



DAGSTUHL REPORTS

Volume 2, Issue 1, January 2012

Foundations for Scripting Languages (Dagstuhl Seminar 12011) <i>Robert Hirschfeld, Shriram Krishnamurthi, and Jan Vitek</i>	1
Computability, Complexity and Randomness (Dagstuhl Seminar 12021) <i>Verónica Becher, Laurent Bienvenu, Rodney Downey, and Elvira Mayordomo</i>	19
Symmetric Cryptography (Dagstuhl Seminar 12031) <i>Frederik Armknecht, Stefan Lucks, Bart Preneel, and Phillip Rogaway</i>	39
Learning in Multiobjective Optimization (Dagstuhl Seminar 12041) <i>Salvatore Greco, Joshua D. Knowles, Kaisa Miettinen, and Eckart Zitzler</i>	50
Analysis of Executables: Benefits and Challenges (Dagstuhl Seminar 12051) <i>Andy M. King, Alan Mycroft, Thomas W. Reps, and Axel Simon</i>	100

ISSN 2192-5283

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.

Online available at <http://www.dagstuhl.de/dagrep>

Publication date

May, 2012

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

License

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported license: CC-BY-NC-ND.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.
- Noncommercial: The work may not be used for commercial purposes.
- No derivation: It is not allowed to alter or transform this work.

The copyright is retained by the corresponding authors.

Aims and Scope

The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.

In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
 - an overview of the talks given during the seminar (summarized as talk abstracts), and
 - summaries from working groups (if applicable).
- This basic framework can be extended by suitable contributions that are related to the program of the seminar, e.g. summaries from panel discussions or open problem sessions.

Editorial Board

- Susanne Albers
- Bernd Becker
- Karsten Berns
- Stephan Diehl
- Hannes Hartenstein
- Frank Leymann
- Stephan Merz
- Bernhard Nebel
- Han La Poutré
- Bernt Schiele
- Nicole Schweikardt
- Raimund Seidel
- Gerhard Weikum
- Reinhard Wilhelm (*Editor-in-Chief*)

Editorial Office

Marc Herbstritt (*Managing Editor*)
Jutka Gasirowski (*Editorial Assistance*)
Thomas Schillo (*Technical Assistance*)

Contact

Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de

Digital Object Identifier: 10.4230/DagRep.2.1.i

www.dagstuhl.de/dagrep

Foundations for Scripting Languages

Edited by

Robert Hirschfeld¹, Shriram Krishnamurthi², and Jan Vitek³

¹ Hasso-Plattner-Institut, Potsdam, DE, hirschfeld@hpi.uni-potsdam.de

² Brown University, Providence, US, sk@cs.brown.edu

³ Purdue University, US, jv@cs.purdue.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12011 on the “Foundations for Scripting Languages”. The choice of “for” rather than “of” is intentional: it is our thesis that scripting languages are in need of foundations to support their extensive use but lack them, and we hope this event consolidated and advanced the state of the art in this direction.

Seminar 02.–06. January, 2012 – www.dagstuhl.de/12011

1998 ACM Subject Classification D.2 Software Engineering, D.2.4 Formal Methods, D.3 Programming Languages, D.3.1 Semantics, D.3.4 Compilers, I.7.2 Scripting Languages

Keywords and phrases scripting languages, programming languages semantics, type systems, verification techniques, security analyses, scalability, rapid software development


Digital Object Identifier 10.4230/DagRep.2.1.1

1 Executive Summary

Robert Hirschfeld

Shriram Krishnamurthi

Jan Vitek

License  Creative Commons BY-NC-ND 3.0 Unported license
© Robert Hirschfeld, Shriram Krishnamurthi, and JanVitek

Common characteristics of scripting languages include syntactic simplicity, a lack of onerous constraints for program construction and deployment, the ability to easily connect to and control systems processes, strong built-in interfaces to useful external objects, extensive library support, and lightweight (and embeddable) implementations. More broadly, these characteristics add up to strong support for effective software prototyping. Due to a combination of these characteristics, common scripting languages like Perl, Python, Ruby, JavaScript, Visual Basic, and Tcl have moved from the fringes to mainstream program development.

To academics, these languages do not appear that different from, say, Scheme or ML. Since languages like Scheme and ML have well-defined semantics and other formal attributes, the mainstream passion for scripting languages may appear to simply be the result of ignorance of better languages amongst mainstream developers. However, the properties that scripting language users claim to find most beneficial are often *not* found in their more academic counterparts, such as a strong orientation towards systems process management, easily extensible objects, specific but useful control operators, etc.

In short, the academic tendency towards reductionism appears to miss some important characteristics. In particular, properties that may appear incidental—and are ignored by



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license
Foundations for Scripting Languages, *Dagstuhl Reports*, Vol. 2, Issue 1, pp. 1–18
Editors: Robert Hirschfeld, Shriram Krishnamurthi, and Jan Vitek



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the formalization of academic languages—may actually be essential. As a result, the formal study of scripting languages is a worthwhile research activity in its own right.

Not only does the study of scripting offer academics fresh problems, their results have the potential for widespread benefit. As scripts grow into programs, the very characteristics that seem an advantage sometimes prove to be disadvantages. If any object can be extended by any other object, it is impossible to reason about its behavior. If any object can access any resources, it is impossible to bound security implications. If programmers can place values of any type into a variable, it is impossible to obtain type guarantees. And so on. In other words, the very flexibility that enables prototyping inhibits the reasoning necessary for programs to grow in scale.

In the early days of scripting, there was an expectation that scripts were not meant to “grow up”. Rather, as a prototype proved valuable, it would be turned into a program in a mainstream language, such as Java. However, reality does not match this vision. First, once a system becomes valuable to an organization, it is not possible to halt development on it while waiting for a full re-implementation. Second, even if the current version is converted to Java, the next version would probably still benefit from the benefits of prototyping. Thus, in both cases, programs that start in a scripting language are likely to remain in it. Finally, even if clients do want to rewrite the program in a more mature language, they would benefit from formal support to enable this conversion.

As a result, the formal study of scripting languages is a worthwhile research activity in its own right. In particular, we hope this seminar had both direct and indirect impact on academia and industry. We also hope that, based on our discussions, academics will identify concrete problems that need solutions and find scripting language experts who they can communicate with. In turn, we hope scripting experts identified knowledge, expertise, and interest from academia and are better aware of how to formulate problems for academics and map their solutions back to practice.

2 Table of Contents

Executive Summary

<i>Robert Hirschfeld, Shriram Krishnamurthi, and Jan Vitek</i>	1
--	---

Overview of Talks





Eval Begone! <i>Gregor Richards</i>	5
Evaluating the Design of the R Language <i>Jan Vitek</i>	5
Reasoning about Javascript <i>Philippa Gardner</i>	5
Language Support for Third-party Code Extensibility <i>Benjamin Lerner</i>	6
Empirical Studies on Static vs. Dynamic Type Systems <i>Stefan Udo Hanenberg</i>	6
Engineering a JavaScript Semantics <i>Arjun Guha</i>	6
AmbientTalk as a Scripting Language <i>Theo D'Hondt</i>	7
Life After main() <i>David Herman</i>	7
RubyX: Symbolic Execution for Security Analysis of Ruby on Rails <i>Jeffrey Foster</i>	7
Languages as Libraries <i>Sam Tobin-Hochstadt</i>	8
Virtual Values for Language Extension <i>Cormac Flanagan</i>	8
Sandboxing Untrusted JavaScript <i>Ankur Taly</i>	9
ADsafety: Type-based Verification of JavaScript Sandboxing <i>Joe Politz</i>	9
Integrating Typed and Untyped Code in a Scripting Language <i>Francesco Zappa Nardelli</i>	9
Using Contracts to Connect Different Scripting Languages <i>Kathryn E. Gray</i>	10
Blame for All <i>Philip Wadler</i>	10
Temporal Higher-order Contracts <i>Cormac Flanagan</i>	11
A Racket Contract Example <i>Robert Bruce Findler</i>	11

Dynamic Inference of Static Types for Ruby <i>Michael Hicks</i>	12
Nested Refinements: A Logic for Duck Typing <i>Ravi Chugh</i>	12
The Ciao Assertions Model <i>Manuel Hermenegildo</i>	13
Occurrence Typing <i>Sam Tobin-Hochstadt</i>	13
Gradual Typing Roundup <i>Jeremy G. Siek</i>	14
(Towards) Gradual Typing for Java <i>Atsushi Igarashi</i>	14
Combining Types and Flow Analysis <i>Arjun Guha</i>	14
Lively Webwerkstatt—A Self-sustaining Web-based Authoring Environment <i>Jens Lincke, Robert Hirschfeld, and Bastian Steinert</i>	15
What Use for Macros / Compile-time Meta-programming? <i>Laurence Tratt</i>	15
Experiences of Implementing a VM with RPython <i>Laurence Tratt</i>	15
Meta-Tracing in the PyPy Project for Efficient Dynamic Languages <i>Carl Friedrich Bolz</i>	15
HipHop – A Synchronous Reactive Extension for Hop <i>Manuel Serrano</i>	16
A Possible End-User Scripting Environment for STEPS <i>Yoshiki Ohshima</i>	16
101companies:101 Ways of Building a Management System With Different Pro- gramming Technologies <i>Ralf Lämmel</i>	16
A Scripting Language for Remote Communication <i>William R. Cook</i>	17
Languages in Racket Demo <i>Matthew Flatt</i>	17
Participants	18

3 Overview of Talks

3.1 Eval Begone!


Gregor Richards (Purdue University, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Gregor Richards

Eval is a common feature in dynamic languages, but an uncommon feature in analyses. Our work measures the real-world use of eval and determines its utility, in search of the “mythical” proper use of eval. We then introduce a system for the automated removal of eval by interactive analysis of its use and dynamic replacement with static suggestions.

3.2 Evaluating the Design of the R Language

Jan Vitek (Purdue University, US)





License     Creative Commons BY-NC-ND 3.0 Unported license
© Jan Vitek

Joint work of Floréal Morandat, Brandon Hill, Leo Osvald, and Jan Vitek

R is a dynamic language for statistical computing that combines lazy functional features and object-oriented programming. This rather unlikely linguistic cocktail would probably never have been prepared by computer scientists, yet the language has become surprisingly popular. With millions of lines of R code available in repositories, we have an opportunity to evaluate the fundamental choices underlying the R language design. Using a combination of static and dynamic program analysis we can assess the impact and success of different language features.

3.3 Reasoning about Javascript

Philippa Gardner (Imperial College London, GB)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Philippa Gardner


Joint work of Philippa Gardner, Sergio Maffei, and Gareth Smith

JavaScript has become the most widely used language for client-side web programming. The dynamic nature of JavaScript makes understanding its code notoriously difficult, leading to buggy programs and a lack of adequate static-analysis tools. We believe that logical reasoning has much to offer JavaScript: a simple description of program behaviour, a clear understanding of module boundaries, and the ability to verify security contracts.

We introduce a program logic for reasoning about a broad subset of JavaScript, including challenging features such as prototype inheritance and with. We adapt ideas from separation logic to provide tractable reasoning about JavaScript code: reasoning about easy programs is easy; reasoning about hard programs is possible. We prove a strong soundness result. All libraries written in our subset and proved correct with respect to their specifications will be well-behaved, even when called by arbitrary JavaScript code.

3.4 Language Support for Third-party Code Extensibility

Benjamin Lerner (University of Washington, Seattle, US)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Benjamin Lerner

Browsers today support extensions, third-party pieces of script and markup that provide new or modified behavior for the underlying system. Likewise, users can inject scripts into web sites to modify them in a similar fashion. However, the idioms used to achieve this injection are cryptic, brittle, and have severe semantic flaws.

In this work we propose adding a new linguistic primitive to JavaScript, namely dynamic aspect weaving, that supports these extensions in a more robust, understandable, and stable way. As a side benefit, the new mechanism often out-performs the original idioms used.

3.5 Empirical Studies on Static vs. Dynamic Type Systems


Stefan Udo Hanenberg (Universität Duisburg-Essen, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Stefan Udo Hanenberg

While static and dynamic type systems are exhaustively discussed by a large number a people, there is still no evidence whether (or in what situations) a static or dynamic type system provides a measurable benefit for software developers. This talk summarizes the results and the underlying ideas for an experiment series which compares the impact of static and dynamic type systems on software developers (based on the measurements of development time). The preliminary results so far are that the possible benefit of static and dynamic type systems is programming task specific. Furthermore, there is some evidence that type casts are no valid argument against static type systems.

3.6 Engineering a JavaScript Semantics

Arjun Guha (Brown University, Providence, US)





License  Creative Commons BY-NC-ND 3.0 Unported license
© Arjun Guha

Joint work of Arjun Guha, Claudiu Saftoiu, and Shriram Krishnamurthi

We reduce JavaScript to LambdaJS, a core calculus structured as a small-step operational semantics. We present several peculiarities of the language and show that our calculus models them. We explicate the desugaring process that turns JavaScript programs into ones in the core. We demonstrate faithfulness to JavaScript using real-world test suites. Finally, we illustrate utility by defining a security property, implementing it as a type system on the core, and extending it to the full language.

3.7 AmbientTalk as a Scripting Language





Theo D'Hondt (Vrije Universiteit Brussel, BE)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Theo D'Hondt

AmbientTalk is a language for mobile ad-hoc networks. It combines actors with effects and promotes failure to the rule rather than the exception. AmbientTalk and its implementation is described and subsequently compared to Python as a scripting language.

3.8 Life After main()





David Herman (Mozilla, Mountain View, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© David Herman

Scripting languages are often embedded in dynamic environments such as editors or browsers, and provide dynamic evaluation through REPL's. When a dynamic language contains static semantics such as static scoping, types, macros, or operator overloading, the interaction between the static and dynamic portions of the language can be fiendishly complicated. In my talk I discuss some of the surprising interactions and describe some of the design landscape for designing scripting languages with static semantics.

3.9 RubyX: Symbolic Execution for Security Analysis of Ruby on Rails


Jeffrey Foster (University of Maryland, College Park, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Jeffrey Foster
Joint work of Jeffrey Foster, Avik Chaudhuri, and Jong-hoon (David) An

Many of today's web applications are built on frameworks that include sophisticated defenses against malicious adversaries. However, mistakes in the way developers deploy those defenses could leave applications open to attack. To address this issue, we introduce Rubyx, a symbolic executor that we use to analyze Ruby-on-Rails web applications for security vulnerabilities. Rubyx specifications can easily be adapted to variety of properties, since they are built from general assertions, assumptions, and object invariants. We show how to write Ruby specifications to detect susceptibility to cross-site scripting and cross-site request forgery, insufficient authentication, leaks of secret information, insufficient access control, as well as application-specific security properties. We used Rubyx to check seven web applications from various sources against our specifications. We found many vulnerabilities, and each application was subject to at least one critical attack. Encouragingly, we also found that it was relatively easy to fix most vulnerabilities, and that Rubyx showed the absence of attacks after our fixes. Our results suggest that Rubyx is a promising new way to discover security vulnerabilities in Ruby-on-Rails web applications.

3.10 Languages as Libraries

Sam Tobin-Hochstadt (Northeastern University, Boston, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Sam Tobin-Hochstadt


Joint work of Sam Tobin-Hochstadt, Robby Findler, Vincent St-Amour, Ryan Culpepper, Eli Barzilay, Matthew Flatt, and Matthias Felleisen

Programming language design benefits from constructs for extending the syntax and semantics of a host language. While C’s string-based macros empower programmers to introduce notational shorthands, the parser-level macros of Lisp encourage experimentation with domain-specific languages. The Scheme programming language improves on Lisp with macros that respect lexical scope.

The design of Racket—a descendant of Scheme—goes even further with the introduction of a full-fledged interface to the static semantics of the language. A Racket extension programmer can thus add constructs that are indistinguishable from “native” notation, large and complex embedded domain-specific languages, and even optimizing transformations for the compiler backend. This power to experiment with language design has been used to create a series of sub-languages for programming with first-class classes and modules, numerous languages for implementing the Racket system, and the creation of a complete and fully integrated typed sister language to Racket’s untyped base language.

3.11 Virtual Values for Language Extension

Cormac Flanagan (University of California, Santa Cruz, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Cormac Flanagan

Joint work of Thomas H. Austin, Tim Disney, and Cormac Flanagan

This paper focuses on extensibility, the ability of a programmer using a particular language to extend the expressiveness of that language. This paper explores how to provide an interesting notion of extensibility by virtualizing the interface between code and data. A virtual value is a special value that supports behavioral intercession. When a primitive operation is applied to a virtual value, it invokes a trap on that virtual value. A virtual value contains multiple traps, each of which is a user-defined function that describes how that operation should behave on that value.

This paper formalizes the semantics of virtual values, and shows how they enable the definition of a variety of language extensions, including additional numeric types; delayed evaluation; taint tracking; contracts; revokable membranes; and units of measure. We report on our experience implementing virtual values for Javascript within an extension for the Firefox browser.

3.12 Sandboxing Untrusted JavaScript

Ankur Taly (Stanford University, US)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Ankur Taly

Joint work of Ankur Taly, John C. Mitchell, Sergio Maffei, Ulfar Erlingsson, Mark S. Miller, and Jasvir Nagra

Most websites today incorporate untrusted JavaScript content in the form of advertisements, maps and social networking gadgets. Untrusted JavaScript, if embedded directly, has complete access to the page’s Document Object Model (DOM) and can therefore steal cookies, navigate the page, maliciously alter the page or cause other harm. In order to combat the above threat, many websites use language-based mechanisms for restricting untrusted JavaScript. Popular examples of such mechanisms are Facebook FBJS, Yahoo! ADSafe and Google Caja. In this talk, I will rigorously define the security goals of such sandboxing mechanisms and then develop principled techniques for designing and analyzing them. I will back the techniques with rigorous guarantees established using an operational semantics for JavaScript. I will also present security vulnerabilities in Facebook FBJS and Yahoo! ADSafe found during the course of this work and principled approaches to fixing those vulnerabilities. The talk will span JavaScript based on 3rd edition of the ECMA262 specification and also the recently released “strict mode” of JavaScript based on 5th edition of the ECMA262 specification.

3.13 ADSafety: Type-based Verification of JavaScript Sandboxing

Joe Politz (Brown University, Providence, US)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Joe Politz

Joint work of Joe Gibbs Politz, Arjun Guha, Spirodon Aristides Eliopolous, and Shriram Krishnamurthi

Web sites routinely incorporate JavaScript programs from several sources into a single page. These sources must be protected from one another, which requires robust sandboxing. The many entry-points of sandboxes and the subtleties of JavaScript demand robust verification of the actual sandbox source. We use a novel type system for JavaScript to encode and verify sandboxing properties. The resulting verifier is lightweight and efficient, and operates on actual source. We demonstrate the effectiveness of our technique by applying it to ADSafe, which revealed several bugs and other weaknesses.

3.14 Integrating Typed and Untyped Code in a Scripting Language

Francesco Zappa Nardelli (Inria, Paris-Rocquencourt, FR)


License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Francesco Zappa Nardelli

Many large software systems originate from untyped scripting language code. While good for initial development, the lack of static type annotations can impact code-quality and performance in the long run. We present an approach for integrating untyped code and typed code in the same system to allow an initial prototype to smoothly evolve into an efficient and robust program. We introduce like types, a novel intermediate point between dynamic and static typing. Occurrences of like types variables are checked statically within their scope

but, as they may be bound to dynamic values, their usage is checked dynamically. Thus like types provide some of the benefits of static typing without decreasing the expressiveness of the language.

3.15 Using Contracts to Connect Different Scripting Languages


Kathryn E. Gray (University of Cambridge, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Kathryn E. Gray

Scripting languages are frequently combined with statically-typed languages, potentially running on different virtual machines. Conventional techniques for writing multi-language programs entail manually inserting data conversions, inter-machine communication, and dynamic checks, which can introduce subtle errors. My previous technique allows values to pass seamlessly from one language to another—for languages with similar dynamic semantics on the same VM. However, with scripting languages these criteria may not be met. So, this talk introduces a framework that supports languages with different runtime systems and semantics, while maintaining type-safety and a free exchange of values.

3.16 Blame for All


Philip Wadler (University of Edinburgh, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Philip Wadler
Joint work of Amal Ahmed, Robert Bruce Findler, Jeremy Siek, and Philip Wadler

[Appeared in POPL 2009] Several programming languages are beginning to integrate static and dynamic typing, including Racket (formerly PLT Scheme), Perl 6, and C# 4.0, and the research languages Sage (Gronski, Knowles, Tomb, Freund, and Flanagan, 2006) and Thorn (Wrigstad, Eugster, Field, Nystrom, and Vitek, 2009). However, an important open question remains, which is how to add parametric polymorphism to languages that combine static and dynamic typing. We present a system that permits a value of dynamic type to be cast to a polymorphic type and vice versa, with relational parametricity enforced by a kind of dynamic selaing along the line proposed by Matthews and Ahmed (2008) and Neis, Dreyer, and Rossberg (2009). Our system includes a notion of blame, which allows us to show that when casting between a more-precise type and a less-precise type, any failure are due to the less-precisely-typed portion of the program. We also show that a cast from a subtype to its supertype cannot fail.

3.17 Temporal Higher-order Contracts

Cormac Flanagan (University of California, Santa Cruz, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Cormac Flanagan


Joint work of Tim Disney, Jay McCarthy, and Cormac Flanagan

Behavioral contracts are embraced by software engineers because they document module interfaces, detect interface violations, and help identify faulty modules (packages, classes, functions, etc). This paper extends prior higher-order contract systems to also express and enforce temporal properties, which are common in software systems with imperative state, but which are mostly left implicit or are at best informally specified. The paper presents both a programmatic contract API as well as a temporal contract language, and reports on experience and performance results from implementing these contracts in Racket.

Our development formalizes module behavior as a trace of events such as function calls and returns. Our contract system provides both non-interference (where contracts cannot influence correct executions) and also a notion of completeness (where contracts can enforce any decidable, prefix-closed predicate on event traces).

3.18 A Racket Contract Example

Robert Bruce Findler (Northwestern University, Evanston, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Robert Bruce Findler

The following contract is an example contract that illustrates why earlier (lax/picky) interpretations of dependent contracts are wrong. See also “Correct Blame for Contracts: No More Scapegoating” in POPL 2011.

```
#lang racket


(provide (contract-out [deriv/c contract?]))
(require (planet cce/fasttest:3/random))

(define n 10)
(define δ 0.01)

(define deriv/c
  (->i ([f (-> real? real?)]
        [ε (and/c real? positive?)])
    (fp (-> real? real?))
    \#:post
    (f fp ε)
    (for/and ([i (in-range 0 n)])
      (define x (random-number))
      (define slope
        (/ (- (f (+ x ε))
              (f (- x ε)))
           (* 2 ε)))
      (<= (abs (- slope (fp x))) δ))))
```

3.19 Dynamic Inference of Static Types for Ruby

Michael Hicks (University of Maryland, College Park, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Hicks

Joint work of Michael Hicks, David An, Jeff Foster, and Avik Chaudhuri


Ruby is a dynamically typed scripting language in the tradition of Smalltalk. We have designed a type system for Ruby and a static type inference algorithm that we have applied to Ruby scripts and libraries. While a useful exercise, we found static type inference extremely difficult to develop: dynamic languages are typically complex, poorly specified, and include features, such as `eval` and reflection, that are hard to analyze.

In response, we developed constraint-based dynamic type inference, a technique that infers static types based on dynamic program executions. In our approach, we wrap each run-time value to associate it with a type variable, and the wrapper generates constraints on this type variable when the wrapped value is used. This technique avoids many of the often overly conservative approximations of static tools, as constraints are generated based on how values are used during actual program runs. Using wrappers is also easy to implement, since we need only write a constraint resolution algorithm and a transformation to introduce the wrappers. The best part is that we can eat our cake, too: our algorithm will infer sound types as long as it observes every path through each method body—note that the number of such paths may be dramatically smaller than the number of paths through the program as a whole.

We have developed `Rubydust`, an implementation of our algorithm for Ruby. `Rubydust` takes advantage of Ruby’s dynamic features to implement wrappers as a language library. We applied `Rubydust` to a number of small programs and found it to be both easy to use and useful: `Rubydust` discovered 1 real type error, and all other inferred types were correct and readable.

3.20 Nested Refinements: A Logic for Duck Typing

Ravi Chugh (University of California, San Diego, US)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Ravi Chugh

Joint work of Ravi Chugh, Pat Rondon, and Ranjit Jhala

Programs written in dynamic languages make heavy use of features—run-time type tests, value-indexed dictionaries, polymorphism, and higher-order functions—that are beyond the reach of type systems that employ either purely syntactic or purely semantic reasoning. We present a core calculus, `System D`, that merges these two modes of reasoning into a single powerful mechanism of nested refinement types wherein the typing relation is itself a predicate in the refinement logic. `System D` coordinates SMT-based logical implication and syntactic subtyping to automatically typecheck sophisticated dynamic language programs. By coupling nested refinements with McCarthy’s theory of finite maps, `System D` can precisely reason about the interaction of higher-order functions, polymorphism, and dictionaries. We also discuss extensions to support imperative updates and inheritance, features commonly found in real-world dynamic languages.

3.21 The Ciao Assertions Model

Manuel Hermenegildo (IMDEA Software, Madrid, ES)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Manuel Hermenegildo

Joint work of M.V. Hermenegildo, M. Carro, P. López-García, J. Morales, F. Bueno, G. Puebla, R. Haemmerlé

We provide a brief overview (and demo!) of Ciao, emphasizing some of the novel aspects and motivations behind its design and implementation.

Ciao is built in layers over a kernel, which is designed to be extensible in a powerful, modular way. Using these facilities, Ciao provides the programmer with a large number of useful features from different programming paradigms and styles. All such features are in libraries, so that the use of each of the features (including those of logic and constraint programming) can be turned on and off at will for each program module. Thus, a given module may be using, e.g., higher order functions and constraints, while another module may be using assignment, predicates, meta-programming, and concurrency. The module system and the extension mechanism together allow user-level design and implementation of powerful extensions and domain specific languages.

Another important objective of Ciao as a “scripting language”—on which the talk and demo concentrate—is to offer the best of the dynamic and static language approaches, i.e., providing the flexibility of dynamic languages, but with guaranteed safety and efficiency. Important elements to this end are the Ciao assertion language and its preprocessor. The assertion language allows expressing many kinds of program properties (ranging from, e.g., moded types to resource consumption), as well as tests and documentation. The preprocessor is capable of statically finding violations of these properties or verifying that programs comply with them, and issuing certificates of this compliance, and also generating run-time tests for (parts of) specifications with which compliance cannot be resolved at compile-time. The compiler performs many types of optimizations (including automatic parallelization), producing code that is highly competitive with other dynamic languages or, with the (experimental) optimizing compiler, even that of static languages, all while retaining the flexibility and interactive development of a dynamic language.

3.22 Occurrence Typing

Sam Tobin-Hochstadt (Northeastern University, Boston, US)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Sam Tobin-Hochstadt

Joint work of Sam Tobin-Hochstadt, Vincent St-Amour, and Matthias Felleisen

Ad-hoc, untagged unions are pervasive in scripting languages. However, traditional type systems do not handle unions well. In this talk, I describe occurrence typing, which provides an effective elimination rule for union types, and enables the type checking of idiomatic scripting language programs. I also describe a surprising application to numeric type checking.

3.23 Gradual Typing Roundup


Jeremy G. Siek (University of Colorado, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jeremy G. Siek

Gradual typing is an approach for integrating static and dynamic typing within the same language. Since it's introduction 5 years ago, many challenges have been overcome, such as how to efficiently represent higher-order casts and how to integrate gradual typing with other features such as objects and generics. This talk gives an example-based survey of the progress in gradual typing and discusses the remaining challenges, with some hints at solutions to some of them.

3.24 (Towards) Gradual Typing for Java


Atsushi Igarashi (Kyoto University, JP)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Atsushi Igarashi
Joint work of Atsushi Igarashi and Lintaro Ina

We have presented our recent work on extending Java with gradual typing. The main focus is on the interaction between type “dynamic” and generic types. We have also discussed how our design constraint that proper Java code should compile to (almost) the same bytecode as javac affected the language feature design and implementation.

3.25 Combining Types and Flow Analysis

Arjun Guha (Brown University, Providence, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Arjun Guha
Joint work of Arjun Guha, Claudiu Saftoiu, and Shiram Krishnamurthi

Programs written in scripting languages employ idioms that confound conventional type systems. In this talk, we highlight one important set of related idioms: the use of local control and state to reason informally about types. To address these idioms, we formalize run-time tags and their relationship to types, and use these to present a novel strategy to integrate typing with flow analysis in a modular way. We demonstrate that in our separation of typing and flow analysis, each component remains conventional, their composition is simple, but the result can handle these idioms better than either one alone.

3.26 Lively Webwerkstatt—A Self-sustaining Web-based Authoring Environment

Jens Lincke, Robert Hirschfeld, and Bastian Steinert (Hasso-Plattner-Institut, Potsdam, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Jens Lincke, Robert Hirschfeld, and Bastian Steinert

Webwerkstatt is an interactive and programmable wiki environment for experimenting with different approaches to End-user Web Development and their boundaries to the application kernel. It is based on the Lively Kernel and incorporates projects such as Lively Wiki (a Wiki of live objects built on an SVN repository) and Lively Fabrik (a dataflow-like GUI builder for Lively). For Webwerkstatt, we developed the context-oriented language extension ContextJS, to explore new concepts for expressing this boundary. Our current research focuses on prototypical scripting and interactive application construction.

- <http://lively-kernel.org/>
- <http://lively-kernel.org/repository/webwerkstatt/webwerkstatt.xhtml>

3.27 What Use for Macros / Compile-time Meta-programming?

Laurence Tratt (King’s College, London, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Laurence Tratt

Is the oft-repeated idea that “all good languages have macros / CTMP” undeniably true? This short talk is intended to make us think about the consequences of this idea.

3.28 Experiences of Implementing a VM with RPython


Laurence Tratt (King’s College, London, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Laurence Tratt

A report on preliminary work on implementing an RPython VM for Converge, which suggests that language designers of the future now have a practical route for making “fast enough” VMs in “fast enough” time. See also <http://convergepl.org/>

3.29 Meta-Tracing in the PyPy Project for Efficient Dynamic Languages

Carl Friedrich Bolz (Universität Düsseldorf, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Carl Friedrich Bolz


Joint work of Carl Friedrich Bolz, Antonio Cuni, Maciej Fijalkowski, Michael Leuschel, Samuele Pedroni, and Armin Rigo

Writing JIT-compilers for recent scripting languages is a hard problem due to their recent semantics. The PyPy project tries to help with that problem by providing a tracing JIT

that operates “one level down”, i.e. below an interpreter. That way the JIT can be reused for a number of languages.

