My Research:

Algorithms for Making Good Decisions

Kevin Leyton-Brown

Computer Science Department

University of British Columbia

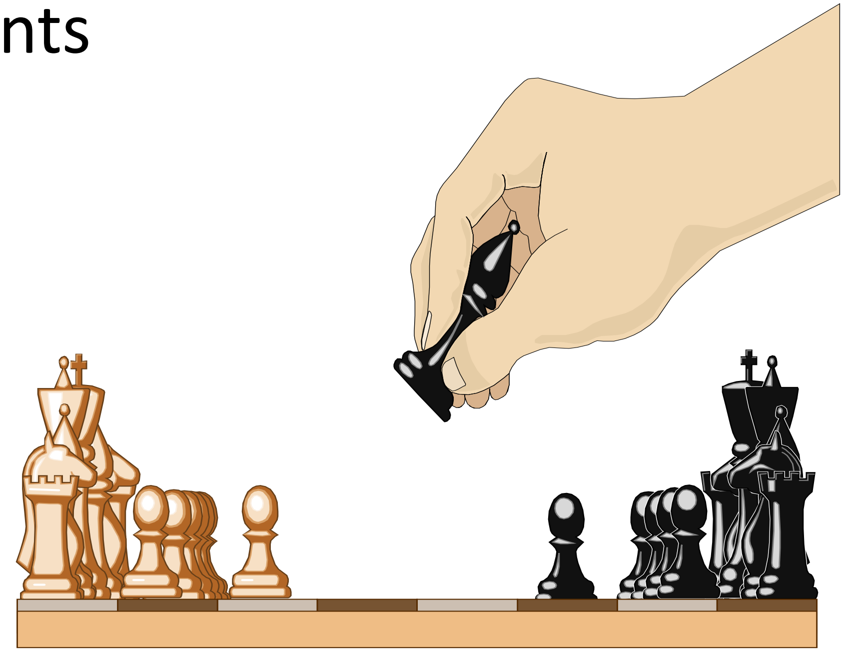
My Field

- Computer Science
  - Hub and spokes model: CS as an **enabling ingredient** for a wide variety of interdisciplinary projects
  - My own **interdisciplinary connections**: Microeconomic Theory, Math, Operations Research, Philosophy, Statistics, Cognitive Science
- **Artificial Intelligence**
  - Getting computers to do things that previously, only people could do
- Computer Science **Theory**
  - Mathematical underpinnings of computer science, particularly in the design and analysis of algorithms
- My own work: **Algorithms for making good decisions**
  - Game theory: decisions depend on what other actors will do
  - Empirical Algorithmics: algorithms that work well in practice

Reasoning about Large Games

Game Theory

- Mathematical study of interaction between **self-interested, rational** agents



- **Game**
  - players/agents
  - actions
  - payoffs
- Strategies:
  - **pure strategy:** picking a single action
  - **mixed strategy:** randomizing over actions

Game Theory

- Should you send your packets using **correctly**-implemented TCP (which has a “backoff” mechanism) or using a **defective** implementation (which doesn’t)?

	C	D
C	$-1, -1$	$-4, 0$
D	$0, -4$	$-3, -3$

- Consider this situati … e:
  - both use a correct in … s delay
  - one correct, one def … t, 0 ms for defective
  - both defective: both …

Analyzing Games

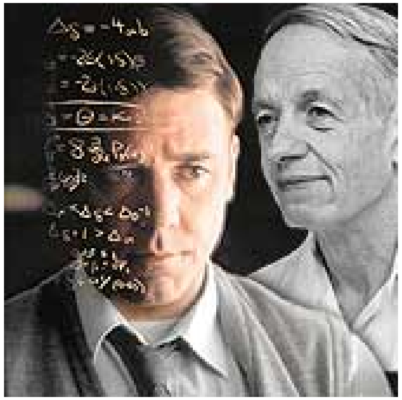
- TCP backoff game is a Prisoner's Dilemma
  - both players have a **dominant strategy**: defective
    - if player 2 plays C, D is player 1's best response
    - if player 2 plays D, D is player 1's best response
    - likewise for player 2

	C	D
C	$-1, -1$	$-4, 0$
D	$0, -4$	$-3, -3$

- Not all games are so simple to analyze
  - the best thing for one player to do can **depend** on what the other player does
    - rock-paper-scissors
    - poker

- What can we say about such games?

Game Theory



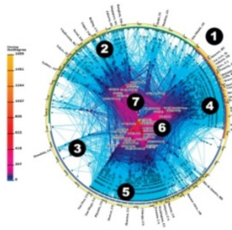
- Key insight:
  - don't just think about single players' actions
  - find strategy profiles where all players **simultaneously** play best responses
- Such a strategy profile is called a **Nash equilibrium**
  - at least one Nash equilibrium exists in every finite game
    - as long as agents are allowed to **randomize** their strategies
  - best known algorithms for finding Nash equilibrium require **exponential time**

The Kind of Games Often Studied

- The analysis of such 2 x 2 games has proven surprisingly interesting, and has had a profound impact both on our understanding of strategic situations and popular culture
  - e.g., google “dark knight game theory” or “strangelove game theory”

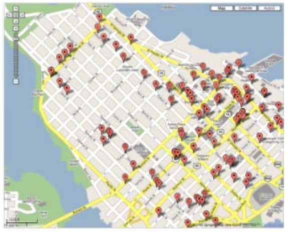
The Kind of Games We'd Like to Study

- When we use game theory to **model real systems**, we'd like to consider games with more than two agents and two actions



- **Some examples** of the kinds of questions we would like to be able to answer:
  - How will heterogeneous users route their traffic in a network?
  - How will advertisers bid in a sponsored search auction?
  - Which job skills will students choose to pursue?
  - Where in a city will businesses choose to locate?

- Most GT work is **analytic, not computational**
- What's holding us back?
  - the size of classical game representations **grows exponentially** in the number of players
    - this makes all but the simplest games infeasible to write down
  - even when games can be represented, the best algorithms tend to have **worst-case performance exponential** in the game's size



Compact Representations

Research program for advancing the computational analysis of games:

1. find representations that can encode games of interest in **exponentially-less space** than the normal form
2. find **efficient algorithms** for working with these representations

- **Action Graph Games:** compactly represent games exhibiting context-specific independence, anonymity or additive structure
- **Generalizes all major, existing compact representations** of simultaneous-move games
- **Fast algorithms** for computing quantities of interest
  - Nash equilibrium, correlated equilibrium, pure-strategy Nash equilibrium, others…

Coffee Shop Game

Google Local Canada

Web Images Groups News Local New! more »

category: Coffee Houses [Search]

e.g., "hotels in calgary" or "5000 dufferin street, toronto"

