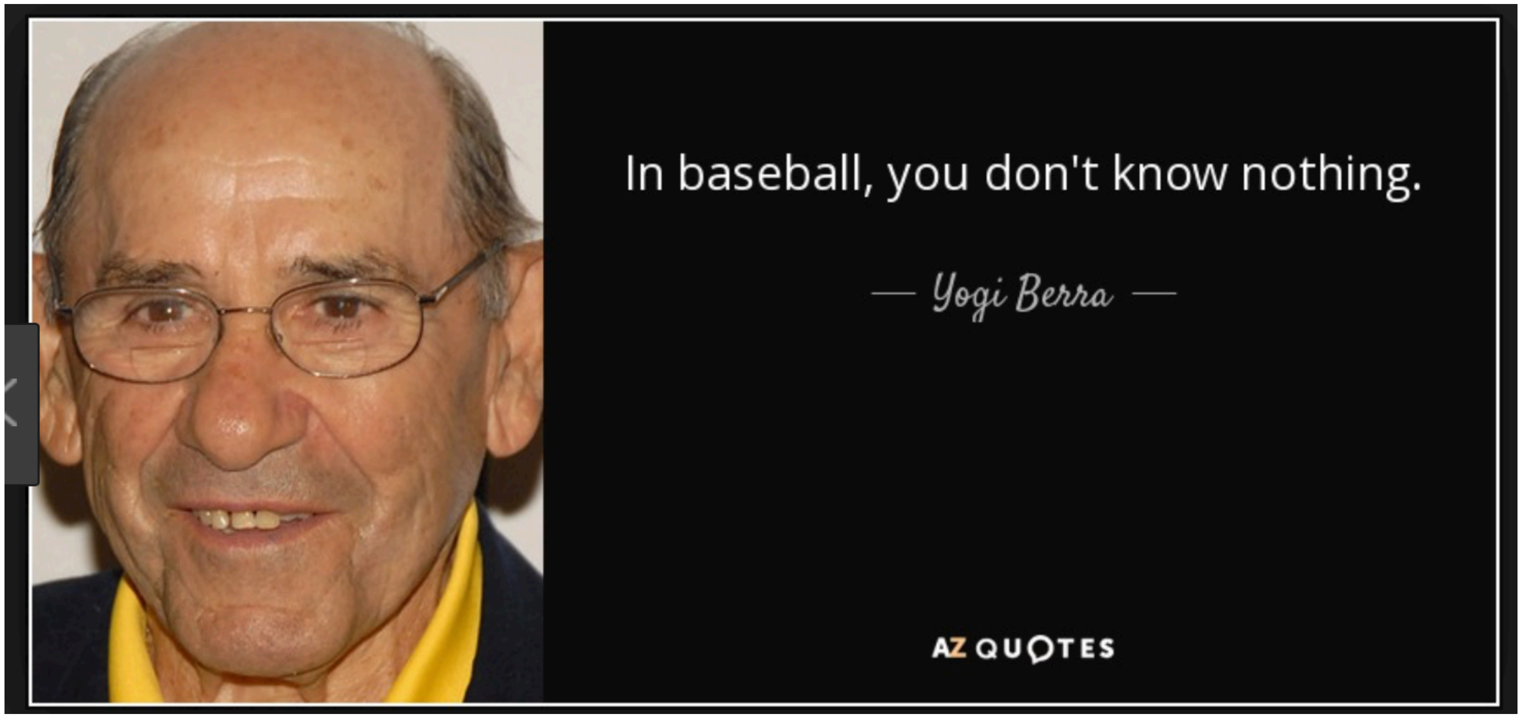
Reproducible AI

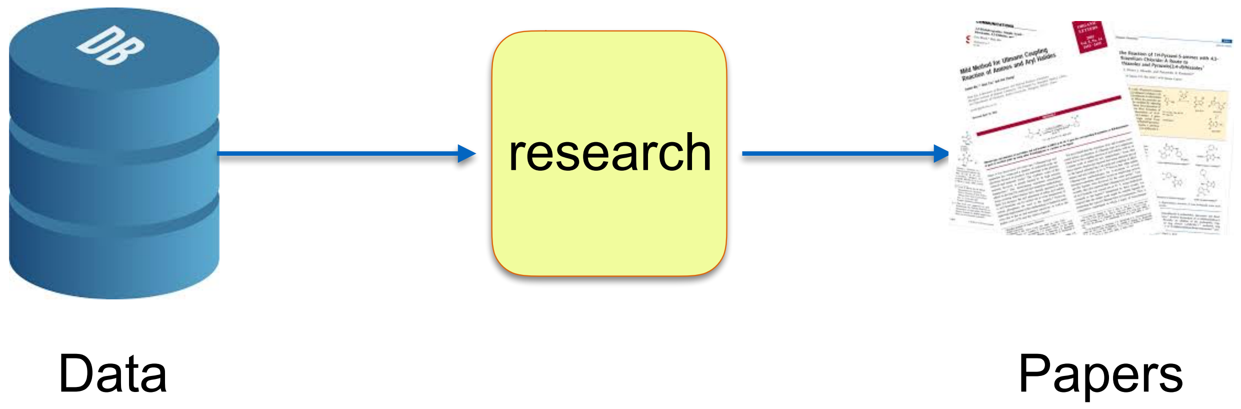
Pascal Van Hentenryck
(and the Ferst Street Band)