3.30 HipHop – A Synchronous Reactive Extension for Hop


Manuel Serrano (Inria, Sophia Antipolis, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Manuel Serrano

HOP is a SCHEME-based language and system to build rich multi-tier web applications. We present HIPHOP, a new language layer within HOP dedicated to request and event orchestration. HIPHOP follows the synchronous reactive model of the Esterel and ReactiveC languages, originally developed for embedded systems programming. It is based on synchronous concurrency and preemption primitives, which are known to be key components for the modular design of complex temporal behaviors. Although the language is concurrent, the generated code is purely sequential and thread-free; HIPHOP is translated to HOP for the server side and to straight JavaScript for the client side. With a music playing example, we show how to modularly build non-trivial orchestration code with HIPHOP.

3.31 A Possible End-User Scripting Environment for STEPS

Yoshiki Ohshima (Viepoints Research Institut, Glendale, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Yoshiki Ohshima


In this talk, a brief overview of the STEPS Project and a possible design and implementation of the end-user scripting environment are presented.

In the STEPS Project, we are set out to explore good abstractions and concise descriptions of the entire personal computing environment. Language execution engines, a graphics engine, a GUI framework and an application framework were created in this philosophy and a universal document editor is created.

However, an end-user scripting system is yet to be written. Drawing from the Functional Reactive Programming work, we are exploring reactive programming in more dynamic setting. A possible implementation of such an end-user scripting system is under development.

3.32 101companies:101 Ways of Building a Management System With Different Programming Technologies

Ralf Lämmel (Universität Koblenz-Landau, DE)





License  Creative Commons BY-NC-ND 3.0 Unported license
© Ralf Lämmel

The open-source 101companies Project is concerned with aggregating, organizing, annotating, and analyzing a corpus of many implementations of a simple Human Resource Management System (the so-called 101companies System) such that the implementations leverage varying programming technologies and varying software languages dedicated to different technological

spaces. The specification of the 101companies System contains several optional features which implementations can choose to implement in the interest of demonstrating specific programming technologies or capabilities thereof. The 101companies Project helps understanding and comparing programming technologies in a manner as it is valuable for different stakeholders such as teachers, learners, developers, software technologists, and ontologists. In this paper, we present the following major aspects of the project: i) an emerging ontology of relevant entities and categories; ii) a list of stakeholders of the project; iii) a feature model of the 101companies System; iv) themes as a grouping concept for implementations of the system; v) the structured documentation of implementations.

3.33 A Scripting Language for Remote Communication

William R. Cook (University of Texas, Austin, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© William R. Cook





Joint work of William R. Cook, Eli Tilevitch, Ben Wiedermann, and Ali Ibrahim

Batches are a new approach to relational database access, remote procedure calls, and web services. Batching employs a simple scripting language to communicate work from a client to a server. Batches also have a new control flow construct, called a Remote Batch statement. A Remote Batch statement combines remote and local execution: all the remote code is executed in a single round-trip to the server, where all data sent to the server and results from the batch are communicated in bulk. Batches support remote blocks, iteration and conditionals, and local handling of remote exceptions. Batches are efficient even for fine-grained interfaces, eliminating the need for hand-optimized server interfaces.

Batch services also provide a simple and powerful interface to relational databases, with support for arbitrary nested queries and bulk updates. One important property of the system is that a single batch statement always generates a constant number of SQL queries, no matter how many nested loops are used.

3.34 Languages in Racket Demo

Matthew Flatt (University of Utah, Salt Lake City, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Matthew Flatt

Racket provides a smooth path from syntactic abstraction, language extension, language implementation, and environment support for languages with or without S-expression notation. In this demonstration, we show how implement a little JavaScript-like language in about 200 lines of code (mostly a parser).

Participants

- Amal Ahmed
Northeastern Univ. – Boston, US
- Carl Friedrich Bolz
Universität Düsseldorf, DE
- Ravi Chugh
University of California – San Diego, US
- William R. Cook
University of Texas – Austin, US
- Theo D’Hondt
Vrije Universiteit Brussel, BE
- Matthias Felleisen
Northeastern University – Boston, US
- Robert Bruce Findler
Northwestern Univ. – Evanston, US
- Cormac Flanagan
University of California – Santa Cruz, US
- Matthew Flatt
University of Utah – Salt Lake City, US
- Jeffrey Foster
University of Maryland – College Park, US
- Andreas Gal
Mozilla – Mountain View, US
- Philippa Gardner
Imperial College London, GB
- Kathryn E. Gray
University of Cambridge, GB
- Arjun Guha
Brown Univ. – Providence, US
- Stefan Udo Hanenberg
Universität Duisburg-Essen, DE
- David Herman
Mozilla – Mountain View, US
- Manuel Hermenegildo
IMDEA Software – Madrid, ES
- Michael Hicks
University of Maryland – College Park, US
- Robert Hirschfeld
Hasso-Plattner-Institut – Potsdam, DE
- Atsushi Igarashi
Kyoto University, JP
- Shriram Krishnamurthi
Brown Univ. – Providence, US
- Ralf Lämmel
Universität Koblenz-Landau, DE
- Benjamin Lerner
University of Washington – Seattle, US
- Jens Lincke
Hasso-Plattner-Institut – Potsdam, DE
- Hidehiko Masuhara
University of Tokyo, JP
- Mark S. Miller
Sunnyvale, US
- Floreal Morandat
Purdue University, US
- Oscar M. Nierstrasz
Universität Bern, CH
- Nathaniel Nystrom
Universität Lugano, CH
- Yoshiki Ohshima
Viepoints Research Institut – Glendale, US
- Joe Politz
Brown Univ. – Providence, US
- Gregor Richards
Purdue University, US
- Manuel Serrano
Inria – Sophia Antipolis, FR
- Jeremy G. Siek
University of Colorado, US
- Bastian Steinert
Hasso-Plattner-Institut – Potsdam, DE
- Ankur Taly
Stanford University, US
- Eric Tanter
Univ. of Chile – Santiago, CL
- Sam Tobin-Hochstadt
Northeastern University – Boston, US
- Laurence Tratt
King’s College – London, GB
- Herman Venter
Microsoft Res. – Redmond, US
- Jan Vitek
Purdue University, US
- Philip Wadler
University of Edinburgh, GB
- Francesco Zappa Nardelli
Inria – Paris-Rocquencourt, FR



Computability, Complexity and Randomness

Edited by

Verónica Becher¹, Laurent Bienvenu², Rodney Downey³, and
Elvira Mayordomo⁴

1 University of Buenos Aires, AR, vbecher@dc.uba.ar

2 University Paris-Diderot, FR, laurent.bienvenu@liafa.jussieu.fr

3 Victoria University of Wellington, NZ, Rodney.Downey@vuw.ac.nz

4 University of Zaragoza, ES, elvira@unizar.es

Abstract

Research on the notions of information and randomness has drawn on methods and ideas from computability theory and computational complexity, as well as core mathematical subjects like measure theory and information theory. The Dagstuhl seminar 12021 “Computability, Complexity and Randomness” was aimed to meet people and ideas in these areas to share new results and discuss open problems. This report collects the material presented during the course of the seminar.

Seminar 08.–13. January, 2012 – www.dagstuhl.de/12021

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.1.1 Computability theory, E.4 Coding and information theory

Keywords and phrases algorithmic randomness, computability theory, computational complexity, Kolmogorov complexity, algorithmic information theory

Digital Object Identifier 10.4230/DagRep.2.1.19

1 Executive Summary

Verónica Becher

Laurent Bienvenu

Rodney Downey

Elvira Mayordomo

License  Creative Commons BY-NC-ND 3.0 Unported license
© Verónica Becher, Laurent Bienvenu, Rodney Downey, Elvira Mayordomo

Randomness and information quantity are central notions in computer science that are still undeveloped. Although classical information theory and probability provide formalizations of these notions they do not allow us to measure the information of a specific string or say that a particular real number is random. The definition of the property of randomness and its connection with a measure of information content was obtained in the 1960s and combines different complexity measures.

As witnessed by the three seminars previously organized in Dagstuhl on complexity and randomness (Seminar 9318, *Descriptive complexity: a multidisciplinary perspective* in 1993; Seminar 03181, *Centennial Seminar on Kolmogorov Complexity and Applications* in 2003; and Seminar 06051 *Kolmogorov Complexity and Applications* in 2006) in recent years there has been an upsurge produced by the people in computability theory that resulted in rapid progress in our understanding of even the most basic notions in randomness, and the solution of old open questions. This has changed and is still changing the landscape and opened up



Except where otherwise noted, content of this report is licensed

under a Creative Commons BY-NC-ND 3.0 Unported license

Computability, Complexity and Randomness, *Dagstuhl Reports*, Vol. 2, Issue 1, pp. 19–38

Editors: Verónica Becher, Laurent Bienvenu, Rodney Downey, and Elvira Mayordomo



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

new avenues of research. An evidence of this activity has been the publication of two new books in the area and the new edition of an already classical one: *Algorithmic Randomness and Complexity*, R. Downey and D. Hirschfeldt, Foundations on Computing, Springer, 2010; *Computability and Randomness*, A. Nies, Oxford University Press, 2009; and *An Introduction to Kolmogorov Complexity and Its Applications*, M. Li and P. Vitanyi, third Edition, Springer Verlag, 2008.

Seminar 12021 has celebrated significant recent research progress. New results connect the theory of algorithmic randomness with computable analysis. We consider them important because they lead to the naturalness of the notions of algorithmic randomness. For instance, Brattka, Miller, and Nies translated the theorem “*every non-decreasing function is almost everywhere differentiable*” to the computable world, by showing that a real x is computably random if and only if every computable non-decreasing function is differentiable at x (this work is has not yet appeared as a publication). Similar investigations identified the notions of randomness that correspond to the Lebesgue density and differentiation theorems. J.Franklin and the work of Gács, Hoyrup, and Rojas related Birkhoff’s pointwise ergodic theorem in connection with Schnorr randomness.

Considerable results have been obtained for problems on Kolmogorov complexity and computable enumerable sets, in particular, in the degree structure that arises from comparing the complexity of the initial segments of two reals. Barmaplias announced the solution of the already long standing open problem posed by Downey and Hirschfeldt *Is there a minimal pair of c.e. reals in the K-degrees?* The answer is no.

Since the start of the discipline, the notion of randomness was defined for infinite sequences, or real numbers. The problem posed by Kolmogorov on a notion of randomness of finite objects remains unsolved. This is also the case for arbitrary countable objects. C.Freer made significant progress on the questions *When is a graph random?* and *What is the connection between quasi-random graphs and pseudorandom bit strings?* He pointed to an emerging theory of continuous limits of finite combinatorial structures that connects graph limits, property testing, and exchangeable relations.

There was a general consensus on the fact that there is yet no adequate solution to the fundamental problem that high-quality independent random bits are in very short supply. And there are many practical applications rely on randomness (for instance, assigning keys to users of a public-key crypto-system). Randomness extractors are algorithms developed “extract” high-quality random bits from low-entropy sources. Construction of such algorithms is foreseen to be an active research area.

The aim of Seminar 12021 was to bring together researchers covering this spectrum of relevant areas, to report their advances and to discuss the relevant research open questions. The seminar had 50 participants, including the most recognized senior specialists as well as young researchers. The atmosphere was very stimulating and led to new research contacts and collaborations.

Concluding remarks and future plans. The seminar was well received, as witnessed by the high rate of accepted invitations, and the exemplary degree of involvement by the participants. Due to the broad scope and depth of the problems on algorithmic randomness and information quantity that have been discussed in the presentations and informal discussions, the organizers regard the seminar as a great success. The organizers wish to express their gratitude towards the Scientific Directors of the Dagstuhl Center for their support of this seminar. We foresee the proposal of a new seminar focusing in the interplay between algorithmic randomness and computable analysis.

Description of the seminar topics

Anti-randomness

The class of sequences with minimal prefix-free Kolmogorov complexity, dubbed K-trivial sequences, were understudied until five years ago. In the seventies, Solovay proved that there is a non computable K-trivial. They are now very well understood, with a number of surprising characterizations and applications. For instance, the “cost function” construction of a K-trivial gives simplest known example of a non computable incomplete computably enumerable set, they also appear in the Kucera-Slaman solution to a well-known question about Turing degrees in Scott sets, also K-triviality has led to a better understanding of the reverse mathematics of the regularity of Lebesgue measure. K-triviality one of the most technically deep subjects in algorithmic randomness, significant questions remain open.

Resource bounded versions

Classical computational complexity theory comes into play defining resource-bounded versions of Kolmogorov complexity, measure, and dimension. This has led to new characterizations of complexity classes involving efficient reducibility to the set of Kolmogorov random strings. Resource-bounded measure and dimension have been used to gain understanding of properties of complexity classes and their complete sets. For instance, they can be used as a probabilistic methods to prove lower bounds on nonuniform complexity.

Derandomization and complexity hierarchies

Derandomization is the study of how to replace probabilistic algorithms with deterministic algorithms. Earlier work by Allender et al. showed that the techniques of derandomization could be viewed through the lens of resource-bounded Kolmogorov complexity theory, and gave significant applications. More recently, they proved that every sufficiently dense set in $NP \cap coNP$ contains strings of low resource-bounded Kolmogorov complexity at every length. In still unpublished work, Allender and his co-authors show that if deterministic and nondeterministic exponential time coincide, this implies a partial collapse of the exponential-time hierarchy, shedding light on a question that has been open for two decades.

Randomness extractors

Randomness extractors have been used and to derive zero-one laws for the packing dimensions of complexity classes and Turing degrees. Recently it has been shown that the converse direction also holds and Kolmogorov extraction is in fact equivalent to randomness extraction.

Computational depth

The computational depth of a string is roughly the difference between its time-bounded Kolmogorov complexity, and its (plain) Kolmogorov complexity. Quite recently, Antunes and Fortnow showed that, under a plausible complexity assumption, computational depth is the right notion to present a “universal” poly-time samplable distribution, in the same way that Kolmogorov complexity allows one to define universal computable semi-measures. They derive a new characterization of algorithms that run in polynomial time on average, and give a relation with their worst-case running time.

Algorithmic randomness and computable analysis

The most accepted definition of randomness for infinite sequences, or real numbers, is based on constructive measure theory and was given by Martin Lőf, 1965. It coincides with the maximal initial segment complexity. Other notions have been proposed since then, by Schnorr, Demuth, Kurtz and others, either via measure theory, or via martingale theory. Most of these definitions have been very well studied in the space of infinite binary sequences, but less is known for other spaces (although there has been some deep founding work by Levin and Gács). Some natural questions are: for a given randomness notion, to what kind of probability space can this notion be extended? To what extent does the chosen space affect the properties of random objects? Then, for every probability space to which we can extend randomness notions, it is interesting to look at classical theorems from a randomness perspective, and try to convert classical theorems of the form “property P holds for μ -almost every sequence” into “property P holds for every μ -random sequence”. This line of study has recently been investigated in a number of different settings: random closed sets, effective ergodic theory, effective brownian motion, etc.

Organization of the seminar and activities

The seminar consisted in nineteen talks, sessions on open questions, and informal discussions among the participants. The organizers selected the talks in order to have comprehensive lectures giving overview of main topics and communications of new research results. Each day consisted of talks and free time for informal gatherings among participants. There were two main sessions on open questions.

2 Table of Contents

Executive Summary

Verónica Becher, Laurent Bienvenu, Rodney Downey, Elvira Mayordomo 19

Overview of Talks





The Strange Link between Kolmogorov complexity and computational complexity classes <i>Eric Allender</i>	25
Kolmogorov complexity and computably enumerable sets <i>George Barmpalias</i>	25
Simple proofs for known inequalities on Kolmogorov complexity using games and symmetry of information <i>Bruno Bauwens</i>	26
Connections between ergodic theory and randomness <i>Johanna Franklin</i>	26
When is a graph random? <i>Cameron Freer</i>	26
Lowness in algorithmic randomness <i>Noam Greenberg</i>	27
Normality is equivalent to incompressibility by finite-state automata <i>Pablo A. Heiber</i>	27
Communication complexity through the lense of Kolmogorov complexity <i>Michal Koucký</i>	27
Constant compression and random weights <i>Wolfgang Merkle</i>	28
Randomness and Lebesgue density theorem <i>Joseph S. Miller</i>	28
Randomness extraction: a computability perspective <i>Benoit Monin</i>	28
Randomness interacts with effective analysis <i>Andre Nies</i>	29
Exponential time vs probabilistic polynomial time <i>Sylvain Perifel</i>	29
Semi-explicit expanders and extractors and their applications <i>Andrej E. Romashchenko</i>	30
Tutorial on randomness extractors <i>Ronen Shaltiel</i>	30
Are random axioms useful? <i>Alexander Shen</i>	30
The graph reachability problem <i>Vinodchandran Variyam</i>	31

Rate-distortion and denoising, of individual sequences by Kolmogorov complexity <i>Paul Vitányi</i>	31
Open Problems	
Questions on the link between Kolmogorov complexity and computational complexity classes <i>Eric Allender</i>	32
Kolmogorov complexity and computably enumerable sets <i>George Barmpalias</i>	32
Normal numbers computable in simple exponential time <i>Verónica Becher</i>	33
Relating computability and logical theories <i>Laurent Bienvenu</i>	33
Order functions and K -triviality <i>Noam Greenberg</i>	34
Questions on K -trivials <i>André Nies</i>	34
Questions on higher randomness <i>André Nies</i>	34
Extraction of mutual information about two strings <i>Alexander Shen</i>	35
Randomness with respect to a semimeasure <i>Alexander Shen</i>	35
What do probabilistic methods tell us about the finite sets? <i>Theodore Slaman</i>	36
On gales combined with computable exponential order functions <i>Ludwig Staiger</i>	36
van Lambalgen-type theorem for time-bounded Kolmogorov complexity <i>Marius Zimand</i>	37
Strong extractors for infinite sequences <i>Marius Zimand</i>	37
Participants	38

3 Overview of Talks

3.1 The Strange Link between Kolmogorov complexity and computational complexity classes

Eric Allender (Rutgers University – Piscataway, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Eric Allender

This talk will survey a body of work that has developed over the last decade, that has led some researchers to suspect that certain important computational complexity classes can be better understood, by studying the computational power of the set of Kolmogorov-random strings.

More specifically, let R denote the set of Kolmogorov-random strings. Let BPP denote the class of problems that can be solved with negligible error by probabilistic polynomial-time computations, and let NEXP denote the class of problems solvable in nondeterministic exponential time.





Conjecture 1: $\text{NEXP} = \text{NP}^R$.

Conjecture 2: BPP is the class of problems non-adaptively polynomial-time reducible to R .

These are not only bold conjectures; they are obviously false! R is not a decidable set, and thus it is absurd to suggest that the class of problems reducible to it constitutes a complexity class. The absurdity fades if, for example, we interpret “ NP^R ” to be “the class of problems that are NP-Turing reducible to R , no matter which universal machine we use in defining Kolmogorov complexity”. We are not yet able to prove that either conjecture (suitably interpreted) is true, but some recent theorems approach this goal. The lecture will highlight several problems that seem ripe for a fruitful blending of techniques from computability theory and complexity theory.

3.2 Kolmogorov complexity and computably enumerable sets

George Barmpalias (Chinese Academy of Sciences, CN)


License     Creative Commons BY-NC-ND 3.0 Unported license
© George Barmpalias

I will start with reporting a solution to a problem of Downey and Hirschfeldt from 2006 as well as further progress that I made on problems on the topic of Kolmogorov complexity of c.e. sets (in particular the structure of the c.e. K-degrees).

After this I will motivate this topic with several open questions which I find natural, yet I haven’t been able to solve.

3.3 Simple proofs for known inequalities on Kolmogorov complexity using games and symmetry of information

Bruno Bauwens (Universidade do Porto, PT)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Bruno Bauwens

First we provide a remarkably simple game-proof that for every n , there is an x of length n such that $C(C(x)|x) \geq \log n - O(1)$ and $C(x) \geq n/2$, slightly improving a result of Gacs and solving a conjecture of Chaitin and Solovay.

As an intermezzo we state symmetry of information for plain complexity as:


$$C(a, b) = K(a|C(a, b)) + C(b|a, C(a, b)),$$

which has two interesting known corollaries: Levin's formula $C(a) = K(a|C(a))$ (taking $b = C(a)$), and every infinitely often C-random real is 2-random.

Finally, we provide a short proof for Solovay's result (a bit improved) stating that for some strings plain complexity can be maximal but prefix-free complexity not. More precise: infinitely many strings x have $C(x) = |x| - O(1)$ and $K(x) = |x| + K(|x|) - \log \log |x| \pm O(1)$. The proof only uses symmetry of information of prefix-free complexity, and Levin's and Gács' results (see above).

3.4 Connections between ergodic theory and randomness

Johanna Franklin (Univ. Of Connecticut, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Johanna Franklin

Since randomness can be defined in terms of measure theory as well as Kolmogorov complexity, it is not surprising that it is related to other areas of mathematics where this concept is fundamental. In this talk, I will introduce the basic principles of ergodic theory, which is the study of the behavior of certain measure-preserving transformations over time, and explain the relationship between ergodic theory and randomness.

3.5 When is a graph random?

Cameron Freer (MIT – Cambridge, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Cameron Freer





Joint work of Ackerman, Nate; Freer, Cameron; Patel, Rehana; Roy, Daniel

What is the connection between quasi-random graphs and pseudorandom bit strings? Can this be used to develop a useful theory of resource-bounded complexity for discrete structures? In the first half of the talk, we will describe the translation by Trevisan between notions in additive combinatorics and computational indistinguishability, and also highlight the emerging theory of continuous limits of finite combinatorial structures that connects graph limits, property testing, and exchangeable relations.

When is a countably infinite graph algorithmically random? In some cases, there is a natural probabilistic construction of the graph that gives rise to an obvious candidate for randomness, but in other cases this is not so clear. In the second half of the talk, we will discuss invariant measures concentrated on a given countable structure, which induces a notion of an algorithmically random copy of that structure.

3.6 Lowness in algorithmic randomness





Noam Greenberg (Victoria University of Wellington, NZ)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Noam Greenberg

I will give a survey of the project of understanding lowness for notions of effective randomness, and pass through some related topics. Characterising a notion of lowness *usually* involves traceability, and is obtained by forcing with an adequate class of closed sets. This, however, fails for the most familiar notion of randomness, namely Martin-Löf's. In this case lowness is inherently enumerable – the opposite of being obtained by forcing. Instead, weakness as an oracle can be measured by interaction with the Turing degrees of random sets (à la Day and Miller, for example).

3.7 Normality is equivalent to incompressibility by finite-state automata





Pablo A. Heiber (University of Buenos Aires, AR)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Pablo A. Heiber

Recall that an infinite sequence over a finite alphabet Σ is *normal* if for any given n , all possible patterns of length n appear in the sequence with equal frequency. We will present a direct and elementary proof of the following fact: an infinite sequence is normal if and only if it cannot be compressed by a finite-state compressor (injective finite state transducer).

3.8 Communication complexity through the lense of Kolmogorov complexity


Michal Koucký (Academy of Sciences – Prague, CZ)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Michal Koucký

In this talk I will survey recent developments in communication complexity related to the notion of information cost and privacy. This development raises interesting questions in the context of Kolmogorov complexity.

3.9 Constant compression and random weights


Wolfgang Merkle (*Universität Heidelberg, DE*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Wolfgang Merkle

We introduce a new characterization of left recursively enumerable (left-r.e.) Martin-Löf random reals: a real is Martin-Löf random and recursively approximable from below if and only if it equals the weight of the compressible strings for some universal prefix-free machine. For sufficiently large intervals $[a; b)$, the weight of strings which are a -compressible strings but not b -compressible is a left-r.e. Martin-Löf random real, and in fact we can use finite intervals of compressibility to characterize the left-r.e. Martin-Löf randoms as well.

3.10 Randomness and Lebesgue density theorem

Joseph S. Miller (*University of Wisconsin – Madison, US*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Joseph S. Miller

Joint work of Bienvenu, Laurent; Day, Adam; Hölzl, Rupert; Miller, Joseph S.; Nies, André

In this talk we will present several recent results on the interactions between effective randomness a Lebesgue differentiability theorem. In joint work with Bienvenu, Hölzl and Nies, we show that a real x is a point of positive density in every Π_1^0 class it belongs to *if and only* it is Martin-Löf random and Turing incomplete (also known as *difference random*). In subsequent joint work with Day, this lead to a solution of a longstanding open question, namely, we prove that a real x is K -trivial if and only if for every incomplete random z , $x \oplus z$ is incomplete.

3.11 Randomness extraction: a computability perspective


Benoit Monin (*University Paris Diderot, FR*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Benoit Monin

Suppose you want to generate a random sequence of zeros and ones and all you have at your disposal is a coin which you suspect to be biased (but do not know the bias). Can “perfect” randomness be produced with this coin? The answer is positive, thanks to a little trick discovered by von Neumann. We will present a generalization of this question: if we have access to a source of bits produced according to some probability measure in a given class of measures, and suppose we know the class but not the measure, can perfect randomness be produced? We will give a positive answer for a large class of probability measures. (as Bernoulli measures or Markov measures). Furthermore, this work naturally has some interesting connections with the Kjos-Hanssen’s concept of Hippocratic randomness. We will actually provide another interesting characterisation of (some) classes of measures for which Hippocratic randomness and Martin-Löf randomness are equivalent.

3.12 Randomness interacts with effective analysis

Andre Nies (University of Auckland, NZ)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andre Nies

Joint work of Bienvenu, Laurent; Brattka, Vasco; Freer, Cameron; Hoelzl, Rupert; Kjos-Hanssen, Bjørn; Kucera, Antonin; Miller, Joseph S.; Nies, André

We seek connections between algorithmic randomness and computable analysis. Tests correspond to computable functions on the unit interval. A real passes a test if and only if the corresponding function is differentiable at the real. In this way, for instance we characterize computable randomness and Schnorr randomness via differentiability of effective Lipschitz functions ([1, 2]; also work of Pathak-Rojas-Simpson, and Rute). We include a historical perspective [3]. The constructivist Osvald Demuth, working on differentiability of effective functions, anticipated major algorithmic randomness notions in the 1970s and 1980. He introduced Demuth randomness which is in the focus of present-day research on lowness properties of oracles. However, in [4] we show that the weaker notion of difference randomness, due to Franklin and Ng already suffices for the application to constructive analysis Demuth had in mind.

We also discuss algorithmic versions of the ergodic theorem. Finally we mention the interaction of higher randomness and differentiability of hyperarithmetical functions.

References

- 1 Brattka, Miller, and Nies. Randomness and differentiability. Submitted.
- 2 Freer, Kjos-Hanssen and Nies. Effective aspects of Lipschitz functions. In preparation.
- 3 Kucera and Nies. Demuth's path to randomness. To appear.
- 4 Bienvenu, Hoelzl, Miller and Nies. The Denjoy alternative for computable functions. Submitted.

3.13 Exponential time vs probabilistic polynomial time


Sylvain Perifel (University Paris-Diderot, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Sylvain Perifel

People usually believe that probabilistic algorithms can be derandomized, meaning that randomness would not give additional power to polynomial-time algorithms. However our current knowledge is despairingly limited, not even ruling out the possibility that incredibly big complexity classes have polynomial probabilistic algorithms. More precisely, we don't know how to separate nondeterministic exponential time NEXP from probabilistic polynomial time BPP, even if we believe that $BPP=P$ (!). After presenting the state of the art, we shall discuss some attempts and strategies to resolve these questions and related circuit lower bounds. The tools will range from resource-bounded Kolmogorov complexity to interactive protocols.

3.14 Semi-explicit expanders and extractors and their applications


Andrej E. Romashchenko (CNRS, Université Montpellier II, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andrej E. Romashchenko

Explicit constructions of graphs with some “random” properties (e.g., expanders and extractors) are known to be a mighty tool in computer science. Despite an impressive progress in this area, the known effective constructions of such graphs still do not always match the parameters achievable by truly random graphs. We are going to discuss constructions of extractors and expanders where the combinatorial parameters are made better while the conventional requirement of “explicitness” is somehow relaxed, e.g., a graph should be constructed in polynomial space but not in polynomial time, or the property of expansion/randomness extraction should hold only for a tiny family of sets of vertices, or a construction may involve some reduced (but not negligible) random seed. We illustrate these methods with several applications: a version of Muchnik’s conditional complexity theorem (for space bounded Kolmogorov complexity), the optimal compression of sets in PSPACE, nearly optimal bit-probe schemes for membership problem (by recent papers of D.Musatov, A.Shen, M.Zimand and A.R.).

3.15 Tutorial on randomness extractors


Ronen Shaltiel (University of Haifa, IL)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Ronen Shaltiel

We give an introduction to the area of “randomness extraction” and survey the main concepts of this area: deterministic extractors, seeded extractors and multiple sources extractors. For each one we briefly discuss background, definitions, explicit constructions and applications.

3.16 Are random axioms useful?

Alexander Shen (Université de Provence, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Alexander Shen


The famous Gödel incompleteness theorem says that for every sufficiently rich formal theory there exist true unprovable statements. Such statements would be natural candidates for being added as axioms, but how can we obtain them? One classical (and well studied) approach is to add (to some theory T) an axiom that claims the consistency of T .

Here we discuss another approach (motivated by Chaitin’s version of the Gödel theorem) where axioms claiming randomness (incompressibility) of some strings are added, and show that it is not really useful (in the sense that it does not help us to prove new interesting theorems). This result answers a question recently asked by Lipton. However, the situation changes if we take into account the size of the proofs: randomly chosen axioms may help to make proofs much shorter (unless $NP=PSPACE$). This result (partially) answers the question asked a while ago by Shen. We also study what can be achieved by adding axioms of type

“complexity of x exceeds n ” for some strings x and numbers n . We show that by adding all true statements of this type, we obtain a theory that proves all true universal statements. Moreover, it is enough to add one statement of this type for each n (or even for infinitely many n) if strings are chosen in a special way. On the other hand, one may add statements of this type for most x of length n (for every n) still having a weaker theory. Finally, we consider a theory that claims Martin-Löf randomness of a given infinite binary sequence. This claim can be formalized in different ways. We show that different formalizations are closely related but not equivalent, and study their properties.

3.17 The graph reachability problem

Vinodchandran Variyam (University of Nebraska – Lincoln, US)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Vinodchandran Variyam

The graph reachability problem, the computational problem of deciding whether there is a path between two given vertices in a graph, is the canonical problem while studying space bounded computations.

Different variations of this problem characterize various important space bounded complexity classes. Understanding the complexity of the reachability problem is a central concern of computational complexity theory. In this talk I will revisit some well known open problems regarding the space complexity of the reachability problem and discuss certain approaches toward them.

3.18 Rate-distortion and denoising, of individual sequences by Kolmogorov complexity

Paul Vitanyi (CWI – Amsterdam, NL)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Paul Vitanyi


Joint work of de Rooij, Steven; Vereshchagin, Nikolay K.; Vitanyi, Paul

The canonical rate-distortion function of a single string is related to the more standard rate-distortion function of Shannon for the given distortion measure. Examples are Hamming distortion, List distortion, and Euclidean distortion. The rate-distortion function for individual sequences can and does assume a wide class of shapes (unlike Shannon’s). Low algorithmic mutual information is related to low Kolmogorov complexity. Destination words having lower distortion to the source word have more properties in common with the source word (hard or impossible to formalize in Shannon’s theory) and this suggests an approach to denoising.