Search the map
Find businesses
Get directions

Local

Print Email Link to this page

Search results for **category: Coffee Houses** in this map

A. Connoisseurs' **Coffee**
1075 Georgia Street West, Vancouver, BC V6E 3C9
(604) 683-1486

B. Melriches Coffeehouse
1244 Davie Street, Vancouver, BC V6E 1N3
(604) 689-5282

C. Hole In The Wall Cappuccino Bar
1030 Georgia Street West, Vancouver, BC V6E 2Y3
(604) 646-4653

D. Starbucks **Coffee** Co
1055 W Georgia, Vancouver, BC V5K 1A1
(604) 685-5882

E. Five Roses Bakery Cafe
1220 Bute Street, Vancouver, BC V6E 1Z8
(604) 669-8989

F. Starbucks **Coffee** Co
1095 Howe Street, Vancouver, BC V6Z 1P6
(604) 685-7083

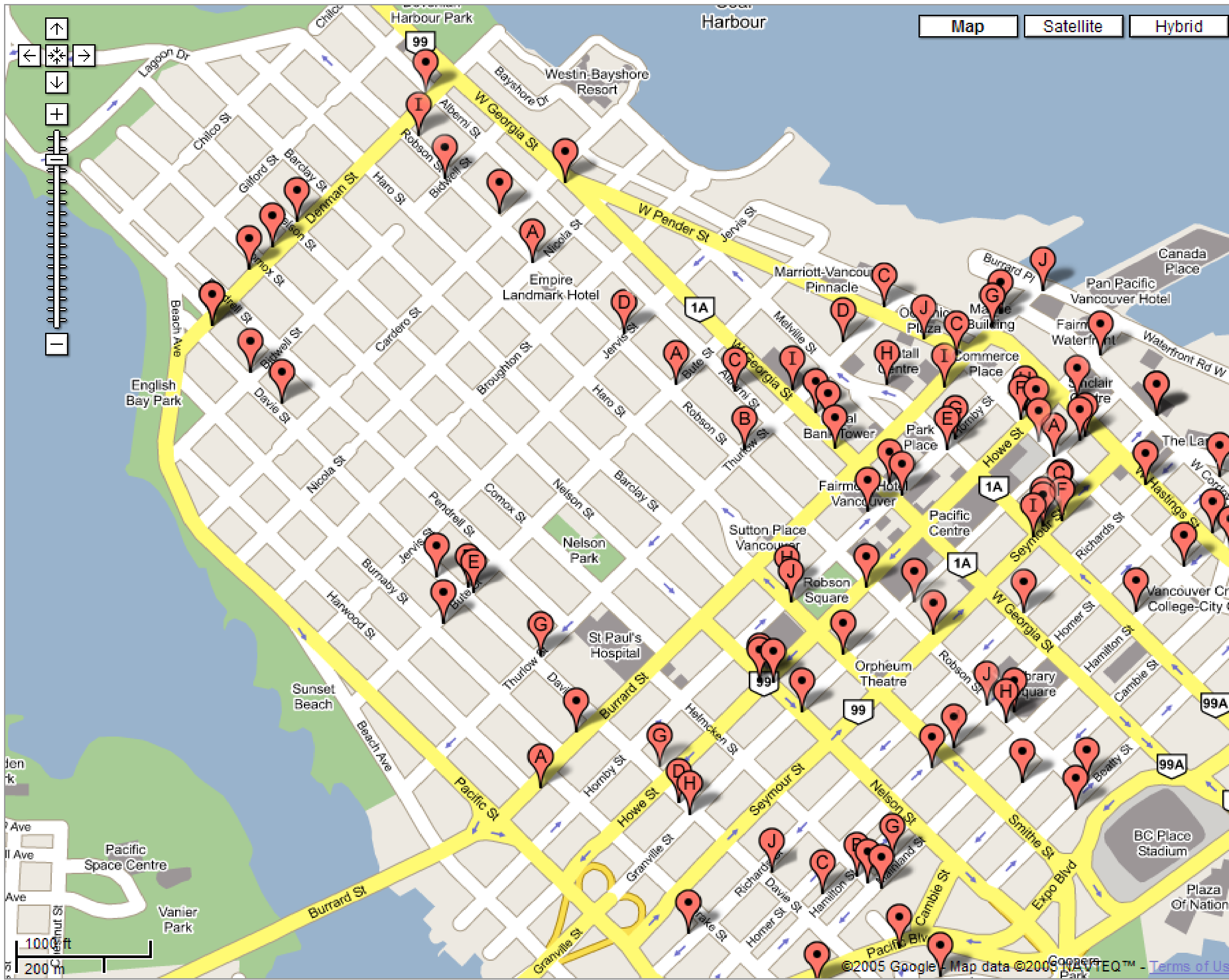
G. Uptown Espresso
808 Nelson Street, Vancouver, BC V6Z 2H2
(604) 689-1920

H. Caffe Artigiano
763 Hornby Street, Vancouver, BC V6Z 1S2
(604) 696-9222

I. Skyline Expresso
900 Howe Street, Vancouver, BC V6Z 2M4
(604) 683-4234

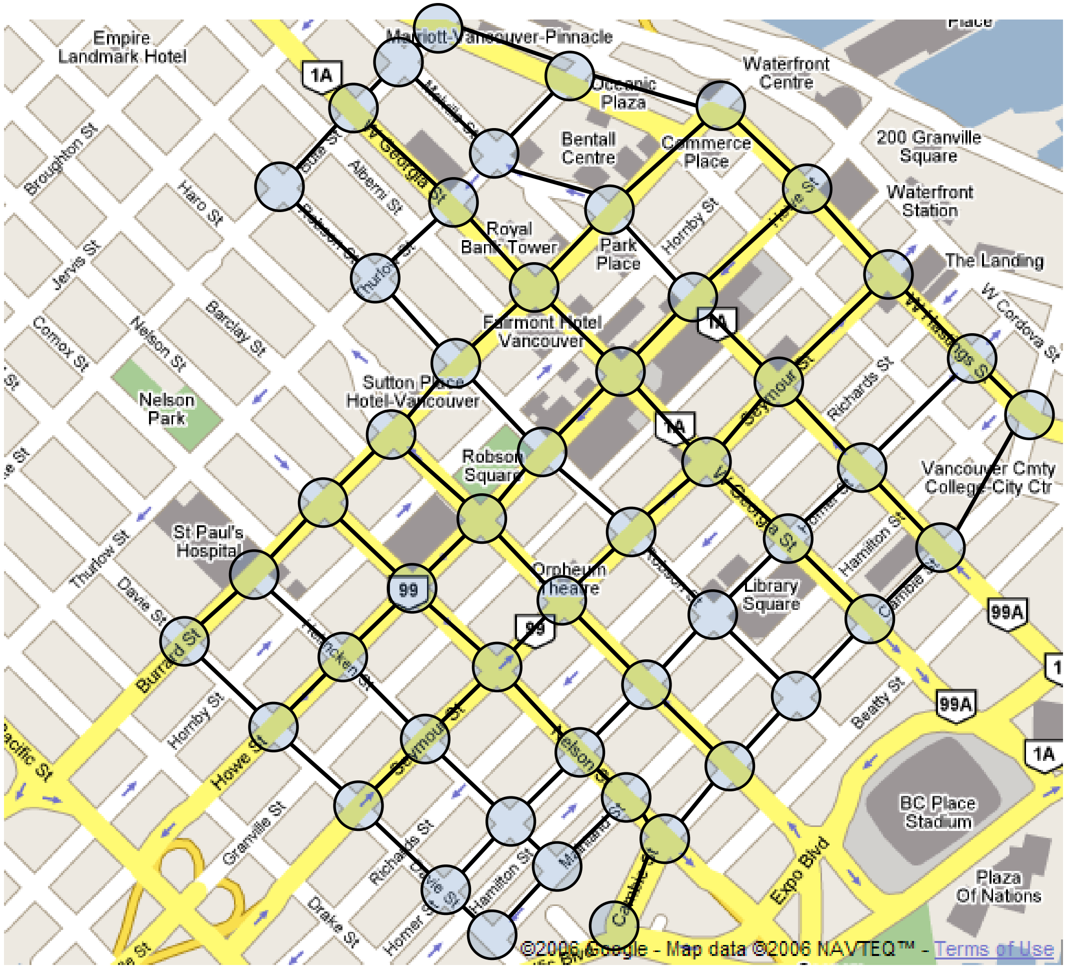
J. Farenheit Celsius **Coffee**
1225 Burrard Street, Vancouver, BC V6Z 1Z5
(604) 682-6675

Chicco Dall Oriente
1504 Robson Street, Vancouver, BC V6G 1C2

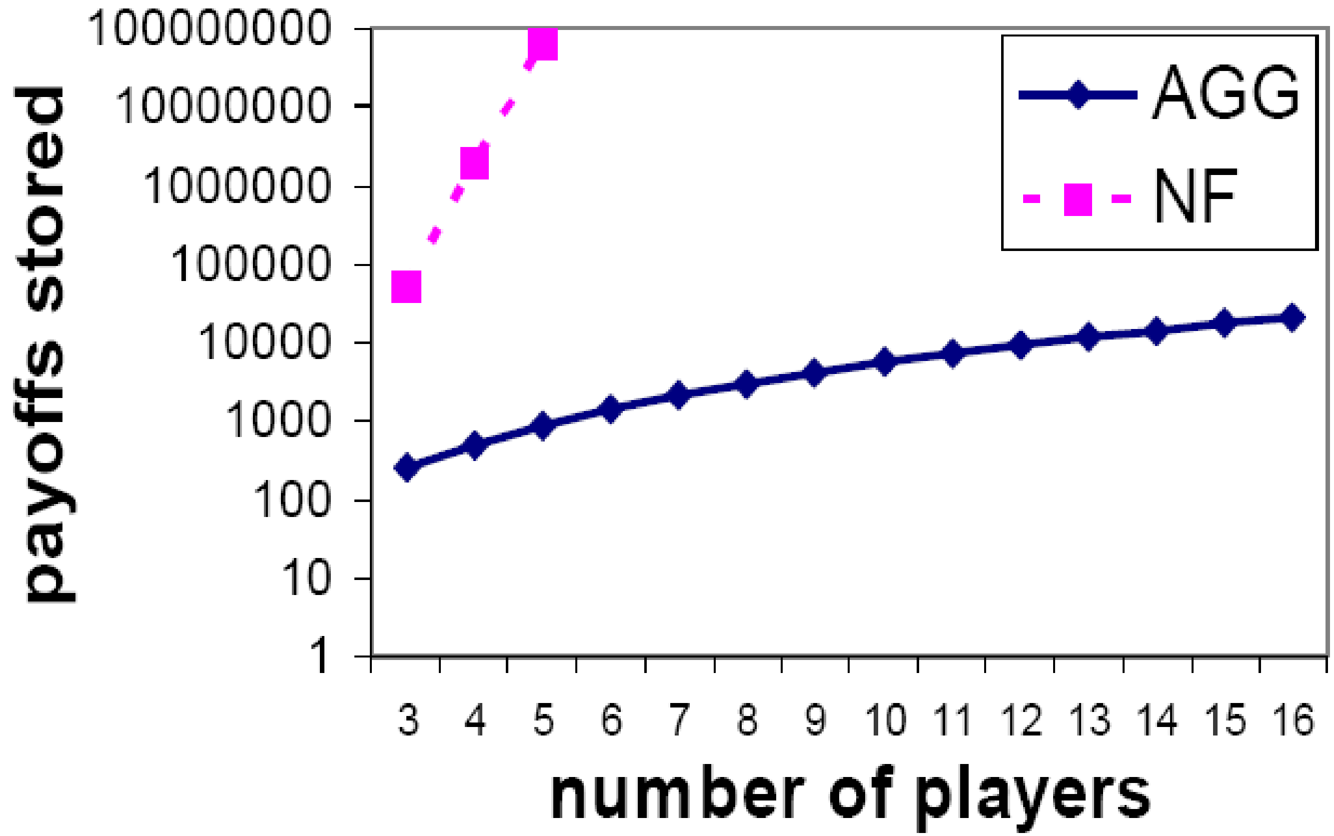


Action-Graph Games

- set of **players:** want to open coffee shops
- **actions:** locations where a shop could be opened
- **utility:** profitability of a location
  - depends only on number of other players who choose same or adjacent location