ISyE, ML Center, SCL Center
Georgia Institute of Technology

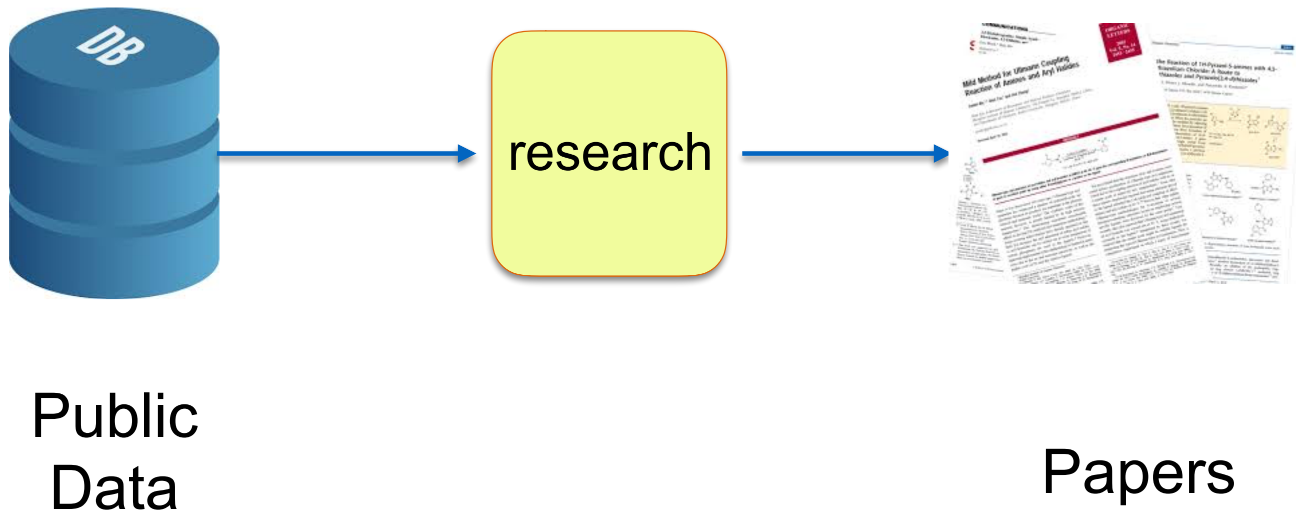
Disclaimer



Not Reproducible AI



Reproducible AI

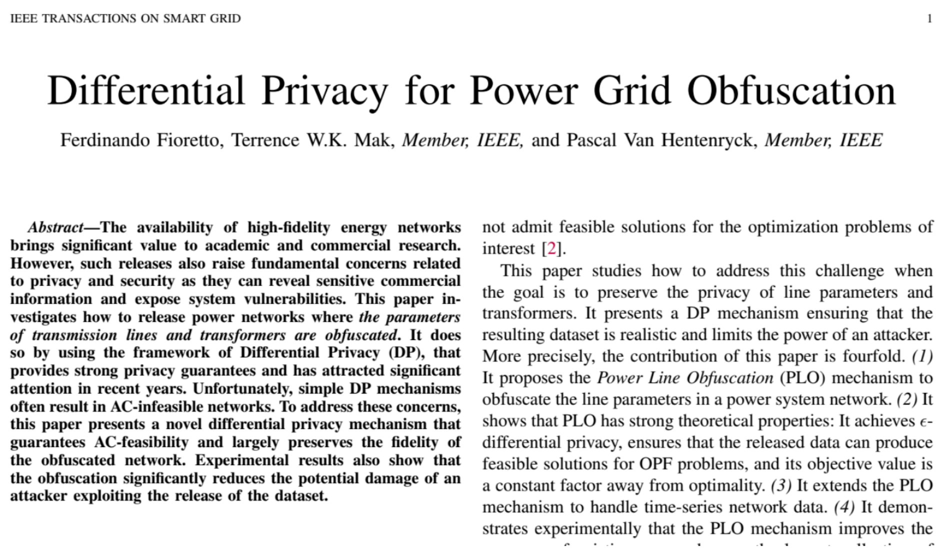


Data Sharing

- Public Data Sets
  - need as many as possible
  - need to make them evolved
- Planning competitions
- Mixed-Integer Programming Library (MIPLIB)
  - de factor standard set of test cases