4 Open Problems

4.1 Questions on the link between Kolmogorov complexity and computational complexity classes

Eric Allender (Rutgers University – Piscataway, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Eric Allender

Recall $\Delta_1^0 \cap \bigcap_U P_{dt}^{R_{C_U}} = P$, where R_{C_U} is the set of random strings using universal machine U : $R_{C_U} = \{x : C_U(x) \geq |x|\}$. We know that it is necessary to take the intersection over all universal machines U ; however, it is not obvious that the other intersection is necessary. This motivates the first two questions below:

Question 1: Does it hold that $\bigcap_U P_{dt}^{R_{C_U}} \subseteq \Delta_1^0$?

Question 2: Do there exist machines U_1, U_2 such that the two sets $R_{K_{U_1}}, R_{K_{U_2}}$ are minimal pairs with respect to \leq_{tt} or \leq_{wtt} ?

Question 3: Recall that there exists U such that the Halting problem H is not in $NP^{R_{K_U}}$. (This is not true if we consider plain Kolmogorov complexity C instead of prefix-free complexity K .) Show that this holds for *every* U .


Question 4: We know that, for all U and for all $t \ll 2^n$, $H \not\leq_{dt}^{Dtime(t)} R_{C_U}$. We also know that, for *some* U , H is dt-reducible to R_{C_U} in doubly-exponential time. Close this gap between exponential and doubly-exponential time.

Question 5: Hitchcock has shown that the exponential time class E contains sets that are not poly-time dt-reducible to R (no matter which universal machine one uses). Does this hold for small time bounds as well? That is, is it true for every superpolynomial $t(n)$, that $Dtime(t(n)) - P_{dt}^R \neq \emptyset$?

Question 6: We know that, for every decidable set A outside PSPACE, there is some U such that $A \notin P_{tt}^{R_{K_U}}$; thus in particular $H \notin P_{tt}^{R_{K_U}}$. Show that this holds for C -complexity as well. That is, show there is a U such that $H \notin P_{tt}^{R_{C_U}}$. [Then try to show that this is true for *every* U .]

4.2 Kolmogorov complexity and computably enumerable sets

George Barmpalias (Chinese Academy of Sciences, CN)

License  Creative Commons BY-NC-ND 3.0 Unported license
© George Barmpalias

Question: Is there a pair of sequences x, y which are not K -trivial and

$$\min(K(x \upharpoonright n), K(y \upharpoonright n)) \leq K(n) + c?$$

Question: Is there a c.e. set where the initial segment complexity is maximal amongst the c.e. sets? The same question holds for the global structure of \leq_K (Miller and Yu).





Also the same question holds for the set of non random strings.

Question: What is the algorithmic independence of c.e. sets? Compare with the work of Levin, Calude and Zimmand on algorithmic independence.

Question: Recall that an order is a strictly increasing computable function $f : \mathbb{N} \rightarrow \mathbb{N}$. Let $X_f = \{f(n) \mid n \in X\}$. X is K -invariant under f if $X \equiv_K X_f$. Characterize their degrees (called K -resolute sequences).

4.3 Normal numbers computable in simple exponential time

Verónica Becher (*Universidad de Buenos Aires, AR*)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Verónica Becher

It is fair to say that Borel's question on providing an example of an absolutely normal number (normal to every integer base) is still unresolved because the few known instances are not completely satisfactory: it is desirable that the number be easily computable, we would like to exhibit the number explicitly.





Turing's algorithm and the computable reformulation of Sierpiński's work are the only known constructions of computable normal numbers. Unfortunately, they both require double exponentially many steps to produce a next digit of the expansion of a constructed number. The existence of normal numbers computable in simple exponential time is ensured by a theorem of Strauss in [1]; however, no specific instances have yet been identified.

References

- 1 Strauss, Martin, 1997. Normal numbers and sources for BPP. *Theoretical Computer Science* 178, 155-169.

4.4 Relating computability and logical theories

Laurent Bienvenu (*Université Paris-Diderot, FR*)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Laurent Bienvenu

Following A. Shen's talk, here are some interesting open questions about the axiomatic power of Kolmogorov complexity:

Question: Is it possible to find an example when some information about Kolmogorov complexity gives us the power to compute \emptyset' , yet not allowing us, on a proof-theoretic level, to prove all true Π_1^0 -statements?


Question: We know from Chaitin's theorem that one can only prove finitely many statements of type " $C(x) > n$ ". How about statements of type " $C(x) \notin [n_1, n_2]$ "?

Question: Can one give a characterization of the sequences (x_n) of strings such that $x_n \in 2^n$ and $C(x_n) \geq n$ such that, adding for each n the axiom " $C(x_n) \geq n$ " for each n , we can prove all true Π_1^0 -statements?

Question: Is there a sequence (x_n) of strings such that $x_n \in 2^n$ and $C(x_n) \geq n$ such that, adding for each n the axiom " $C(x_n) \geq n/2$ " for each n , we can prove all true Π_1^0 -statements?

4.5 Order functions and K -triviality

Noam Greenberg (Victoria University of Wellington, NZ)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Noam Greenberg


The goal is to find a combinatorial (or discrete) characterisation of K -triviality. That is, one that does not mention measure, Kolmogorov complexity, or randomness. Such dual characterisations are available for example for lowness for Schnorr randomness and for strong jump-traceability.

One possible approach is via traceability. Let h be an order function (a computable, non-decreasing, and unbounded function from ω to $\omega - \{0\}$). Recall that a Turing degree \mathbf{a} is h -jump-traceable if every \mathbf{a} -partial computable function has a c.e. trace bounded by h . The aim is to identify a collection \mathcal{H} of order functions such that a degree is K -trivial if and only if it is h -jump-traceable for all $h \in \mathcal{H}$. We have some approximations of such a result. For example, it is known that if \mathbf{a} is $\sqrt{\log n}/9$ -jump-traceable then it is K -trivial; and that every K -trivial degree is $O(h)$ -jump-traceable for any summable order function h ($\sum 2^{-h(n)} < \infty$). The latter result comes from a characterisation of K -triviality (by Hölzl, Kräling and Merkle) using jump-traceability with respect to a collection of bounds which is defined using Solovay functions and Kolmogorov complexity K .

The dividing line may be some constant multiple of the logarithm function. Here we have a related result: if every K -trivial degree is $(\log n)/10$ -jump-traceable, then there is no minimal pair of LR-hard c.e. degrees.

4.6 Questions on K -trivials

André Nies (University of Auckland, NZ)

License  Creative Commons BY-NC-ND 3.0 Unported license
© André Nies

Question (open since 2005): Let A be K -trivial. Is there a T -incomplete Martin Löf random Z such that $Z \geq_T A$?

Question (open since 2006): Let \mathcal{K} be the ideal of K -trivial degrees. Are there c.e. \mathbf{a}, \mathbf{b} such that $\mathcal{K} = [\mathbf{0}, \mathbf{a}] \cap [\mathbf{0}, \mathbf{b}]$?

Question (open since 2011): A function $f : \omega \rightarrow \omega$ is K -trivial if there is c such that $\forall n [K(f \upharpoonright n) \leq K(0^n) + c]$. Can we compute, from a K -trivial constant from the graph of f (as a set) a K -trivial constant for f ?

4.7 Questions on higher randomness

André Nies (University of Auckland, NZ)

License  Creative Commons BY-NC-ND 3.0 Unported license
© André Nies

Recall the definitions:

Z is Π_1^1 -random if Z is in no null Π_1^1 set.

Z is higher weakly 2-random if Z passes all Π_1^1 weak 2-tests (i.e., $Z \notin \bigcap_m G_m$, where “ $[\sigma] \subseteq G_m$ ” is Π_1^1 , and $\lim_{m \rightarrow \infty} \lambda G_m = 0$).

Question (open since 2005): Is there a non hyperimmune set that is low for Π_1^1 -random?





Question (posed in Chapter 9, *Computability and Randomness*, A. Nies, Oxford University Press, 2009): Show the properness of these implications.

Π_1^1 -random \Rightarrow higher weakly 2-random \Rightarrow Π_1^1 -Martin Lőf random.

The last implication was recently announced by Yu Liang.

4.8 Extraction of mutual information about two strings

Alexander Shen (*Université de Provence, FR*)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Alexander Shen





Let A_1, \dots, A_n be a tuple of strings. If X is a random oracle, with high probability it does not change significantly the complexities of A_i , of pairs (A_i, A_j) , etc. The question is whether the same is true for other properties expressed in terms of complexity.

A specific question: assume that for a random X the strings A_1, A_2 have common information (extractable mutual information): there exists a string B such that $C(B|A_1, X) \approx 0$, $C(B|A_2, X) \approx 0$, and $C(B|X) \approx I(A_1 : A_2|X)$. Is the same true without an oracle?

Another question about oracles and tuples of strings: is it always possible for given A_1, \dots, A_n to find some oracle X such that $C(A_i|X) \approx 0.5 C(A_i)$?

4.9 Randomness with respect to a semimeasure

Alexander Shen (*Université de Provence, FR*)


License     Creative Commons BY-NC-ND 3.0 Unported license
© Alexander Shen

Let L be some probabilistic machine that uses the internal random bits generator to produce a sequence of output bits. Such a machine L has an output distribution which corresponds to a semimeasure: $l(x)$ equals the probability that the output of L has x as a prefix. In this way we can obtain all semimeasures on the binary tree (lower semicomputable functions on finite strings with nonnegative values such that $l(\Lambda) = 1$ for the empty string Λ and $l(x) \geq l(x0) + l(x1)$ for every string x). Now consider the infinite outputs of L for all Martin-Lőf random sequences used as random bits.

Question: is this set of sequences determined by l or different machines with the same output distributions can lead to different sets? (If determined by l , this set can be considered as the set of random sequences with respect to a semimeasure l . This would extend the Martin-Lőf definition of randomness to semimeasures.)

4.10 What do probabilistic methods tell us about the finite sets?

Theodore Slaman (University of California – Berkeley, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Theodore Slaman

I would like to propose an investigation of the heuristic question, “What do probabilistic methods tell us about the finite sets?” For this, we would like both lower bounds, saying that certain properties P of the finite sets can be established by probabilistic methods, and upper bounds, saying that any theorem about the finite sets established probabilistically has an alternate proof from purely number-theoretic properties Q . Still speaking informally, we would like to know the power of and limitations on probabilistic methods as applied to number-theoretic questions.

For example, we might express a version of this question using the formalism of second-order arithmetic, in which one has the language appropriate to express properties of the natural numbers n with addition, multiplication, and order, and also to refer to subsets X of the natural numbers with the relation “element of” allowing formulas of the form “ n is an element of X .” It is standard to use the theory RCA_0 to formalize computable methods, where RCA_0 includes the basic properties of $+$ and $\times(P-)$, the principle of induction for Σ_1^0 sets of numbers (to allow for the definition of total computable functions by recursion), and the property that the sets of numbers are closed under relative computation.

Now consider augmenting RCA_0 by postulating the existence of relative random reals. Let 1 – RAN be the formal statement that for every set X there is a set R which is Martin-Löf relative to X . Let 2 – RAN be the analogous statement for 2-randoms. Applying a theorem of Harrington, if φ is an arithmetic sentence which is provable from “ $RCA_0 + 1 - RAN$,” then φ is provable from RCA_0 . In other words, the use of randomness can be eliminated.


Recent results with Conidis, show that there is an arithmetic sentence which is provable from “ $RCA_0 + 2 - RAN$ ” and not provable from RCA_0 . So, this use of randomness cannot be eliminated. However, if φ is an arithmetic sentence which is provable from “ $RCA_0 + 2 - RAN$,” then φ is provable from “ $RCA_0 + B - \Sigma_2^0$.” Here, $B - \Sigma_2$ is the assertion that if F is a finite set and ψ is a Σ_2^0 formula relative to the set X such that ψ holds for every number in F , then there is a bound on the existential witnesses needed to verify ψ on F .

Specific question: It is also known that $B - \Sigma_2$ is not provable from “ $RCA_0 + 2 - RAN$,” and it would be very interesting to obtain a natural number-theoretic axiomatization of the number-theoretic consequences of “ $RCA_0 + 2 - RAN$.” The same is true for “ $RCA_0 + k - RAN$,” for larger values of k .

Heuristic question. Identify the natural contexts, beyond purely computable, in which randomness is used to shed light on the finite and determine in which cases the arguments based on concepts of measure and randomness cannot be removed.

4.11 On gales combined with computable exponential order functions

Ludwig Staiger (Martin-Luther-Universität Halle-Wittenberg, DE))

License  Creative Commons BY-NC-ND 3.0 Unported license
© Ludwig Staiger





Lutz’s s -(super-)gales are (super-)martingales combined with exponential order functions. They are mainly considered as computable or left-computable functions having a (weakly) com-

putable value of s . This corresponds to computable or left-computable (super-)martingales combined with (weakly) computable exponential order functions.

Question: Are there computable or left-computable s -(super-)gales for non-(weakly) computable values of s which are not s' -(super-)gales for a value $s' < s$?

4.12 van Lambalgen-type theorem for time-bounded Kolmogorov complexity

Marius Zimand (Towson University, US)





License     Creative Commons BY-NC-ND 3.0 Unported license
© Marius Zimand

For unrestricted Kolmogorov complexity, it holds that if we put together two sequences (or strings) such that each one of them is random given the other one the result is random. More precisely if $x \in \{0, 1\}^\omega$ is (Martin-Löf, Schnorr, computable) random conditioned by y , and $y \in \{0, 1\}^\omega$ is random conditioned by x , then $x \oplus y$ is random (van Lambalgen Theorem). The same holds for finite strings x and y that are c -random conditioned by each other (meaning $C(x | y) \geq |x| - c$, $C(y | x) \geq |y| - c$, and also if we replace C by K). For time-bounded Kolmogorov complexity this question is open. More precisely, the question is:

Question: Let x, y be n -bit strings such that for some constant c and some polynomial-time bound $p(n)$, $C^{p(n)}(x | y) \geq n - c$ and $C^{p(n)}(y | x) \geq n - c$. What can we say about the $C^{poly(n)}(xy)$? (Perhaps, under some computational complexity assumption, one can show that it is $\ll 2n$.)

4.13 Strong extractors for infinite sequences

Marius Zimand (Towson University, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Marius Zimand

It is known that Kolmogorov extractors for two independent sequences exist. For example there exists a Turing reduction (even truth-table reduction) such that for each sequences x and y that have each effective dimension, say $1/2$, and are independent, it holds that that $f^{x \oplus y}$ has effective dimension 1.

Question: Is it possible to have a Turing-reduction f such that for all x and y as above, computes a sequence that has effective dimension 1 even conditioned by x , and also conditioned by y ?

For x and y finite strings (or finite distributions) the corresponding f exists and is called strong Kolmogorov extractor (and respectively strong extractor).

Participants

- Eric Allender
Rutgers Univ. – Piscataway, US
- Klaus Ambos-Spies
Universität Heidelberg, DE
- George Barmpalias
Chinese Academy of Sciences, CN
- Bruno Bauwens
Universidade do Porto, PT
- Verónica Becher
University of Buenos Aires, AR
- Laurent Bienvenu
University Paris-Diderot, FR
- Harry Buhrman
CWI – Amsterdam, NL
- Douglas Cenzer
University of Florida –
Gainesville, US
- Chris J. Conidis
University of Waterloo, CA
- Quinn Culver
Univ. of Notre Dame, US
- David Diamondstone
Victoria Univ. of Wellington, NZ
- Rodney Downey
Victoria Univ. of Wellington, NZ
- Lance Fortnow
Northwestern University –
Evanston, US
- Johanna N. Y. Franklin
Univ. Of Connecticut, US
- Cameron Freer
MIT – Cambridge, US
- Noam Greenberg
Victoria Univ. of Wellington, NZ
- Serge Grigorieff
University Paris-Diderot, FR
- Pablo A. Heiber
University of Buenos Aires, AR
- John Hitchcock
University of Wyoming, US
- Rupert Hölzl
University Paris-Diderot, FR
- Michal Koucký
Academy of Sciences –
Prague, CZ
- Thorsten Kräling
Universität Heidelberg, DE
- Antonin Kucera
Charles University – Prague, CZ
- Sophie Laplante
INRIA Saclay – Orsay, FR
- Andrew Lewis
University of Leeds, GB
- Bruno Loff
CWI – Amsterdam, NL
- Elvira Mayordomo
University of Zaragoza, ES
- Wolfgang Merkle
Universität Heidelberg, DE
- Joseph S. Miller
University of Wisconsin –
Madison, US
- Benoit Monin
University Paris-Diderot, FR
- Philippe Moser
Nat. University of Ireland, IE
- Satyadev Nandakumar
Indian Inst. of Technology –
Kanpur, IN
- Andre Nies
University of Auckland, NZ
- Sylvain Perifel
University Paris-Diderot, FR
- Christopher P. Porter
Univ. of Notre Dame, US
- Robert Rettinger
FernUniversität in Hagen, DE
- Andrej E. Romashchenko
CNRS, Univ. Montpellier II, FR
- Ronen Shaltiel
University of Haifa, IL
- Alexander Shen
Université de Provence, FR
- Theodore A. Slaman
University of California –
Berkeley, US
- Ludwig Staiger
Martin-Luther-Universität
Halle-Wittenberg, DE
- Antoine Tavenaux
University Paris-Diderot, FR
- Leen Torenvliet
University of Amsterdam, NL
- Daniel Turetsky
Victoria Univ. of Wellington, NZ
- Vinodchandran N. Variyam
Univ. of Nebraska – Lincoln, US
- Stijn Vermeeren
University of Leeds, GB
- Paul M. B. Vitanyi
CWI – Amsterdam, NL
- Vladimir Viyugin
IITP – Moscow, RU
- Osamu Watanabe
Tokyo Institute of Technology, JP
- Marius Zimand
Towson University, US



Symmetric Cryptography

Edited by

Frederik Armknecht¹, Stefan Lucks², Bart Preneel³, and
Phillip Rogaway⁴

1 Universität Mannheim, DE, armknecht@uni-mannheim.de

2 Bauhaus-Universität Weimar, DE, stefan.lucks@uni-weimar.de

3 K.U. Leuven, BE, Bart.Preneel@esat.kuleuven.be

4 University of California, Davis, US, rogaway@cs.ucdavis.edu

Abstract

From 15.01.2012 to 20.01.2012, the Seminar 12031 in *Symmetric Cryptography* was held in Schloss Dagstuhl–Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Seminar 15.–20. January, 2012 – www.dagstuhl.de/12031

1998 ACM Subject Classification E.3 Data Encryption.

Keywords and phrases Hash functions, Feistel networks, BLAKE, KLEIN, Keccak, IDEA, GCM, EAXprime, TLS, KISS

Digital Object Identifier 10.4230/DagRep.2.1.39

Edited in cooperation with Ewan Fleischmann


1 Executive Summary

Frederik Armknecht

Stefan Lucks

Bart Preneel

Phillip Rogaway

License  Creative Commons BY-NC-ND 3.0 Unported license
© Armknecht, Frederik; Lucks, Stefan; Preneel, Bart; Rogaway, Phillip

Research in Symmetric Cryptography is quickly evolving. The seminar was the third of its kind, the first one took place in 2007, the second in 2009. We observe a steadily increasing interest in Symmetric Cryptography, as well as a growing practical demand for symmetric algorithms and protocols. The seminar was very successful in discussing recent results and sharing new ideas. Furthermore, it inspired the participants to consider how Symmetric Cryptography has evolved in the past, and how they would like it to evolve in the future.

Two intense discussions dealt with Authenticated Encryption and the issue of a 'valid' attack on a symmetric primitive. The participants agreed on Authenticated Encryption becoming a major research topic for Symmetric Cryptography in the next few years, because current Authenticated Encryption Schemes are not always suitable for practical demands – especially are the relevant attack modes and models not yet well-understood (e.g., misuse attacks, blockwise adaptive attacks, etc.). Regarding the issue of 'valid' attacks, the participants



Except where otherwise noted, content of this report is licensed
under a Creative Commons BY-NC-ND 3.0 Unported license

Symmetric Cryptography, *Dagstuhl Reports*, Vol. 2, Issue 1, pp. 39–49

Editors: Frederik Armknecht, Stefan Lucks, Bart Preneel, and Phillip Rogaway



DAGSTUHL
REPORTS

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

agreed that the current development of academic cryptanalysis with a growing number of increasingly 'marginal' attacks, is unsatisfactory.

2 Table of Contents

Executive Summary

Armknecht, Frederik; Lucks, Stefan; Preneel, Bart; Rogaway, Phillip 39

Overview of Talks

BLAKE SIMD: past, present, future

Jean-Philippe Aumasson 42

Attacking KLEIN

Jean-Philippe Aumasson 42

Practical Collisions in Round-Reduced Keccak

Itai Dinur 43

Getting Results under Weak Expectations

Yevgeniy Dodis 43

An IDEA to Consider

Orr Dunkelman 43

Oracle Reducibility of Hash Functions

Marc Fischlin 44

GCM Security, Revisited

Tetsu Iwata 44

Cryptanalysis of EAXprime

Tetsu Iwata 45

On The Distribution of Linear Biases: Three Instructive Examples

Gregor Leander 45

New Results on EAX-Prime

Stefan Lucks 46

The Preimage Security of Double-Block-Length Compression Functions

Frederik Armknecht 46

Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol

Kenneth G. Paterson 46

KISS: A Bit Too Simple

Greg Rose 47

Bounds for Balanced Feistel Networks

Kyoji Shibutani 47

Collisions are not Incidental: A Compression Function Exploiting Discrete Geometry


Martijn Stam 48

Participants 49

3 Overview of Talks

3.1 BLAKE SIMD: past, present, future

Jean-Philippe Aumasson (Nagravision – Cheseaux, CH)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jean-Philippe Aumasson

Joint work of Aumasson, Jean-Philippe; Neves, Samuel


Main reference Third SHA-3 Conference (to appear)

The SHA-3 candidate hash function BLAKE is based on a keyed permutation whose data-level parallelism allows implementers to exploit SIMD instructions sets, as available in popular general-purpose processors. We will first review previous implementations that used Intel’s streaming SIMD extensions (SSE), as well as recent implementations using ARM’s NEON SIMD instruction set.

We will then present the recent 256-bit-wide AVX and the upcoming AVX2 extensions (expected in 2013 in Intel’s Haswell microarchitecture) and how we used them to write new assembly implementations of BLAKE.

3.2 Attacking KLEIN

Jean-Philippe Aumasson (Nagravision – Cheseaux, CH)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jean-Philippe Aumasson

Joint work of Aumasson, Jean-Philippe; María Naya-Plasencia; Markku-Juhani O. Saarinen

Main reference J.-P. Aumasson, M. Naya-Plasencia, M.-J. O. Saarinen, “Practical Attack on 8 Rounds of the Lightweight Block Cipher KLEIN,” INDOCRYPT 2011, pp. 134–145, LNCS, vol. 7107, Springer, 2011.

URL http://dx.doi.org/10.1007/978-3-642-25578-6_11

KLEIN is a family of lightweight block ciphers presented at RFIDSec 2011 that combines a 4-bit Sbox with Rijndael’s byte-oriented MixColumn. This approach allows compact implementations of KLEIN in both low-end software and hardware. We show that interactions between those two components lead to the existence of differentials of unexpectedly high probability: using an iterative collection of differential characteristics and neutral bits in plaintexts, we find conforming pairs for four rounds with amortized cost below 2^{12} encryptions, whereas at least 2^{30} was expected by the preliminary analysis of KLEIN. We exploit this observation by constructing practical ($\approx 2^{35}$ encryptions), experimentally verified, chosen-plaintext key-recovery attacks on up to 8 rounds of KLEIN-64 – the instance of KLEIN with 64-bit keys and 12 rounds. We also investigate the extension of the attack to 9 rounds.

3.3 Practical Collisions in Round-Reduced Keccak

Itai Dinur (Weizmann Institute – Rehovot, IL)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Itai Dinur

Main reference I. Dinur, O. Dunkelman, A. Shamir, “New attacks on Keccak-224 and Keccak-256,” Cryptology ePrint Archive: Report 2011/624.

URL <http://eprint.iacr.org/2011/624.pdf>

The Keccak hash function is one of the five finalists in NIST’s SHA-3 competition, and so far it showed remarkable resistance against practical collision finding attacks: After several years of cryptanalysis and a lot of effort, the largest number of Keccak rounds for which actual collisions were found was only 2.

We describe improved collision finding techniques which enable us to double this number. More precisely, we can now find within a few minutes on a single PC actual collisions in standard Keccak-224 and Keccak-256, where the only modification is to reduce their number of rounds to 4. When we apply our techniques to 5-round Keccak, we can get in a few days excellent near collisions, where the Hamming distance is 5 in the case of Keccak-224 and 10 in the case of Keccak-256. Our new attack combines differential and algebraic techniques, and uses the fact that each round of Keccak is only a quadratic mapping in order to efficiently find pairs of messages which follow a high probability differential characteristic.

3.4 Getting Results under Weak Expectations

Yevgeniy Dodis (New York University, US)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Yevgeniy Dodis

Recently, there has been renewed interest in basing cryptographic primitives on weak secrets, where the only information about the secret is some non-trivial amount of (min-)entropy.

From a formal point of view, such results require to upper bound the expectation of some function $f(X)$, where X is a weak source in question. We show an elementary inequality which essentially upper bounds such “weak expectation” by two terms, the first of which is *independent* of f , while the second only depends on the “variance” of f under *uniform* distribution. Quite remarkably, as relatively simple corollaries of this elementary inequality, we obtain some “unexpected” results, in several cases noticeably simplifying/improving prior techniques for the same problem. Examples include non-malleable extractors, leakage-resilient symmetric encryption, seed-dependent condensers and improved entropy loss for the leftover hash lemma.

3.5 An IDEA to Consider

Orr Dunkelman (University of Haifa, IL)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Orr Dunkelman

Joint work of Biham, Eli; Keller, Nathan; Shamir, Adi


IDEA is a 64-bit block cipher with 128-bit keys which is widely used due to its inclusion in several cryptographic packages such as PGP. After its introduction by Lai and Massey in

1991, it was subjected to an extensive cryptanalytic effort, but so far the largest variant on which there are any published attacks contains only 6 of its 8.5-rounds. The first 6-round attack, described in the conference version of this paper in 2007, was extremely marginal: It required essentially the entire codebook, and saved only a factor of two compared to the time complexity of exhaustive search.

In 2009, Sun and Lai reduced the data complexity of the 6-round attack from 2^{64} to 2^{49} chosen plaintexts and simultaneously reduced the time complexity from 2^{127} to $2^{112.1}$ encryptions. In this revised version of our paper, we combine a highly optimized meet-in-the-middle attack with a keyless version of the Biryukov-Demirci relation to obtain new key recovery attacks on reduced-round IDEA, which dramatically reduce their data complexities and increase the number of rounds to which they are applicable. In the case of 6-round IDEA, we need only two known plaintexts (the minimal number of 64-bit messages required to determine a 128-bit key) to perform full key recovery in $2^{123.4}$ time. By increasing the number of known plaintexts to sixteen, we can reduce the time complexity to $2^{111.9}$, which is slightly faster than the Sun and Lai data-intensive attack. By increasing the number of plaintexts to about one thousand, we can now attack 6.5 rounds of IDEA, which could not be attacked by any previously published technique. By pushing our techniques to extremes, we can attack 7.5 rounds using 2^{63} plaintexts and 2^{114} time, and by using an optimized version of a distributive attack, we can reduce the time complexity of exhaustive search on the full 8.5-round IDEA to $2^{126.8}$ encryptions using only 16 plaintexts.

3.6 Oracle Reducibility of Hash Functions

Marc Fischlin (TU Darmstadt, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Marc Fischlin

Recently, Baecher and Fischlin (Crypto 2011) used the notion of random oracle reducibility to relate the random oracles in different schemes. Roughly, a random oracle in a scheme B reduces to that in another scheme A if any (oracle-based or standard-model based) instantiation of the hash function making scheme A secure, also makes scheme B secure.

Here we discuss that the same idea applies to other oracle objects such as the ideal cipher model. In particular, we look at the constructions of hash functions (resp. compression functions) out of ideal ciphers, and how the ideal ciphers in different constructions such as the PGV schemes, or (Tandem-)DM compared to Hirose, relate. Our results concerning reducibility are partially positive, and in some cases negative, showing that the hash function constructions rely on different properties of the cipher.

3.7 GCM Security, Revisited

Tetsu Iwata (Nagoya University, JP)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Tetsu Iwata

Joint work of Iwata, Tetsu; Ohashi, Keisuke; Minematsu, Kazuhiko

GCM is the authenticated encryption mode developed by McGrew and Viega. In 2007, GCM was adopted as a recommendation mode by NIST, and it is widely used in practice.

The designers presented proofs of security, and despite extensive security analyses by the cryptographic community, its provable security results are considered to be sound.

In this talk, we revisit the provable security results of GCM, and discuss in detail their correctness.

3.8 Cryptanalysis of EAXprime

Tetsu Iwata (Nagoya University, JP)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Tetsu Iwata

Joint work of Minematsu, Kazuhiko; Morita, Hiraku; Iwata, Tetsu

Main reference K. Minematsu, S. Lucks, H. Morita, T. Iwata, “Cryptanalysis of EAXprime,” Cryptology ePrint Archive: Report 2012/018.

URL eprint.iacr.org/2012/018.pdf

EAX’ (EAXprime) is an authenticated encryption (AE) specified by ANSI C12.22 as a standard security function used for a smart grid. EAX’ is based on EAX, a provably secure AE proposed by Bellare, Rogaway, and Wagner.

In this talk, we present simple and efficient forgery and distinguishing attacks against EAX’ using one-block cleartext and plaintext.

3.9 On The Distribution of Linear Biases: Three Instructive Examples

Gregor Leander (Technical University of Denmark – Lyngby, DK)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Gregor Leander


Joint work of Abdelraheem, Mohamed Ahmed; Ågren, Martin; Beelen, Peter; Leander, Gregor

Despite the fact that we evidently have very good block ciphers at hand today, many fundamental questions on their security are still unsolved.

One such fundamental problem is to precisely assess the security of a given block cipher with respect to linear cryptanalysis. In by far most of the cases we have to make (clearly wrong) assumptions, e.g., assume independent round-keys. Besides being unsatisfactory from a scientific perspective, the lack of fundamental understanding has a direct consequence on the performance of the ciphers we use. As we do not understand the security sufficiently enough, we are forced to embed a security margin – from an efficiency perspective nothing else than wasted performance. The aim of this paper is to stimulate research on the fundamental lack of understanding of block ciphers. We do this by presenting three examples of ciphers that behave differently to what is normally assumed. Thus, on the one hand these examples serve as counter examples to common beliefs and on the other hand serve as a guideline for future work.