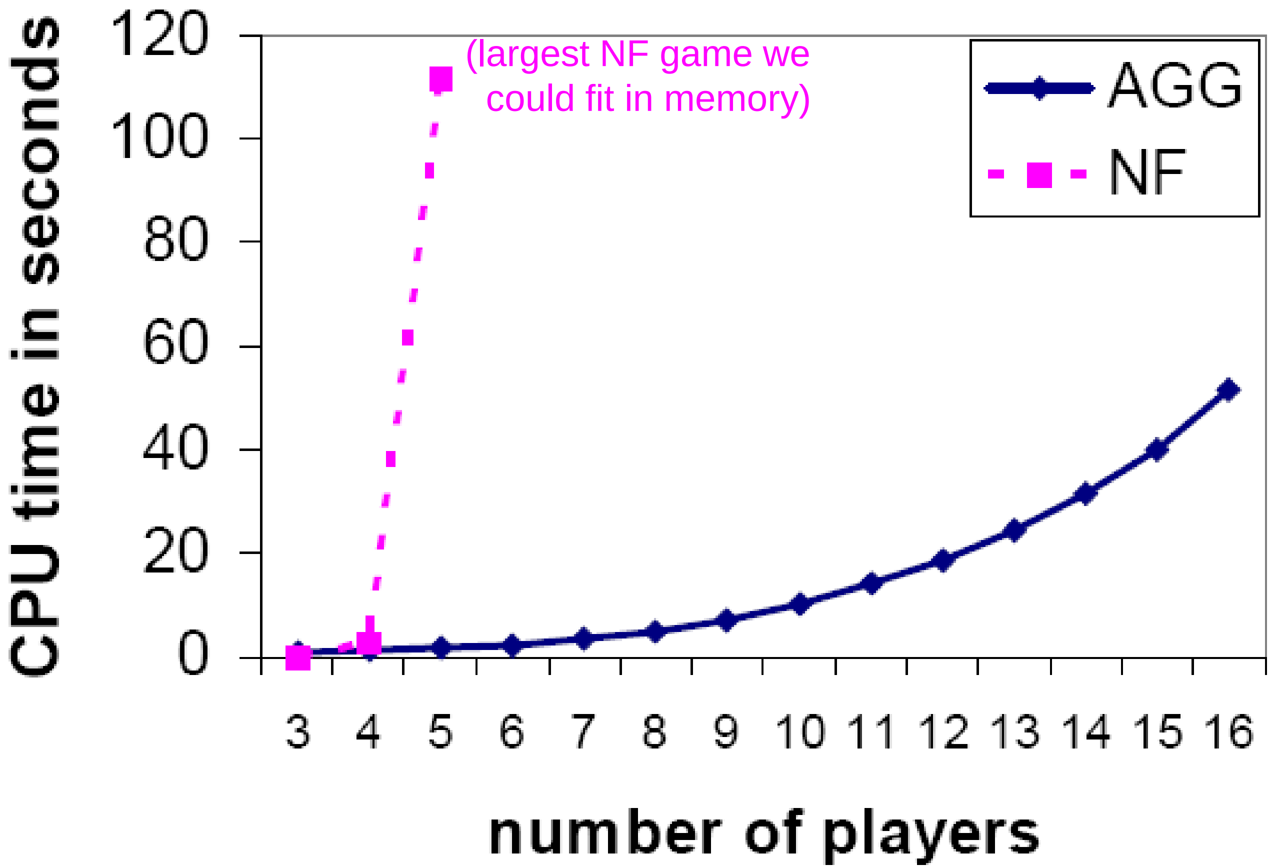
Experimental Results: Representation Size



Coffee shop game, 5 x 5 grid

NF grows exponentially; AGG grows polynomially

Experimental Results: Expected Payoff



Coffee Shop Game, 5 x 5 grid, 1000 random strategy profiles

NF grows exponentially; AGG grows polynomially

Auctions and Market Design

Auctions: why do computer scientists care?

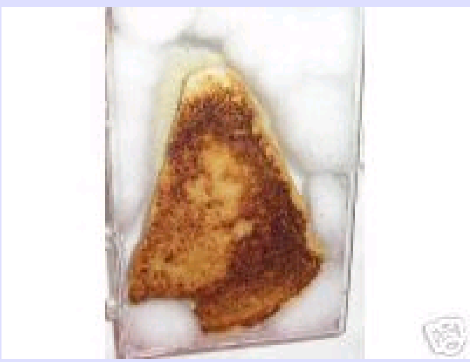
- **Efficient resource allocation**
  - a core interest of computer science
  - auctions solve this problem when agents are self interested
- They're **big** ($$$)
  - and the internet is changing the way they're used

Virgin Mary In Grilled Cheese NOT A HOAX ! LOOK & SEE ! Item number: 5535890757

Bidder or seller of this item? Sign in for your status | Email to a friend | **Watch this item** in My eBay

Note: This listing is restricted to pre-approved bidders or buyers only.

Email the seller to be placed on the pre-approved bidder/buyer list.



Larger Picture

Current bid:	**US $7,600.00**
	Place Bid >
Time left:	**3 days 23 hours** 7-day listing Ends Nov-22-04 17:22:07 PST
Start time:	Nov-15-04 17:22:07 PST
History:	4 bids (US $3,000.00 starting bid)
High bidder:	User ID kept private

Seller information

dltdesigns2002 (47)

Feedback Score: 47
Positive Feedback: 96.1%
Member since Jul-03-02 in United States

Read feedback comments
Add to Favorite Sellers
Ask seller a question
View seller's other items

Safe Buying Tips

Auctions: a key application of game theory

- A **broader category** than often perceived
- Generally, auctions are **markets** in which:
  - agents make binding declarations of interest in one or more resources
  - these resources are allocated according to known rules
  - payments to/from agents may be imposed

- Modeled using **game theory**. Some new wrinkles:
  - infinite action space
  - imperfect information about payoffs (other agents' valuations)

- How do sellers choose the **particular auctions** they do?
  - mechanism design (Nobel prize 2007): "inverse game theory"

Second-Price Auctions

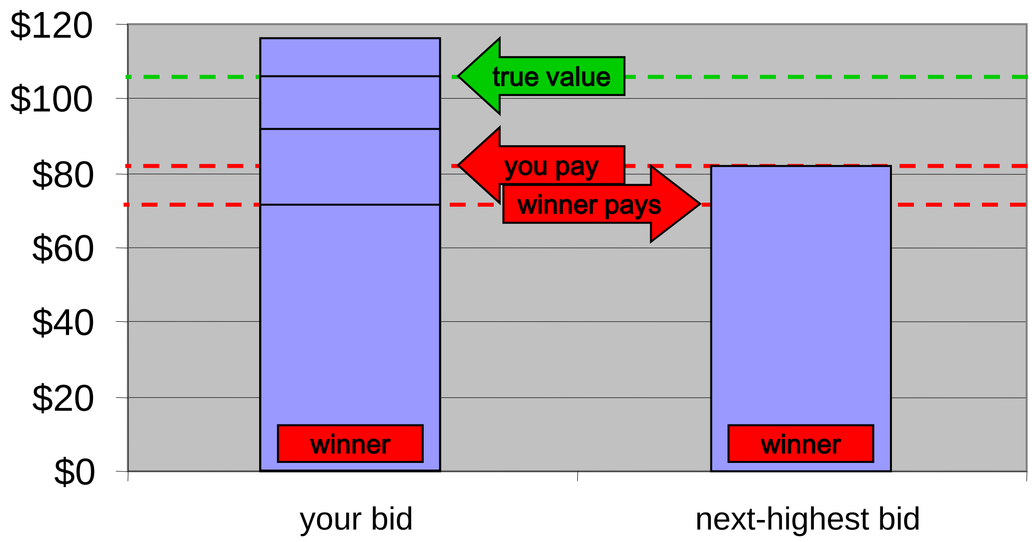
- An auction that might initially seem strange: **second-price**
  1. all bidders submit sealed bids
  2. the high bid wins
  3. the winner pays the **second-highest** bid amount

- **Theorem:** it is a dominant strategy in a second-price auction to bid your true value for the good.
- **Proof:**
  - **Case 1:** bidding truthfully **would** make you the high bidder
    - you can't gain by changing your bid
  - **Case 2:** bidding truthfully **would not** make you the high bidder
    - you can't gain by changing your bid

Second-Price Auctions

- **Theorem**: it is a dominant strategy in a second-price auction to bid your true value for the good.

