredi et al., 2001]
- SACO [T'Kindt et al., 2002]
- Multiobjective Network ACO [Cardoso et al., 2003]
- Multicriteria Population-based ACO [Guntsch & Middendorf, 2003]
- MACS [Barán & Schaerer, 2003]
- COMPETants [Doerner et al., 2003]
- Pareto ACO [Doerner et al., 2004]
- Multiple Objective ACO Metaheuristic [Gravel et al., 2002]
- MOACO-bQAP [López-Ibáñez et al., 2004]
- MOACO-ALBP [Baykasoglu et al., 2005]
- mACO-$\{1, 2, 3, 4\}$ [Alaya et al., 2007]
- Population-based ACO [Angus, 2007]

A flexible MOACO framework

- High-level design independent from the problem
  ⇒ Easy to extend to new problems
- Instantiates 9 MOACO algorithms from the literature
- Multi-objective algorithmic design: 10 parameters
- Hundreds of potential ~~papers~~ algorithm designs
- *Underlying ACO settings* are also configurable
- Implemented for bi-objective TSP and bi-objective Knapsack

Automatic Design: Results

Tuning done with **irace** (1 000 runs)

Worst	Best MOACO of literature + default ACO settings
	Tuned MOACO design + default ACO settings
	Best MOACO of literature + *tuned* ACO settings
	Tuned MOACO design + *tuned* ACO settings
Best	*Tuned* (MOACO design + ACO settings)

Automatic Design: Summary

- ~~We propose a new MOACO algorithm that. . .~~

- We propose an approach to automatically design MOACO algorithms:
  1. Synthesize state-of-the-art knowledge into a flexible MOACO framework
  2. Explore the space of potential designs automatically using irace

- Other examples:
  - Single-objective solvers for MIP: CPLEX, SCIP
  - Single-objective algorithmic framework for SAT: SATenstein *[KhudaBukhsh, Xu, Hoos & Leyton-Brown, 2009]*
  - Multi-objective framework for PFSP, TP+PLS *[Dubois-Lacoste, López-Ibáñez & Stützle, 2011]*
  - Multi-objective Evolutionary Algorithms (MOEAs) *[Bezerra, López-Ibáñez & Stützle, 2015]*

Automatic Design of Algorithms: The End of the Game

> “*The Journal of Heuristics does not endorse the up-the-wall game.* [Policies on Heuristic Search Research]”

> “*True innovation in metaheuristics research therefore does not come from yet another method that performs better than its competitors, certainly if it is not well understood why exactly this method performs well.* [Sörensen, 2013]”

- Finding a state-of-the-art algorithm is “easy”:

  problem modeling + algorithmic components + computing power

- *What* novel components? *Why* they work? *When* they work?

Conclusions

- Hyper-heuristics is a cool name, but not very explanatory
- Interesting works that may fall within Hyper-heuristics, but not called as such
- Automatic algorithm configuration is working today: Use it!
- Automatic design will be the end of the up-the-wall game
- Paradigm shift in optimisation research:

From monolithic algorithms
to flexible frameworks of algorithmic components

Automatic (Offline) Configuration & Design of Optimisation Algorithms

Manuel López-Ibáñez
manuel.lopez-ibanez@manchester.ac.uk

University of Manchester

October 27, 2015
CREST Open Workshop, London

http://iridia.ulb.ac.be/irace

MANCHESTER 1824
The University of Manchester
Alliance Manchester Business School

References I

I. Alaya, C. Solnon, and K. Ghédira. Ant colony optimization for multi-objective optimization problems. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 1, pages 450–457. IEEE Computer Society Press, Los Alamitos, CA, 2007.

D. Angus. Population-based ant colony optimisation for multi-objective function optimisation. In M. Randall, H. A. Abbass, and J. Wiles, editors, *Progress in Artificial Life (ACAL)*, volume 4828 of *Lecture Notes in Computer Science*, pages 232–244. Springer, Heidelberg, Germany, 2007. doi: 10.1007/978-3-540-76931-6_21.

C. Ansótegui, M. Sellmann, and K. Tierney. A gender-based genetic algorithm for the automatic configuration of algorithms. In I. P. Gent, editor, *Principles and Practice of Constraint Programming, CP 2009*, volume 5732 of *Lecture Notes in Computer Science*, pages 142–157. Springer, Heidelberg, Germany, 2009. doi: 10.1007/978-3-642-04244-7_14.

B. Barán and M. Schaerer. A multiobjective ant colony system for vehicle routing problem with time windows. In *Proceedings of the Twenty-first IASTED International Conference on Applied Informatics*, pages 97–102, Insbruck, Austria, 2003.