3.10 New Results on EAX-Prime

Stefan Lucks (Bauhaus-Universität Weimar, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Stefan Lucks


Main reference K. Minematsu, S. Lucks, H. Morita, T. Iwata, “Cryptanalysis of EAXprime,” Cryptology ePrint Archive: Report 2012/018.

URL eprint.iacr.org/2012/018.pdf

Starting from previous results presented by Tetsu Iwata at this Seminar, we present an improved cryptanalysis of EAX-Prime. The main observation is that the forgery attacks presented by Tetsu can be extended and turned into Chosen Ciphertext Message Recovery Attacks. These results have been found during the Seminar.

3.11 The Preimage Security of Double-Block-Length Compression Functions

Frederik Armknecht (University Mannheim, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Frederik Armknecht

Joint work of Fleischmann, Ewan; Krause, Matthias; Lee, Jooyoung; Stam, Martijn; Steinberger, John P.


Main reference F. Armknecht, E. Fleischmann, M. Krause, J. Lee, M. Stam, J. Steinberger, “The Preimage Security of Double-Block-Length Compression Functions,” ASIACRYPT 2011, pp. 233–251, LNCS, vol. 7073, Springer, 2011.

URL http://dx.doi.org/10.1007/978-3-642-25385-0_13

We present new techniques for deriving preimage resistance bounds for block cipher based double-block-length, double-call hash functions. We give improved bounds on the preimage security of the three ‘classical’ double-block-length, double-call, block cipher-based compression functions, these being Abreast-DM, Tandem-DM and Hirose’s scheme. For Hirose’s scheme, we show that an adversary must make at least 2^{2n-5} block cipher queries to achieve chance 0.5 of inverting a randomly chosen point in the range. For Abreast-DM and Tandem-DM we show that at least 2^{2n-10} queries are necessary. These bounds improve upon the previous best bounds of $\Omega(2^n)$ queries, and are optimal up to a constant factor since the compression functions in question have range of size 2^{2n} .

3.12 Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol

Kenneth G. Paterson (RHUL – London, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Kenneth G. Paterson

Joint work of Paterson, Kenneth G.; Ristenpart, Tom; Shrimpton, Tom

Main reference K.G. Paterson, T.E. Shrimpton, T. Ristenpart, “Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol,” Asiacypt 2011, pp. 372-389, LNCS, vol. 7073, Springer, 2011.

URL http://dx.doi.org/10.1007/978-3-642-25385-0_20

We analyze the security of the TLS Record Protocol, a MAC-then-Encode-then-Encrypt (MEE) scheme whose design targets confidentiality and integrity for application layer communications on the Internet. Our main results are twofold. First, we give a new distinguishing

attack against TLS when variable length padding and short (truncated) MACs are used. This combination will arise when standardized TLS 1.2 extensions (RFC 6066) are implemented.

Second, we show that when tags are longer, the TLS Record Protocol meets a new length-hiding authenticated encryption security notion that is stronger than IND-CCA.

3.13 KISS: A Bit Too Simple

Greg Rose (Qualcomm Inc. – San Diego, US)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Greg Rose

Main reference G. Rose, “KISS: A Bit Too Simple,” Cryptology ePrint Archive: Report 2011/007.

URL <http://eprint.iacr.org/2011/007.pdf>

KISS (‘Keep it Simple Stupid’) is an efficient pseudo-random number generator specified by G. Marsaglia and A. Zaman in 1993. G. Marsaglia in 1998 posted a C version to various USENET newsgroups, including `sci.crypt`. Marsaglia himself has never claimed cryptographic security for the KISS generator, but many others have made the intellectual leap and claimed that it is of cryptographic quality. In this paper we show a number of reasons why the generator does not meet the KISS authors’ claims, why it is not suitable for use as a stream cipher, and that it is not cryptographically secure. Our best attack requires about 70 words of generated output and a few hours of computation to recover the initial state. A further attack on a newer version of KISS is also presented.

3.14 Bounds for Balanced Feistel Networks




Kyoji Shibutani (Sony – Tokyo, JP)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Kyoji Shibutani

Feistel ciphers are among the most popular block cipher constructions in use today. We explore the optimality of balanced Feistel networks with SP-type F-functions with respect to their resistance against differential and linear cryptanalysis. Instantiations of Feistel ciphers with the wide class of (SP)u and (SP)uS F-functions is considered: One F-function can contain an arbitrary number of S-box layers interleaved with linear diffusion. For the matrices with maximum diffusion, it is proven that SPS and SPSP F-functions are optimal in terms of the proportion of active S-boxes in all S-boxes – a common efficiency metric for substitution-permutation ciphers. Interestingly, one SP-layer in the F-function is not enough to attain optimality whereas taking more than two S-box layers does not increase the efficiency either.

3.15 Collisions are not Incidental: A Compression Function Exploiting Discrete Geometry

Martijn Stam (University of Bristol, GB)

License    Creative Commons BY-NC-ND 3.0 Unported license

© Martijn Stam

Joint work of Jetchev, Dimitar; Özen, Onur; Stam, Martijn

We present a new construction of a compression function $h: \{0, 1\}^{3n} \rightarrow \{0, 1\}^{2n}$ that uses two parallel calls to an ideal primitive (an ideal blockcipher or a public random function) from $2n$ to n bits. This is similar to the well-known MDC-2 or the recently proposed MJH by Lee and Stam (CT-RSA'11). However, unlike these constructions, we show already in the compression function that an adversary limited (asymptotically in n) to $O(2^{2n(1-\delta)/3})$ queries (for any $\delta > 0$) has disappearing advantage to find collisions.

A key component of our construction is the use of the Szemerédi–Trotter theorem over finite fields to bound the number of full compression function evaluations an adversary can make, in terms of the number of queries to the underlying primitives.

Moreover, for the security proof we rely on a new abstraction that refines and strengthens existing techniques.

We believe that this framework elucidates existing proofs and we consider it of independent interest.

Participants

- Elena Andreeva
K.U. Leuven, BE
- Frederik Armknecht
Universität Mannheim, DE
- Jean-Philippe Aumasson
Nagravision – Cheseaux, CH
- Daniel J. Bernstein
Univ. of Illinois – Chicago, US
- Eli Biham
Technion – Haifa, IL
- Alex Biryukov
University of Luxembourg, LU
- Andrey Bogdanov
K.U. Leuven, BE
- Joan Daemen
STMicroelectronics –
Zaventem, BE
- Itai Dinur
Weizmann Inst. – Rehovot, IL
- Yevgeniy Dodis
New York University, US
- Orr Dunkelman
University of Haifa, IL
- Marc Fischlin
TU Darmstadt, DE
- Ewan Fleischmann
Bauhaus-Universität Weimar, DE
- Christian Forler
Bauhaus-Universität Weimar, DE
- Matthias Hamann
Universität Mannheim, DE
- Tetsu Iwata
Nagoya University, JP
- Antoine Joux
University of Versailles, FR
- Lars Ramkilde Knudsen
Technical Univ. of Denmark –
Lyngby, DK
- Matthias Krause
Universität Mannheim, DE
- Rudolphe Lampe
University of Versailles, FR
- Gregor Leander
Technical Univ. of Denmark –
Lyngby, DK
- Stefan Lucks
Bauhaus-Universität Weimar, DE
- Florian Mendel
K.U. Leuven, BE
- Vasily Mikhalev
Universität Mannheim, DE
- Tilo Müller
Univ. Erlangen-Nürnberg, DE
- Maria Naya-Plasencia
University of Versailles, FR
- Kaisa Nyberg
Aalto University, FI
- Jacques Patarin
University of Versailles, FR
- Kenneth G. Paterson
RHUL – London, GB
- Bart Preneel
K.U. Leuven, BE
- Christian Rechberger
ENS – Paris, FR
- Phillip Rogaway
Univ. of California – Davis, US
- Sondre Ronjom
NSM Norway, NO
- Greg Rose
Qualcomm Inc. – San Diego, US
- Yu Sasaki
NTT Labs. – Tokyo, JP
- Adi Shamir
Weizmann Inst. – Rehovot, IL
- Kyoji Shibusaki
Sony – Tokyo, JP
- Martijn Stam
University of Bristol, GB
- John Steinberger
Univ. of British Columbia, CA
- Deniz Toz
K.U. Leuven, BE
- Kerem Varici
K.U. Leuven, BE
- Bogdan Warinschi
University of Bristol, GB
- Jakob Wenzel
Bauhaus-Universität Weimar, DE
- Kan Yasuda
NTT Labs. – Tokyo, JP
- Erik Zenner
Hochschule Offenburg, DE



Learning in Multiobjective Optimization

Edited by

Salvatore Greco¹, Joshua D. Knowles², Kaisa Miettinen³, and
Eckart Zitzler⁴

1 Università di Catania, IT, salgreco@unict.it

2 University of Manchester, GB, j.knowles@manchester.ac.uk

3 University of Jyväskylä, FI, and Royal Institute of Technology, Stockholm, SE,
kaisa.miettinen@jyu.fi

4 University of Teacher Education, Bern, CH, eckart.zitzler@phbern.ch

Abstract

This report documents the programme and outcomes of the Dagstuhl Seminar 12041 *Learning in Multiobjective Optimization*. The purpose of the seminar was to bring together researchers from the two main communities studying multiobjective optimization, Multiple Criteria Decision Making and Evolutionary Multiobjective Optimization, to take part in a wide-ranging discussion of what constitutes learning in multiobjective optimization, how it can be facilitated, and how it can be measured. The outcome was a deeper, more integrated understanding of the whole problem-solving process in multiobjective optimization from the viewpoint of learning, and several concrete research projects directly addressing different aspects of learning.

Seminar 23.–27. January, 2012 – www.dagstuhl.de/12041

1998 ACM Subject Classification G.1.6 Optimization, H.4.2 Types of Systems, I.2.6 Learning, I.2.8 Problem Solving, Control Methods, and Search, I.5.1 Models

Keywords and phrases multiple criteria decision making, evolutionary multiobjective optimization

Digital Object Identifier 10.4230/DagRep.2.1.50

Edited in cooperation with Richard Allmendinger

1 Executive Summary

Salvatore Greco

Joshua D. Knowles

Kaisa Miettinen

Eckart Zitzler

License  Creative Commons BY-NC-ND 3.0 Unported license
© Salvatore Greco, Joshua D. Knowles, Kaisa Miettinen and Eckart Zitzler

Multiobjective optimization is the study of optimization under competing interests, goals or criteria; it concerns the search for *nondominated* solutions (or Pareto optima) that offer different trade-offs of the competing criteria, as well as methods for choosing among the alternative solutions by the consideration of *preferences*. Multiobjective optimization problems arise naturally in several areas: engineering, economics, operations research/management, and the natural sciences, and today a significant portion of research into optimization is concerned with these problems. The present seminar, the fourth in a series on Multiobjective Optimization (following 04461, 06501 and 09041) dating back to 2004, renewed its ambitions to unite researchers from the two main communities studying multiobjective optimization, MCDM



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license

Learning in Multiobjective Optimization, *Dagstuhl Reports*, Vol. 2, Issue 1, pp. 50–99

Editors: Salvatore Greco, Joshua D. Knowles, Kaisa Miettinen, and Eckart Zitzler



Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

(multiple criteria decision making) and EMO (evolutionary multiobjective optimization) to stimulate new research directions crossing these discipline boundaries.

As with earlier meetings in the series, we chose a strong theme for the seminar, which this time was *Learning*. In multiobjective optimization, learning has a key role to play because, uniquely to the multiobjective case, optimization involves both an exploration of trade-offs and a consideration of user (or decision maker) *preferences*, which are usually implicit in the mind(s) of decision maker(s) at the start of the solution process. Solving a problem therefore involves at least two simultaneous learning processes: the decision maker (DM) learning about the problem, and the optimization process itself learning about the DM's preferences (to achieve a steering of the search toward a preferred solution). Our aim in the seminar was to focus centrally on this learning aspect to give it, for the first time, due attention, as in previous seminars it arose rather peripherally to other themes.

The seminar took place January 22nd–27th 2012. The main goals of the seminar were to explore in depth three different aspects of learning in multiobjective optimization which may be briefly summarized as:

Focus 1: User preferences What should be learnt from user interactions and how should user preferences be captured?

Focus 2: Problem understanding What should be learnt about the problem structure and how can useful information for the DM be extracted?

Focus 3: The problem solving process How do we know if a decision maker has learned? How does a decision maker learn? What factors influence how and what a decision maker learns?

Participants were given some written materials [1, 2] prior to the seminar to orient them to these different aspects and to help them prepare relevant contributions to the seminar programme.

During the seminar, the programme was updated on a daily basis to maintain flexibility and, through this system, we were able to give adequate time both to prepared material and to evolving discussions, mostly taking place in working groups. In particular, breakout working groups were organized initially by lottery (to be purposely disruptive of existing groupings) and then by forming subtopics that individuals could sign up to for the remainder of the week. Six groups emerged in this way. (In the appendix, the complete list of topics suggested can be seen).

The prepared part of the programme included four invited talks of forty-five minutes each and sixteen contributed talks of twenty minutes each. These were spaced to allow time for discussion, and the evenings were kept free to allow further reflection and relaxation. The full programme can be found in Section 5, and the abstracts of all talks are given in the sequel to this summary.

Other notable events during the week included: (i) an interactive demonstration given by Pekka Korhonen on rationality in decision making, which reminded us all of the limits of human (including our own “expert”) rationality in the face of complex data; (ii) a presentation session to allow us to share details of upcoming events in our research community; and (iii), rather less formally, a wine and cheese party was offered by Dagstuhl in the name of ESTECO to express appreciation to ESTECO for giving a donation to the Dagstuhl Foundation.

Outcomes

The outcomes of each of the working groups can be seen in the sequel, but a number of key findings are worth brief mention:

DM Sense working group outlined the design for a system that could aid decision-makers rationalize their learning and decisions *in natural language* by pulling together both recent and older research in artificial intelligence and decision making systems.

Pareto Sense working group established a critical agenda of research to undertake in learning and knowledge representation of the combined spaces of Pareto sets and fronts.

Quantifying Learning working group formalized a method for quantifying the learning associated with decision makers steering a search process, and compared this with the algorithmic learning that occurs in some key model-learning MCDM methods.

Navigation working group developed a detailed understanding of search and decision making approaches to identify the most-preferred solution among the Pareto-set (termed “Navigation”), using this to categorize current methods, and identify applications.

Representation working group considered learning in multiobjective optimization from a machine perspective, proposing that learning could be viewed as the process of obtaining parsimonious representations that enable efficient query-answering in support of (particular) search algorithms or decision processes.

Algorithm Design Methods working group considered formally how algorithms for search and decision making should be selected based on information about the decision maker, as well as the problem, and were able to produce first bounds on the number of function evaluations and queries to a decision maker needed to solve a problem.

These findings were reported to the main group during the seminar, and led to lively debate. Further work within the groups (by email correspondence) following the end of the seminar is planned, including several proposals for joint conference and journal papers.

At the wrap-up session of the seminar, we invited written comments from all the participants concerning how the seminar may be improved, what should be maintained, and inviting topics for future seminars. Comments included ‘working groups were a great opportunity to discuss [...] common features from different perspectives’, ‘Not too many talks — very good; staying in focus — very good; atmosphere — very good’, ‘atmosphere ... is very fruitful, encouraging’, and ‘maintain: the diversity of the experts / participants; good balance between presentations and group discussions, like this time’.

In summary, the seminar made for a very productive and enjoyable week. It has revealed a number of research problems that need careful consideration and detailed further study. It has allowed us to begin this work in earnest, and make some significant first steps.

Acknowledgments

Many thanks to the Dagstuhl office and its helpful and patient staff; huge thanks to the organizers of the previous seminars in the series for setting us up for success; and thanks to all the participants, who worked hard and were amiable company all week.

In the appendix, we also give special thanks to Kaisa Miettinen and Eckart Zitzler as they step down from the organizer role.

References

- 1 S. Greco, J. Knowles, K. Miettinen, E. Zitzler, Dagstuhl Seminar 12041: Learning in Multiobjective Optimization. Seminar Proposal Document, 2011.
- 2 V. Belton, J. Branke, P. Eskelinen, S. Greco, J. Molina, F. Ruiz, R. Słowiński, Interactive multiobjective optimization from a learning perspective. Chapter 15 [in]: J. Branke, K. Deb, K. Miettinen, R. Słowiński (eds.), Multiobjective Optimization: Interactive and Evolutionary Approaches. Springer-Verlag, Berlin, pp.405-434, 2008.

2 Table of Contents

Executive Summary

Salvatore Greco, Joshua D. Knowles, Kaisa Miettinen and Eckart Zitzler 50

Overview of Talks

Interactive Multiobjective Optimization From a Learning Perspective
Jürgen Branke and Roman Słowiński 56

A General Framework for Integrating User Preferences With Evolutionary Multiobjective Optimization: Towards Making the Weighted Hypervolume Approach User-Friendly
Dimo Brockhoff 56

Innovization: Learning Problem Knowledge Through Multi-Objective Optimization
Kalyanmoy Deb 57

Risk and return in multiobjective optimization
Carlos M. Fonseca 57

Cynefin: Learning, Problem Formulation and MCDA
Simon French 58

A Comparison of Hypervolume- and Approximation-Guided MOEAs
Tobias Friedrich 58

Adapting MOEAs to solve practical complex engineering problems
Antonio Gaspar-Cunha 59

Optimization in Logistics from a Learning Perspective: The Case of the Multi-Objective Vehicle Routing Problem
Martin Josef Geiger 60

Learning-Oriented Method Pareto Navigator for Interactive Nonlinear Multiobjective Optimization
Jussi Hakanen 61

Extreme ranking analysis and rank related requirements in multiple objective optimization
Milosz Kadzinski 61

Can a Linear Value Function Explain Choices? An Experimental Study
Pekka Korhonen 62

Offline Automatic Configuration in Multi-Objective Optimization
Manuel López-Ibáñez 62

User preferences in EMO: What can be learned from preference elicitation?
Vincent Mousseau 65

Simulation-Based Innovization using Data Mining and Visual Analytics for Production Systems Analysis
Amos H. C. Ng 65

Problem Understanding with Data Mining of Pareto-Optimal Designs in Space Engineering
Akira Oyama 66

Problem solving process in engineering applications: multiobjective optimization and user preferences <i>Silvia Poles</i>	66
Modelling bipolar interactions in robust ordinal regression: the UTAGSS method <i>Johannes Siebert</i>	67
Learning from Pareto-Front Approximations of Real-World Optimization Problems – A Clustering Approach <i>Tamara Ulrich</i>	68
Hybrid Evolutionary Multi-Objective Optimization: Different Interaction Styles and an Approach <i>Jyrki Wallenius</i>	69
Learning Tradeoffs in Multiobjective Optimization: A Cone-based Approach <i>Margaret M. Wiecek</i>	69
Multiobjective optimization in self-optimizing systems and applications <i>Katrin Witting</i>	71
Working Groups	
Drafting a Manifesto for DM-DSS Interaction (Working Group “DM Sense”) <i>Salvatore Corrente, Simon French, Salvatore Greco, Milosz Kadzinski, Joshua Knowles, Vincent Mousseau, Johannes Siebert, Roman Słowiński</i>	72
What and how can we learn from Pareto fronts and sets? (Working Group “Pareto Sense”) <i>Susanne Bürklen, Kalyanmoy Deb, Michael Emmerich, Karlheinz Kuefer, Boris Naujoks, Amos H. C. Ng, Akira Oyama, Silvia Poles, Tamara Ulrich, Katrin Witting</i>	81
Evaluating, Measuring, Quantifying Learning (Working Group “Quantifying Learning”) <i>Jürgen Branke, Jussi Hakanen, Markus E. Hartikainen, Hisao Ishibuchi, Enrico Rigoni, Karthik Sindhya, Theodor J. Stewart, Margaret M. Wiecek</i>	82
Navigation in Multi Objective Optimization Methods (Working Group “Navigation”) <i>Richard Allmendinger, Heinrich Braun, Matthias Ehrgott, Xavier Gandibleux, Martin J. Geiger, Kathrin Klamroth, Pekka Korhonen, Mariano Luque, Eckart Zitzler</i>	86
Representations (Working Group “Representation”) <i>Carlos A. Coello Coello, José Rui Figueira, Carlos M. Fonseca, António Gaspar-Cunha, Kaisa Miettinen, Sanaz Mostaghim, Dmitry Podkopaev, Pradyumn Kumar Shukla, El-ghazali Talbi, Margaret M. Wiecek</i>	92
Which questions should be asked to find the most appropriate method for decision making and problem solving? (Working Group “Algorithm Design Methods”) <i>Anne Auger, Dimo Brockhoff, Manuel López-Ibáñez, Kaisa Miettinen, Boris Naujoks, Günter Rudolph</i>	92
Seminar schedule	94
Topics emerging from discussions in working groups on Day One	96
Changes in the seminar organization body	97

Kaisa Miettinen steps down as co-organizer	97
Eckart Zitzler leaves us for pastures new – Pedagogy in Bern	97
Welcome to Kathrin Klamroth and Günter Rudolph	98
Participants	99

3 Overview of Talks

3.1 Interactive Multiobjective Optimization From a Learning Perspective

Jürgen Branke (University of Warwick, GB) and Roman Słowiński (Poznan University of Technology, PL)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Jürgen Branke and Roman Słowiński

Joint work of Belton, Valerie; Branke, Jürgen; Eskelinen, Petri; Greco, Salvatore; Molino, Julian; Ruiz, Francisco; Słowiński, Roman

Main reference V. Belton, J. Branke, P. Eskelinen, S. Grecco, J. Molina, F. Ruiz, R. Słowiński, “Interactive Multiobjective Optimization from a Learning Perspective,” Chapter 15 in *Branke, Deb, Miettinen, Słowiński*, (Eds.): Multiobjective Optimization: Interactive and Evolutionary Approaches. LNCS 5252, Springer, Berlin, 2008.

URL <http://dx.doi.org/10.1007/978-3-540-88908-3>

Learning is inherently connected with Interactive Multiobjective Optimization (IMO), therefore, a systematic analysis of IMO from the learning perspective is worthwhile. After an introduction to the nature and the interest of learning within IMO, we consider two complementary aspects of learning: individual learning, i.e., what the decision maker can learn, and model or machine learning, i.e., what the formal model can learn in the course of an IMO procedure. Finally, we discuss how one might investigate learning experimentally, in order to understand how to better support decision makers.

Experiments involving a human decision maker or a virtual decision maker are considered.

3.2 A General Framework for Integrating User Preferences With Evolutionary Multiobjective Optimization: Towards Making the Weighted Hypervolume Approach User-Friendly

Dimo Brockhoff (INRIA Nord Europe – Lille, FR)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Dimo Brockhoff

Joint work of Auger, Anne; Bader, Johannes; Brockhoff, Dimo; Kaci, Souhila; Hamadi, Youssef; Thiele, Lothar; Zitzler, Eckart

Hypervolume-based selection is nowadays considered a standard technique in multiobjective evolutionary algorithms (MOEAs). In 2007, a generalization of the standard hypervolume indicator to the so-called weighted hypervolume indicator has been proposed and it has been showed how this new indicator can be used in the selection of MOEAs to steer the search towards solutions preferred by the user. In the meantime, several studies both about improving the approach’s efficiency for many-objective optimization problems and about understanding its theoretical foundations have been published.


Since its beginnings, the weighted hypervolume indicator approach has been criticized as the definition of the indicator’s weight functions might not be intuitive to the user—in particular not if more than two objectives are to be optimized. Two recent studies deal with this criticism and in my talk I presented the main ideas behind both of them. The first study presents a general weight function toolkit with which the user is not only able to define complex weight functions from simple, easy-to-understand and efficient-to-compute basis functions but also to simulate several classical user preference approaches such as weighted Tchebycheff or desirability functions within the same algorithmic framework. The second

study aims at interactively changing the weight functions and presents a novel way how a weight function can be extracted from the user's input.

More specifically, in the last study, we allow the user to formalize her preferences by explicit preference statements and corresponding semantics which are then automatically translated into a partial order on the current solutions and further transformed into a weight function for the indicator. As this approach contains the intermediate step of visualizing the user's abstract preference statements and the formal, but difficult to interpret semantics as partial orders in an interactive way, it can help the user to learn how to express intrinsic informal preferences in terms of formal preference statements.

3.3 Innovization: Learning Problem Knowledge Through Multi-Objective Optimization


Kalyanmoy Deb (Indian Inst. of Technology – Kanpur, IN)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Kalyanmoy Deb

In optimization studies, often researchers are interested in finding one or more optimal or near-optimal solutions. In this talk, I describe a systematic optimization-cum-analysis procedure which performs a task beyond simply finding optimal solutions, but first finds a set of near-Pareto-optimal solutions and then analyses them to unveil salient knowledge about properties which make a solution optimal. The proposed 'innovization' task is explained and its working procedure is illustrated on a number of engineering design tasks. The variety of problems chosen and the resulting innovations obtained for each problem amply demonstrate the usefulness of the proposed innovization task. The procedure is a by-product of performing a routine multiobjective optimization for a design task and in our opinion portrays an important process of knowledge discovery which may not be possible to achieve by other means.

3.4 Risk and return in multiobjective optimization

Carlos M. Fonseca (University of Coimbra, PT)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Carlos M. Fonseca

Joint work of Fonseca, Carlos M.; Yevseyeva, Iryna; Emmerich, Michael T. M.


The task of selecting a diverse subset of (non-dominated) solutions from a larger set of candidate solutions according to Decision Maker preference information in evolutionary algorithms is reinterpreted as a (financial) portfolio selection problem. Fitness assignment may then be performed by finding an optimal, risk-adjusted portfolio of candidate solutions, e.g., based on the Sharpe-ratio performance index, which amounts to solving a convex quadratic programming problem in the simplest case.

One particular instance of this general paradigm combines Fonseca and Fleming's preferability relation with the hypervolume indicator in order to arrive at a goal-driven, diversity-promoting, combined fitness-assignment and bounded-archiving procedure for evolutionary multiobjective optimization (EMO) algorithms. Experimental results show that the resulting optimizer is highly competitive with NSGA II and SMS-EMOA on a number of multiobjective

knapsack problem instances, and motivate further research on the connection between risk modelling and diversity promotion in EMO.

3.5 Cynefin: Learning, Problem Formulation and MCDA

Simon French (University of Warwick, GB)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Simon French

Main reference S. French. Cynefin, Statistics and Decision Analysis. Journal of the Operational Research Society, 2012 (In Press).

David Snowden’s Cynefin framework, introduced to articulate discussions of sense-making, knowledge management and organisational learning, also has much to offer discussion of problem and issue formulation, value elicitation and learning. In the seminar, I explored its value in helping recognise different problem contexts and which analytic and modelling methodologies are most likely to offer appropriate support. What approaches to optimisation might be relevant? How might this affect our approach to eliciting or capturing decision maker’s values?

3.6 A Comparison of Hypervolume- and Approximation-Guided MOEAs

Tobias Friedrich (MPI für Informatik – Saarbrücken, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Tobias Friedrich

Joint work of Bringmann, Karl; Friedrich, Tobias; Neumann, Frank; Wagner, Markus
Main reference K. Bringmann, T. Friedrich, F. Neumann, M. Wagner, “Approximation-Guided Evolutionary Multi-Objective Optimization,” in Proc. of the 22nd Int’l Joint Conf. on Artificial Intelligence (IJCAI 2011), pp. 1198–1203, 2011.
URL http://www.mpi-inf.mpg.de/~fried/paper/2011IJCAI_AGE.pdf

We propose to measure the quality of a set of solutions of a multi-objective problem by its approximation factor. The theoretical analysis of the approximation factor of single-objective problems is well established and extends nicely to many objectives problems. In the first part of the talk we use this concept to analyze the quality achieved by sets maximizing the hypervolume indicator [1, 2, 3]. In the second part of the talk we present a new MOEA which is directly guided by the approximation factor and has a runtime which scales linearly in the dimension [4].

References

- 1 T. Friedrich, K. Bringmann, T. Voß, and C. Igel. The logarithmic hypervolume indicator. In *Proceedings of the 11th International Workshop on Foundations of Genetic Algorithms (FOGA 2011)*, pages 81–92, Schwarzenberg, Austria, 2011. ACM Press.
- 2 K. Bringmann and T. Friedrich. Tight bounds for the approximation ratio of the hypervolume indicator. In *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN XI)*, volume 6238 of *Lecture Notes in Computer Science*, pages 607–616, Krakow, Poland, September 2010.
- 3 K. Bringmann and T. Friedrich. The maximum hypervolume set yields near-optimal approximation. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO 2010)*, pages 511–518. ACM Press, 2010.

- 4 K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *Proceedings of the 22nd International Joint Conferences on Artificial Intelligence (IJCAI 2011)*, pages 1198–1203. IJCAI/AAAI, 2011.

3.7 Adapting MOEAs to solve practical complex engineering problems

Antonio Gaspar-Cunha (University of Minho – Guimarães, PT)

License © ⓘ ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Antonio Gaspar-Cunha

Joint work of Gaspar-Cunha, Antonio; Ferreira, Jose; Fonseca, Carlos; Covas, Jose

Main reference A. Gaspar-Cunha, J.A. Covas, Eds., “Optimization in Polymer Processing,” 1st ed., Nova Science Publishers, 2011.

URL https://www.novapublishers.com/catalog/product_info.php?products_id=20015

In general, real engineering design problems are complex and multidisciplinary and thus difficult to solve adequately within reasonable timings. The scientific and technological advances in some fields (e.g., computational fluid dynamics, heat transfer, structural mechanics), together with the availability of highly performing computing techniques (e.g., parallel and/or grid computing) and facilities, provide the possibility of considering more problem aspects, thus generating improved solutions. However, since significant computational resources must be available, their efficient use must be guaranteed.

Multidisciplinary Design Optimization (MDO) can be defined as a methodology to design complex integrated engineering structures, which combines different disciplines and takes into account in a synergistic manner the interaction between the various subsystems. Examples of its practical application include aircrafts, cars, building structures and manufacturing systems.


A practical way to deal with engineering problems consists of using Multi-Objective Evolutionary Algorithms (MOEA), since at a certain point of the design process it will be necessary to provide information regarding the relative importance of every problem objective, i.e., the preferences of a Decision Maker (DM) must be considered. Furthermore, the solutions must also be robust, i.e., the performance of the prospective optimal solution(s) should be only slightly affected by perturbations of the design variables, or of environmental parameters. Two additional issues concerning the application of MOEAs to complex engineering problems are: i) the large number of objective functions evaluations that are necessary to attain an acceptable solution and ii) the high number of objectives to be taken in simultaneously. The former can be dealt with through the hybridization of MOEAs with local search procedures, while the latter involves the application of techniques to reduce the number of objectives.

The aim of this work is to present and discuss approaches to solve complex problems by employing tools that are able to simultaneously deal with multiple objectives, decision making and robustness of the solutions, among others, with a view to demonstrating that multi-objective engineering problems can be solved efficiently through the combination of optimization methodologies with engineering and design tools.

Two examples, from the fields of polymer engineering and aesthetic design, will be used to illustrate the methodologies proposed above.