- **Case 1:** bidding truthfully, you're the high bidder

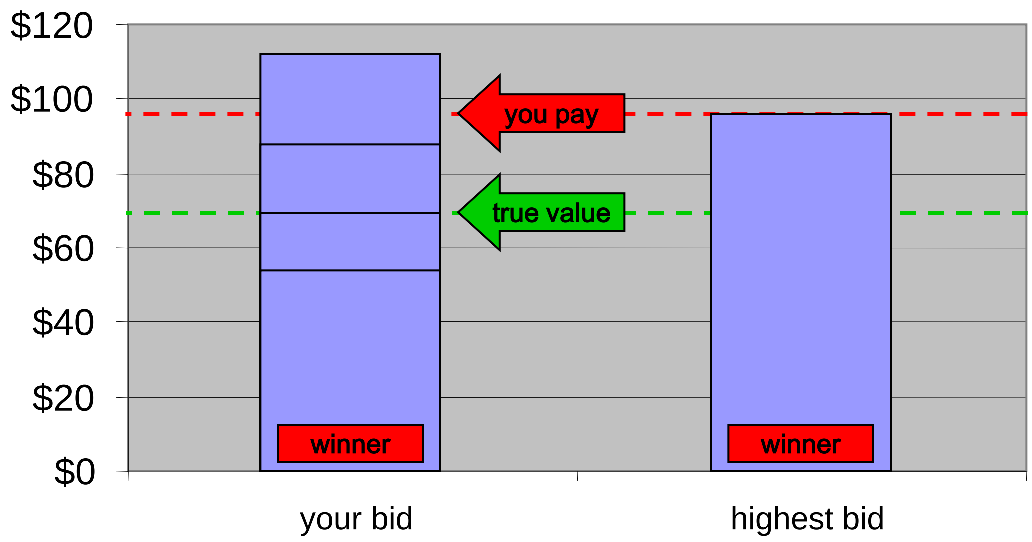


- bid more:
  - no difference (still win, pay same)
- bid less:
  1. no difference
  2. you lose

Second-Price Auctions

- **Theorem**: it is a dominant strategy in a second-price auction to bid your true value for the good.

- **Case 2:** bidding truthfully, you're **not** the high bidder



- bid **less**:
  - no difference (still lose, pay nothing)
- bid **more**:
  1. no difference
  2. you win, pay too much

eBay Help

Help Topics
A-Z Index
eBay Acronyms
eBay Glossary

Contact Us

Related Links

Learning Center
eBay University
Security Center
About Customer Support
Community Answer Center

Bidding on eBay

eBay uses an automatic bidding system to make bidding on auctions more convenient and less time-consuming for buyers. There is nothing you have to set up in order to bid in this way. When you bid on an auction style listing you will be placing bids using this method. Practice bidding on eBay from this test auction!

Here's how bidding on eBay works:

1. When you place a bid, you enter the maximum amount you'd be willing to pay for the item. Your maximum amount is kept confidential from other bidders and the seller.
2. The eBay system compares your bid to those of the other bidders.
3. The system places bids on your behalf, using only as much of your bid as is necessary to maintain your high bid position (or to meet the reserve price). The system will bid up to your maximum amount.
4. If another bidder has a higher maximum, you'll be outbid. BUT, if no other bidder has a higher maximum, you win the item. And you could pay significantly less than your maximum price! This means you don't have to keep coming back to re-bid every time another bid is placed.

Google AdWords

kevinlb@gmail.com | Help | Contact us | Sign Out
Customer ID:696-0296

- Ranking: descending by (quality score) x (bid amount)
  - quality score is click-through rate plus other measures of advertisement relevance

- “The AdWords Discounter will charge you the lowest CPC you can be charged while still maintaining your position”

Campaig... ...ed Search
Campaign S... Search
Tools > Tra...

Traffic E...
« Revise se...
All estimate... ...ositions may vary. To vie... ...opriate Ad Group. Lea...

Average C...
Estimated clicks per day: **309** - **445** (at a daily budget of $100.00)

Maximum CPC: .25 **Daily budget:** 100 Get New Estimates

Keywords ▼	Search Volume	Estimated Avg. CPC	Estimated Ad Positions	Estimated Clicks / Day	Estimated Cost / Day
Search Total		**$0.10 - $0.23**	**4 - 6**	**309 - 445**	**$30 - $100**
computer science		$0.09 - $0.20	1 - 3	8 - 11	$1 - $3
game theory		$0.08 - $0.12	1 - 3	2	$1
insurance		$0.11 - $0.25	4 - 6	156 - 235	$20 - $60
kevin		$0.08 - $0.16	1 - 3	84 - 108	$7 - $20
mortgage		$0.11 - $0.25	4 - 6	59 - 88	$7 - $30
university of british columbia		$0.08 - $0.15	1 - 3	1	$1

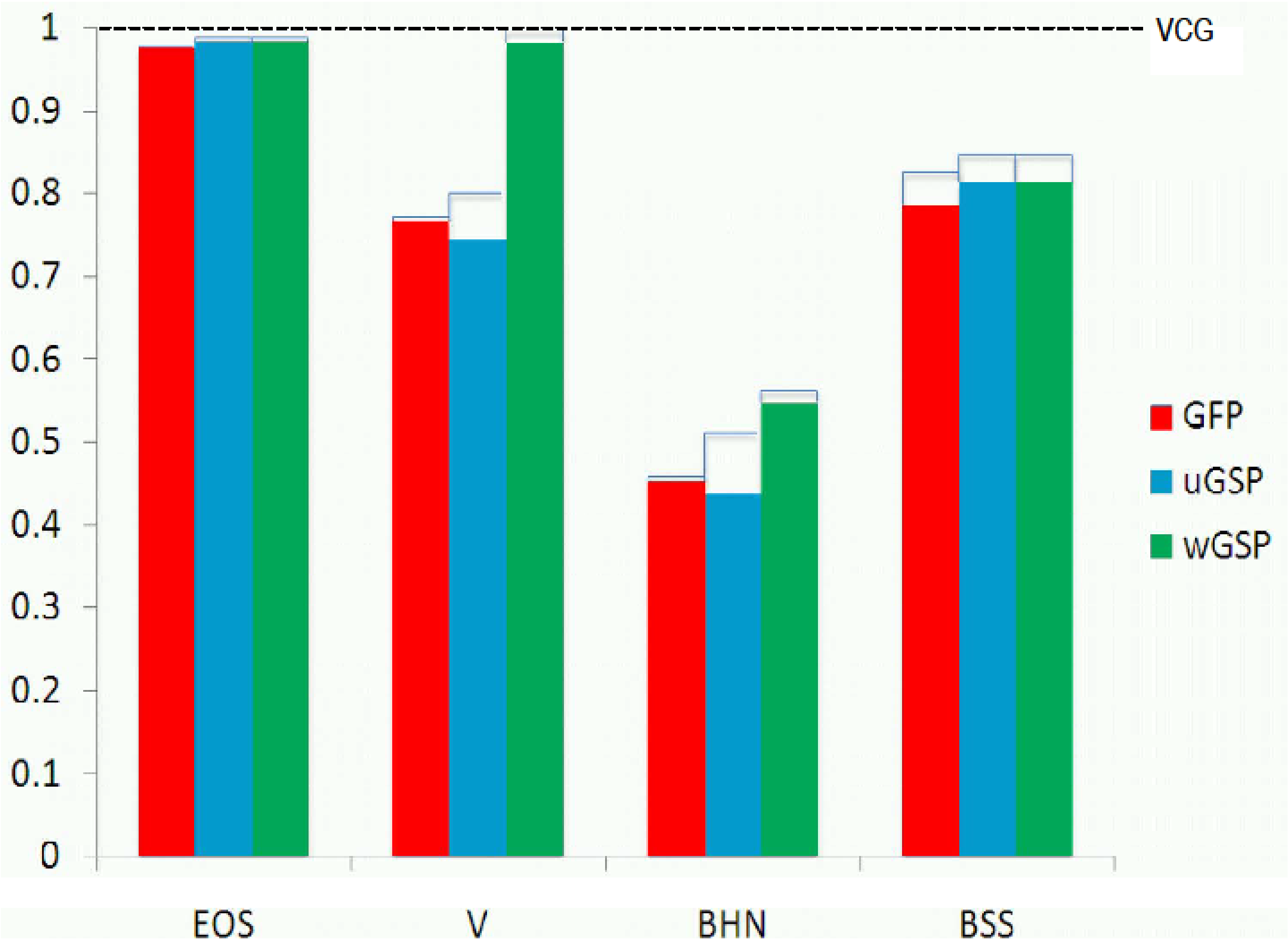
Estimates for these keywords are based on clickthrough rates for current advertisers. Some of the keywords above are subject to review by Google and may not trigger your ads until they are approved. Please note that your traffic estimates assume your keywords are approved.

