Doing a PhD in AI: a Case Study

Frank Hutter

Postdoctoral Fellow
Department of Computer Science
University of British Columbia
Vancouver, Canada

PhD thesis (September 2009):
Automated Configuration of Algorithms
for Solving Hard Computational Problems

PhD supervisors:
Holger Hoos, Kevin Leyton-Brown & Kevin Murphy

AI is driven by applications

- AI is everywhere
  - AI in space: *e.g.*, Mars rovers
  - AI in homes: *e.g.*,automatic vacuum cleaners
  - AI in mobile devices: *e.g.*,face detection in digital cameras
  - AI in industry: *e.g.*, software and hardware verification
  - ...

AI is driven by applications

- AI is everywhere
  - AI in space: *e.g.*, Mars rovers
  - AI in homes: *e.g.*,automatic vacuum cleaners
  - AI in mobile devices: *e.g.*,face detection in digital cameras
  - AI in industry: *e.g.*, software and hardware verification
  - ...
- Gap between theory and practice
  - E.g. "NP-hard $\rightarrow$ hopeless"
  - But we solve SAT-encoded verification instances with 100000s of variables in seconds

AI is driven by applications

- AI is everywhere
  - AI in space: *e.g.*, Mars rovers
  - AI in homes: *e.g.*,automatic vacuum cleaners
  - AI in mobile devices: *e.g.*,face detection in digital cameras
  - AI in industry: *e.g.*, software and hardware verification
  - ...
- Gap between theory and practice
  - E.g. "NP-hard $\rightarrow$ hopeless"
  - But we solve SAT-encoded verification instances with 100000s of variables in seconds
- Need good research in theory
  - Average case analysis
  - Identify tractable subclasses
  - Approximation algorithms

Few theoretical results $\rightarrow$ need solid experiments

- Almost every AI paper has some empirical component to it
  - SAT solving, machine learning, NLP, computer vision, you name it

Few theoretical results $\rightarrow$ need solid experiments

- Almost every AI paper has some empirical component to it
  - SAT solving, machine learning, NLP, computer vision, you name it
- We are not well prepared for this
  - Course in empirical methods/how to do experiments?

Few theoretical results $\rightarrow$ need solid experiments

- Almost every AI paper has some empirical component to it
  - SAT solving, machine learning, NLP, computer vision, you name it
- We are not well prepared for this
  - Course in empirical methods/how to do experiments?
  - Learning by doing/indirect feedback from reviewers
  - Very different from e.g. life sciences

Few theoretical results → need solid experiments

- Almost every AI paper has some empirical component to it
  - SAT solving, machine learning, NLP, computer vision, you name it
- We are not well prepared for this
  - Course in empirical methods/how to do experiments?
  - Learning by doing/indirect feedback from reviewers
  - Very different from e.g. life sciences
  - Paul Cohen's book: Empirical Methods for Artificial Intelligence

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm
  - Report *test set* performance instead
  - Watch out for *overtuning* to a benchmark

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm
  - Report *test set* performance instead
  - Watch out for *overtuning* to a benchmark
- Doing a few experiments and reporting their mean

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm
  - Report *test set* performance instead
  - Watch out for *overtuning* to a benchmark
- Doing a few experiments and reporting their mean
  - Do enough experiments to gain confidence in your results
  - Also report measures of variation (stddev/quantiles)
  - Perform statistical tests

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm
  - Report *test set* performance instead
  - Watch out for *overtuning* to a benchmark
- Doing a few experiments and reporting their mean
  - Do enough experiments to gain confidence in your results
  - Also report measures of variation (stddev/quantiles)
  - Perform statistical tests
- "I didn't have time to do more experiments"

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm
  - Report *test set* performance instead
  - Watch out for *overtuning* to a benchmark
- Doing a few experiments and reporting their mean
  - Do enough experiments to gain confidence in your results
  - Also report measures of variation (stddev/quantiles)
  - Perform statistical tests
- "I didn't have time to do more experiments"
  - Automate the experimental setup
    (so it's just CPU time, not your time)

Simple problems to avoid

- Reporting results on problem instances used to develop algorithm
  - Report *test set* performance instead
  - Watch out for *overtuning* to a benchmark
- Doing a few experiments and reporting their mean
  - Do enough experiments to gain confidence in your results
  - Also report measures of variation (stddev/quantiles)
  - Perform statistical tests
- "I didn't have time to do more experiments"
  - Automate the experimental setup
    (so it's just CPU time, not your time)
  - Run it on a compute cluster
    (ask your supervisor, it should be free)

Once you have automated your experiments

- Easy to study variants of the algorithm
  - Does performance improve when you change an algorithm component?
  - Does performance improve with different parameter settings?