  - contains medium, hard, and open instances
  - revised every couple of years
  - fundamental driver of innovation

Privacy should not be an excuse

- Differential privacy
  - release of privacy-preserving data sets



IEEE TRANSACTIONS ON SMART GRID 1

Differential Privacy for Power Grid Obfuscation

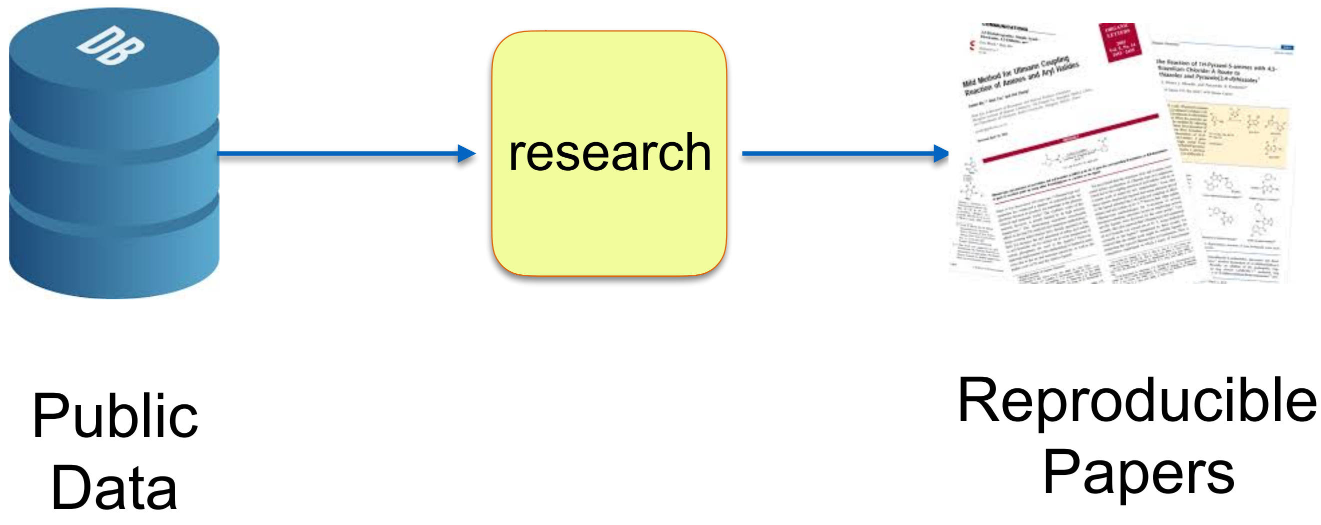
Ferdinando Fioretto, Terrence W.K. Mak, *Member, IEEE,* and Pascal Van Hentenryck, *Member, IEEE*