Analyzing Ad Auctions

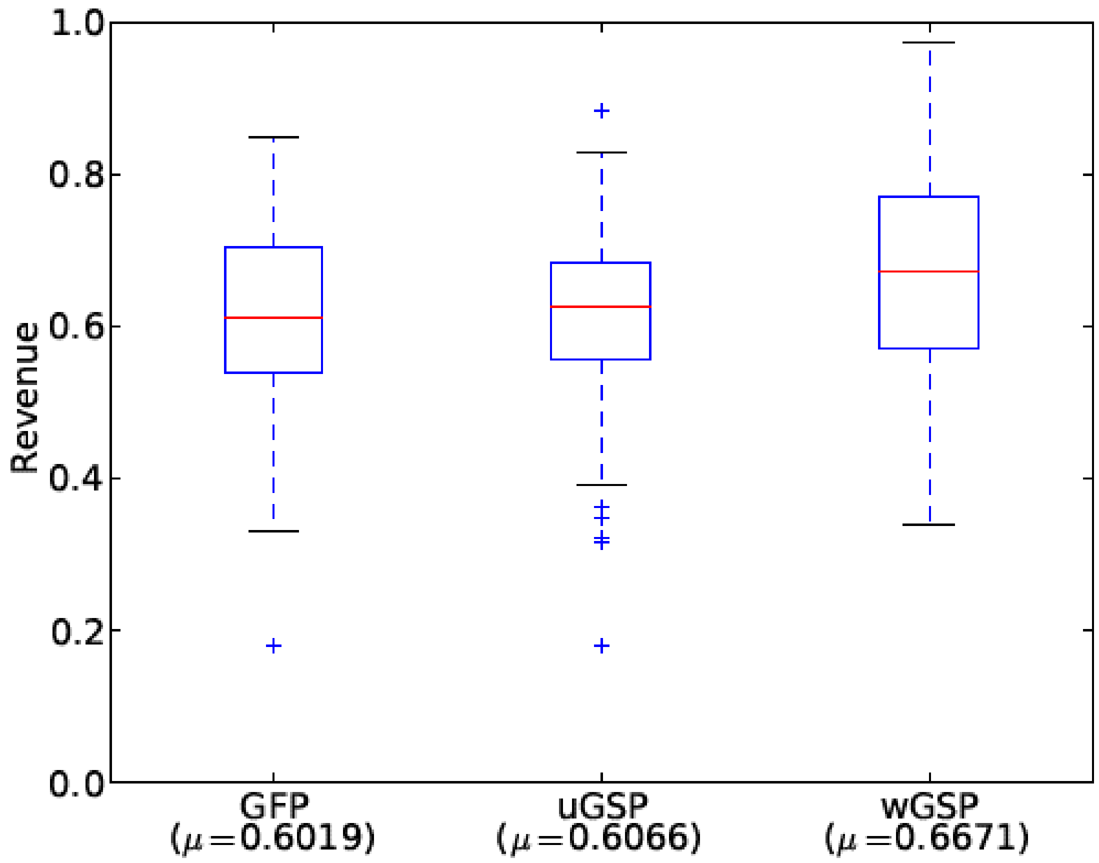
- Search engines used **different auctions** over the years
  - GFP: Yahoo! and Overture 1997-2002
  - uGSP: Yahoo! 2002-2007
  - wGSP: Google, Microsoft, Yahoo! 2007-present
- Question
  - Is **wGSP better** than GFP and uGSP?
- Better by **what metric**:
  - revenue?
  - efficiency?

- Answer this question by representing the ad auction **as an AGG**, and computing Nash equilibria

Analyzing Ad Auctions: Efficiency



Analyzing Ad Auctions: Revenue



Designing Algorithms to Work Well in Practice

Empirical Algorithmics

- Many **important problems** are computationally hard
  - circuit verification, planning, protein folding, probabilistic inference, vehicle routing, ...
- We need to be able to **solve hard problems** in practice, even if their worst-case complexity is exponential
  - Luckily, many instances are **easy in practice**
- Overall research agenda:
  - bypass the theoretical question of worst-case hardness
  - design algorithms that **do well on "typical" inputs**

One Motivating Question

"How hard is it to solve a given problem in practice, using the best available methods?"

The best available methods tend

- to offer **no interesting theoretical guarantees**
- work **astoundingly well** in practice
- often exhibit **exponentially varying performance** (e.g., milliseconds to days) even on fixed-size problems

Our Key Finding

Even in settings where **formal analysis seems hopeless**:

- algorithms are complex black boxes
- instance distributions are heterogeneous or richly structured

…it is possible to apply **rigorous statistical methods** to answer such questions with high levels of confidence.

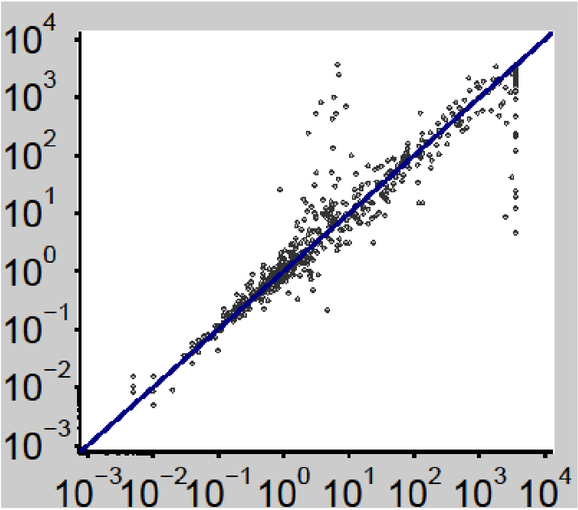
Empirical Hardness Models

- Predict how long an algorithm will take to run, given:
  - A **set of instances** D
  - For each instance $i \in D$, a vector $\boldsymbol{x}_i$ of **feature values**
  - For each instance $i \in D$, a **runtime observation** y_i

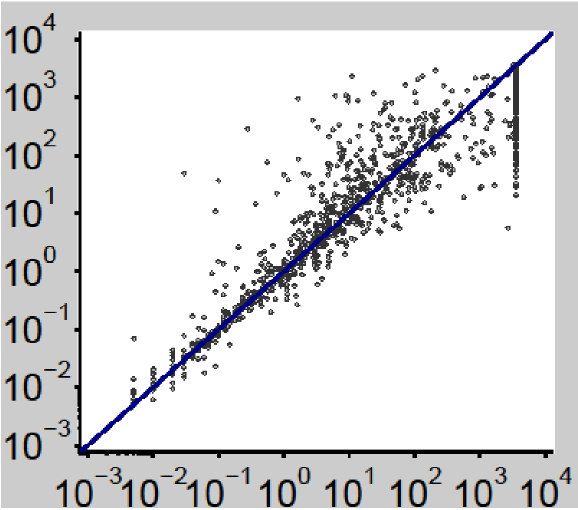
- We want a mapping $f(x) \mapsto y$ that **accurately predicts** y_i given $\boldsymbol{x}_i$
  - This is a **regression** problem

- The amazing thing: **this works at all**!

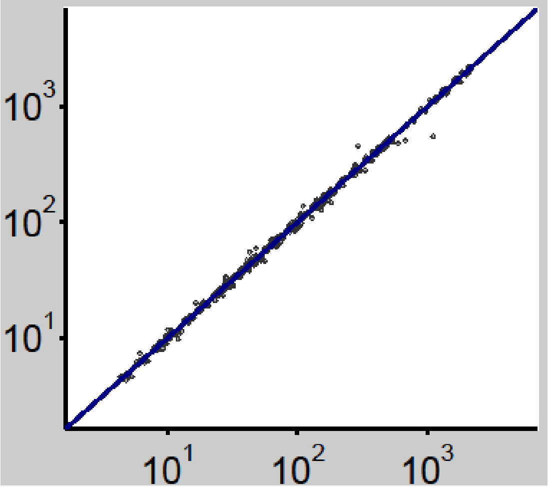
Examples



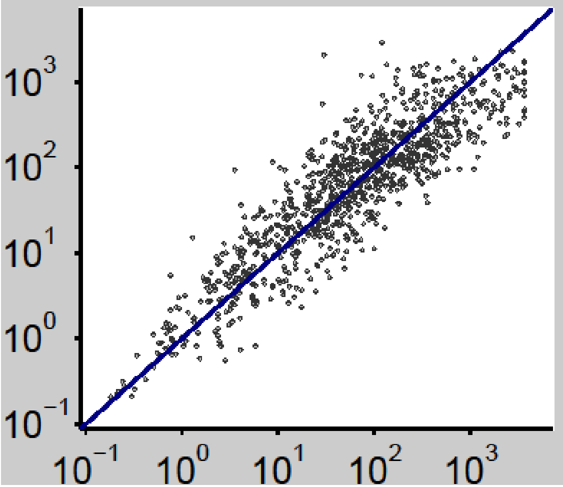
Circuit Verification, SPEAR



MIPLIB, CPLEX



Red Cockaded Woodpecker Habitat, CPLEX



Travelling Salesperson, Concorde

Design Patterns

- It’s a lot of work to **design new heuristic algorithms**
  - Algorithms that do well on instances arising from a given application often **perform poorly elsewhere**
- Solution: **automatic analysis and design patterns**
  - general methods for predicting algorithm performance and constructing new algorithms, based on representative sets of “typical” problem instances
  - exchange expensive human expertise for cheap computer time