Once you have automated your experiments

- Easy to study variants of the algorithm
  - Does performance improve when you change an algorithm component?
  - Does performance improve with different parameter settings?
  - Computer can try out many possible variants

Once you have automated your experiments

- Easy to study variants of the algorithm
  - Does performance improve when you change an algorithm component?
  - Does performance improve with different parameter settings?
  - Computer can try out many possible variants
  - $\rightsquigarrow$ My thesis topic

Outline

1. My PhD in a nutshell

2. Some general points

Motivation for my PhD thesis

Most algorithms have parameters

- Decisions that are left open during algorithm design
  - Numerical parameters (*e.g.*, real-valued thresholds)
  - Categorical parameters (*e.g.*, which heuristic to use)

Motivation for my PhD thesis

Most algorithms have parameters

- Decisions that are left open during algorithm design
  - Numerical parameters (*e.g.*, real-valued thresholds)
  - Categorical parameters (*e.g.*, which heuristic to use)
- Set to maximize empirical performance

Motivation for my PhD thesis

Most algorithms have parameters

- Decisions that are left open during algorithm design
  - Numerical parameters (*e.g.*, real-valued thresholds)
  - Categorical parameters (*e.g.*, which heuristic to use)
- Set to maximize empirical performance
- Can we use AI techniques to set these parameters?

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...
  - 26 user-specifiable parameters:
    10 categorical, 12 continuous, 4 integer parameters

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...
  - 26 user-specifiable parameters:
    10 categorical, 12 continuous, 4 integer parameters

- Minimize expected run-time

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...
  - 26 user-specifiable parameters:
    10 categorical, 12 continuous, 4 integer parameters
- Minimize expected run-time
- Problems:
  - Good performance on a few instances does not generalise well
  - Many possible configurations (8.34×10^{17} after discretization)

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...
  - 26 user-specifiable parameters:
    10 categorical, 12 continuous, 4 integer parameters

- Minimize expected run-time

- Problems:
  - Good performance on a few instances does not generalise well
  - Many possible configurations (8.34×10^{17} after discretization)

- High-dimensional discrete optimization problem
  - Often performed *manually*

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...
  - 26 user-specifiable parameters:
    10 categorical, 12 continuous, 4 integer parameters

- Minimize expected run-time

- Problems:
  - Good performance on a few instances does not generalise well
  - Many possible configurations (8.34×10^{17} after discretization)

- High-dimensional discrete optimization problem
  - Often performed *manually*
  - **Humans are not good at this!**

Real-world example for algorithm configuration: Tree search for SAT-encoded software verification

- New DPLL-type SAT solver (SPEAR)
  - Variable/value heuristics, clause learning, restarts, ...
  - 26 user-specifiable parameters:
    10 categorical, 12 continuous, 4 integer parameters

- Minimize expected run-time

- Problems:
  - Good performance on a few instances does not generalise well
  - Many possible configurations (8.34×10^{17} after discretization)

- High-dimensional discrete optimization problem
  - Often performed *manually*
  - **Humans are not good at this!**

  $\rightsquigarrow$ Automate!