***Abstract*—The availability of high-fidelity energy networks brings significant value to academic and commercial research. However, such releases also raise fundamental concerns related to privacy and security as they can reveal sensitive commercial information and expose system vulnerabilities. This paper investigates how to release power networks where *the parameters of transmission lines and transformers are obfuscated*. It does so by using the framework of Differential Privacy (DP), that provides strong privacy guarantees and has attracted significant attention in recent years. Unfortunately, simple DP mechanisms often result in AC-infeasible networks. To address these concerns, this paper presents a novel differential privacy mechanism that guarantees AC-feasibility and largely preserves the fidelity of the obfuscated network. Experimental results also show that the obfuscation significantly reduces the potential damage of an attacker exploiting the release of the dataset.**

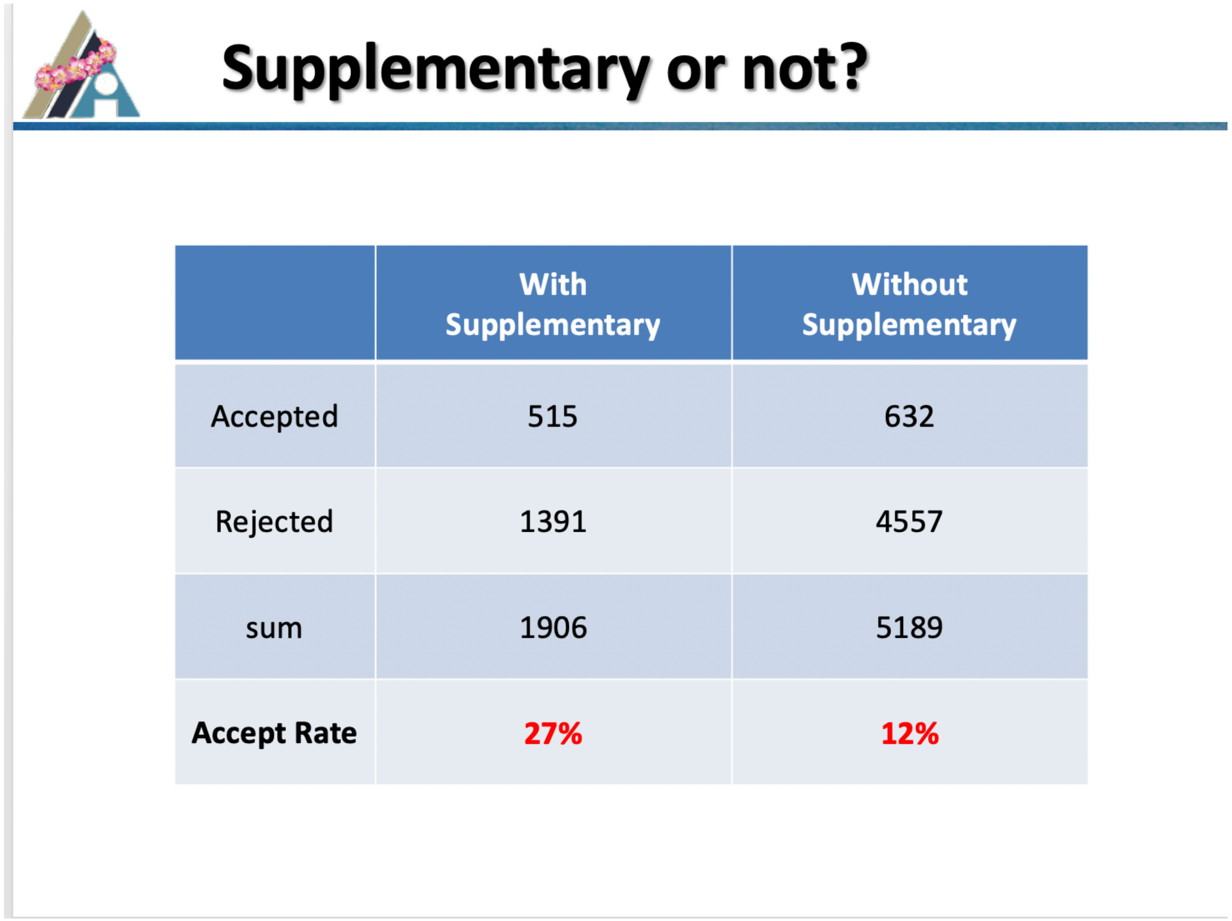
not admit feasible solutions for the optimization problems of interest [2].