T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss. Sequential parameter optimization. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pages 773–780, Piscataway, NJ, Sept. 2005. IEEE Press.

A. Baykasoglu, T. Dereli, and I. Sabuncu. A multiple objective ant colony optimization approach to assembly line balancing problems. In *35th International Conference on Computers and Industrial Engineering (CIE35)*, pages 263–268, Istanbul, Turkey, 2005.

L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle. Automatic component-wise design of multi-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2015. doi: 10.1109/TEVC.2015.2474158.

P. Cardoso, M. Jesus, and A. Marquez. MONACO: multi-objective network optimisation based on an ACO. In *Proc. X Encuentros de Geometría Computacional*, Seville, Spain, 2003.

K. F. Doerner, R. F. Hartl, and M. Reimann. Are COMPETants more competent for problem solving? The case of a multiple objective transportation problem. *Central European Journal for Operations Research and Economics*, 11(2):115–141, 2003.

References II

K. F. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, 131:79–99, 2004.

J. Dubois-Lacoste, M. López-Ibáñez, and T. Stützle. Automatic configuration of state-of-the-art multi-objective optimizers using the TP+PLS framework. In N. Krasnogor and P. L. Lanzi, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2011*, pages 2019–2026. ACM Press, New York, NY, 2011. ISBN 978-1-4503-0557-0. doi: 10.1145/2001576.2001847.

G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pinciroli, F. Mascia, V. Trianni, and M. Birattari. AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 2015. doi: 10.1007/s11721-015-0107-9.

L. M. Gambardella, É. D. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 63–76. McGraw Hill, London, UK, 1999.

C. García-Martínez, O. Cordón, and F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, 180(1): 116–148, 2007.

M. Gravel, W. L. Price, and C. Gagné. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, 143(1):218–229, 2002. doi: 10.1016/S0377-2217(01)00329-0.

M. Guntsch and M. Middendorf. Solving multi-objective permutation problems with population based ACO. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-criterion Optimization, EMO 2003*, volume 2632 of *Lecture Notes in Computer Science*, pages 464–478. Springer, Heidelberg, Germany, 2003.

F. Hutter, H. H. Hoos, and T. Stützle. Automatic algorithm configuration based on local search. In *Proc. of the Twenty-Second Conference on Artifical Intelligence (AAAI '07)*, pages 1152–1157. AAAI Press/MIT Press, Menlo Park, CA, 2007.

References III

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In C. A. Coello Coello, editor, *Learning and Intelligent Optimization, 5th International Conference, LION 5*, volume 6683 of *Lecture Notes in Computer Science*, pages 507–523. Springer, Heidelberg, Germany, 2011.

S. Iredi, D. Merkle, and M. Middendorf. Bi-criterion optimization with multi colony ant algorithms. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Evolutionary Multi-criterion Optimization, EMO 2001*, volume 1993 of *Lecture Notes in Computer Science*, pages 359–372. Springer, Heidelberg, Germany, 2001.

A. R. KhudaBukhsh, L. Xu, H. H. Hoos, and K. Leyton-Brown. SATenstein: Automatically building local search SAT solvers from components. In C. Boutilier, editor, *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 517–524. AAAI Press, Menlo Park, CA, 2009.

M. Lang, H. Kotthaus, Marwedel, C. Weihs, J. Rahnenführer, and B. Bischl. Automatic model selection for high-dimensional survival analysis. *Journal of Statistical Computation and Simulation*, 2014. doi: 10.1080/00949655.2014.929131.

M. López-Ibáñez, L. Paquete, and T. Stützle. On the design of ACO for the biobjective quadratic assignment problem. In M. Dorigo et al., editors, *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004*, volume 3172 of *Lecture Notes in Computer Science*, pages 214–225. Springer, Heidelberg, Germany, 2004. doi: 10.1007/978-3-540-28646-2_19.

M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari. The irace package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011. URL http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-004.pdf.

C. E. Mariano and E. Morales. MOAQ: An Ant-Q algorithm for multiple objective optimization problems. In W. Banzhaf, J. M. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. J. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 1999*, pages 894–901. Morgan Kaufmann Publishers, San Francisco, CA, 1999.

K. Sörensen. Metaheuristics - the metaphor exposed. *International Transactions in Operational Research*, 2013. doi: 10.1111/itor.12001.

V. T'Kindt, N. Monmarché, F. Tercinet, and D. Laügt. An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research*, 142(2):250–257, 2002.