Outline

1. My PhD in a nutshell
    - Iterated Local Search in Configuration Space
    - Sequential Model-Based Search in Configuration Space

2. Some general points

Simple manual approach for configuration

Start with some parameter configuration;

Simple manual approach for configuration

Start with some parameter configuration;

Modify a single parameter;

Simple manual approach for configuration

Start with some parameter configuration;

Modify a single parameter;
if *results on benchmark set improve* **then**
  keep new configuration;

Simple manual approach for configuration

Start with some parameter configuration;
repeat
    Modify a single parameter;
    if *results on benchmark set improve* **then**
        keep new configuration;
until *no more improvement possible (or "good enough");*

Simple manual approach for configuration

Start with some parameter configuration;
repeat
    Modify a single parameter;
    if *results on benchmark set improve* **then**
        keep new configuration;
until *no more improvement possible (or "good enough");*

$\rightsquigarrow$ Manually-executed **local search**

Simple manual approach for configuration

Start with some parameter configuration;
repeat
  Modify a single parameter;
  if *results on benchmark set improve* **then**
    keep new configuration;
until *no more improvement possible (or "good enough");*

$\rightsquigarrow$ Manually-executed **local search**

ParamILS [Hutter, Hoos & Stützle, AAAI '07]:
Iterated local search: biased random walk over local optima

Configuration of Spear for Verification

[Hutter, Babić, Hoos & Hu; FMCAD '07]

- SPEAR, tree search solver for industrial SAT instances
  - Developed by Domagoj Babić at UBC
  - 26 parameters, 8.34×10^{17} configurations
  - Competitive with the state of the art

Configuration of Spear for Verification

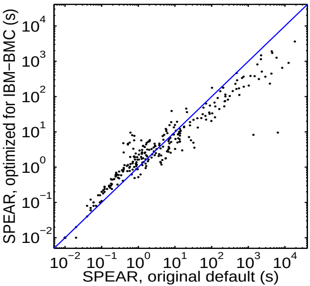
[Hutter, Babić, Hoos & Hu; FMCAD '07]

- SPEAR, tree search solver for industrial SAT instances
  - Developed by Domagoj Babić at UBC
  - 26 parameters, 8.34×10^{17} configurations
  - Competitive with the state of the art
- Ran PARAMILS for 2 days on 10 machines

Configuration of Spear for Verification

[Hutter, Babić, Hoos & Hu; FMCAD '07]

- SPEAR, tree search solver for industrial SAT instances
  - Developed by Domagoj Babić at UBC
  - 26 parameters, 8.34×10^{17} configurations
  - Competitive with the state of the art
- Ran PARAMILS for 2 days on 10 machines
- Results (on test instances)

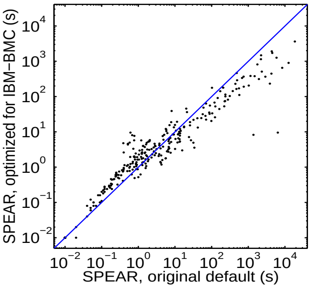


IBM Bounded Model Checking:
4.5-fold speedup

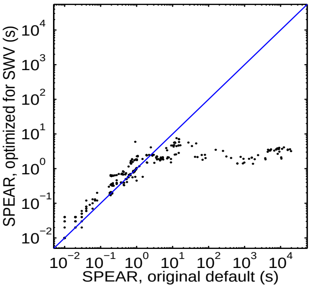
Configuration of Spear for Verification

[Hutter, Babić, Hoos & Hu; FMCAD '07]

- SPEAR, tree search solver for industrial SAT instances
  - Developed by Domagoj Babić at UBC
  - 26 parameters, 8.34×10^{17} configurations
  - Competitive with the state of the art
- Ran PARAMILS for 2 days on 10 machines
- Results (on test instances)



IBM Bounded Model Checking: 4.5-fold speedup



Software verification: 500-fold speedup
$\rightsquigarrow$ won 2007 SMT competition

Configuration of MIP Solvers

[Hutter, Hoos & Leyton-Brown; CP-AI-OR '10]

MIP (mixed integer programming)

- $\mathcal{NP}$-hard optimization problem
- Highly relevant in industry

Configuration of MIP Solvers

[Hutter, Hoos & Leyton-Brown; CP-AI-OR '10]

MIP (mixed integer programming)

- $\mathcal{NP}$-hard optimization problem
- Highly relevant in industry

MIP Solvers

- CPLEX (76 parameters)
- GUROBI (25 parameters)
- LPSOLVE(47 parameters)

Configuration of MIP Solvers

[Hutter, Hoos & Leyton-Brown; CP-AI-OR '10]

MIP (mixed integer programming)

- $\mathcal{NP}$-hard optimization problem
- Highly relevant in industry

MIP Solvers

- Cplex (76 parameters)
- Gurobi (25 parameters)
- lpsolve(47 parameters)