This paper studies how to address this challenge when the goal is to preserve the privacy of line parameters and transformers. It presents a DP mechanism ensuring that the resulting dataset is realistic and limits the power of an attacker. More precisely, the contribution of this paper is fourfold. *(1)* It proposes the *Power Line Obfuscation* (PLO) mechanism to obfuscate the line parameters in a power system network. *(2)* It shows that PLO has strong theoretical properties: It achieves ϵ-differential privacy, ensures that the released data can produce feasible solutions for OPF problems, and its objective value is a constant factor away from optimality. *(3)* It extends the PLO mechanism to handle time-series network data. *(4)* It demonstrates experimentally that the PLO mechanism improves the

Reproducible AI



Supplementary Material at AAAI



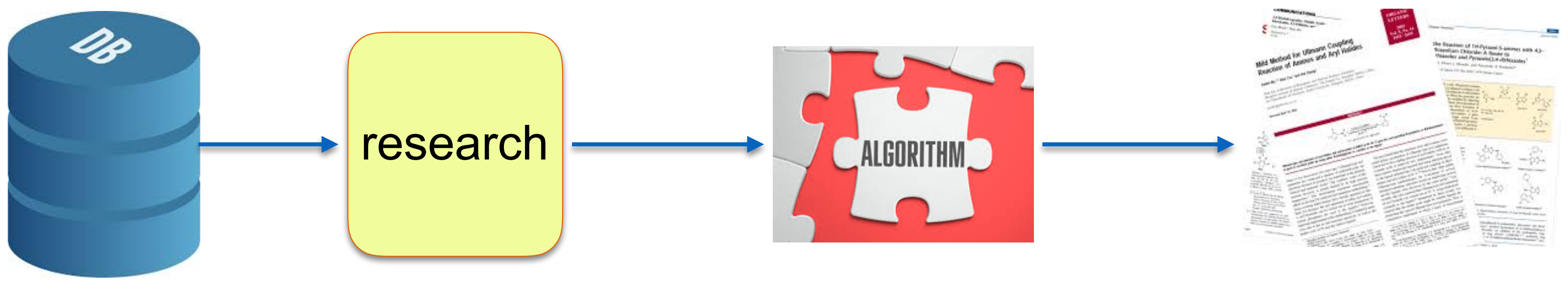
Supplementary or not?