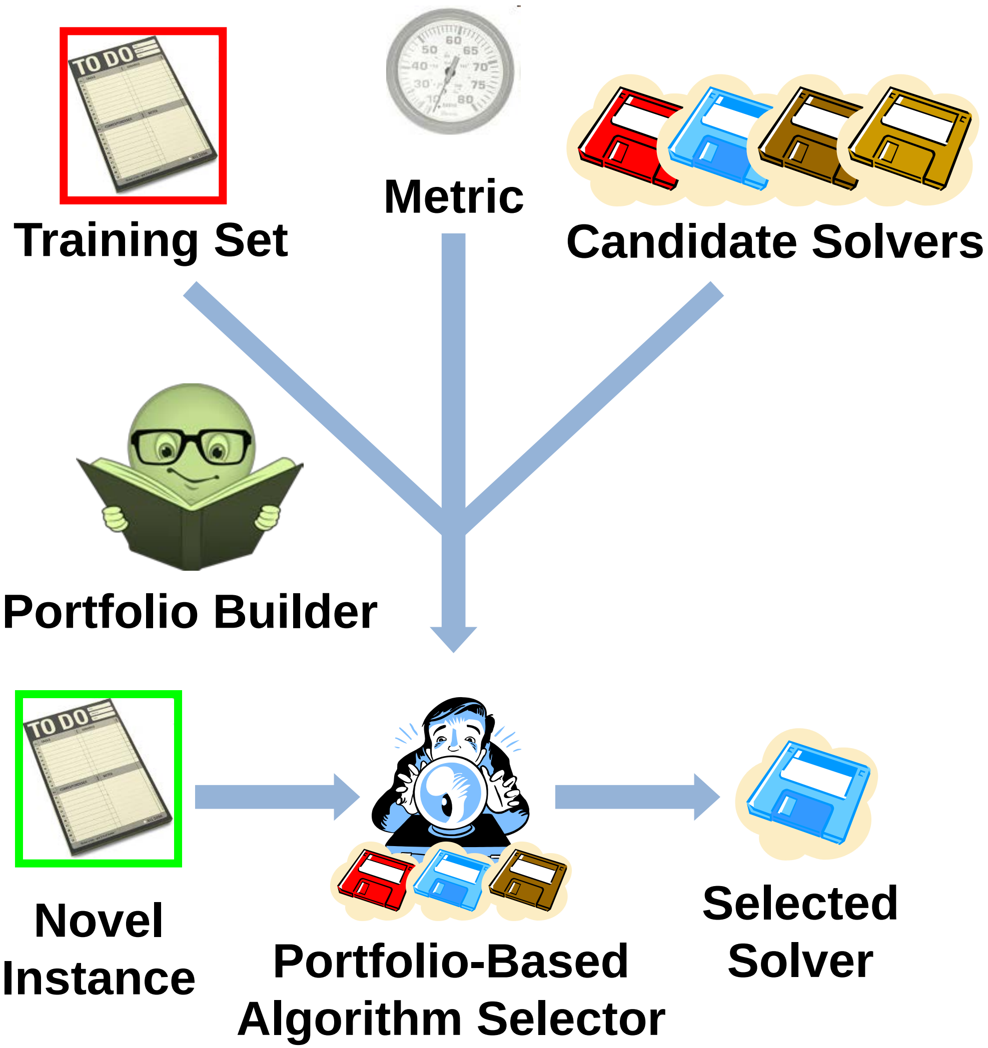
Satisfiability (SAT) Solvers

- There are **many high performance SAT solvers**
  - indeed, for years a biannual international competition has received >20 submissions in each of 9 categories
- However, **no solver is dominant**
  - different solvers work well on different problems
    - hence the different categories
  - even within a category, the best solver varies by instance

SATzilla

- The idea: an **algorithm portfolio**, leveraging the power of all available algorithms
- **SATzilla**:
  - an algorithm portfolio constructed from all available state-of-the-art complete and incomplete SAT solvers
  - it won 5 medals in each of the 2007 and 2009 SAT competitions

SATzilla



- **Given:**
  - **training set of instances**
  - **performance metric**
  - **candidate solvers**
  - **portfolio builder** ***(incl. instance features)***
- Training:
  - collect performance data
  - portfolio builder learns empirical hardness models
- At Runtime:
  - predict performance
  - select solver

Algorithm Design Philosophies

- Traditional approach
  - **Hard-code** various design choices
  - Iteratively conduct **small experiments** to improve the design
- Our approach
  - Make all design options explicit, encoding them as **parameters**
    - Results in a generalized, highly parameterized algorithm
    - Instantiation produces many different solvers
  - Given a distribution, set the parameters using an **automatic algorithm configuration** procedure

SATenstein?

- **Frankenstein's** goal:
  - Create "perfect" human being from scavenged body parts
- **SATenstein's** goal:
  - Create high-performance SAT solvers using components scavenged from existing solvers

How does SATenstein work?

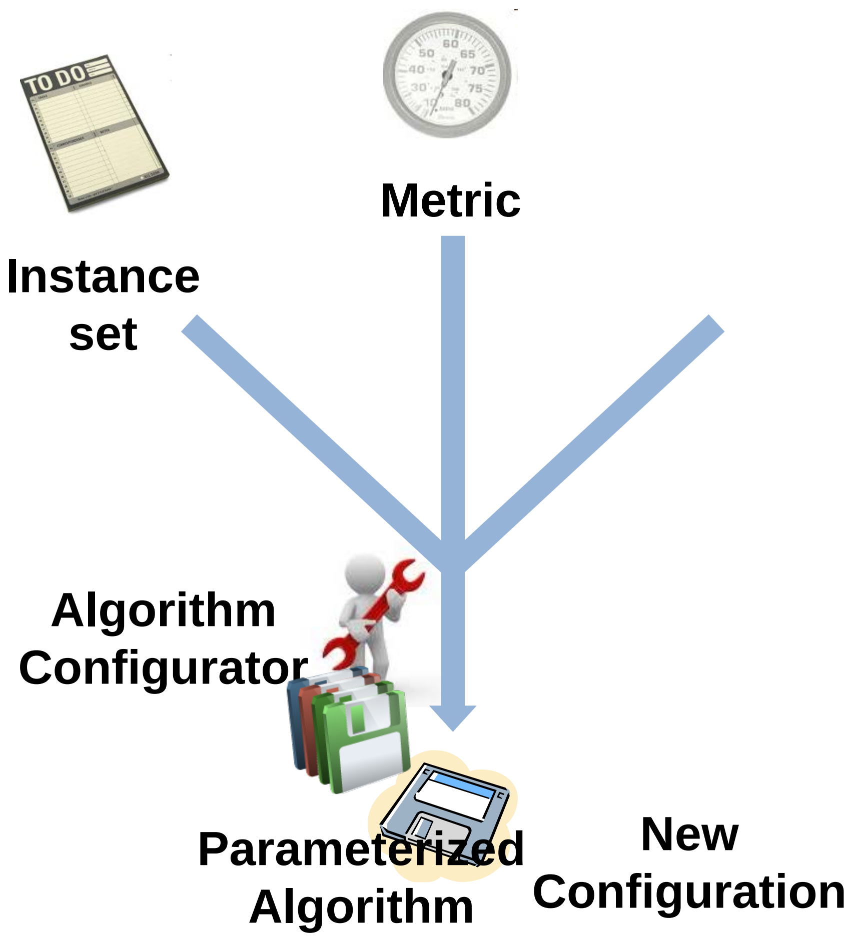
Existing Algorithm Components

Domain Expert

Parameterized Algorithm

- **Designer creates highly-parameterized algorithm from existing components**
- Given:
  - training set of instances
  - performance metric
  - parameterized algorithm
  - algorithm configurator
- Configure algorithm:
  - run configurator on training instances
  - output is a configuration that optimizes metric

How does SATenstein work?