Comparison against default algorithm configurations

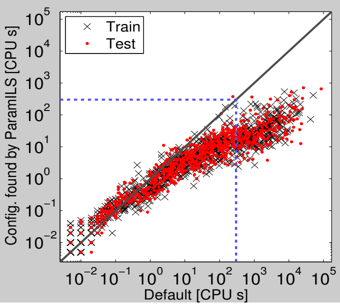
> "A great deal of algorithmic development effort has been devoted to establishing default ILOG CPLEX parameter settings that achieve good performance on a wide variety of MIP models." [Cplex 12.1 user manual, page 478]

Configuration of MIP Solvers

- Ran ParamILS for 2 days on 10 machines

Configuration of MIP Solvers

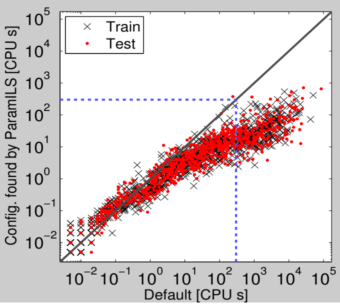
- Ran ParamILS for 2 days on 10 machines
- Speedups (on test instances)
  - Cplex 2x to 52x



Cplex on SUST instances

Configuration of MIP Solvers

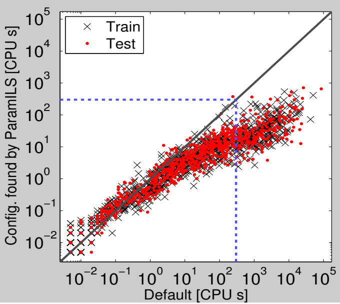
- Ran ParamILS for 2 days on 10 machines
- Speedups (on test instances)
  - Cplex 2x to 52x
  - Gurobi 1.2x to 2.3x



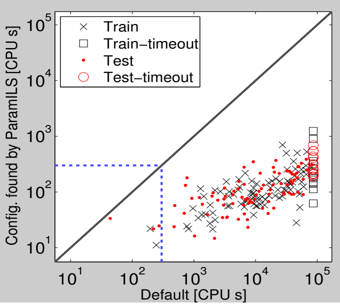
Cplex on SUST instances

Configuration of MIP Solvers

- Ran PARAMILS for 2 days on 10 machines
- Speedups (on test instances)
  - CPLEX 2x to 52x
  - GUROBI 1.2x to 2.3x
  - LPSOLVE 1x (no speedup) to 150x



CPLEX on SUST instances



LPSOLVE on WDP instances

Other Successful Applications of ParamILS

- Probabilistic Reasoning [Hutter, Hoos & Stützle, '07]
- Protein Folding [Thatchuk, Shmygelska & Hoos '07]
- Time-tabling [Fawcett, Hoos & Chiarandini '09]
- Local Search for SAT [Khudabukhsh, Xu, Hoos, & Leyton-Brown '09]
- ...

Outline

1. My PhD in a nutshell
   - Iterated Local Search in Configuration Space
   - Sequential Model-Based Search in Configuration Space

2. Some general points

Why Model-Based Approaches?

Model-free techniques are limited

- Only return a good parameter setting
- Do not provide additional information
  - How important is each of the parameters?
  - Which parameters interact?
  - For which types of instances is a parameter setting good?

Why Model-Based Approaches?

Model-free techniques are limited

- Only return a good parameter setting
- Do not provide additional information
  - How important is each of the parameters?
  - Which parameters interact?
  - For which types of instances is a parameter setting good?

Model-based approaches can help

- Construct predictive model of algorithm performance
- Use model to answer the questions above
- Use model in sequential approach for algorithm configuration

Challenges

- Technique from statistics literature on black-box function optimization
- Wanted to use for algorithm configuration

Challenges

- Technique from statistics literature on black-box function optimization
- Wanted to use for algorithm configuration
- Issues
  - Computational complexity
  - Scaling to many dimensions (parameters)
  - Handling categorical parameters
  - Handling configuration for benchmark *set*
  - ...

Challenges

- Technique from statistics literature on black-box function optimization
- Wanted to use for algorithm configuration
- Issues
  - Computational complexity
  - Scaling to many dimensions (parameters)
  - Handling categorical parameters
  - Handling configuration for benchmark *set*
  - ...