3.8 Optimization in Logistics from a Learning Perspective: The Case of the Multi-Objective Vehicle Routing Problem

Martin Josef Geiger (*Helmut-Schmidt-Universität – Hamburg, DE*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Martin Josef Geiger

Joint work of Geiger, Martin Josef; Wenger, Wolf

Main reference W. Wenger, M.J. Geiger, “Hybrid Interactive Planning Under Many Objectives: An Application to the Vehicle Routing Problem,” in Proc. of the 8th Int’l. Conf. on Hybrid Intelligent Systems (HIS’08), Sept. 10–12, 2008, Barcelona, Spain, pp. 302-307, ISBN 978-0-7695-3326-1.

URL <http://dx.doi.org/10.1109/HIS.2008.91>

Many logistical problems are characterized by numerous, often conflicting objectives. In combination with the underlying, often NP-hard optimization problems, this leads to a combination of search (for efficient outcomes) and decision making, i.e. choice of a most-preferred alternative.

Interactive systems supporting such a process should possess at least two characteristics. On the one hand, an adaptivity must be present, so that the presented results change w. r. t. evolving preference statements. On the other hand, the results should be of high quality, i.e. Pareto-optimal (or close to the efficient outcomes) [1].

In the talk, we consider the case of the multi-objective vehicle routing problem, for which an interactive optimization and decision making system has been developed [2, 3]. On the basis of benchmark data taken from the literature, the adaptivity of the system is investigated for different types of decision makers, i.e. decision makers with different preferences for the considered objectives.

In the interactive, alternating process of optimization and choice of a most-preferred solution, learning takes place both from the point of view of the optimization system (algorithm) and the decision maker. For the algorithm, ‘learning’ is a simple, adaptive process, improving the current solution in a direction given by the decision maker. For the decision maker, the ‘learning’ has two components.


First, preferred characteristics of the most-preferred solution have to be detected. Second, disadvantageous properties should be learned, that have to be avoided in the final solution. Both together describe a process of preference building, in which the expert is presented a series of alternatives. A convergence can be detected once the decision maker does not alter his/her preference statements any more, and thus does not seek for alternatives in another direction of the outcome space.

References

- 1 M.J. Geiger and W. Wenger. Market based allocation of transportation orders to vehicles in adaptive multi-objective vehicle routing. In *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*, pp. 119-132. Springer Verlag, Berlin, Heidelberg, 2008.
- 2 M.J. Geiger and W. Wenger. On the interactive resolution of multi-objective vehicle routing problems. In *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007*, volume 4403 of *Lecture Notes in Computer Science*, pp. 687-699. Springer Verlag, Berlin, Heidelberg, New York, 2007.
- 3 M.J. Geiger, W. Wenger, and W. Habenicht. Interactive utility maximization in multi-objective vehicle routing problems: A “decision maker in the loop”-approach. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)*, pp. 178-184, Hilton Hawaiian Village, Honolulu, Hawaii, USA, April 2007.

3.9 Learning-Oriented Method Pareto Navigator for Interactive Nonlinear Multiobjective Optimization

Jussi Hakanen (University of Jyväskylä, FI)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jussi Hakanen

Joint work of Eskelinen, Petri; Miettinen, Kaisa; Klamroth, Kathrin; Hakanen, Jussi


Main reference P. Eskelinen, K. Miettinen, K. Klamroth, J. Hakanen, “Pareto Navigator for Interactive Nonlinear Multiobjective Optimization,” *OR Spectrum*, pp. 211–227, 32, 2010.

URL <http://dx.doi.org/10.1007/s00291-008-0151-6>

We describe a new interactive learning-oriented method called Pareto navigator for convex multiobjective optimization. In the method, first a polyhedral approximation of the Pareto optimal set is formed in the objective function space using a relatively small set of Pareto optimal solutions representing the Pareto optimal set. Then the decision maker can navigate around the polyhedral approximation and direct the search for promising regions where the most preferred solution could be located. In this way, the decision maker can learn about the interdependencies between the conflicting objectives and possibly adjust one’s preferences. Once an interesting region has been identified, the polyhedral approximation can be made more accurate in that region or the decision maker can ask for the closest counterpart in the actual Pareto optimal set. If desired, (s)he can continue with another interactive method from the solution obtained. Pareto navigator can be seen as a nonlinear extension of the linear Pareto race method. After the representative set of Pareto optimal solutions has been generated, Pareto navigator is computationally efficient because the computations are performed in the polyhedral approximation and for that reason function evaluations of the actual objective functions are not needed. Thus, the method is well suited especially for problems with computationally costly functions. Furthermore, thanks to the visualization technique used, the method is applicable also for problems with three or more objective functions, and in fact it is best suited for such problems. After introducing the method, we demonstrate how it works with an implementation which has been created as a part of the IND-NIMBUS multiobjective optimization framework.

3.10 Extreme ranking analysis and rank related requirements in multiple objective optimization

Milosz Kadzinski (Poznan University of Technology, PL)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Milosz Kadzinski

We present a new interactive procedure for multiple objective optimization. The procedure is composed of two alternating stages. In the first stage, a representative sample of solutions from the Pareto optimal set is generated. In the second stage, the Decision Maker (DM) is asked to provide preference information concerning some solutions from the generated sample. In particular, (s)he may refer to the holistic judgments concerning these solutions such as, e.g., pairwise comparisons or desired ranks. As far as the latter option is concerned, real-life experience indicates that people willingly refer to the range of allowed ranks that a particular solution should attain (e.g., a should take place on the podium, b should be ranked in the upper/lower half, c should be among the 10% of best/worst solutions). Referring to the rank-related requirements, the DM rates a given solution individually, at the same time

collating it with all the remaining solutions jointly. This preference information is used to build a preference model composed of all general additive value functions compatible with the obtained information. The set of compatible value functions is then applied on the whole Pareto optimal set. The recommendation which can be obtained for any compatible value function can vary substantially. An interesting way to examine this diversity is to determine the best and the worst rank that each solution can attain. In this way, we are able to assess its performance relative to all the solutions considered simultaneously, and not only in terms of pairwise comparisons, as it is the case in the original multiple objective optimization methods based on the principle of robust ordinal regression, such as GRIP. These extreme results are used to select a new sample of solutions, which is presented to the DM, and the procedure cycles until a satisfactory solution is selected from the sample or the DM comes to conclusion that there is no satisfactory solution for the current problem setting.

3.11 Can a Linear Value Function Explain Choices? An Experimental Study

Pekka Korhonen (Aalto University, FI)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Pekka Korhonen

We investigate in a simple bi-criteria experimental study, whether subjects are consistent with a linear value function while making binary choices.

Many inconsistencies appeared in our experiment. However, the impact of inconsistencies on the linearity vs. non-linearity of the value function was minor. Moreover, a linear value function seems to predict choices for bi-criteria problems quite well. This ability to predict is independent of whether the value function is diagnosed linear or not. Inconsistencies in responses did not necessarily change the original diagnosis of the form of the value function. Our findings have implications for the design and development of decision support tools for Multiple Criteria Decision Making problems.

3.12 Offline Automatic Configuration in Multi-Objective Optimization

Manuel Lopez-Ibanez (Université Libre de Bruxelles, BE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Manuel López-Ibáñez
Joint work of López-Ibáñez, Manuel; Stutzle, Thomas

Most of the current literature on machine learning in multi-objective optimization concerns the problem of learning while solving a particular problem instance, that is, *online learning* for the purposes of learning the decision maker's preferences [1], adapting the parameters of an algorithm [2] or approximating the landscape for expensive multi-objective optimization problems [3]. Few works in multi-objective optimization deal with *offline learning*, that is, learning during a training phase and repeatedly using what has been learned in a secondary production (or testing) phase.

One of the most prominent applications of offline learning in single-objective optimization is offline tuning [4], and more generally, automatic configuration [5] and programming by optimization [6]. The key idea behind automatic configuration is to automatically learn

from examples the best design choices to build a fully-specified optimization algorithm tailored for a particular user context. An example could be to tune the parameters of a general-purpose solver, such as an evolutionary algorithm, to solve instances of a particular family of optimization problems, such as the traveling salesman problem. In single-objective optimization, this approach has led to notable successes. One notable example is the application of an automatic configuration tool to a framework of SAT solvers that won several prizes in the International SAT competition [7].

Existing automatic configuration tools may be used for multi-objective optimization algorithms by means of unary quality measures, such as the hypervolume [8]. Using this approach, Wessing et al. [9] have presented results for configuring the variation operator of a multi-objective evolutionary algorithm to a continuous function, and López-Ibáñez & Stützle [10, 11] automatically configured a flexible multi-objective ant colony optimization framework for tackling the bi-objective traveling salesman problem.

Despite these initial successes, it is currently an open research question how to effectively carry out offline automatic configuration in a multi-objective context without relying on unary quality measures. In order to achieve this goal, several challenging issues must be tackled, which are closely related to the question of how to design meaningful experiments in order to investigate learning [12].

The first challenge is how to assess the relative performance of multi-objective optimizers in an automatic fashion, not on an individual application, but over a series of training instances/examples. This is straightforward if the preference information available is enough to reduce the multi-objective problem to a single utility value, since then classical techniques from experimental design and statistical inference are applicable [13]. However, such preference information may not be always available, and although there are some initial results on extending statistical inference methodologies to the multi-objective context [14], there are no methods equivalent to those used in automatic configuration tools.

The various preference models pose an additional challenge. If preferences are defined a priori, and are common to all training examples, then it becomes possible to tune the optimization method for that particular preference model. However, one can easily imagine that each training example may have its own preference model, or even that the goal may be to choose the preference model itself, e.g., each training instance involving a different, possibly virtual, decision maker (DM). The challenge here is how to evaluate and compare different preference models.

Lastly, how to include the role of the DM in an offline learning procedure is far from clear. Perhaps the most straightforward strategy is to reuse the knowledge available about interactive approaches, making the offline configuration process a semi-automatic approach where a DM is asked about her preferences w.r.t. the quality of alternative algorithmic configurations. The automatic configuration tool may implicitly build a model of the DM preferences and use it to guide the automatic configuration process. However, the preference elicitation process will likely be more complex than in classical interactive approaches, since the DM will be asked to decide over a number of training examples. Moreover, there is the issue of how to define training examples of decision-making that correlate to the expected behavior of the, possibly multiple, DMs that will use the final system in the production phase.

The field of multi-objective optimization has advanced to the point that there are many high-quality approaches to solve problems. The choice of the most appropriate approach depends greatly on the user's context. However, users do not have the knowledge and expertise to make informed choices in order to choose and adapt existing approaches to solve


their own problem. What they often have is examples of the kind of problems they want to solve. The ideal automatic configuration method for multi-objective optimization problem will not only be able to automatically design an algorithm given the user context, but also to choose the most appropriate preference model.

References

- 1 R. Battiti and A. Passerini. Brain-computer evolutionary multiobjective optimization. A genetic algorithm adapting to the decision maker. In *IEEE Transactions on Evolutionary Computation*, 14(5):671–687, 2010.
- 2 G. Toscano Pulido and C.A. Coello Coello. The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization. In *Evolutionary Multi-criterion Optimization (EMO 2003)*, Lecture Notes in Computer Science, vol. 2632, pp. 252–266. Springer, Heidelberg, Germany, 2003.
- 3 J.D. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. In *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- 4 M. Birattari. Tuning Metaheuristics: A Machine Learning Perspective. In *Studies in Computational Intelligence*, vol. 197. Springer, Berlin / Heidelberg, 2009.
- 5 F. Hutter. *Automated Configuration of Algorithms for Solving Hard Computational Problems*. PhD thesis, University of British Columbia, Department of Computer Science, Vancouver, Canada, October 2009.
- 6 H.H. Hoos. Programming by optimisation: Towards a new paradigm for developing high-performance software. In *MIC 2011, the 9th Metaheuristics International Conference*, Plenary talk, 2011. [urlhttp://mic2011.diegm.uniud.it/uploads/plenaries/Hoos-MIC2011.pdf](http://mic2011.diegm.uniud.it/uploads/plenaries/Hoos-MIC2011.pdf)
- 7 A.R.KhudaBukhsh, L. Xu, H.H Hoos, and K. Leyton-Brown. SATenstein: Automatically building local search SAT solvers from components. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*. pp. 517–524, 2009. <http://ijcai.org/papers09/Papers/IJCAI09-093.pdf>
- 8 E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto evolutionary algorithm. In *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- 9 S. Wessing, N. Beume, G. Rudolph, and B. Naujoks. Parameter tuning boosts performance of variation operators in multiobjective optimization. In *Parallel Problem Solving from Nature, PPSN XI*, Lecture Notes in Computer Science, vol. 6238, pp. 728–737. Springer, Heidelberg, Germany, 2010.
- 10 M. López-Ibáñez and T. Stützle. Automatic configuration of multi-objective ACO algorithms. In *Swarm Intelligence, 7th International Conference, ANTS 2010*, Lecture Notes in Computer Science, vol. 6234, pp. 95–106. Springer, Heidelberg, Germany, 2010.
- 11 M. López-Ibáñez and T. Stützle. The automatic design of multi-objective ant colony optimization algorithms. In *IEEE Transactions on Evolutionary Computation*, accepted, 2012.
- 12 V. Belton, J. Branke, P. Eskelinen, S. Greco, J. Molina, F. Ruiz, and R. Słowiński. Interactive multiobjective optimization from a learning perspective. In *Multi-objective Optimization – Interactive and Evolutionary Approaches*, Lecture Notes in Computer Science, vol. 5252, pp. 405–433. Springer, Heidelberg, Germany, 2008.
- 13 T. Bartz-Beielstein. Experimental Research in Evolutionary Computation. The New Experimentalism. Springer, Berlin, Germany, 2006.
- 14 V. Grunert da Fonseca and C.M. Fonseca. The attainment-function approach to stochastic multiobjective optimizer assessment and comparison. In *Experimental Methods for the Analysis of Optimization Algorithms*. Springer, Berlin, Germany, 2010.

3.13 User preferences in EMO: What can be learned from preference elicitation?

Vincent Mousseau (Ecole Centrale – Paris, FR)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Vincent Mousseau

An important perspective in EMO concerns the optimization process that interacts with the user and tries to infer formal information to guide the search and adapt the model. The key questions here are “What can and should be learnt from user interactions? how can user preferences be inferred? how can such user preference model guide the search?” The field of multiple criteria preference elicitation has developed a variety of concepts and procedures to capture DMs preferences from holistic preferences. The proposed elicitation techniques propose interaction protocols and algorithms to infer a formal preference model from assertions made by DMs.

In this presentation, we will show on two examples, how preference elicitation ideas can be integrated into evolutionary multiobjective optimization algorithms so as to focus the computation of solutions judged as good by the DM. The first example involves a utility based preference model while the second represents preferences using a binary (outranking) relation.

3.14 Simulation-Based Innovization using Data Mining and Visual Analytics for Production Systems Analysis

Amos H. C. Ng (University of Skövde, SE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Amos H. C. Ng

Joint work of Ng, Amos H. C.; Dudas, Catarina; Deb, Kalyanmoy

Main reference A.H.C. Ng, C. Dudas, K. Deb, “Simulation-Based Innovization using Data Mining for Production Systems Analysis,” in *L. Wang, A. Ng, K. Deb*, (eds.): *Evolutionary Multi-objective Optimization in Product Design and Manufacturing*, Springer, pp. 401–430, 2011.


URL http://dx.doi.org/10.1007/978-0-85729-652-8_15

The aim of this talk is to introduce a novel methodology for the optimization, analysis and decision support in production systems development. The methodology is based on the innovization procedure, originally introduced for unveiling new and innovative design principles in engineering design problems. Although the innovization method is based on multi-objective optimization and post-optimality analyses of optimized solutions, it stretches the scope beyond an optimization task and attempts to discover new design/operational rules/principles relating to decision variables and objectives, so that a deeper understanding of the design problem can be obtained (i.e. problem understanding). By integrating the concept of innovization with discrete-event simulation and data mining techniques, a new set of powerful tools can be developed for general systems analysis, particularly suitable for production systems development. The uniqueness of the integrated approach introduced in this talk lies on applying data mining and visual analytics to the data sets generated from simulation-based multi-objective optimization, in order to automatically or semi-automatically discover and interpret the hidden relationships and patterns for optimal production systems design/reconfiguration and then present to the decision maker in an interactive manner. After describing such a simulation-based innovization (SBI) using data mining procedure and

its difference from conventional simulation analysis methods, results from several industrial-based case studies for production systems design and/or improvement will be presented. As illustrated with the experience learnt from the decision making process in these industrial case studies, the talk will convince that SBI not only helping production managers/engineers to explore optimal design and decision variable settings, but also gaining better knowledge and insight about production systems development in general.

3.15 Problem Understanding with Data Mining of Pareto-Optimal Designs in Space Engineering


Akira Oyama (JAXA – Sagami, JP)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Akira Oyama

Multiobjective design exploration (MODE) is a framework that can obtain useful knowledge for design optimization problems. MODE finds Pareto-optimal solutions with a multiobjective design optimization method and then extracts useful knowledge to understand the problem from the solution database with data mining approaches. In this presentation, how MODE are used to understand real-world design problems that Japan Aerospace Exploration Agency actually have is presented.

3.16 Problem solving process in engineering applications: multiobjective optimization and user preferences

Silvia Poles (EnginSoft S.p.A. – Padova, IT)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Silvia Poles

Designing real products is an enormous task that requires a multiobjective and multidisciplinary perspective involving many decision makers and departments.


At later stages of the design phase, engineers are forced to respect predefined characteristics. Conversely, it is exactly in the early phase that designers can look for product innovation making decisions that can have a great influence in the final design.

What is necessary during the entire phase is a common framework in which decision makers can interact, run multiobjective optimizations, construct models, extract values and plot meaningful charts for exploring the cross influences of the design, discovering patterns and similarities between different configurations.

The most important part is the feedback/learning phase in which engineers can gain knowledge of the problem at hand. By clicking on charts it is possible to filter solutions or to run the optimizer to explore more deeply a specific area of the Pareto front. In this way, engineers are not just waiting for optimization solutions, they are part of the optimization process, they are learning on the job.

3.17 Modelling bipolar interactions in robust ordinal regression: the UTAGSS method

Johannes Siebert (Universität Bayreuth, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Johannes Siebert

We present in analogy to Figueira et al. [1] an interactive method for multiobjective optimization, which is based on the use of a set of value functions as a preference model built by an ordinal regression method. Initially we generate a sample of solutions from the Pareto optimal set (or from its approximation). Subsequently the DM has to provide additional preference information in terms of holistic pairwise comparisons of some solutions from the generated sample. Based on this information we build a preference model composed of all general additive value functions compatible with the obtained information under consideration of bipolar interactions between criteria. The set of compatible value functions is then applied on the whole Pareto optimal set, which results in possible and necessary rankings of Pareto optimal solutions.

Using these rankings a new sample of solutions has to be pairwise evaluated by the DM. This interactive cycle stops when the DM comes to conclusion that there is no satisfactory solution for the current problem setting. The set of compatible value functions is constructed using ordinal regression methods called UTAGSS, the most general approach in the UTAGMS/GSS family. This method generalizes UTA-like methods and is competitive to AHP and MACBETH methods.

The problem of representing interactions has been dealt with different methodologies, such as polynomial conjoint measurement, multilinear value functions, and nonadditive integrals, like Choquet integral and Sugeno integral.

Recently Greco, Mousseau and Słowiński [2] presented a decision model able to represent interaction by adding to the classical additive utility function some additional terms expressing a bonus or a penalty related to evaluations of pairs, triples and, in general, n-tuples of criteria. [2] presents a method called UTAGMS-INT in which the decision model is assessed using robust ordinal regression. This means that starting from some preference information given by the Decision Maker (DM), the set of compatible value functions is defined such that alternative a is necessarily weakly preferred to alternative b if a is at least as good as b for all compatible value functions, while a is possibly weakly preferred to b if a is at least as good as b for at least one compatible value function. The interactions modelled in [2] are synergy and redundancy, which yield a bonus or a penalty, respectively, when values of the considered n-tuple of criteria improve together.

UTAGSS, that extends the UTAGMS-INT method by considering criteria values under consideration of the idea of bipolarity [4]. In this case, synergy and redundancy (i.e. bonus and penalty) depend on the relative position of values of the considered n-tuple of criteria with respect to a neutral level. To gain the highest degrees of freedom we use the idea of bipolarity to distinguish between different areas of interaction effects. This allows considering different neutral levels for each pair of interacting criteria. Considering so called bipolar interactions we are able to get a representation of DM's preferences, which is more faithful with respect to the information supplied by the DM. If the DM has no idea about the interactions, then we use a mixed integer linear program to determine sets of pairs of interacting criteria. In UTAGSS it is possible but not necessary to consider bipolar scales for all criteria. UTAGSS is a generalization of UTAGMS-INT, because it produces in the special case, all neutral levels regarding interactions between two criteria are located at the worst performance of the


criteria, the same results as UTAGMS-INT. UTAGSS is the most flexible method able to represent the most complex interactions. We introduce an example of a bipolar interaction which takes effect like a two-dimensional knock-out criterion, a straightforward extension of vetos and pushers based on one criterion [3]. This phenomenon can only be modelled with UTAGSS and not with UTAGMS-INT.

References

- 1 J. Figueira, S. Greco, V. Mousseau, and R. Słowiński. Interactive Multiobjective Optimization using a Set of Additive Value Functions. In *Multiobjective optimization*, Springer, 2008.
- 2 S. Greco, V. Mousseau, and R. Słowiński. UTAGMS-INT: Robust Ordinal Regression of Value Functions Handling Interacting Criteria, submitted.
- 3 J.-L. Marichal. Tolerant or intolerant character of interacting criteria in aggregation by the choquet integral. In *European Journal of Operational Research*, 155(3):771–791, 2004.
- 4 A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. In *Journal of Risk and Uncertainty*, 5(4):297–323, 1992.

3.18 Learning from Pareto-Front Approximations of Real-World Optimization Problems – A Clustering Approach

Tamara Ulrich (ETH Zürich, CH)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Tamara Ulrich


Multiobjective problems usually contain conflicting objectives. Therefore, there is no single best solution, but a set of solutions that represent different tradeoffs between these objectives. For real-world problems, an interpretation of the front is usually not straightforward.

We have proposed a method to help the decision maker by clustering a given set of tradeoff solutions. We do so by extending the standard approach of clustering the solutions in objective space, such that it finds clusters which are compact and well separated both in decision and in objective space. It is not the goal of the method to provide the decision maker with a single preferred solution.

Instead, it helps the decision maker by eliciting information from the front about what design types lead to what regions in objective space. The novelty of the presented approach over existing work is its general nature, as it does not require the identification of distinct design variables. Instead, our method only requires that a distance measure between a given pair of solutions can be calculated both in decision and in objective space. This makes it applicable to any real-world problem.

3.19 Hybrid Evolutionary Multi-Objective Optimization: Different Interaction Styles and an Approach


Jyrki Wallenius (Aalto University, FI)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jyrki Wallenius

We describe different man-computer interaction styles, which are commonly embedded within existing Multiple Criteria Decision Making techniques to elicit the Decision-Maker's preferences in problems involving more than two objective functions. The elicitation process reflects Decision-Maker's learning of his/her preferences, based on realizing what is possible and feasible to achieve regarding objective function values. A case in point is pairwise comparisons, which have been found easy to elicit. Two example methods, which are representatives of hybrid Evolutionary Multiobjective Optimization methods, are explained in some detail. We also discuss computational results. The talk concludes with a discussion of future research questions. The talk highlights the importance of the role of a human decision-maker, and more broadly understanding the behavioural foundations of decision making, in Evolutionary Multiobjective Optimization.

3.20 Learning Tradeoffs in Multiobjective Optimization: A Cone-based Approach

Margaret M. Wiecek (Clemson University, USA)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Margaret M. Wiecek

Optimality in multiobjective optimization problems (MOPs) is governed by a partial order in the objective space and produces a set of solutions to the optimization problem rather than a unique optimal solution. The partial order implied by the binary relation of the componentwise comparison of two vectors has traditionally been used and is known as the Pareto optimality. In the process of multicriteria decision making (MCDM), the decision maker faces the challenge of choosing a preferred solution from the set of Pareto solutions. While Pareto solutions are equivalent in the mathematical sense, they are not equivalent for the decision maker (DM) because they are not equally preferred with respect to preferences that the DM may have or be developing in the course of decision making. The DM's preferences may be changing due to a learning process he or she is engaged in while searching through the Pareto solutions.

In the MCDM literature there is a great variety of models of DM's preferences on the Pareto set and there are numerous procedures making use of those models. In this talk we are interested in models developed with convex cones since we believe that the concept of cone is inherent to multiobjective optimization. After Yu [1] developed grounds for relating cones to the Pareto optimality, cones turned out to be an effective concept for modeling DM's preferences from the perspective of tradeoffs associated with the Pareto solutions in the objective space.

We will review the state of the art in cone-based modeling of preferences. Berman and Naumov [2] are perhaps the first to use interval tradeoffs and construct a matrix of a cone to represent DM's preferences. Noghin [3] uses weights as the coefficients of relative importance between criteria, constructs a direction in the objective space, and models DM's

preferences by the convex hull of the Pareto cone and this direction. The ideas of Noghin [3] are extended in [4] and [5] to construct an estimate of the Pareto set, and in [6] to derive conditions for consistency of relative importance information. Hunt and Wiecek [7] and Hunt et al. [8] build on Noghin's approach and allow more directions to be appended to the Pareto cone to construct a new preference cone. DMs preferences are quantified by the so called allowable tradeoffs between objectives, or the maximum amount the DM is willing to allow one objective to decay to obtain one unit of improvement in one other objective. Using these values, convex polyhedral cones are constructed and their complete algebraic descriptions are derived.

In the second part of the talk we will discuss the use of cone-based models in decision making [9]. They reduce the Pareto set to a subset of decisions that are representative for the DM's preferences and satisfy certain bounds on tradeoffs. In this way the models offer a tool being a compromise between the models relying on scalarizing approaches and set-oriented methods. The former reduce the Pareto set to a singleton, which may be rather limiting for the DM, while the latter (e.g., evolutionary methods) yield a representation of the Pareto set in the form of many points, which can be overwhelming and difficult to use.

The models can be incorporated into the MCDM process either a priori, a posteriori, or interactively because they can work in concert with any MCDM method. The advantage of the a priori approach is that Pareto solutions that do not satisfy the DM's preferences are never considered. If DMs are unfamiliar with the problem and/or unsure of their preferences, they have the freedom to interactively explore the set of feasible solutions by adjusting the models. This exploration allows them to familiarize themselves with the problem and learn about which solutions are the least sensitive to small changes in preferences. The models extract the solutions from the Pareto producing a short list of "strong" or "privileged" solutions with preferred tradeoffs. The short list may be long or even include one solution. In any case, DMs retain the right to choose and exercise their right within a small subset of candidate solutions.

We will also report on the applications of Hunt et al. [10, 11] and [12] models in engineering design and present the accompanying learning process an automotive designer is engaged in. We will conclude the talk with future research directions on the development of cone-based models of preferences for MCDM.

References

- 1 P.L. Yu. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. In *Journal of Optimization Theory and Applications*, 14(3):319–377, 1974
- 2 V.P. Berman and G.E. Naumov. Preference relation and interval value tradeoffs in criterion space. In *Automation and Remote Control*, 50(3):398–410, 1989.
- 3 V. D. Noghin. Relative importance of criteria: a quantitative approach. In *Journal of Multicriteria Decision Analysis*, 6:355–363, 1997.
- 4 V.D. Noghin and I.V. Tolstykh. Using quantitative information on the relative importance of criteria for decision making. In *Computational Mathematics and Mathematical Physics*, 40(11):1529–1536, 2000.
- 5 V.D. Noghin. A logical justification of the Edgeworth-Pareto principle. In *Computational Mathematics and Mathematical Physics*, 42(7):915–920, 2001.
- 6 O.N. Klimova and V.D. Noghin. Using interdependent information on the relative importance of criteria in decision making. In *Computational Mathematics and Mathematical Physics*, 46(12):2178–2190, 2006.

- 7 B.J. Hunt and M.M. Wiecek. Cones to aid decision making in multicriteria programming. In *Multi-Objective Programming and Goal-Programming*, pp. 153-158, 2003.
- 8 B.J. Hunt, M.M. Wiecek, and C. Hughes. Relative importance of criteria in multiobjective programming: a cone-based approach. In *European Journal of Operational Research*, 207:936–945, 2010.
- 9 M.M. Wiecek. Advances in cone-based preference modeling for decision making with multiple criteria. In *Decision Making in Manufacturing and Services*, 1(1-2):153–173, 2007.
- 10 B.J. Hunt, M.M. Wiecek, and G. Fadel. Matrices as preference modeling tools in bi-criteria engineering design. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, August-September, 2004.
- 11 B.J. Hunt, V.Y. Blouin, and M.M. Wiecek. Relative Importance of Design Criteria: A Preference Modeling Approach. In *Journal of Mechanical Design*, 129(9):907–914, 2007.
- 12 V. Blouin, B.J. Hunt, and M.M. Wiecek. MCDM with relative importance of criteria: application to configuration design of vehicles. In *Multiple Criteria Decision Making '08*, pp. 11-40, 2008.

3.21 Multiobjective optimization in self-optimizing systems and applications

Katrin Witting (Universität Paderborn, DE)

License © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license
© Katrin Witting

Joint work of Witting, Katrin; Dellnitz, Michael; Trächtler, Ansgar; Geisler, Jens; Böcker, Joachim; Schulz, Bernd; Fröhleke, Norbert

Main reference J. Geisler, A. Trächtler, K. Witting, M. Dellnitz, “Multiobjective optimization of control trajectories for the guidance of a rail-bound vehicle,” 17th IFAC World Congress, Seoul, Korea, 2008.

URL <http://dx.doi.org/10.3182/20080706-5-KR-1001.00738>

In the Collaborative Research Center “Self-optimizing concepts and structures in mechanical engineering” (SFB614) at the University of Paderborn, Germany, methods for the design of tomorrow’s mechanical engineering products are developed. The concept of self-optimization developed within this research project goes beyond the classical adaptation techniques for mechatronical systems. It includes three steps that are repeated during operation time: (i) Analysis of the current situation, (ii) Determination of the system’s objectives, (iii) Adaptation of the system behaviour.

For model-based self-optimization of mechatronical systems, multiobjective optimization is an important approach. Having formulated suitable objectives the determination of the system’s objectives in step (ii) of the self-optimization process can be seen as decision making on the Pareto set. Depending on the current situation, adequate Pareto points have to be chosen. For several technical applications like for example the operating point assignment of a linear drive [1] and the guidance of a rail-bound vehicle [2] we have constructed special heuristics that allow to choose Pareto points fitting to the current situation during operation time. These heuristics have been developed in close cooperation with the engineers who developed the technical systems. In case of the driving module the Pareto optimal solution is adapted during operation time making use of numerical path following methods.