- Designer creates highly-parameterized algorithm from existing components
- Given:
  - training set of instances
  - performance metric
  - parameterized algorithm
  - algorithm configurator
- Configure algorithm:
  - run configurator on training instances
  - output is a configuration that optimizes metric

Summary of Results

- Performance summary:
  - Factor of **70 - 1300** performance improvement over best challenger on QCP, HGEN, CBMC-SE
  - Factor of **1.4 - 2** performance improvement over best challenger on SW-GCP, R3SAT and FAC

- Impact on **state of the art**:
  - in all cases, generated the best SLS algorithm we're aware of
  - for some distributions, our new algorithm is the very best of which we're aware

Advantages and Disadvantages

SATzilla
portfolio-based algorithm selection

SATenstein
algorithm design via automatic configuration

Advantages and Disadvantages

SATzilla
portfolio-based algorithm selection

Exploit per-instance variation between solvers using learned runtime models

- **practical:** e.g., won competition medals
- **fully automated:** requires only cluster time rather than human design effort

Key drawback:

- requires a set of **strong, relatively uncorrelated** candidate solvers
- **can't be applied** in domains for which such solvers do not exist

Advantages and Disadvantages

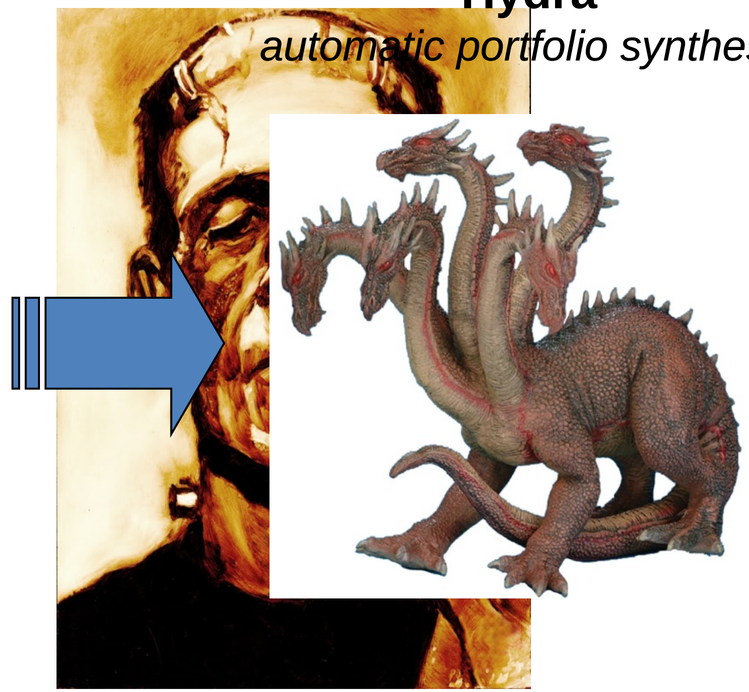
SATenstein

[KhudaBukhsh, Xu, Hoos, Leyton-Brown, 2009]

algorithm design via automatic configuration

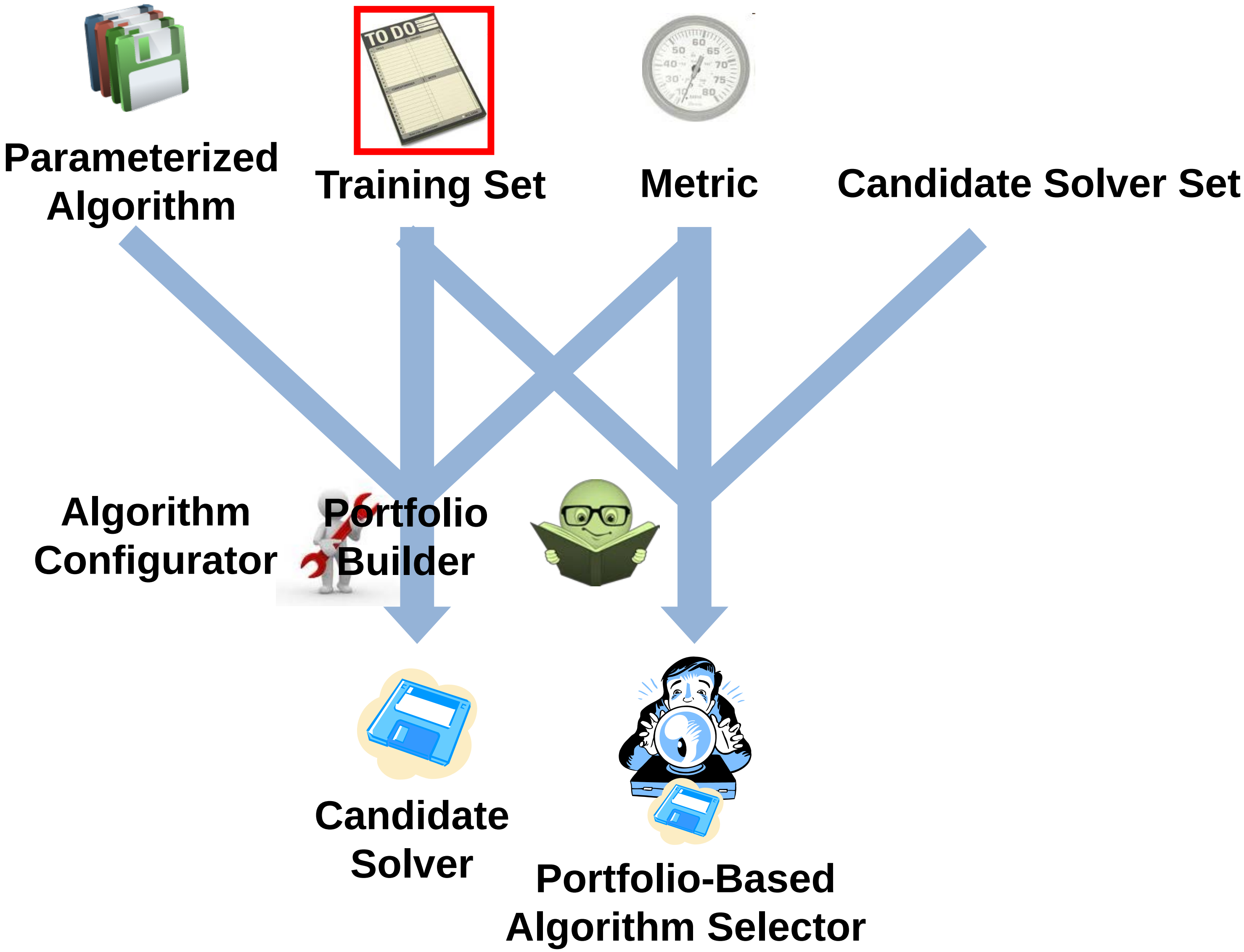
- **Instead of manually exploring a design space, build a highly-parameterized algorithm and then configure it automatically**
- Can find **powerful, novel designs**
- But: **only produces single algorithms** designed to perform well on the entire training set