  $\rightsquigarrow$ Did not work right away

Mistake: tried to tackle all issues at once

- Wanted to wrap up
  (German funding for the "regular" PhD duration of 3 years)

Mistake: tried to tackle all issues at once

- Wanted to wrap up
  (German funding for the "regular" PhD duration of 3 years)
- Trying to rush it did not help
  - 2008: didn't even submit a single paper

Mistake: tried to tackle all issues at once

- Wanted to wrap up
  (German funding for the "regular" PhD duration of 3 years)
- Trying to rush it did not help
  - 2008: didn't even submit a single paper
- New years resolution 2008/09: be more scientific
  - Broke problem into pieces
  - Rapid progress, published on one issue at a time
  - Got things working in the end

Mistake: tried to tackle all issues at once

- Wanted to wrap up
  (German funding for the “regular” PhD duration of 3 years)
- Trying to rush it did not help
  - 2008: didn’t even submit a single paper
- New years resolution 2008/09: be more scientific
  - Broke problem into pieces
  - Rapid progress, published on one issue at a time
  - Got things working in the end
  - But still some unsolved problems to date; that’s ok!

Outline

1. My PhD in a nutshell

2. Some general points

Choosing a topic (Finding your niche)

- Am I excited about it?

Choosing a topic (Finding your niche)

- Am I excited about it?
- Will it have impact?

Choosing a topic (Finding your niche)

- Am I excited about it?
- Will it have impact?
- Am I the right person to do it?
  - Do I have the skill sets needed?
  - Are there others who could do this much easier than me?

Choosing a topic (Finding your niche)

- Am I excited about it?
- Will it have impact?
- Am I the right person to do it?
  - Do I have the skill sets needed?
  - Are there others who could do this much easier than me?
- Do I have the right people to do this with?
  - Supervisor should
    - + know more about the topic/methods than me (in the beginning)
    - + be excited about the topic

Choosing a topic (Finding your niche)

- Am I excited about it?
- Will it have impact?
- Am I the right person to do it?
  - Do I have the skill sets needed?
  - Are there others who could do this much easier than me?
- Do I have the right people to do this with?
  - Supervisor should
    - know more about the topic/methods than me (in the beginning)
    - be excited about the topic
  - Other collaborators (PhD/MSc students, industry, collaborating groups, etc)

Build your tool box

- Empirical algorithmics: do your experiments right

Build your tool box

- Empirical algorithmics: do your experiments right
- Machine learning comes in handy all the time
  - General concepts are important
  - E.g. standard regression/classification
  - Average user does not need to know every detail

Build your tool box

- Empirical algorithmics: do your experiments right
- Machine learning comes in handy all the time
  - General concepts are important
  - E.g. standard regression/classification
  - Average user does not need to know every detail
- Discrete optimization: local search is very general
- ...

Try to add to others' tool boxes

- Put your code on the web !
  - Reproducibility of experiments
  - Much (!) more impact
  - Papers are only one mode of scientific progress

Try to add to others' tool boxes

- Put your code on the web !
  - Reproducibility of experiments
  - Much (!) more impact
  - Papers are only one mode of scientific progress
- Automated algorithm configuration now one of those tools:
  `http://www.cs.ubc.ca/labs/beta/Projects/AAC/`

Try to add to others' tool boxes

- Put your code on the web !
  - Reproducibility of experiments
  - Much (!) more impact
  - Papers are only one mode of scientific progress
- Automated algorithm configuration now one of those tools: `http://www.cs.ubc.ca/labs/beta/Projects/AAC/`
  - Don't waste your time tuning parameters manually anymore...

Conclusion

- Importance of good empirical work
- My thesis: automated algorithm configuration
- Find your niche!
- Everybody goes through tough times

Thanks to

- Thesis supervisors
  - Holger Hoos
  - Kevin Leyton-Brown
  - Kevin Murphy
- Further collaborators
  - Domagoj Babić
  - Thomas Bartz-Beielstein
  - Youssef Hamadi
  - Alan Hu
  - Thomas Stützle
  - Dave Tompkins
  - Lin Xu