References


- 1 K. Witting, B. Schulz, M. Dellnitz, J. Böcker, and N. Fröhleke. A new approach for online multiobjective optimization. In *Int. Journal on Software Tools for Technology Transfer (STTT)*, 10(3):223–231, 2008. <http://dx.doi.org/10.1007/s10009-008-0066-1>

- 2 J. Geisler, A. Trächtler, K. Witting, and M. Dellnitz. Multiobjective optimization of control trajectories for the guidance of a rail-bound vehicle. In *17th IFAC World Congress*, Seoul, Korea, 2008. <http://dx.doi.org/10.3182/20080706-5-KR-1001.00738>

4 Working Groups

4.1 Drafting a Manifesto for DM-DSS Interaction (Working Group “DM Sense”)

Salvatore Corrente, Simon French, Salvatore Greco, Milosz Kadzinski, Joshua Knowles, Vincent Mousseau, Johannes Siebert, Roman Słowiński

License  Creative Commons BY-NC-ND 3.0 Unported license
© Salvatore Corrente, Simon French, Salvatore Greco, Milosz Kadzinski, Joshua Knowles, Vincent Mousseau, Johannes Siebert, Roman Słowiński

4.1.1 Introduction

The DM Sense group, as it was called, met several times during the Dagstuhl Seminar to discuss intelligent user interactions in decision support with an emphasis on the need to create dialogues between decision makers and their decision support tools which explained the process and the underlying reasoning so bringing understanding and insight to the decision makers. The challenge was to develop dialogues that facilitated the user’s thinking. In a way the challenge paralleled that of Turing’s test: could a machine interact with a decision maker in a way that was indistinguishable from how a decision analyst might interact?

Naturally the task we set ourselves in the opening discussion was somewhat simpler. We decided that our aims could be summarised as:

Aims and Objectives

Construct a system able to generate contextual explanations in natural language in support of the decision.

- Generate sentences which materialise the explanations and support further interactions with the Decision Maker.
- Keep trace of arguments that led to the decision in order to present them to other stakeholders.
- Expressing preference information should not require great cognitive effort from the Decision Maker.
- The explanations should be accessible to inexperienced and unsophisticated users.

By the end of our discussions we felt that there was need for much more work on this topic that recognised its importance if multi-objective decision support systems were truly to support the growth in decision makers’ understanding and their confidence in the final decision. We needed to prepare a *Manifesto for Interactions between Decision Makers and Decision Support Systems*.

The following are our notes as they were generated during the several sessions of the Dagstuhl Seminar and reported back to the plenary session on the final day.

4.1.2 Assumptions

During our discussions we made a number of assumptions. Some we explored in detail, others we left for further discussion after Dagstuhl (see Section 4.1.7 below)

- Decision making is a stepwise learning process.
- At each step the Decision Maker interacts with the system:
 - providing specific preference information,
 - getting explanations in terms of consequences of this information on a preference structure in a sample of solutions,
 - being informed by the system of inconsistencies of preference statements with respect to the model,
 - being able to revise previous preference statements.
- The generation of the explanations relies on a preference model.
- Expressed preference information is either solution-based (indirect) or model-based (direct).
- The set of solutions is fixed or is progressively discovered along the iterative process.

One assumption that we did not make explicit in our discussions but was implicit throughout is that we assumed that all interactions should be in natural language supported by tables and charts, exactly as they would be if a human decision analyst led them. We also recognised that the process of interaction needs to be driven by both sides. The System needs to elicit judgements and explore issues, moving the decision maker through a series of stages defined by a multi-objective decision analytic methodology. But equally the Decision Maker needs to be able to interrupt the flow and ask for explanation of a particular point in the reasoning or perhaps volunteer information that he or she believes is relevant. There also needs to be the possibility for the Decision Maker to reject the preference model being used — or, equivalently, its assumptions — and similarly for the System to recognise that a different model may be needed and adjust the interaction strategy accordingly.

4.1.3 Previous work

The group were aware of several pieces of earlier work in this area including [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

4.1.4 Questions that might arise during interactions

We were concerned to think about the types of question that might be asked during the interaction. Some might be posed by the analyst with the aim of eliciting judgements or stimulating reflection on the part of the decision maker; others might be asked by the decision maker to seek an explanation for part of the developing reasoning or step in the process.

The following lists are not intended to be exhaustive.

Questions that might be asked by the decision maker

- Why should I consider a instead of b ? Give me an explanation that involves this kind of preference information? Why is the model not able to compare a and b ?
- Why should I consider a as the best solution?
- Is the best solution unique?
- Why do I have to work with that set of solutions?
- If I could change constraints on the decision space, what is the best I could do?
- How much of the work have I already done? If I stop now, what can you tell me?
- What will happen if I change some preference information?
- What should be changed now that b is equal to a ?
- Questions regarding inconsistencies:
 - Why did you tell me I am inconsistent?

- Why I am not acting rationally?
- When have I provided inconsistent preference information? Which alternatives have I already seen?
- When did I tell you this?

Questions that might be asked by the System

- What is the argument to eliminate these solutions? (e.g. from the search)
- Please confirm you earlier choice because it is particular influential on the solution.
- You are departing from linearity... are you really like this?
- Do you agree that there could be interactions?
- Could it be possible to have a pair to explain the reason for non-independence of criteria?
- How important this particular criteria was in the selection (at the end)?
- “Do you really think safety is unimportant?”

4.1.5 Two examples

To give some substance to our discussions we divided into two subgroups and developed two examples of interactions between a Decision Maker and the System.

Even Swap

The first is based on Keeney, Hammond and Raiffa’s Even-Swap approach [11, 12]. The example below suggests how a System might use this approach in supporting a Decision Maker in choosing the location of an office. Remember that it is simplified: real problems and real interactions would need be more complex.

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

DM: which is the best alternative among b, c, d ?

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Your earlier answers suggest that c and c' are worth the same to you because increasing the commute time from 20 to 25 is balanced by increasing the clients from 70 to 78.

Is that right ?

[If no, it would be a more complex interaction]

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Hence we can compare c' to b and d instead of comparing c .

System: Notice that Commute is now 25 for all three options.

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Your earlier answers suggest that c' and c'' are worth the same to you because improving the service from C to B is balanced by an increase in cost from 1500 to 1750.

Is that right ?

[If no, it would be a more complex interaction]

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Hence we can compare c'' to b and d instead of comparing c' .

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Now compare b and c'' . Notice that b is as good as or better than c'' on all criteria. So, b dominates c'' , and c'' can be eliminated.

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: So c'' can be eliminated (hence c' and c)

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Your earlier answers suggest that d and d' are worth the same to you because reducing service level from A to B is balanced by an decrease in cost from 1900 to 1800.

Is that right ?

[If no, it would be a more complex interaction]

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Hence we must compare b to d' instead of comparing b to d .

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Your earlier answers suggest that b and b' are worth the same to you because an increase in size from 700 to 950 is balanced by an increase in cost from 1700 to 1950.

Is that right ?

[If no, it would be a more complex interaction]

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Hence we must compare b' to d' instead of comparing b to d' .

	b	b'	c	c'	c''	d	d'
Commute	25	25	20	25	25	25	25
Clients	80	80	70	78	78	85	85
Services	B	B	C	C	B	A	B
Size	700	950	500	500	500	950	950
Cost	1700	1950	1500	1500	1750	1900	1800

System: Now compare b' and d' . Notice that b' is as good as or better than d' on all criteria. So, b' dominates d' , and d' can be eliminated.

System: So b is the best solution among b , c , and d .

This even-swap approach could be programmed up into a decision support system with some effort, but without significant conceptual difficulty. There are even-swap codes already developed but these run through the analysis without ‘explaining’ the logic fully. Nonetheless adding such explanation would be achievable task with today’s language processing methods. Generally, however, we were thinking of a much less structured process which worked through the entire decision analysis process: identifying the issues and choice, structuring them into a decision model, identifying appropriate constraints on preference models to ‘fit’ the decision maker’s preferences subject to some rationality requirements, eliciting judgements, and identifying the optimal choice or choices, all the time explaining what the system was doing. The second group discussed this more sophisticated approach.

4.1.6 User-System interaction: possible scope and example

We distinguish five different types of information which has to be communicated between system and user:

- Information regarding concepts and methods and procedures.
- Problem related information (data).
- User related information.
- Preference information.
- Self-evident output and detailed explanations derived from prior articulated preference information.

Table ?? illustrates based on a possible interaction five different types of information which has to be communicated between system and user.

In the following paragraphs, the numbering (n) refers to the interaction identifier in Table ??.

The first type is the *information regarding concepts, methods and procedures*. For a user friendly navigation through the decision process, the system should upon request explain

■ **Table 1** Table illustrates based on a possible interaction five different types of information which has to be communicated between system and user.

‘Speaker’	Interaction	Inter- action Iden- tifier.
System:	Hello user. How can I help you?	1
User:	I want to buy a car. But I do not know which one is the best for me.	2
System:	Are you experienced in exercising this kind of a decision?	3
User:	Yes, I am.	4
System:	Could you describe the decision problem?	5
User:	Yes.[user describes the problem].	6
User:	I would like to identify the best five alternatives. Six criteria are relevant.	7
System:	Are you interested in a rough approximation which could realized with not much effort or are you interested in an optimized solution which would cost you time and much cognitive effort?	8
User:	I do not have that much time. I am interested in a rough approximation.	9
System:	Are you familiar with holistic pairwise comparisons?	10
User:	No.	11
System:	In a holistic pairwise comparison you have to compare two alternatives. For example you could prefer a to b, you are indifferent or you are not able to do the comparison.	12
User:	Why you are asking me about pairwise comparisons?	13
System:	Psychologists confirm that DMs are more confident exercising their decisions than explaining them. The most natural is a holistic pairwise comparison of some alternatives relatively well known to the DM. You told me earlier that you do not have much time. The use of pairwise comparisons in preference elicitation could achieve this best.	14
System:	Please compare this six pairs [system provides six pairs].	15
User:	[Does five pairwise comparisons].	16
User:	I am not able to execute the sixth pairwise comparison. Please provide me another one.	17
User:	Which alternatives are the best?	18
System:	There are 10 alternatives which could be possibly the best.	19
User:	I do not understand this. Could you explain it?	20
System:	Based on the information you gave me these ten alternatives are not dominated by other ones. They all could possibly be the best alternative.	21
System:	Would you like me to suggest further pairs for comparison?	22
User:	Yes.	23
System:	Please compare this four pairs.	24
User:	Why these particular four pairs?	25
System:	They will cut the number of alternatives that could be considered best.	26
User:	[Does comparisons].	27
System:	Please consider these five alternatives which could be the best.	28
User:	Why these five alternatives are the best?	29
System:	Gives explanation by values. Alternatives X, Y, Z have at least value 80 on criterion speed.	30
User:	Could you explain this in terms of preference information that I have provided?	31
System:	Gives a chain for each. The screen fills with 15 chains]. D > Q because you said that R > T.	32
User:	I would like to include alternative Q which is not in the five best you proposed. Is this possible?	33
System:	Yes, you need to revise R better T. Do you agree?	34
User:	Yes, I agree.	35
System:	Now these seven alternatives are the best ones. Would you provide preferences for these three pairs?	36
User:	How many more would I have to do until I am finished?	37
System:	Probably overall six to eight additional pairs.	38
User:	Please show the result if I would stop now.	39
System:	The result is the following...	40
User:	I like the results.	41
User:	Please print a report.	42

which options the user has (8) and which input will be required if she chooses this optional procedure (22, 37). All user questions and commands are part of this task. If the user is not familiar with a method or concept she should be able to ask the system (13) and get an answer of the system (12). It is important that the user decides on the progress through the procedure and that she never faces a situation in which the system does not offer help or alternatives. For instance, if the user is not able to make a suggested pairwise comparison, the system has to suggest another pairwise comparison (18). Also, if she does not want to provide additional preference information the system must offer the option of asking for the preliminary results based on already given preference information (39) and allow ending the procedure (42) if the user is satisfied with these results. The information that the system provides the user already exists or should relatively easily be created. There are different accepted and proven forms for the system to transmit the relevant information to the user. The flow of information in the other direction, from user to system, is more difficult because the system has to understand what the user wants. The challenge for this type of information lies in the individual tailoring.

The second type is *problem related information (data)*. For example the description of the decision problem and constraints which do not depend on the preferences of the user (2, 6). The challenge is to provide an interface which is able to deal with and interpret whatever kind of information. The more structured and based on numbers this information are, the easier is the further processing for the system.

The third type is *user related information* for example whether the user is experienced in exercising decisions (3) or whether she knows special concepts or methods (11). This information is necessary for individual tailored explanations and an efficient as well as effective procedure. An expert for decisions is able to understand more complicated argumentations in comparison to a layman. This requests that the code adopts itself to different types and differently experienced users require the use of different approaches, i.e. either selecting different approaches or limiting the validity of the results. The difficulty here consists of the elicitation of the relevant information and the integration in the decision process.

The fourth type is *preference information*. The user can articulate her goals (9) and carry out some given pair wise comparisons (16, 27). Thereupon the user should have the opportunity to articulate her preferences proactively, for example, if she does not agree on preliminary results based on her earlier articulated preferences (33) or if she confirms preference information (35). The structured questions of the system can relatively easily been modeled since they can be derived from the used models. The challenge lies in the proactively provided information by the user. The system has to be able to deal with any kind of preference information articulated by the user.

The fifth type of information contains *self-evident output and detailed explanations derived from prior articulated preference information*. If a system should be accepted by the user it “must give plausible and credible recommendations and provide convincing justification for those recommendations using terminology and logic understood and trusted by the users” [4].

Greer et al. [4] summarize the following by Kass [5, 6] raised issues regarding explanations:

1. “A good explanation is relevant, convincing, and understandable to the user.
2. A relevant explanation answers the immediate question and addresses the user’s higher goals.
3. An explanation must convince the user that the recommendation is correct.
4. An explanation can be readily understood if:
 - it is appropriate to the user’s knowledge of the domain
 - it is economical and concise

- it is organized
- it is expressed in terms of familiar concepts, and
- it requires little cognitive processing or indirect inference by the user.”

Most simple is the communication of (intermediate) results (19, 40). The system can justify which method it suggests based on the information the user has earlier provided. For example that pairwise comparisons should be used for the elicitation of the preferences because they could lead fast to a rough approximation as requested by the user (14) or the system can provide the user a pairwise comparison which the user should confirm if she is consistent in her preferences (34). Thereupon the system can explain results (21). Such an explanation can be more complicated and based on information the user has not directly articulated (30). The system can also explain how long the whole process will take based on the already elicited preference information (38).

4.1.7 Future work

The group agreed to continue work after the Seminar and develop these ideas into a *Manifesto for Interactions between Decision Makers and Decision Support Systems*. The intention is to prepare such a paper and submit it to a mainstream journal with a view to stimulating further work.

Since the Workshop two papers have been drafted:

1. Salvatore Corrente, Salvatore Greco and Roman Slovinski (2012) “Rough set and rule-based explanatory decision support”.
2. Milosz Kadzinski (2012) “Review of some explanatory decision support systems and underlying methods”.

References

- 1 V. Bertsch, S. French, J. Geldermann, R. P. Hämäläinen, N. Papamichail and O. Rentz. Multi-criteria Decision Support and Evaluation of Strategies for Environmental Remediation Management. In *OMEGA*, 37(1):238–251, 2009.
- 2 G. Carenini and J. D. Moore. Generating and evaluating evaluative arguments. In *Artificial Intelligence*, 170:925–952, 2006.
- 3 T. Dodson, N. Mattei, and J. Goldsmith. A Natural Language Argumentation Interface for Explanation Generation in Markov Decision Processes. *Presented at ADT2011*, DIMACS, Rutgers, New Jersey, 2011. http://adt2011.org/papers/paper_12.pdf
- 4 J. E. Greer, S. Falk, K. J. Greer and M. J. Bentham. Explaining and justifying recommendations in an agriculture decision support system. In *Computers and Electronics in Agriculture* 11(2-3):195–214, 1994.
- 5 R. Kass and T. Finin. The need for user models in generating expert system explanation. In *International Journal of Expert Systems*, 1(4):345–375, 1988.
- 6 A. M. Kass and D. B. Leake. Case-Based Reasoning Applied to Constructing Explanations. In *Proceedings of 1988 Workshop on Case-Based Reasoning*, pp. 190-208, 1988.
- 7 D. A. Klein. Decision-Analytic Intelligent Systems: Automated Explanation and Knowledge Acquisition. *Lawrence Erlbaum Associates*, 1994.
- 8 C. Labreuche, N. Maudet and W. Ouerdane. Minimal and Complete Explanations for Critical Multi-attribute Decisions. In *Algorithmic Decision Theory*, pp. 121-134, 2011.
- 9 K. N. Papamichail and S. French. Explaining and Justifying the Advice of a Decision Support System: A Natural Language Generation Approach. *Expert Systems with Applications*, pp. 35-48, 2003. http://adt2011.org/papers/paper_39.pdf

- 10 K. N. Papamichail and S. French. Design and evaluation of an intelligent DSS for nuclear emergencies. In *Decision Support Systems*, 41:84–111, 2005.
- 11 J. S. Hammond, R. L. Keeney and H. Raiffa. *Smart Choices: a Practical Guide to Making Better Decisions*. Boston, Harvard Business School Press, 1998.
- 12 J S Hammond, R L Keeney, and H Raiffa. Even swaps: a rational method for making trade-offs. In *Harvard Business Review*, 76(2):137–138, 1998.

4.2 What and how can we learn from Pareto fronts and sets? (Working Group “Pareto Sense”)

Susanne Bürklen, Kalyanmoy Deb, Michael Emmerich, Karlheinz Kuefer, Boris Naujoks, Amos H. C. Ng, Akira Oyama, Silvia Poles, Tamara Ulrich, Katrin Witting

License  Creative Commons BY-NC-ND 3.0 Unported license

© Susanne Bürklen, Kalyanmoy Deb, Michael Emmerich, Karlheinz Kuefer, Boris Naujoks, Amos H. C. Ng, Akira Oyama, Silvia Poles, Tamara Ulrich, Katrin Witting

Our working group was concerned with the question what and how a decision maker can learn from the Pareto set, Pareto front and the mapping between these sets. We discussed several topics:

1. What and how can we learn from Pareto fronts?
2. What and how can we learn from Pareto sets in decision space and from the mapping into objective space?
3. How do constraints influence the solution sets?

The discussions resulted in a broad collection of properties of Pareto fronts and how these can be interpreted in a decision context. Moreover a range of methods, in particular visualization methods, for measuring properties and analysing Pareto optimization results were brought together put into a structured view. Interesting question for future research were identified and ideas for an extended report (white paper) with a collection of properties and analysis methods with examples for the interpretation and explanation of the observed properties.

Already when looking at only the Pareto front many structural properties can be observed that have an interpretation for decision making. Among others these are special points and regions (knees, bents, gaps, elbows, cusp points, etc.), correlation between objectives, convex and concave shapes of the Pareto front or projections of it. Well-balanced compromise solutions are often located at knees. Gaps and cusp points often indicate structural transitions (e.g. discrete choices, hysteresis, bifurcations, etc.).

To gain a better understanding of these can reveal interesting insights into the structure of the optimization problems or help the decision maker in navigation across the Pareto front. In the engineering context design principles could be derived from this. Additional analysis focusing on critical regions can be used to identify parameters that are responsible for their occurrence and this can reveal interesting design principles for instance in the context of innovation.

There are various tools for visualizing Pareto fronts. 2-D and 3-D scatter plots and surface plots are very common. In particular in 3-D, plotting also the attainment surface can help to visually locate the position of points in a 3-D plot. In 4-D and higher dimensions we may use shapes, colors, size of points and even blinking patterns to indicate additional objective function values in 3-D projections (as for instance done in the LIONSolver). For dense approximation sets it can occur that the points that are overshadowed are not visible. Slicing can be used in this case, or even movies that remove layers of non-dominated solutions

in the 3-D projection. In N -dimensions, techniques from multidimensional data visualization can be used and a variety of methods is available, such as Parallel coordinates diagrams, heatmaps, interactive decision diagrams, clustering-based approaches. Besides, textual and rule based descriptions of the Pareto front might reveal its structure and interesting patterns.


In order to learn from Pareto optimization, decision space information should be combined with information from the objective space. In particular, the preimage of the Pareto front is of interest. In parametric spaces (e.g. decision space is real valued) it is possible to combine decision variables and objective function variables in diagrams, for instance in the Parallel coordinates diagram. A challenging question is how to gain intuition about the mapping and decision space in case of non-parametric decision spaces or structures, such as molecules, bridge constructions or airfoil shapes. For this case it was rendered to be a good approach to show animations along the Pareto front (2-D case), across the Pareto front (higher dimensional Pareto fronts). Moreover, viewing animations moving from non-optimal subspaces towards the Pareto front can teach intuition of what makes solutions Pareto optimal.

An important information is, whether Pareto optimal solutions occur at the boundary of constraints, and if so, which constraints are active. It is important to know this, because it might be possible to relax constraint boundaries, for instance if a constraint occurs at the preferred solution and relaxing the constraint can further improve it. There are even techniques that relax the constraints until an ideal solution can be found, but it is questionable whether these techniques have a wide scope. In addition, solutions at the constraint boundary are often not robust solutions, and a decision maker might prefer a solution, if less constraints are active.

Therefore, constraints data is important for decision making.

4.3 Evaluating, Measuring, Quantifying Learning (Working Group “Quantifying Learning”)

Jürgen Branke, Jussi Hakanen, Markus E. Hartikainen, Hisao Ishibuchi, Enrico Rigoni, Karthik Sindhya, Theodor J. Stewart, Margaret M. Wiecek

License  Creative Commons BY-NC-ND 3.0 Unported license

© Jürgen Branke, Jussi Hakanen, Markus E. Hartikainen, Hisao Ishibuchi, Enrico Rigoni, Karthik Sindhya, Theodor J. Stewart, Margaret M. Wiecek

4.3.1 First Phase of Discussions

It was agreed to focus, initially at least, on learning experienced by the decision maker in using the methods, in contrast to learning by the algorithm (but see the final section). The purpose of measuring learning is perhaps primarily to assess and to compare methods, but it was recognized that inevitably there would also be an evaluation of decision makers.

Within this context, three issues were discussed, namely: What can be learnt? What information is potentially available to provide measures of learning? How can such information be exploited to provide an operational measure of learning?

What can be learnt?

The following items were identified:

1. Whether a solution is Pareto-optimal
2. Whether a target is achievable

3. Range of each objective
4. Shape of frontier/Identification of knees
5. Own preferences, relative importance of objectives
6. Criteria that may be missing
7. Constraints that may be missing
8. The absence of a satisfying solution, and the need to increase the search space
9. Causes for trade-offs and the mapping between decision and objective spaces (to give insight into problem and to support “innovization”)
10. Mapping between preference and objective spaces (to identify what preferences lead to what areas in the objective space and ultimately the corresponding decision space).

What information is potentially available to provide measures of learning?

A wealth of information is potentially available to assess learning, varying from quantitative performance measures of the algorithmic implementation to subjective assessments of the extent of learning experienced. Types of information available in principle includes:

1. The process, or sequence of interactions followed, e.g.:
 - Number of solutions (Pareto or non-Pareto optimal) visited
 - Inconsistency of responses
 - Backtracking
2. Number of relationships identified/explored (if available in method)
3. Rate of change in response times
4. Does the DM prefer the solution found to all in a sample of Pareto optimal solutions?
5. Subjective evaluation of learning demonstrated, as assessed by the analyst or an external observer
6. Can the DM explain the rationale behind the choice (judged by analysts or external observer)
7. Direct questions posed to the DM before and/or after process, e.g.:
 - Expressed preferences before and after process
 - Compare final solution with prior assessments of attainable outcomes
 - Sketches of perceptions of 2-dim slices through the Pareto frontier
 - Statements of importance of objectives (before and after process)
 - Confidence and satisfaction in solution found
 - Is the DM still happy with answer two weeks later?
 - Other questions in a structured questionnaire
8. Changes in process or result with repeated analysis using the same or a different method

How can such information be exploited to provide an operational measure of learning?

This is the primary challenge to future research. Some of the issues identified are the following:

1. How should we set up hypothetical (simple but realistic) test problems, on which experiences with different methods on sets of “decision makers” (e.g. students) can be evaluated.
2. We should develop a variety of test cases for different contexts (e.g. business, engineering, environment)
3. The operational feasibility of the potential measures needs to be investigated, for example:

- How do we seek the right balance between “objective” and “subjective” measures?
- How should we interpret even objective measures, e.g. whether visiting a larger number of solutions is an indicator of poor or rich learning.
- The design of an effective questionnaire?

The group split into two sub-groups on the last day, in order to probe some of the above issues further. The results of these discussions are summarized in the next sections.

4.3.2 Second Phase of Discussions

Sub-group 1: Evaluating/measuring DM’s learning by monitoring him/her

This sub-group looked into evaluating and/or measuring DM’s learning by monitoring him/her while he/she is using an interactive method to solve the multiobjective optimization problem. The idea was that if the analyst had kinds of rules, the analyst could further develop methods to support learning, to determine whether the DM has learned (without asking him/her), to help the DM learn by guiding him/her and even to suggest a change of method if the analyst can determine that the DM is not learning.

First, the sub-group made different hypotheses on what the behavior of the DM could look like when he/she learns and what distinguishes it from one that is not learning. However, this turned out rather difficult and it was concluded by the sub-group that it seems to be hard to distinguish between DM’s learning and his/her growing confusion – both of these may lead to changes in the DM’s behavior. Thus, the sub-group decided to pursue an alternative direction of thought.

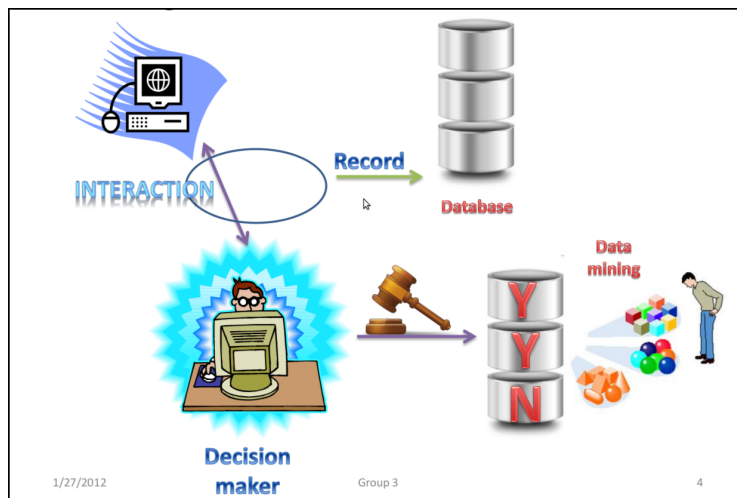
It was concluded that one should set up an experiment for determining the rules for whether whether the DM is learning. In the experiment, all the interaction between the DM and the interactive method should be recorded and whether the DM has learned or not should be determined through a questionnaire. With a sample that is large enough, data mining techniques can be used to derive rules that distinguish the ones that have not learned from the ones that have. Particular attention must naturally be paid to designing the questionnaire and to designing the monitoring tools and there are also other issues to resolve before doing the experiment. However, the benefits of the rules that could be the result of this experiment would be great, as explained also previously. An outline of the designed experiment is shown in Figure 1.

Even though the sub-group decided that they could not come up with the rules without an experiment, they were able to give some hypotheses on what distinguishes the behavior of a learning DM from a one that is not learning. The sub-group agreed on the following rules for a learning DM:

1. There is change in search direction and consistency after that.
2. Step size of the interactive method decreases when the area of preferred solution has been found.
3. On the later stages of using the method, there is an almost monotonic convergence to the final solution.
4. Response time of the DM decreases in the end.

On the other hand, it was agreed that learning may not happen, when the following rules apply:

1. The DM stays in a small area for the whole time that he/she is using the method.
2. There are continuous (almost random) changes in the search direction



■ **Figure 1** A graphical illustration of the experiment to derive rules that determine whether the DM has learned

The rules for a learning DM imply a change in the DM’s thinking that then stays consistent after the change and the rules for a non-learning DM imply either a continuous changes or unwillingness to try anything new. However, as stated earlier the validity of these hypothesis must be evaluated through the experiment shown in Figure 1.

Sub-group 2: Quantifying algorithmic learning

Our first observation when considering the learning of algorithms was that only some approaches learn explicitly a model of the user preferences.

Examples include

- The Zionts/Wallenius method
- MACBETH
- UTA/GRIP/ ...
- AHP in the absolute measurement mode.

Other approaches don’t learn explicitly, but rely more on the user to learn from the interaction in order to guide the search. Examples include

- Reference point methods
- ELECTRE
- Geoffrion/Dyer/Feinberg.

Because the concept of algorithmic learning makes more sense in the first, explicit model learning group of algorithms, we decided to focus on these.

We concluded that the learned model is only really useful if it can generalize to previously unseen alternatives.

So we assume that a method is “trained” (used) on a particular training set (e.g., a given set of preference relations) or interactively with a particular DM.

It is then validated on an additional set of solutions, preferably a representative sample of Pareto optimal solutions with known total preference information.

Then, the following things could be measured:

- How many preference relations can be decided?


- How many of those are decided correctly?
- If they are incorrect, by how much? Note that it may be interesting here to look at the DM's opinion as well as the difference in estimated value.
- Even if the preference relation is determined correctly, does the magnitude of value difference match the DM's?

With all these measures, some relationships may be considered more important than others. For example, a correct ranking of the best solutions may be more important than a correct ranking of the worst solutions.

These concepts open the way to empirical (experimental) research in which the approaches can be evaluated (within student groups for example) on the basis of the above measures.

4.4 Navigation in Multi Objective Optimization Methods (Working Group “Navigation”)

Richard Allmendinger, Heinrich Braun, Matthias Ehrgott, Xavier Gandibleux, Martin J. Geiger, Kathrin Klamroth, Pekka Korhonen, Mariano Luque, Eckart Zitzler

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Richard Allmendinger, Heinrich Braun, Matthias Ehrgott, Xavier Gandibleux, Martin J. Geiger, Kathrin Klamroth, Pekka Korhonen, Mariano Luque, Eckart Zitzler

4.4.1 Introduction

Many practical problems can best be described involving several criteria. In the case of optimization problems, this leads to the loss of the formal, but straight-forward definition of optimality. Contrary to the existence of a single optimal solution, an entire set of Pareto-efficient outcomes might exist that ‘optimizes’ the considered criteria. Besides the complexity of such problems, this raises the problem of making a selection of a, from a decision makers point of view, most-preferred solution.

Numerous different preference elicitation methods are available to facilitate the process of constructing representations of the decision makers’ preferences. Besides, other techniques exists that allow an interactive search for a most-preferred solution, without necessarily relying on the construction of an explicit notion of the actual preferences. With the rise of human-machine-interfaces, and the availability of powerful computer hardware, we believe such techniques to play in increasingly important role in the future. Consequently, some formal considerations of this field of research are needed, which ultimately should lead to a structuring of existing approaches, and a stipulation of future research.