	With Supplementary	Without Supplementary
Accepted	515	632
Rejected	1391	4557
sum	1906	5189
Accept Rate	**27%**	**12%**

Traditional Complaints

- Evaluated on old test cases
  - not clear how it behaves on new data sets
- Evaluated on small data sets
  - not clear how it scales
- Evaluated against a weak baseline
  - we will come back to this
- Evaluated against private datasets
  - reviewers dislike this
- Not clear why the algorithms work and what is important
  - reviewers dislike this as well

Reproducible Algorithms



public Data

reproducible algorithms

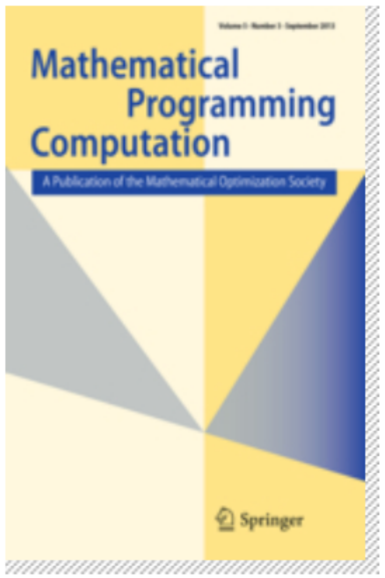
reproducible papers

Reproducible Algorithms

- Mathematical Programming C
  - flagship journal of the mathematical programming community
  - focus on computational optimization
- Publishing in MPC
  - requires the code to be released

Reproducible Algorithms

- Mathematical Programming C
  - flagship journal of the mathematical programming communit
  - focus on computational optimization


Mathematical Programming Computation
A Publication of the Mathematical Optimization Society
Springer

- Offers original research on computational issues in mathematical programming
- Article submissions are accompanied by software and data, subject to review and verification processes
- Coverage includes integer programming, linear programming, convex optimization, nonlinear programming, stochastic and robust optimization and much more

Reproducible Algorithms

- Mathematical Programming C
  - flagship journal of the mathematical programming community
  - focus on computational optimization
- Publication in MPC
  - requires the code to be released
  - requires the code to be working on the infrastructure of MPC

Reproducible Algorithms

- Mathematical Programming C
  - flagship journal of the mathematical programming community
  - focus on computational optimization
- Publication in MPC
  - requires the code to be released
  - requires the code to be working on the infrastructure of MPC
  - requires the code to be reproducing the results

Reproducible Algorithms

- Mathematical Programming C
  - flagship journal of the mathematical programming community
  - focus on computational optimization
- Publication in MPC
  - depend on having the code released
  - depend on the code working on the infrastructure of MPC
  - depend on the code reproducing the results
- The source code
  - does not have to be public

Reproducible Algorithms

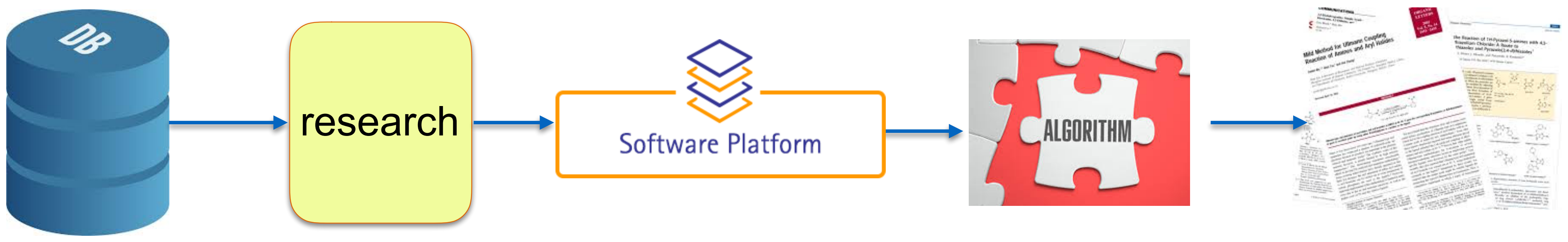
- Can you imagine if the Facebook AI team has the binaries of AlphaZero to help them?