Hydra

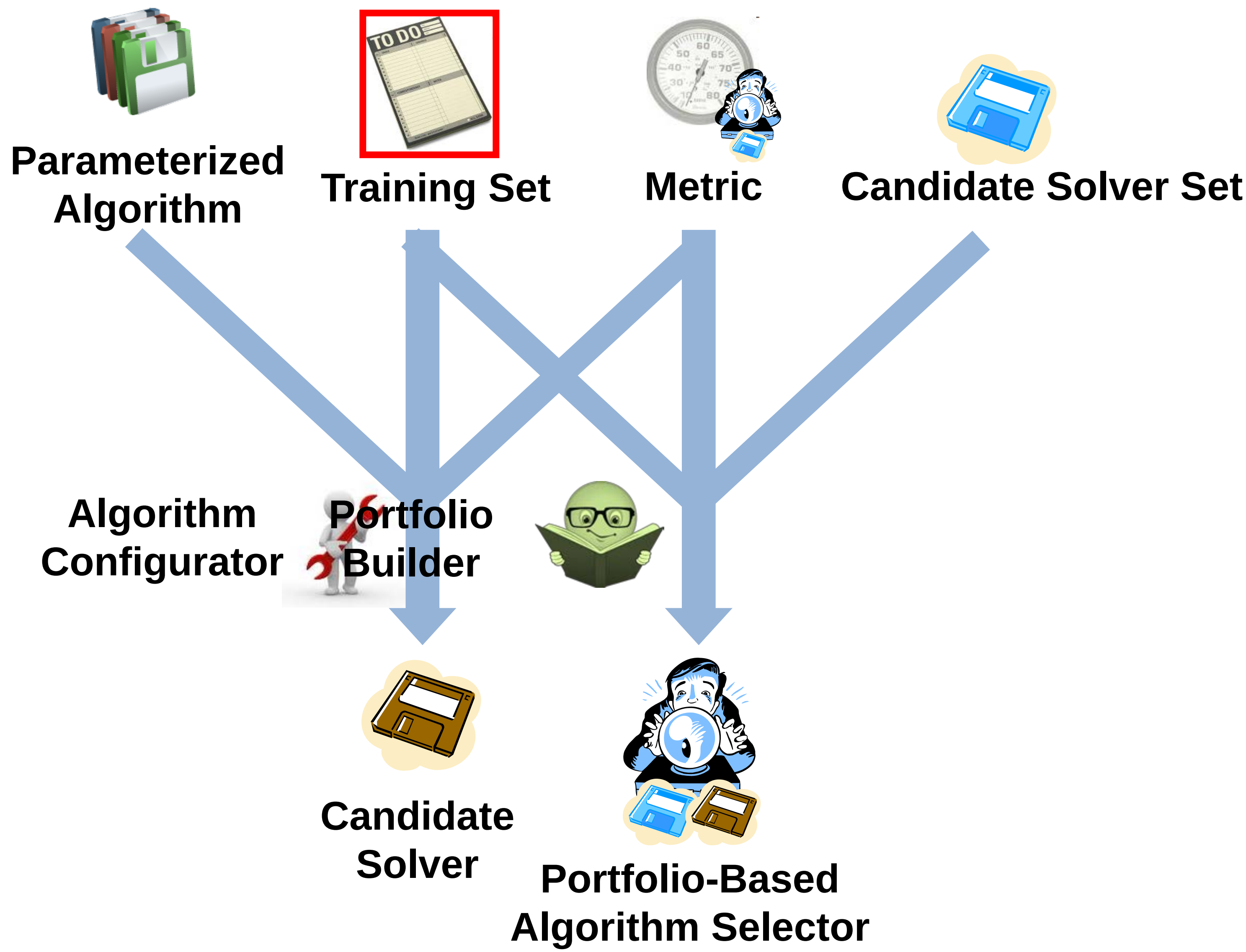


Starting from a **single parameterized algorithm**, automatically find a set of **uncorrelated configurations** that can be used to build a **strong portfolio.**

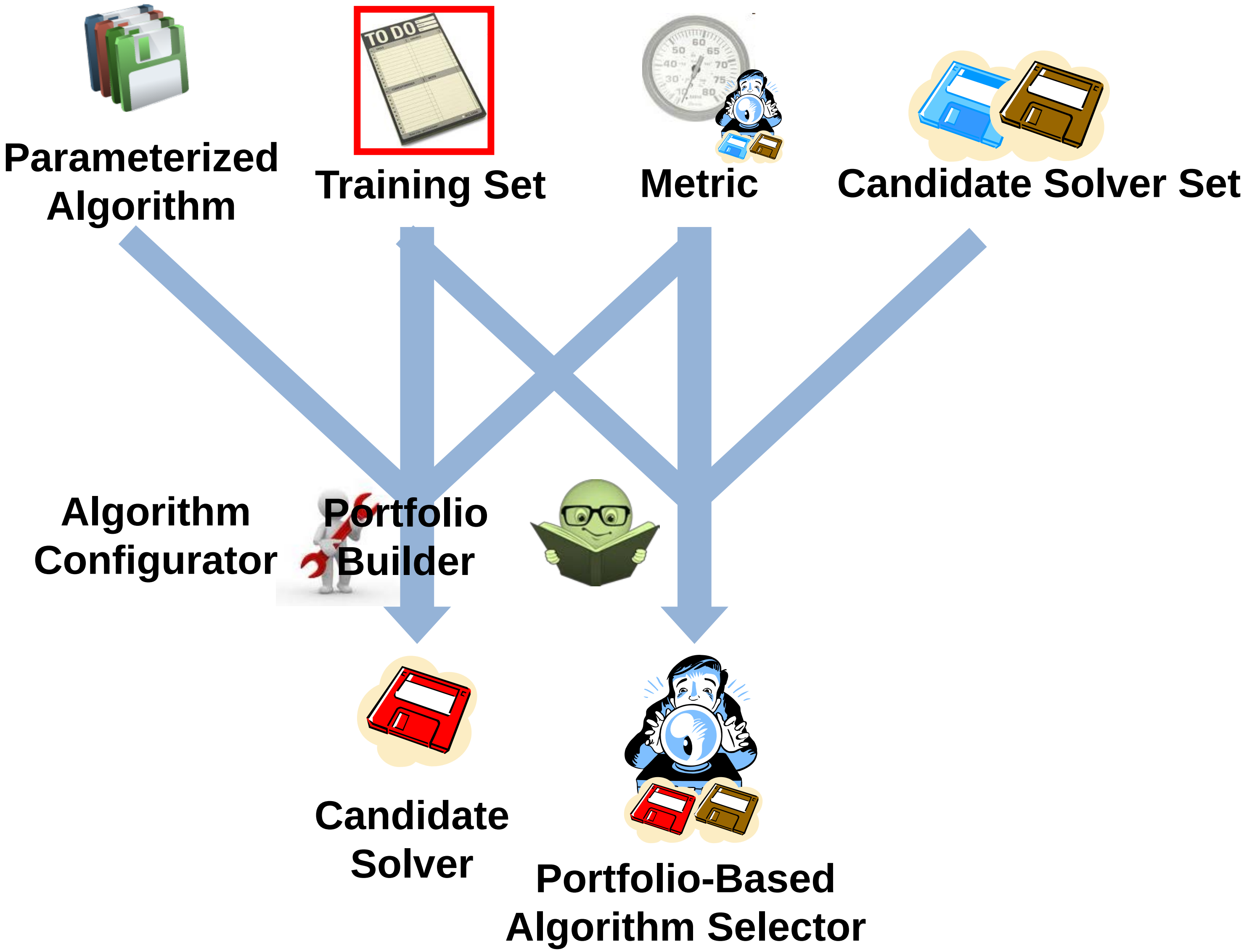
Hydra Procedure: Iteration 1



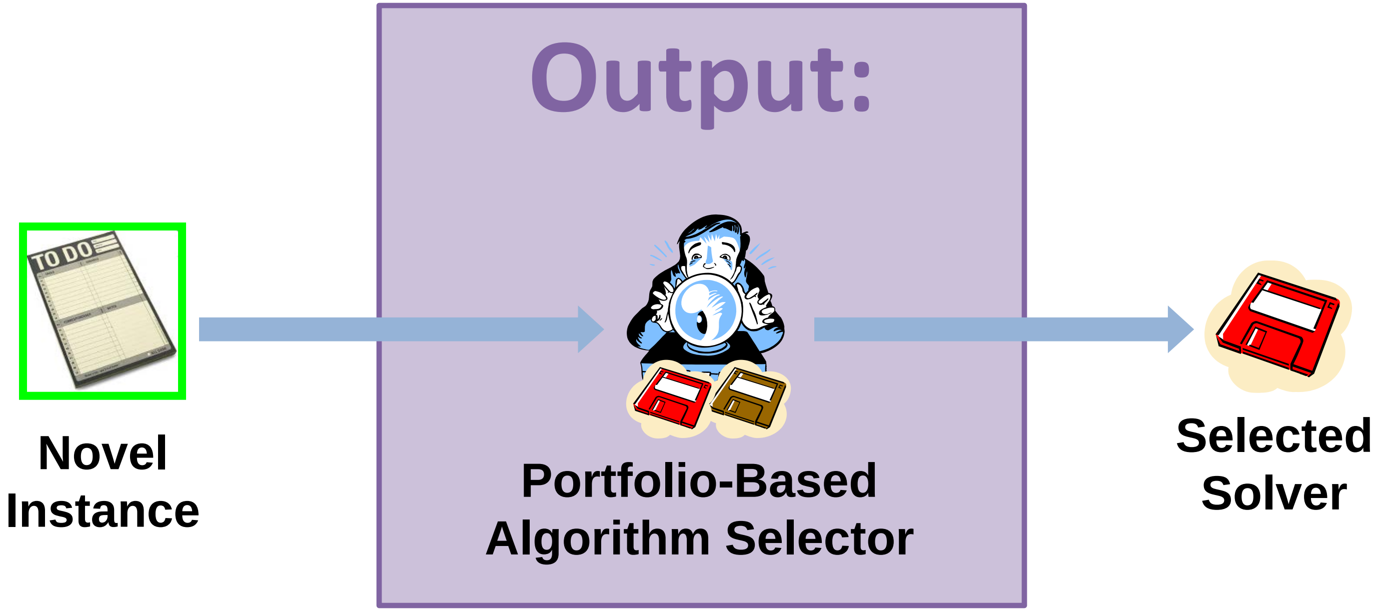
Hydra Procedure: Iteration 2



Hydra Procedure: Iteration 3



Hydra Procedure: After Termination



Performance Summary

Solver	RAND	HAND	BM	INDU
Best Challenger (of 17)	1128.63	2960.39	224.53	11.89

** Statistically insignificant performance difference (sign rank test).*
Hydra's performance was significantly better in all other pairings.

Algorithms for Making Good Decisions

- **Reasoning about Large Games:** can compute equilibria (etc.) of large game-theoretic interactions by representing them as action-graph games.

- **Auction Design and Analysis:** game theory can be leveraged to construct protocols that work even if agents aren't cooperative. Computational techniques can help us understand what will happen under a new design.

- **Empirical Algorithmics:** algorithms that work well in practice
  - **Empirical hardness models:** predict algorithm behavior
  - **SATzilla:** use these models to build algorithm portfolios
  - **SATenstein:** solve the design problem using automatic configuration
  - **Hydra:** design portfolios from a single parameterized algorithm