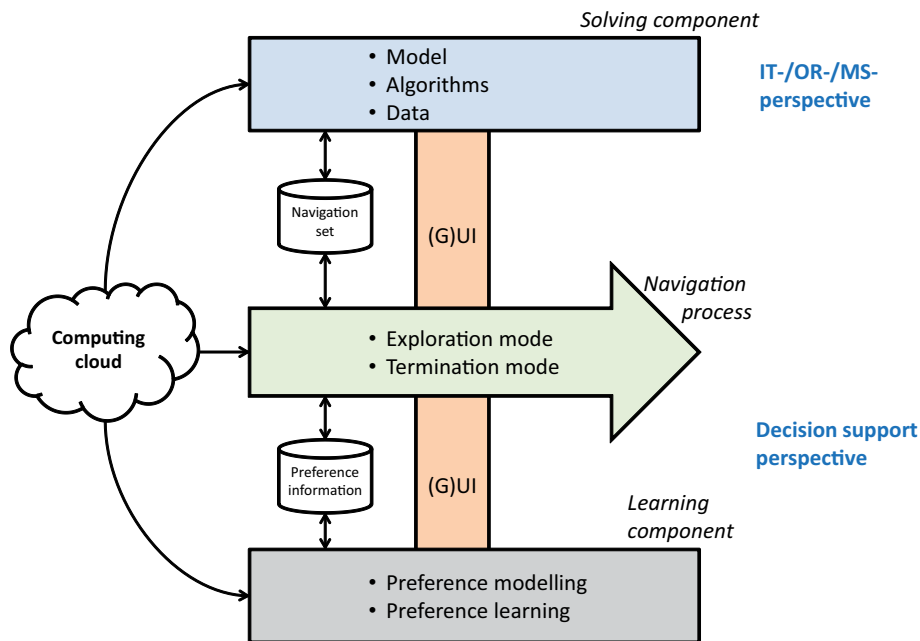
One way of approaching the above sketched topic can be found in the introduction of the concept of *navigation*, which we define in the following.

► **Definition 1** (Navigation). *Navigation* is the interactive procedure of traversing through a set of points in objective space guided by a decision maker (DM). The ultimate goal of this procedure is to identify the most-preferred Pareto-optimal solution.

The following Figure 2 depicts the concept of *navigation*, its’ integration in an IT-landscape, and its’ relation to reasoning/preference learning.

4.4.2 Key Aspects of Navigation

Following the rather general definition above, some more precise elaborations are needed in order to fully understand the concept of *navigation*. We believe the following integral aspects



■ **Figure 2** A framework of navigation.

to be of relevance:

1. *What is the set to be navigated?*

The set could be: The entire Pareto-front, a true subset of the Pareto-front, or any other set of points. Consequently, this includes *a posteriori* approaches in which the Pareto-front is identified/approximated before the navigation phase. A further special case is found in most classical *interactive approaches*, which consider a single outcome and progress from there.

2. *How to navigate?*

In general, the iterations of the *navigation procedure* are triggered by actions of the decision maker. Once such statements become apparent, the system reacts such that new points are computed in real time/selected and presented to the decision maker. Potentially, this action modifies the navigation set.

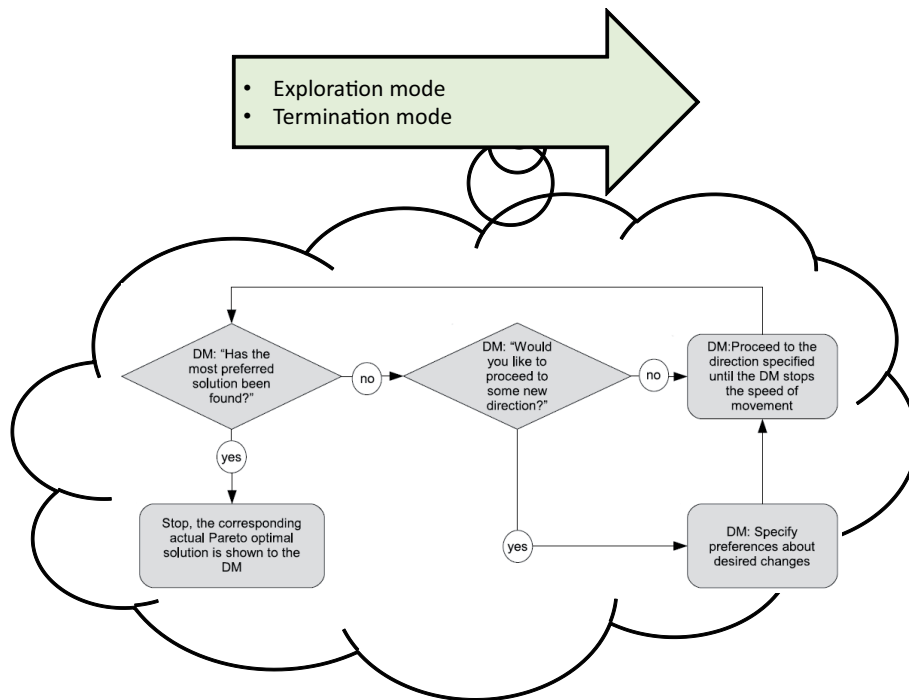
Consequently, any navigation procedure therefore describes an alternation of *move-* and *dialog-* phases.

Optionally, the DM also takes into consideration the information from the decision space. Prominent examples are found in engineering design applications, vehicle routing, and other complex decision problems.

3. *Guidance provided by the navigation*

Exploration mode: During the exploration process, the control is fully in the hand of the decision maker. In this mode, the decision maker learns about the problem. Guidance/support provided by the method can facilitate this process. Relevant examples of such guidance include (i) cycle detection, (ii) information about the possible alternatives, (iii) direction derived from the navigation history, (iv) statistics of the navigation history, (v) intensification/diversification characteristics of navigation steps.

Termination mode: At the moment of termination, the decision maker might ask for strong support in order to be convinced that he/she has found the most-preferred



■ **Figure 3** Navigation including *move-* and *dialog-* phases.

solution. This information could be provided by the use of value functions which are e. g. extracted from the statements made by the decision maker. In this mode, the system learns from the DM.

4.4.3 Features of Navigation

Out of the integral properties of *navigation*, several features arise.

- Pareto vs. non-Pareto search?

On the one hand, when navigating in the Pareto set only, any navigation direction implies the worsening of at least a single objective. On the other hand, navigation between feasible (non Pareto-optimal) points may allow for a simultaneous improvement without ‘sacrificing’ the current values.

This has some implications for possible navigation directions of the actions given by the DM. In any case, and ultimately, the final outcome of navigation should be a point of the current navigation set for which there is no other point known dominating it.

- Different starting points

A key question is whether the same ultimate point is reached when starting from different points. We believe this to be the case if certain assumptions are made with respect to the value function of the DM and the consistency of the navigation and the preference/direction statements.

- Behavioral aspects (e. g. inconsistent behavior)

Especially in the exploration phase, a certain amount of ‘inconsistent’ behavior is to be expected. This stems from the fact that the DM explores the navigation set in order to learn about the problem. As a consequence, any method implementing *navigation* should account for this issue.

Following the implications from prospect theory, decision makers may not judge symmetrically with respect to gains and losses of previously obtained outcomes. Navigation methods can take this into account by selecting a dominated starting point.

- Discrete vs. continuous, linear vs. nonlinear, convex vs. non-convex
The precise properties of the problem are important. Whether the considered problem is discrete or continuous influences the type of navigation which can be used. In both cases, discrete representations satisfying different aspects (hypervolume, uniformity, coverage, approximation error, ...) can be used as the basis for navigation.

4.4.4 Previous Research Related to Navigation

Methods

- Pareto navigator
Pareto Navigator [1] extends the ideas of Pareto Race [2] to nonlinear convex and mildly nonconvex problems with multiple objectives. 1. In a preprocessing (*initialization phase*), a convex polyhedral approximation of the nondominated set is computed using an appropriate approximation method. In this way, the Pareto Race concept can be transferred to nonlinear problems, and expensive objective function evaluations can be avoided during the interactive navigation phase. 2. After specifying an initial solution (e.g., from the previously computed approximation), the decision maker can explore the nondominated set and collect trade-off information by navigating in the polyhedral approximation.

In each iteration of this *navigation phase*, the decision maker specifies a search direction, for example, by a classification approach or by directly specifying a reference point. The movement towards this direction is realized using parametric linear programming on the polyhedral approximation, and is visualized using, for example, value paths with appropriate steplengths. 3. At any time during the navigation, the decision maker can change the speed of the movement, the direction, or request the computation of the closest nondominated point, i.e., the projection of the current solution to the actual Pareto optimal set. This point can then be included in the approximation and the search can be continued, or the decision maker may choose to terminate the search at this point.

When the decision maker has completed the learning phase with the Pareto Navigator, he or she may wish to continue with some other interactive method to complete the decision phase, or simply stop with the final solution found.

- Nautilus
NAUTILUS [3] is an interactive method based on an unusual set of navigation: through a set of points that can be feasible or unfeasible and where all points are dominated by at least one non-dominated objective vector except the last solution. This last solution will be an efficient solution and should result the most preferred solution.
Plenty of the interactive methods for multiobjective optimization are based on the sequential determination of non-dominated objective vectors, by introducing new preferential information at each iteration. This means that the decision maker must always allow the impairment of at least one objective function to produce the next iteration. The main purpose of this method is to eliminate 'the sacrifice' of at least one objective function at each iteration, due to the the psychological assumption that people do not react symmetrically to gains and losses.

Other important purpose is to avoid the anchoring effect mainly due to the starting point.

In this method, each solution dominates the previous one, whereupon the navigation is always carried out improving all objective functions in a given direction. This direction is obtained through the consideration of preferential weights that reflect the DM's preferences and where, by minimizing an achievement scalarizing function, the search is oriented towards the part of the Pareto front that the DM prefers. In this navigation process, useful information for the DM is the range of the attainable values for each objective function at each iteration (upper and lower bounds). These ranges are contracted at each iteration, allowing guide the search toward the desired part of the Pareto front.

Applications

- Closed-loop optimization scenarios

Closed-loop optimization scenarios are characterized by the feature that the evaluation of candidate solutions involves to conduct real experiments, e.g. physical or biochemical experiments, and/or to run expensive computer simulations [4, 5, 6, 7, 8]. Examples of such applications include many scientific and technological problems including in areas like drug discovery and manufacturing [9, 10], analytical biochemistry [11], experimental quantum control [12], robotics [13], electronics design [14], food science [15]. In addition to expensive evaluations, closed-loop problems are often subject to multiple objectives, limited resources, and user preferences may be available too (further challenges include noisy fitness values, uncertainty, and constraints). A common situation in closed-loop optimization is that the Pareto-front of a problem needs to be approximated within a relatively small number of evaluations (due to limited resources). Subsequently, navigation can be employed (offline) to explore the front (e.g. correlations between objective values and/or decision variables) and to account for user preferences in an interactive manner. Ultimately, navigation supports an experimentalist in the (i) process of selecting the most-preferred solution in the Pareto-front, which is then realized in real-world, and in (ii) understanding the importance of specific control variables and certain (manufacturing) processes.

- Multi-objective vehicle routing

Applications of the *vehicle routing problem* (VRP) are typically found in the physical distribution of goods. Customers are visited by vehicles which ship/collect certain goods from/to one or several depots. Obviously, cost criteria are important, with the minimization of the traveled distances as a prominent example of an objective function. Besides, the service provided by the logistical companies comes into play, often being expressed as the agreement of service with promised delivery dates or time windows. Consequently, vehicle routing presents itself as a multi-objective problem, in which the balancing of the considered objectives is of importance [16].

Interactive approaches involving concepts of navigation have recently been adopted to the application domain of the multi-objective VRP. In the work of [17, 18], the decision maker is given the opportunity to state his/her preferences by means of an overall utility function, combining different objectives into an overall evaluation. The system then computes an alternative maximizing the currently stated utility function, and reports it back to the DM. In a subsequent *navigation phase*, changes to the utility function are permitted, and an adaptation of the presented solution to the altered utility function definition is tried by the optimization approach. In this spirit, the search for

alternatives follows the directions given by the DM. The traveled navigation set depends on the properties of the global utility function. In case of a function employing a convex combination of criteria, the search navigates towards solutions lying on the convex hull of the Pareto front. However, and this is due to the heuristic nature of the implemented optimization approach which relies on local search, sub-optimal alternatives might be reported back also.

Interactive search finally terminates when the DM chooses so. In a practical application, this is the case when the DM has visited enough alternatives to build his/her preferences, thus converging towards a most-preferred solution.


References

- 1 P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 23:211–227, 2010.
- 2 P. Korhonen and J. Wallenius. A Pareto race. *Naval Research Logistics*, 35:615–623, 1988.
- 3 K. Miettinen, P. Eskelinen, F. Ruiz, and M. Luque. NAUTILUS method: an interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, 206(2):426–434, 2010.
- 4 G.E.P. Box. Evolutionary operation: A method for increasing industrial productivity. *Applied Statistics*, 6(2):81–101, 1957.
- 5 H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. PhD thesis, Technical University of Berlin, 1975.
- 6 I. Rechenberg. Case studies in evolutionary experimentation and computation. *Computer Methods in Applied Mechanics and Engineering*, 2-4(186):125–140, 2000.
- 7 J. Knowles. Closed-loop evolutionary multiobjective optimization. *IEEE Computational Intelligence Magazine*, 4(3):77–91, 2009.
- 8 R. Allmendinger. *Tuning Evolutionary Search for Closed-Loop Optimization*. PhD thesis, University of Manchester, Manchester, UK, 2012.
- 9 S.S. Farid. Process economics of industrial monoclonal antibody manufacture. *Journal of Chromatography B*, 848:8–18, 2007.
- 10 B.G. Small, B.W. McColl, R. Allmendinger, J. Pahle, G. López-Castejón, N.J. Rothwell, J. Knowles, P. Mendes, D. Brough, and D.B. Kell. Efficient discovery of anti-inflammatory small molecule combinations using evolutionary computing. *Nature Chemical Biology*, 7:902–908, 2011.
- 11 S. O’Hagan, W.B. Dunn, M. Brown, J. Knowles, and D.B. Kell. Closed-loop, multiobjective optimization of analytical instrumentation: Gas chromatography / time-of-flight mass spectrometry of the metabolomes of human serum and of yeast fermentations. *Analytical Chemistry*, 77(1):290–303, 2005.
- 12 O.M. Shir. *Niching in Derandomized Evolution Strategies and Its Applications in Quantum Control: A Journey from Organic Diversity to Conceptual Quantum Designs*. PhD thesis, University of Leiden, 2008.
- 13 I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi. Evolutionary robotics: The Sussex approach. *Robotics and Autonomous Systems*, 20(2-4):205–224, 1996.
- 14 A. Thompson. *Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution*. PhD thesis, University of Sussex, 1996.
- 15 M. Herdy. Evolutionary optimization based on subjective selection — evolving blends of coffee. In *European Congress on Intelligent Techniques and Soft Computing*, pp. 640-644, 1997.
- 16 N. Jozefowicz, F. Semet, and E.-G. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309, 2008.

- 17 M.J. Geiger and W. Wenger. On the interactive resolution of multi-objective vehicle routing problems. In *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007*, volume 4403 of Lecture Notes in Computer Science, pp. 687-699, 2007.
- 18 M.J. Geiger, W. Wenger, and W. Habenicht. Interactive utility maximization in multi-objective vehicle routing problems: A “decision maker in the loop”-approach. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)*, pp. 178-184, 2007.

4.5 Representations (Working Group “Representation”)


Carlos A. Coello Coello, José Rui Figueira, Carlos M. Fonseca, António Gaspar-Cunha, Kaisa Miettinen, Sanaz Mostaghim, Dmitry Podkopaev, Pradyumn Kumar Shukla, El-ghazali Talbi, Margaret M. Wiecek

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Carlos A. Coello Coello, José Rui Figueira, Carlos M. Fonseca, António Gaspar-Cunha, Kaisa Miettinen, Sanaz Mostaghim, Dmitry Podkopaev, Pradyumn Kumar Shukla, El-ghazali Talbi, Margaret M. Wiecek

This working group focused on the issue of learning about the Pareto-optimal set in both decision and objective space from a machine perspective. In this context, learning was understood as the process of obtaining a parsimonious representation of the Pareto-optimal set and/or front, either explicitly by storing points or implicitly by building a model, so as to allow relevant information to be produced in response to Decision Maker queries. A taxonomy of representations was outlined, raising awareness of the distinct requirements of approximate optimization methods, such as evolutionary multiobjective optimizers, and exact optimization methods.

4.6 Which questions should be asked to find the most appropriate method for decision making and problem solving? (Working Group “Algorithm Design Methods”)

Anne Auger, Dimo Brockhoff, Manuel López-Ibáñez, Kaisa Miettinen, Boris Naujoks, Günter Rudolph

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Anne Auger, Dimo Brockhoff, Manuel López-Ibáñez, Kaisa Miettinen, Boris Naujoks, Günter Rudolph

The group started with a general discussion about two different perspectives when designing methods for decision making and problem solving. One perspective is to define clearly delimited goals, benchmarks and/or evaluation criteria, and analyze existing methods according to these criteria, acknowledging their characteristics as a simplification of the real-world. This is the approach typically followed in single-objective optimization, where the selection of an optimization method can be even done automatically in some cases. In multi-objective optimization, such selection would require not only information about the optimization problem, but also about the decision maker (DM).

A different perspective is to address the more challenging issues in real-world applications. For example, objectives may be unknown a priori, there is no well-defined utility function, there may exist noise or inconsistencies in the objective functions or the behavior of the DM.

Another challenging issue is how to help the DM to learn, but this was a topic covered by a different working group.

The consensus reached was that the selection among existing method, or the design of new methods, should be guided by:

- The goals of the DM. For example, (i) explore trade-offs, (ii) identify the most preferred solution, or (iii) maximize the confidence in the preferred solution. We can only compare different methods as long as they have the same goals.
- The constraints of the problem. There may be a budget for the algorithm (time per iteration) and a different budget for the DM (in terms of number of queries to the DM).
- Assumptions about the DM behavior. For example, we can probably assume rationality and the existence of domain knowledge, but not consistency. Moreover, any practical model of learning must assume that human learning can be (theoretically) modeled by a *learning algorithm*.
- The semantics of the DM's answers, that is, DM's preferences are "*values with semantic*".

The outcome of the above consensus was a general model of the interaction of the DM and an optimization algorithm. This model assumes that the DM has an internal (but unknown) utility function. This internal utility function is not necessarily static, but it can evolve according to the set of visited solutions. A preference model is how the DM communicates her internal utility function to the algorithm, which may be a ranking of solutions, a search direction, aspiration levels, an explicit model, etc. The algorithm (implicitly or explicitly) tries to build a model of the DM's utility function. The ideal algorithm would be an oracle that gives always the same answer as the DM, that is, an algorithm that is able to predict the answer of the DM, taking into account the set of visited solutions, and the previous interactions with the DM.

With respect to this model, it would be very useful to have simulated/virtual DMs that may be used to define different benchmark scenarios for interactive algorithms, such as, (i) goal-driven, (ii) exploring trade-offs, (iii) find knee-points, (iv) find infeasible regions, etc.

The final goal would be to have open source benchmarks for interactive algorithms, such that different algorithms may compete using different virtual DMs on a particular benchmark scenario. The final conclusion on this topic was that the first step should be a survey of the literature about DM models, including the literature of multi-criteria decision-making, machine learning and artificial intelligence.

The second part of the discussion focused on what can be said starting from the most simplified version of the above model. Possible questions are: "*How fast can an algorithm identify the most preferred solution?*" and "*How many times the algorithm has to ask the DM to identify it?*". In this manner, we defined a very simple DM and a (1+1) interactive EA (iEA) applied to a simple binary problem (Leading-Ones-Trailing-Zeroes). The conclusion is that one can compute an expected number of function evaluations to identify the most preferred solution, and the expected number of queries to the DM. Further work should focus on extending these initial results to more complex DM models, algorithms and problems.

5 Seminar schedule

Monday 23rd January (Theme: Learning and Interaction)	
7.30-8.45	Breakfast
8.45-9.15	Opening welcome and introduction (Joshua Knowles)
9.15-9.45	Round of personal introductions (all participants)
9.45-10.30	Opening Invited Talk – Interactive Multiobjective Optimization from a Learning Perspective (Jürgen Branke, Julian Molina, Roman Słowiński)
10.30-10.45	Questions and discussion
10.45-11.00	Coffee break
11.00-11.45	Keynote Talk – User preferences in EMO: what can be learned from preference elicitation? (Vincent Mousseau)
11.45-12.00	Questions and discussion
12.15-13.15	Lunch
13.15-13.45	Contributed Talk – Modelling bipolar interactions in robust ordinal regression: the UTA GSS method (Salvatore Greco, Johannes Siebert, Roman Słowiński) (20min talk + 10 min questions and discussion)
13.45-14.15	Contributed Talk – Pareto Navigator: Learning-Oriented Method for Interactive Multiobjective Optimization (Jussi Hakanen, Kathrin Klamroth, Kaisa Miettinen, Vesa Ojalehto) (20 + 10min)
14.15-15.30	Participants to suggest questions as topics for working groups
15.30-16.00	Coffee break
16.00-17.00	Discussion of proposed questions and arrangement of participants into working groups
17.00-18.00	First meeting of discussion/working groups.
18.00-19.00	Dinner
Tuesday 24th January (Theme: User Preferences)	
7.30-8.45	Breakfast
8.45-9.15	Contributed Talk – Can a Linear Value Function Explain Choices? An Experimental Study (Pekka Korhonen, Kari Silvennoinen, Jyrki Wallenius and Anssi Öörni) (20 + 10min)
09.15-10.00	Keynote Talk – Cynefin: Problem Formulation and Uncertainty (Simon French)
10.00-10.15	Questions and discussion
10.15-10.45	Coffee break
10.45-11.15	Contributed Talk – A General Framework for Integrating User Preferences With Evolutionary Multiobjective Optimization – Towards Making the Weighted Hypervolume Approach User-Friendly (Dimo Brockhoff) (20 + 10min)
11.15-11.45	Contributed Talk – Optimization in Logistics from a Learning Perspective: The Case of the Multi-Objective Vehicle Routing Problem (Martin J Geiger) (20 + 10min)
11.45-12.15	Contributed Talk – Extreme ranking analysis and rank related requirements in multiple objective optimization (Milesz Kadzinski, Salvatore Greco and Roman Słowiński) (20 + 10min)
12.15-13.45	Lunch
13.45-14.15	Contributed Talk – Risk and Return in Multiobjective Optimization (Carlos Fonseca, Iryna Yevseyeva and Michael Emmerich) (20 + 10min)
14.15-14.45	Contributed Talk – Approximation Factor as the Aim of Multiobjective Optimization and the Hypervolume Indicator (Tobias Friedrich) (20 + 10min)
14.45-18.00	Second meeting of working groups (includes coffee break)
18.00-19.00	Dinner

Wednesday 25th January (Theme: Problem Understanding)	
7.30-8.45	Breakfast
8.45-9.15	Contributed Talk – Learning from Pareto-Front Approximations of Real-World Optimization Problems — A Clustering Approach (Tamara Ulrich) (20+10min)
09.15-10.00	<i>Keynote Talk</i> – Innovization: Learning Problem Knowledge Through MultiObjective Optimization (Kalyanmoy Deb)
10.00-10.15	Questions and discussion
10.15-10.45	Contributed Talk – Adapting MOEAs to solve practical complex engineering problems (António Gaspar-Cunha, José Carlos Ferreira, Carlos M. Fonseca, José A. Covas) (20min talk + 10 min questions and discussion)
10.45-11.00	Coffee break
11.00-11.30	Contributed Talk – Simulation-Based Innovization using Data Mining and Visual Analytics for Production Systems Analysis (Amos HC Ng) (20 + 10min)
11.30-12.00	Contributed Talk – Problem Understanding with Data Mining of Pareto-Optimal Designs in Space Engineering (Akira Oyama) (20 + 10min)
12.15-13.15	Lunch
13.30	Group photo outdoors Excursion (Hike)
18.00-19.00	Dinner
19.30-20.30	Summaries of Working Group Discussions and Next Steps

Thursday 26th January (Theme: The Problem Solving Process)	
7.30-8.45	Breakfast
8.45-9.15	Contributed Talk – Learning Tradeoffs in Multiobjective Optimization: A Cone-based Approach (Margaret M Wiecek) (20 + 10min)
09.15-10.00	<i>Keynote Talk</i> – Hybrid Evolutionary Multi-Objective Optimization: Different Interaction Styles and an Approach (Jyrki Wallenius)
10.00-10.15	Questions and discussion
10.15-10.45	Contributed Talk – Offline Automatic Configuration in Multi-Objective Optimization (Manuel López-Ibáñez and Thomas Stützle) (20 + 10 min)
10.45-11.00	Coffee break
11.00-11.30	Contributed Talk – Problem solving process in engineering applications: multiobjective optimization and user preferences (Silvia Poles) (20 + 10min)
11.30-12.00	Contributed Talk – Multiobjective optimization in self-optimizing systems and applications (Katrin Witting) (20+10min)
12.15-13.15	Lunch
13.30-18.00	Working groups (includes coffee break)
18.00-19.00	Dinner
20.00	Wine and Cheese Event (Music Room)

Friday 27th January (Wrap-Up)	
7.30-8.45	Breakfast
8.45-10.15	Working Group Presentations
10.15-10.30	Coffee
10.30-12.00	Whole Group Discussion and Wrap-Up
12.15-13.15	Lunch and goodbye

Eckart will leave us in a more fundamental way too, as he explained to the floor during the Seminar. He is now working in a new academic role and direction in his career as a Professor of Pedagogy at PHBern — University of Teacher Education, and has now ceased activities in optimization research. We all wish him the best in this new venture and thank him for his great contributions to our field of study. It is no exaggeration to state that Eckart's research — and that of his collaborators — has shaped much of the landscape in evolutionary multiobjective optimization over the last dozen or more years. He has made very significant contributions to both theory and practice, which we're sure will prove of enduring worth, he has built bridges between communities, and he has nurtured a very large number of young researchers to success.

Ecki, we all wish you well in your new work and position, and hope that you will find the time to join us again at Dagstuhl in the future.

7.3 Welcome to Kathrin Klamroth and Günter Rudolph

Finally, joining the organizing team for next time, the current organizers wish to welcome our esteemed colleagues, Kathrin Klamroth and Günter Rudolph.

Participants

- Richard Allmendinger
University College London, GB
- Anne Auger
INRIA Saclay – Orsay, FR
- Jürgen Branke
University of Warwick, GB
- Heinrich Braun
DHBW – Karlsruhe, DE
- Dimo Brockhoff
INRIA Nord Europe – Lille, FR
- Susanne Buerklen
Robert Bosch GmbH –
Stuttgart, DE
- Carlos A. Coello Coello
CINVESTAV del IPN, MX
- Salvatore Corrente
Università di Catania, IT
- Kalyanmoy Deb
Indian Institute of Technology –
Kanpur, IN
- Matthias Ehrgott
University of Auckland, NZ
- Michael Emmerich
Leiden University, NL
- Jose Rui Figueira
Instituto Superior Tecnico –
Lisboa, PT
- Joerg Fliege
University of Southampton, GB
- Carlos M. Fonseca
University of Coimbra, PT
- Simon French
University of Warwick, GB
- Tobias Friedrich
MPI für Informatik –
Saarbrücken, DE
- Xavier Gandibleux
University de Nantes, FR
- Antonio Gaspar-Cunha
Univ. of Minho – Guimarães, PT
- Martin Josef Geiger
Helmut-Schmidt-Universität –
Hamburg, DE
- Salvatore Greco
Università di Catania, IT
- Jussi Hakanen
University of Jyväskylä, FI
- Markus E. Hartikainen
University of Jyväskylä, FI
- Hisao Ishibuchi
Osaka Prefecture University, JP
- Milosz Kadzinski
Poznan Univ. of Technology, PL
- Kathrin Klamroth
Bergische Univ. Wuppertal, DE
- Joshua D. Knowles
University of Manchester, GB
- Pekka Korhonen
Aalto University, FI
- Karlheinz Kuefer
Fraunhofer ITWM –
Kaiserslautern, DE
- Manuel López-Ibáñez
Université Libre de Bruxelles, BE
- Mariano Luque
University of Malaga, ES
- Kaisa Miettinen
KTH – University of Jyväskylä,
FI and KTH – Stockholm, SE
- Sanaz Mostaghim
KIT – Karlsruhe Institute of
Technology, DE
- Vincent Mousseau
Ecole Centrale – Paris, FR
- Boris Naujoks
FH Köln, DE
- Amos H. C. Ng
University of Skövde, SW
- Akira Oyama
JAXA – Sagami-hara, JP
- Dmitry Podkopaev
University of Jyväskylä, FI
- Silvia Poles
EnginSoft S.p.A. – Padova, IT
- Enrico Rigoni
ESTECO SRL – Trieste, IT
- Günter Rudolph
TU Dortmund, DE
- Pradyumn Kumar Shukla
KIT – Karlsruhe Institute of
Technology, DE
- Johannes Siebert
Universität Bayreuth, DE
- Karthik Sindhya
University of Jyväskylä, FI
- Roman Słowiński
Poznan Univ. of Technology, PL
- Theodor J. Stewart
University of Cape Town, ZA
- El-Ghazali Talbi
INRIA Nord Europe – Lille, FR
- Tamara Ulrich
ETH Zürich, CH
- Jyrki Wallenius
Aalto University, FI
- Margaret M. Wiecek
Clemson University, US
- Katrin Witting
Universität Paderborn, DE
- Eckart Zitzler
PH Bern, CH



Analysis of Executables: Benefits and Challenges

Edited by

Andy M. King¹, Alan Mycroft², Thomas W. Reps³, and
Axel Simon⁴

- 1 University of Kent, GB, A.M.King@kent.ac.uk
- 2 University of Cambridge, GB, am@c1.cam.ac.uk
- 3 University of Wisconsin – Madison, US, reps@cs.wisc.edu
- 4 TU München, DE, Axel.Simon@in.tum.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12051 “Analysis of Executables: Benefits and Challenges”. The seminar had two focus groups: security engineers who need to find bugs in existing software systems and people in academia who try to build automated tools to prove correctness. The meeting of these diverse groups was beneficial and productive for all involved.

Seminar 29. January – 03. February, 2012 – www.dagstuhl.de/12051

1998 ACM Subject Classification B.2.2 Worst-case analysis, D.2.4 Formal methods, D.3.2 Macro and assembly languages, D.3.4 Debuggers and Interpreters, D.4.5 Fault-tolerance and Verification, D.4.6 Information flow controls and Invasive software, D.4.8 Modelling and prediction, D.4.9 Linkers and Loaders, F.3.2 Operational semantics and Program analysis, I.2.2 Program modification

Keywords and phrases Executable analysis, reverse engineering, malware detection, control flow reconstruction, emulators, binary instrumentation.