Reproducible Algorithms

- The power of scientific platforms
  - ArXiv

- Think about what we could do with releasing code!

Platforms: Enabling AI



public Data

Enabling platforms

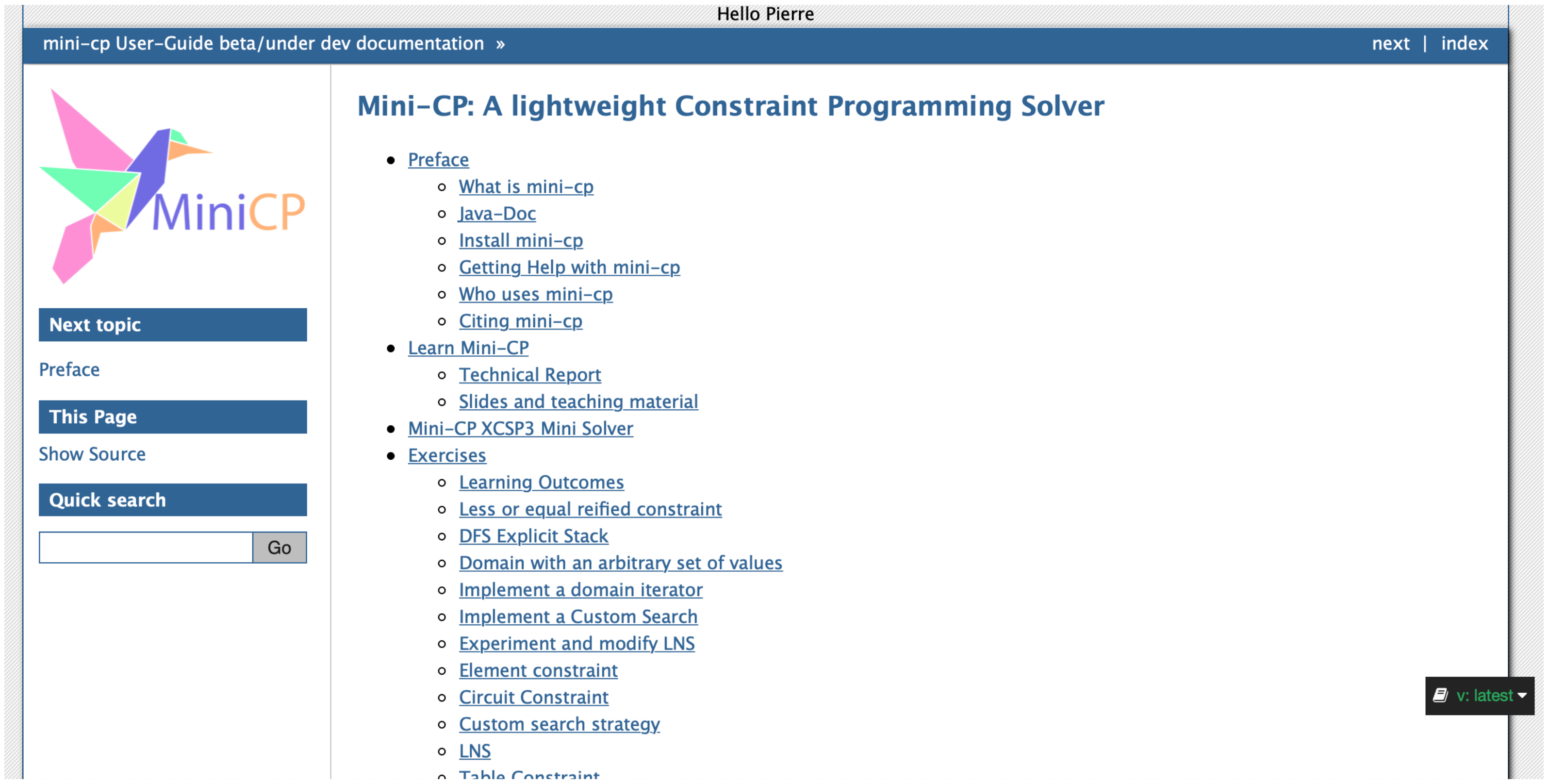
reproducible algorithms

reproducible papers

Software Platforms

- RoboCup Competition
  – code sharing induced substantially faster progress
- Reinforcement Learning
  – RLIB, Open AI gym, …
- Games
  – ELF