Digital Object Identifier 10.4230/DagRep.2.1.100

Edited in cooperation with Edward Barrett

1 Executive Summary

Axel Simon


Andy King

License  Creative Commons BY-NC-ND 3.0 Unported license
© Axel Simon and Andy King

The analysis of executables is concerned with extracting information from a binary program typically, though not exclusively, with program analysis techniques based on abstract interpretation. This topic has risen to prominence due to the need to audit code, developed by third parties for which the source is unavailable. Moreover, compilers are themselves a source of bugs, hence the need to scrutinise and systematically examine executables.

Seminar topics

The theme of the analysis of executables is an umbrella term adopted for this seminar, covers, among other things, the following topics:

 Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license
Analysis of Executables: Benefits and Challenges, *Dagstuhl Reports*, Vol. 2, Issue 1, pp. 100–116
Editors: Andy M. King, Alan Mycroft, Thomas W. Reps, and Axel Simon



DAGSTUHL REPORTS
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- specifying the semantics of native instructions, intermediate languages and the synthesis of transfer functions from blocks of instructions;
- abstract domains for binary analysis and how to combine them; type synthesis;
- control-flow graph (CFG) reconstruction, which is a prerequisite for many program analysis, and CFG matching, which is useful for detecting piracy;
- self-modifying code, characterising its semantics and detecting malware.

Chronological overview of the discussion

For practical reasons, all talks on Monday were held by the four organizers, including an overview of various known tools created by Thomas Reps and his group. His talk was followed by synthesis of transfer functions (the semantics of basic blocks) using SAT solving by Andy King, type reconstruction by Alan Mycroft and the combination of several abstract domains by Axel Simon. These rather varied topics gave a good introduction. Thomas Reps suggested that we identify common goals through a group discussion, which we could not complete on Monday due to the lack of time. Instead, we scheduled mostly industrial talks on Tuesday in order to find out about the problems that security engineers face in their everyday work and which tools they developed themselves. With this information, a group discussion on Tuesday afternoon quickly raised specific issues and their priorities: analyses must be scalable, preferably to some 12.5 billion instructions that large and vulnerable applications such as Adobe Reader are comprised of. This focus begs the question of whether we can afford a sound analysis or, as was suggested on the last day of the talk on CFG reconstruction, if an engineer can afford to work on a CFG in which not all indirect jumps are resolved precisely. In general, we should be aware of what assumptions we are making, for instance, about the correctness of CPU hardware, and possibly focus more on tools that are sound only under certain assumptions. This would still be an improvement since most security engineers nowadays even use unsound tools if they are helpful. A laudable long-term goal is the verification of a browser.

A more technical topic was the way we think about the control flow of a program, in the sense that associating a program counter address with a control flow graph node is inadequate in the presence of self-modifying code. Similarly, it is not clear what constitutes a function (due to for example, tail sharing) and how to reliably identify a function in the presence of obfuscated or optimized code that does not adhere to any standard ABI. It was pointed out that functions can have hundreds of entries with a large common body, implying that duplicating this body for each entry might create a considerable code size increase for an analysis.

To contrast the applied side of binary analysis with a theoretical view on static analysis, we scheduled the more theoretic talks on Wednesday morning. The speakers addressed how mutating malware could be classified (Roberto Giacobazzi) and how to treat memory allocated from a static array as independent heap cells (Xavier Rival). These topics gave an outlook on the challenges that lie beyond the already complicated reconstruction of the control flow graph.

Thursday and Friday featured talks mostly from the academic community who presented their current state-of-the-art. One particular debate arose on how the semantics of assembler instructions are best expressed. During an informal meeting on Thursday evening we agreed that the community would benefit from a common infrastructure to decode executable code. The way in which we should specify the semantics of native instructions was more difficult

to agree upon. Thus, we set up a mailing list to discuss a common decoder infrastructure that should be able to accommodate several platforms (say ARM and x86). The design of a decoder should feature a domain specific language that allows for a human readable specification of decoding instructions. This DSL should ideally be usable to also express the semantics of instructions, even if the various groups might want to implement their own semantic interpretation depending on their analysis needs.

Participation

In all, 42 researchers, both senior and more junior, from 10 countries attended the meeting. This high number shows the strong interest in this emerging field. The feedback from the participants was also very positive.

Directions for the future

Thus, one of the tangible outcomes is that the community set out to create a common piece of infrastructure. Beyond this, it was agreed that another seminar about the analysis of executables in two years time would be most welcome. We discussed what topics this new seminar should focus on and we distilled that malware, obfuscation, interpreters and self-modifying code should be major topics, as these constitute challenges that the community needs to address.

2 Table of Contents

Executive Summary

<i>Axel Simon and Andy King</i>	100
---	-----

Overview of Talks





A Tale of Two Tools: BEST & GIRA <i>Gogul Balakrishnan</i>	105
Refinement-based CFG Reconstruction from Unstructured Programs <i>Sebastien Bardin</i>	105
Model Checking PLC Programs <i>Sebastian Biallas</i>	105
On Backward Analysis in Binary Code using SAT/SMT <i>Jörg Brauer</i>	106
Evaluating Binary Code Diversification <i>Bjorn De Sutter</i>	106
Comparison, Navigation, Classification <i>Thomas Dullien</i>	107
Insight Framework: Yet Another Executable Binary Analysis Framework... <i>Emmanuel Fleury</i>	107
Fast Linear Two Variable Equalities <i>Andrea Flexeder</i>	107
Metamorphic Code Analysis by Abstract Interpretation <i>Roberto Giacobazzi</i>	108
Emulator Design, Traps and Pitfalls <i>Paul Irofti</i>	108
Jakstab & Alternating Control Flow Reconstruction <i>Johannes Kinder</i>	109
Transfer Function Synthesis at the Bit-level <i>Andy M. King</i>	109
Context Sensitive Analysis Without Calling Context <i>Arun Lakhotia</i>	109
In Situ Reuse of Functional Components of Binaries <i>Arun Lakhotia</i>	110
TSL: A System for Automatically Creating Analysers and its Applications <i>Junghee Lim</i>	110
Scalable Vulnerability Detection in Machine Code <i>Alexey Loginov</i>	111
Analysis of Binaries: An Industrial Perspective <i>Florian Martin</i>	111
PEASOUP: Preventing Exploits Against Software of Uncertain Provenance <i>David Melski</i>	111

Binary Code Analysis and Modification with Dyninst <i>Barton P. Miller</i>	112
Decompilation, Type Inference and Finding Code <i>Alan Mycroft</i>	112
A Formal ARM Model and Its Use <i>Magnus Myreen</i>	113
There’s Plenty of Room at the Bottom: Analyzing and Verifying Machine Code <i>Thomas W. Reps</i>	113
Race Condition Detection in Compiled Programs <i>Andrew Ruef</i>	114
Combining Several Analyses Into One or What Is a Good Intermediate Language for the Analysis of Executables? <i>Axel Simon</i>	114
Constraint-Based Static Analysis of Java Bytecode <i>Fausto Spoto</i>	114
A Method for Symbolic Computation of Abstract Operations <i>Aditya Thakur</i>	115
Adversarial Program Analysis and Malware Genomics <i>Andrew Walenstein</i>	115
Participants	116

3 Overview of Talks

3.1 A Tale of Two Tools: BEST & GIRA

Gogul Balakrishnan (NEC Laboratories America, Inc. – Princeton, US)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Gogul Balakrishnan

I will describe the BEST & GIRA tools developed at NEC Labs America.

BEST (Binary-instrumentation-based Error-directed Symbolic Testing) is a tool for finding problems in multi-threaded C/C++/Java programs. BEST uses binary-instrumentation to extract traces of execution runs, and uses SMT-based symbolic techniques to explore alternate schedules not visited during the given execution run. BEST can be used during testing to predict program failures, or during debugging to replay program failures.

GIRA (Generation of Intermediate Representation for Analysis) is a framework for analysing C++ programs. When describing GIRA, I will demonstrate that an executable compiled from C++ is very static-analysis unfriendly, and show how GIRA can alleviate the problem.

3.2 Refinement-based CFG Reconstruction from Unstructured Programs

Sebastien Bardin (CEA – Gif-sur-Yvette, FR)





License     Creative Commons BY-NC-ND 3.0 Unported license
© Sebastien Bardin

We address the issue of recovering a both safe and precise approximation of the Control Flow Graph (CFG) of a program given as an executable file. CFG reconstruction is a cornerstone of safe binary-level analysis: if the recovery is unsafe, subsequent analyses will be unsafe too; if it is too rough, they will be blurred by too many unfeasible branches and instructions. The problem is tackled with a refinement-based static analysis working over finite sets of constant values. The refinement mechanism allows to adjust the domain precision only where it is needed, resulting in precise CFG recovery at moderate cost.

First experiments, including an industrial case study from aeronautics, give promising results in terms of precision and efficiency.

3.3 Model Checking PLC Programs

Sebastian Biallas (RWTH Aachen, DE)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Sebastian Biallas

Programmable Logic Controllers (PLCs) are control devices used in the automation industry for operating robots, machines and plants. This talk presents the ArcadePLC (Aachen Rigorous Code Analysis and Debugging Environment for PLC) framework to verify PLC programs, written in various languages used in industry.

ArcadePLC provides a model checker and static analysis to prove safety properties and aid in program understanding. PLCs usually operate in the cycling scanning mode, which consists of three atomically and repeatedly executed phases: (1) reading input variables from sensors, (2) executing the program and (3) write-back of output variables which are connected to actuators. To verify such programs, the user can specify relations of inputs/outputs for the model checker in ACTL and ptLTL logic, which are evaluated at the end of each cycle (which corresponds to the observable behaviour).


To allow for model checking larger programs, we use abstract and symbolic simulation of the program. Our key idea is to exploit the cyclic operation mode of PLCs: In the first phase, will build successors by performing symbolic execution.

For ambiguous control flow, we use this symbolic information to infer weakest preconditions on the inputs. This allows for successively refining input values until the control flow is deterministic. Then, we discard the symbolic information and store only interval and bit-set information in state space. In the second phase, we use a CEGAR technique: Possible counterexamples are analysed and – if necessary – used to further refine the state space.

We used ArcadePLC to successfully verify different libraries of function blocks used in industry.

3.4 On Backward Analysis in Binary Code using SAT/SMT

Jörg Brauer (RWTH Aachen, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Jörg Brauer

Over the past decade, a variety of techniques have been invented that automatically compute optimal abstractions in the abstract interpretation framework. Impressive progress on decision procedures such as SAT and SMT solvers has made these techniques a practical proposition. However, it is important to note that automatic abstraction has thus far concentrated on forward abstraction.

Our presentation focuses on problems and techniques that operate in both, forward and backward direction. We identify domain-theoretic properties which explain the problems involved in backward analyses, and propose a framework based on the computational domain of Boolean formulae to circumvent these problems. Further, we report on a method that computes value-set approximations alternately in forward and backward directions. This technique allows us to reconstruct an accurate control flow graph from binary code using incremental SAT solving.

3.5 Evaluating Binary Code Diversification

Bjorn De Sutter (Ghent University, BE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Bjorn De Sutter

Software diversity has been proposed as a mechanism to support renewability in a range of software protection techniques, as well as a direct defence against collusion attacks or against the automation of attack scripts. This paper evaluates the potential of software diversity to protect against collusion attacks on security patches, such as the attacks commonly referred

to as “exploit Wednesday” attacks. Those attacks build on patches released on “Microsoft patch Tuesday” and rely on the fact that security fixes are easy to identify in undiversified software. This paper evaluates the feasibility of adapting the (semi-)automated attacks described in literature to diversified software, for a range of diversifying transformations of different strengths. We found that all existing tools can easily be thwarted, thus making the automation of the existing attacks on diversified software infeasible.

3.6 Comparison, Navigation, Classification


Thomas Dullien (Google Switzerland – Zürich, CH)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Thomas Dullien

This talk discusses the algorithms and ideas used in BinDiff, BinNavi, VxClass which were tools distributed by zynamics prior to the acquisition by Google.

3.7 Insight Framework: Yet Another Executable Binary Analysis Framework...


Emmanuel Fleury (Université Bordeaux, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Emmanuel Fleury

The Insight framework is a executable binary analysis framework for UNIX platforms and aiming at validation, verification and reverse-engineering binaries. The framework comes with a proposal of a machine-code independent intermediate representation that allows manipulation (e.g. for deobfuscation).

3.8 Fast Linear Two Variable Equalities


Andrea Flexeder (TWT GmbH, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andrea Flexeder

We present a novel interprocedural analysis of linear two-variable equalities which has a worst-case complexity of $\mathcal{O}(nk^4)$, where k is the number of variables and n is the program size. The analysis can be applied for identifying local variables and thus for interprocedurally observing stack pointer modifications as well as for an analysis of array index expressions, when analysing low-level code.

3.9 Metamorphic Code Analysis by Abstract Interpretation


Roberto Giacobazzi (Università degli Studi di Verona, IT)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Roberto Giacobazzi

Metamorphic code includes self-modifying semantics-preserving transformations to exploit code diversification. The impact of metamorphism is growing in security and code protection technologies, both for preventing malicious host attacks, e.g., in software diversification for IP and integrity protection, and in malicious software attacks, e.g., in metamorphic malware self-modifying their own code in order to foil detection systems based on signature matching. In this paper we consider the problem of automatically extracting metamorphic signatures from metamorphic code. We introduce a semantics for self-modifying code, later called phase semantics, and prove its correctness by showing that it is an abstract interpretation of the standard trace semantics. Phase semantics precisely model the metamorphic code behaviour by providing a set of traces of programs which correspond to the possible evolutions of the metamorphic code during execution. We show that metamorphic signatures can be automatically extracted by abstract interpretation of the phase semantics. In particular, we introduce the notion of regular metamorphism, where the invariants of the phase semantics can be modelled as finite state automata representing the code structure of all possible metamorphic changes of a metamorphic code, and we provide a static signature extraction algorithm for metamorphic code where metamorphic signatures are approximated in regular metamorphism.

3.10 Emulator Design, Traps and Pitfalls

Paul Irofti (FileMedic Ltd., PL)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Paul Irofti

During the last two years I've been researching the field of dynamic analysis in regards to emulating obfuscated and/or malevolent binaries. The result is an emulator that translates code blocks of binary samples from different platforms (operating systems and machine types) into an intermediate representation where information retrieval, data analysis and behaviour observations are made. After a code block is compiled and executed on the host platform and the entire environment is updated accordingly. Unless a verdict has been reached, a new cycle begins.

During the Dagstuhl Seminar I will present in-depth the design of this emulator and exchange ideas with people involved in similar activities.

3.11 Jakstab & Alternating Control Flow Reconstruction

Johannes Kinder (EPFL – Lausanne, CH)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Johannes Kinder

Unresolved indirect branch instructions are a major obstacle for statically reconstructing a control flow graph (CFG) from machine code. If static analysis cannot compute a precise set of possible targets for a branch, the necessary conservative over-approximation introduces a large amount of spurious edges, leading to even more imprecision and a degenerate CFG.

We propose to leverage under-approximation to handle this problem. We provide an abstract interpretation framework for control flow reconstruction that alternates between over- and under-approximation. Effectively, the framework imposes additional preconditions on the program on demand, allowing to avoid conservative over-approximation of indirect branches. We implemented the framework on top of our binary analysis tool Jakstab and present very promising results from using only constant propagation and a single concrete execution trace per target.

3.12 Transfer Function Synthesis at the Bit-level


Andy M. King (University of Kent, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andy M. King

In this talk we review how concrete semantics of blocks, represented as SAT or SMT instances, can be used to distil transfer functions that operate over systems of congruences and octagons. The reoccurring idea is to repeatedly solve an instance, collect different solutions, and then merge them to derive a summary for a block as a whole. We show how this technique can be applied to deobfuscate blocks to recover their meaning as well as derive transfer functions that can be composed so as to derive invariants from binary code.

3.13 Context Sensitive Analysis Without Calling Context

Arun Lakhotia (University of Louisiana – Lafayette, US)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Arun Lakhotia

Classic methods of interprocedural analysis are based on reachable paths defined over interprocedural control flow graph (ICFG). Adapting these methods to binaries require static identification of procedure 'call' and 'ret' instructions. There are many instances when a binary may not use such instructions to call (or return from) a procedure, such as, with tail-merge or body-merge operations performed by optimizing compilers or obfuscations used by malware.

We present a method to perform context-sensitive analysis using a 'stack graph' instead of 'call graph'. This method removes the need for identifying atomic instructions that modify the stack as well as transfer control. Instead our method requires only the ability to statically identify statements that modify the stack pointer.

3.14 In Situ Reuse of Functional Components of Binaries

Arun Lakhotia (University of Louisiana – Lafayette, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Arun Lakhotia


A complex binary is a composition of many behaviours. Access to these behaviours is provided through a user interface chosen by the programmers. There are times when one may need to access some part of the binary’s behaviour or access its behaviour in ways that were not imagined by the original designers. One way to achieve this is to replicate the specific behaviour of the binary in another, independent program and use it. Such ex situ methods can be challenging, since they require creating code that can be independently compiled.

We present a method to use the functionality of the binary in situ, that is, directly within the binary without physical extraction. The architecture consists of three parts: a LEFC (logical extraction of functional component) identifier, a LEFC compiler, and a LEFC execution monitor. A functional component is defined as an entry point, a collection of exit points, a list of parameters (registers, locations), pre-condition state of the program required for the FC to behave well, and types of the parameters. The extraction of this information may be done manually or automatically. The LEFC compiler compiles this descriptor into a library, that provides a standard function call interface to the FC. To reuse the FC, a programmer links with this library. When the function is invoked, the LEFC Monitors executes the original program and communicates with its process to executes the required code directly in the program’s address space.

We discuss a prototype implementation of this concept using OllyDbg. The LEFC compiler creates script for OllyDbg’s scripting plug-in. A user may use these scripts to access an FC.

3.15 TSL: A System for Automatically Creating Analysers and its Applications

Junghee Lim (University of Wisconsin – Madison, US)




License  Creative Commons BY-NC-ND 3.0 Unported license
© Junghee Lim

In this talk, I presented the design and implementation of system, called TSL that provides a systematic solution to the problem of creating retargetable tools for analysing machine code. TSL is a meta-tool; a tool generator that automatically creates different abstract interpreters for machine code instruction sets. TSL advances the state of the art in program analysis by providing a YACC-like mechanism for creating the key components of machine code analysers from a description of the concrete operational semantics of a given instruction set. TSL automatically creates implementations of different abstract interpreters for the instruction set.

I also briefly talked about various application tools developed via the TSL system.

3.16 Scalable Vulnerability Detection in Machine Code




Alexey Loginov (*GrammaTech Inc. – Ithaca, US*)

License    Creative Commons BY-NC-ND 3.0 Unported license
© Alexey Loginov

This talk describes the design and implementation of a scalable and precise tool for detecting vulnerabilities in machine code. The talk presents project goals, an overview of the tool architecture, the evaluation strategy for the tool, as well as how the evaluation strategy evolved as we gained experience during broader application of the tool. The talk will conclude with a discussion of a few challenges that may require the combined efforts of this community.

3.17 Analysis of Binaries: An Industrial Perspective

Florian Martin (*AbsInt – Saarbrücken, DE*)




License    Creative Commons BY-NC-ND 3.0 Unported license
© Florian Martin

In safety-critical systems a worst case execution time (WCET) analysis is vital, as it is the prerequisite for schedulability analysis. aiT is a sound WCET analyser, which is available for many different target processors. As the execution time is influenced greatly by the compiler and even can be influenced by the linker, the analyser works on fully linked binaries.

This talk will present the basic architecture of aiT. It will discuss some of the challenges and benefits which arise from analyzing executables, and the methods to cope with them.

3.18 PEASOUP: Preventing Exploits Against Software of Uncertain Provenance


David Melski (*GrammaTech Inc. – Ithaca, US*)

License    Creative Commons BY-NC-ND 3.0 Unported license
© David Melski

We present ongoing research on PEASOUP, a technology that enables the safe execution of software executables of uncertain provenance. PEASOUP (Preventing Exploits Against Software Of Uncertain Provenance) provides multi-level protection against the exploitation of multiple vulnerability classes. PEASOUP's operation is divided into an offline analysis phase and an online monitoring phase. The analysis phase builds an IR for the subject executable, produces multiple hardened, diversified variants of the subject executable, and tests the variants for resistance to attack and conformance with the original executable. The execution monitoring stage selects a variant of the subject, transforms the subject into the variant on demand during execution, and monitors the runtime execution for attempted exploits. This work is sponsored by the US Air Force Research Labs.

3.19 Binary Code Analysis and Modification with Dyninst

Barton P. Miller (University of Wisconsin – Madison, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Barton P. Miller

The Dyninst suite of toolkits provides a platform on which to build a wide variety of tools for operating on binary programs. Such tools include those for debugging, tracing, performance profiling, code optimization, testing, modelling, and cyber forensics.

Dyninst provides both control and data flow analyses of code, including live register analysis and slicing. The control flow analysis will identify functions, loops, basic blocks and instructions. As part of this analysis, Dyninst identifies (and can use for instrumentation) function entry and exit points; call sites; and loop entry, exit and body. Analysis occurs both at start time and during execution as new code is discovered (loaded dynamically, unpacked, or found based on tracking obfuscated control flow operations).

Instrumentation and modification of the code is based on patching the new operations into the code. Only the code that is being instrumented or modified is effected. Dyninst is a major customer of its own analyses, using them to generate efficient instrumentation code. Code modification as done in terms of editing the program's control flow graph and updating individual instructions in basic blocks. All instruction-level code changes are in terms of an abstract syntax tree representation, so are platform independent and portable.

For analysing and instrumenting malicious code, Dyninst has the ability to detect and deactivate defensive checks, and capture obfuscated control flow such as those based on return address manipulation, exceptions, run-time unpacking of code, and instruction overwriting. This defensive mode of Dyninst has been tested with code generated by most of the popular code packers and obfuscators.


Dyninst is actually a suite of toolkit libraries that can be used separately or in combination. These libraries support such functionality as code parsing, instruction cracking, symbol table reading and modifying (a surprisingly complex and tricky package), dataflow analysis and symbolic execution, code patching, dynamic code generation, process control, stack walking, and a C-like language interface to instrumentation code specification.

Dyninst will operate on executables and libraries, both statically and dynamically linked). While Dyninst operates happily on stripped binaries, it will also make best use available symbols (both static and dynamic) and debugging information. Supported platforms for Dyninst include x86 (32 and 64 bit) on Linux and Windows, Power (32 and 64 bit) on Linux and BlueGene.

Dyninst is also a platform for research into new techniques in program forensics (determining the provenance and authorship of the binary), vulnerability assessment of the code, and fault diagnosis.

3.20 Decompilation, Type Inference and Finding Code

Alan Mycroft (University of Cambridge, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Alan Mycroft

Decompilation is a mechanism for attempting to understand lower-level code by reconstructing source code of similar functionality. For type-unsafe languages such as C this is inherently




problematic since C’s ‘undefined behaviour’ allows return addresses etc. to be modified in a way which cannot be portably expressed as C source by a decompiler.

We highlight this decompiler choice between functionality and beauty and note that it occurs at all levels in the decompiler pipeline from executable to binary payload to assembler source to high-level code and is particularly an issue in malware.

The second topic notes that many techniques for compilation and decompilation are common, e.g. SSA removes aliasing performed by register allocation. In particular, for assembler code in SSA, we show how a variant of Hindley-Milner type reconstruction can construct C-level types, including recursive structs, *ab initio*.

3.21 A Formal ARM Model and Its Use

Magnus Myreen (University of Cambridge, GB)

License    Creative Commons BY-NC-ND 3.0 Unported license
© Magnus Myreen

Joint work of Fox, Anthony; Sewell, Thomas; Klein, Gerwin

I presented a formal model of the ARM ISA developed by Anthony Fox. This model has its roots in a project on hardware verification, has been extensively tested and covers all current versions of the ARM ISA: ARMv4-v7.




I also showed how I’ve used this model in proofs inside the HOL4 theorem prover.

My main tool is a proof-producing decompiler which takes machine code (e.g. ARM) and provides the user with a concise functional description of the machine code.

This tool has been used in an extension of the L4.verified project which proved functional correctness of the seL4 microkernel.

3.22 There’s Plenty of Room at the Bottom: Analyzing and Verifying Machine Code

Thomas W. Reps (University of Wisconsin – Madison, US)


License    Creative Commons BY-NC-ND 3.0 Unported license
© Thomas W. Reps

Computers do not execute source code programs; they execute machine code programs that are generated from source code. Consequently, some of the elements relevant to understanding a program’s capabilities and potential flaws may not be visible in its source code. The differences in outlook between source code and machine code can be due to layout choices made by the compiler or optimizer, or because transformations have been applied subsequent to compilation (e.g., to make the code run faster or to insert software protections).

The talk discussed the obstacles that stand in the way of using static, dynamic, and symbolic analysis to understand and verify properties of machine-code programs. Compared with analysis of source code, the challenge is to drop all assumptions about having certain kinds of information available (variables, control-flow graph, call-graph, etc.) and also to address new kinds of behaviours (arithmetic on addresses, jumps to “hidden” instructions starting at positions that are out of registration with the instruction boundaries of a given reading of an instruction stream, self-modifying code, etc.). In addition to describing the challenges, the talk will also describe what can be done about them.

3.23 Race Condition Detection in Compiled Programs


Andrew Ruef (University of Maryland – College Park, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andrew Ruef

Race conditions in multi-threaded programs are especially troublesome. They can manifest as deadlocks, faults, or semantic errors in program function. The nondeterminism inherent in multi-threaded programs presents challenges to testing and verifying them, especially once compiled. We present some approaches to use program rewriting to attempt to identify race conditions in compiled applications, without the assistance of any symbol information or user assistance. These systems are intended to increase the ability of quality assurance and allows developers to locate and reproduce concurrency errors in multi-threaded programs.

3.24 Combining Several Analyses Into One or What Is a Good Intermediate Language for the Analysis of Executables?


Axel Simon (TU München, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Axel Simon

The implementation of a static analysis is a complex undertaking when several domains should be combined to yield a more precise result. We contrast the Astree approach of implementing mostly a partially reduced cardinal product versus using only functor domains (which we interpret as reduced cardinal power domains). We illustrate how affine equations, congruences and intervals can be combined this way, thereby requiring less communication and, more importantly, a simpler communication infrastructure. The advantage of functor domains is that the API of a domain can change. However, for software engineering reasons it is sensible to settle for a few APIs between domains since then an analysis is flexible in re-arranging domains. We identify four APIs (and thus intermediate languages) that we use to address the analysis of executables including the treatment of wrapping of finite integer arithmetic.

3.25 Constraint-Based Static Analysis of Java Bytecode


Fausto Spoto (Università degli Studi di Verona, IT)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Fausto Spoto

I will present the constraint-based static analysis technique implemented in the Julia analyser for Java and Android. Examples will be taken from field initialization analysis and reachability analysis between program variables. I will conclude with future developments and open problems.

3.26 A Method for Symbolic Computation of Abstract Operations

Aditya Thakur (University of Wisconsin – Madison, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Aditya Thakur


In 1979, Cousot and Cousot gave a specification of the best (most-precise) abstract transformer possible for a given concrete transformer and a given abstract domain. Unfortunately, their specification does not lead to an algorithm for obtaining the best transformer. In fact, algorithms are known for only a few abstract domains.

Motivated by this problem, we developed a parametric framework that, for a given abstract domain A and logic L , computes increasingly better abstract values in A that over-approximate the set of states defined by an arbitrary formula in logic L . Because the method approaches the most-precise abstract value from “above”, if the computation takes too much time it can be stopped to yield a sound answer. For certain domains and logics, the framework is capable of computing the most-precise abstract value in the limit.

Our framework can be used to compute the best abstract transformer for a given abstract domain and concrete transformer represented by a formula in L . We describe instantiations of our framework for well-known abstract domains, such as intervals, polyhedra, and affine relations over bit-vectors.

3.27 Adversarial Program Analysis and Malware Genomics

Andrew Walenstein (University of Louisiana at Lafayette, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andrew Walenstein

Three challenges for binary analysis are presented. One challenge is that of robustness of analysis, and an experiment is reported that illustrates how fusing multiple tracer outputs can yield improved automated classification.

Another challenge is of fair evaluation of robustness, and an experiment is reported that illustrates how authentic (wild) malware are likely to be poor tests of the robustness of an analysis since the analysis is not being targeted.

The final challenge presented is that of malware relationship recovery. A model-driven evaluation of the different ways in which malicious files can be derived suggests complications for relationship recovery that may be surprising to some.

Participants

- Gogul Balakrishnan
NEC Lab. America, Inc. –
Princeton, US
- Sébastien Bardin
CEA – Gif-sur-Yvette, FR
- Edward Barrett
University of Kent, GB
- Sebastian Biallas
RWTH Aachen, DE
- Jörg Brauer
RWTH Aachen, DE
- Doina Bucur
INCAS3, NL
- Mihai Christodorescu
IBM TJ Watson Res. Center –
Hawthorne, US
- Bjorn De Sutter
Ghent University, BE
- Thomas Dullien
Google Switzerland – Zürich, CH
- Emmanuel Fleury
Université Bordeaux, FR
- Andrea Flexeder
TWT GmbH, DE
- Roberto Giacobazzi
Univ. degli Studi di Verona, IT
- Sean Heelan
Immunity Inc., US
- Paul Irofti
FileMedic Ltd., PL
- Johannes Kinder
EPFL – Lausanne, CH
- Andy M. King
University of Kent, GB
- Arun Lakhotia
Univ. of Louisiana – Gifette, US
- Jerome Leroux
Université Bordeaux, FR
- Junghee Lim
University of Wisconsin –
Madison, US
- Alexey Loginov
GammaTech Inc. – Ithaca, US
- Florian Martin
AbsInt – Saarbrücken, DE
- David Melski
GammaTech Inc. – Ithaca, US
- Bogdan Mihaila
TU München, DE
- Barton P. Miller
University of Wisconsin –
Madison, US
- Martin Murfitt
Trustwave Ltd., London, GB
- Alan Mycroft
University of Cambridge, GB
- Magnus Myreen
University of Cambridge, GB
- Michael Petter
TU München, DE
- Thomas W. Reps
University of Wisconsin –
Madison, US
- Xavier Rival
ENS – Paris, FR
- Edward Robbins
University of Kent, GB
- Daniel Roelker
DARPA – Arlington, US
- Andrew Ruef
University of Maryland – College
Park, US
- Alexander Sepp
TU München, DE
- Holger Siegel
TU München, DE
- Axel Simon
TU München, DE
- Fausto Spoto
Univ. degli Studi di Verona, IT
- Aditya Thakur
University of Wisconsin –
Madison, US
- Christopher Vick
Qualcomm Corp.R&D – Santa
Clara, US
- Aymeric Vincent
Université Bordeaux, FR
- Andrew Walenstein
University of Louisiana –
Lafayette, US
- Florian Zuleger
TU Wien, AT