Software Platforms



Hello Pierre

mini-cp User-Guide beta/under dev documentation » next | index

Next topic

Preface

This Page

Show Source

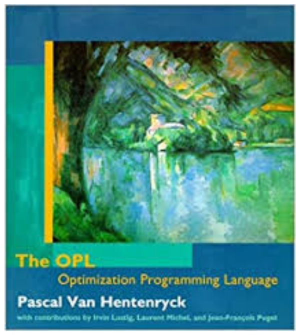
Quick search

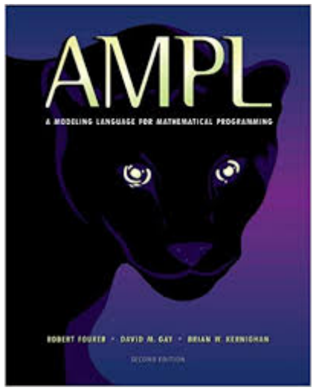
Go

Mini-CP: A lightweight Constraint Programming Solver

v: latest

Models versus Platforms versus Algorithms

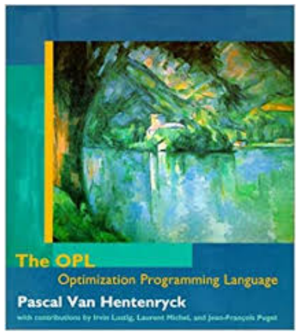




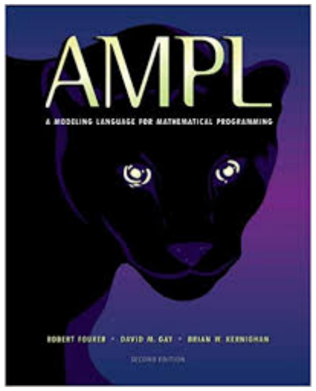
```
int n = ...;
range Men = 1..n;
range Women = 1..n;
int wrank[Men][Women] = ...;
int mrank[Women][Men] = ...;
dvar int wife[Men] in Women;
dvar int husband[Women] in Men;

constraints {
 forall(m in Men)
   husband[wife[m]] == m;
 forall(w in Women)
   wife[husband[w]] == w;
 forall(m in Men, w in Women)
   wrank[m,w]<wrank[m,wife[m]]=>mrank[w,husband[w]]<mrank[w,m];
 forall(w in Women,m in Men)
   mrank[w,m]<mrank[w,husband[w]]=>wrank[m,wife[w]]<wrank[m,w];
}
```

Models versus Platforms versus Algorithms


The OPL
Optimization Programming Language
Pascal Van Hentenryck
with contributions by Irvin Lustig, Laurent Michel, and Jean-François Puget

IBM
CPLEX


AMPL
A MODELING LANGUAGE FOR MATHEMATICAL PROGRAMMING
ROBERT FOURER · DAVID M. GAY · BRIAN W. KERNIGHAN
SECOND EDITION

SCIP • SOPLEX
GCG • UG
ZIMPL
THE SCIP OPTIMIZATION SUITE

Stochastic Behavior

- Replicable algorithmic behavior
  - Deterministic behavior
    - same random seeds
  - Deterministic parallel implementation
    - available in the main optimization solvers
    - available in the simulation world

How to Move Forward?

- Links reproducibility to publication?

How to Move Forward?

- Links reproducibility to publication?

- Focus on platforms

How to Move Forward?

- Links reproducibility to publication?

CONTROVERSIAL

- Focus on platforms

- Distinguish models and algorithms