Algorithmic Analysis of Code-Breaking Games

Miroslav Klimoš

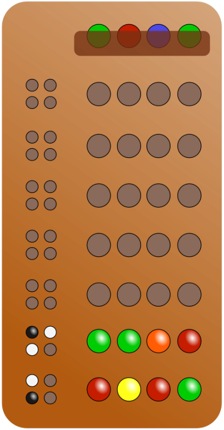
Advisor: prof. RNDr. Antonín Kučera Ph.D.

June 2014

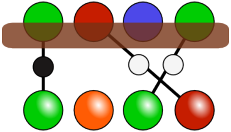
Code-breaking games

- 2 players: **codemaker + codebreaker**
- codemakers selects a **secret code**
- codebreaker strives to reveal the code through **experiments**
- experiments provide **partial information** about the code

Mastermind



- Code: 4 pegs $\times$ 6 colours
- Experiment $=$ guess



The counterfeit coin

- n coins + balance scale
- All coins except one have the same weight
- Identify the odd-weight coin

Challenges

1. Formal model of code-breaking games
2. Strategies for experiment selection
3. Method for symmetry detection
4. Algorithms for strategy evaluation and synthesis
5. Computer language for game specification
6. Implementation of proposed algorithms

Formal model

- Game description
  - set of propositional variables X
  - initial constraint φ
  - set of **experiments** E
- Secret code: valuation of X (satisfying φ)
- Partial information: formula in X
- Strategy (memory-less): function $\textsc{form}_X \to E$

Example – the counterfeit coin with 4 coins

- variables $\{x_1, x_2, x_3, x_4, y\}$
- initial constraint $\text{EXACTLY}_1(x_1, x_2, x_3, x_4)$
- experiment "coin 1 $\times$ coin 2" can result in

  "$<$": $(x_1 \wedge \neg y) \vee (x_2 \wedge y)$

  "$>$": $(x_1 \wedge y) \vee (x_2 \wedge \neg y)$

  "$=$": $\neg x_1 \wedge \neg x_2$

Strategies

Max-models strategy

Select an experiment that minimizes the maximal number of possibilities for the code in the next round.

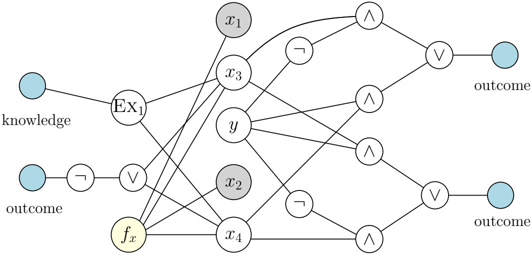
- in Mastermind 4×6, this is worst-case optimal
- Generalization: one-step look-ahead strategies

Symmetry detection

- Problem: symmetries enlarge the state space
- Solution: experiment equivalence

Symmetry detection

- Problem: symmetries enlarge the state space
- Solution: experiment equivalence



- Reduction to isomorphism of labelled graphs
- Tools available for graph canonization (Bliss)

Algorithmic problems

Strategy analysis

Compute the average-case/worst-case number of experiments required to reveal the code by a given strategy.

Optimal strategy synthesis

Synthesise the average-case/worst-case optimal strategy.

Algorithmic problems

Strategy analysis

Compute the average-case/worst-case number of experiments required to reveal the code by a given strategy.

Optimal strategy synthesis

Synthesise the average-case/worst-case optimal strategy.

$\Longrightarrow$ intelligent backtracking

Game specification language

- Based on the formal model
- Python preprocessing for easier generation

Game specification language

- Based on the formal model
- Python preprocessing for easier generation

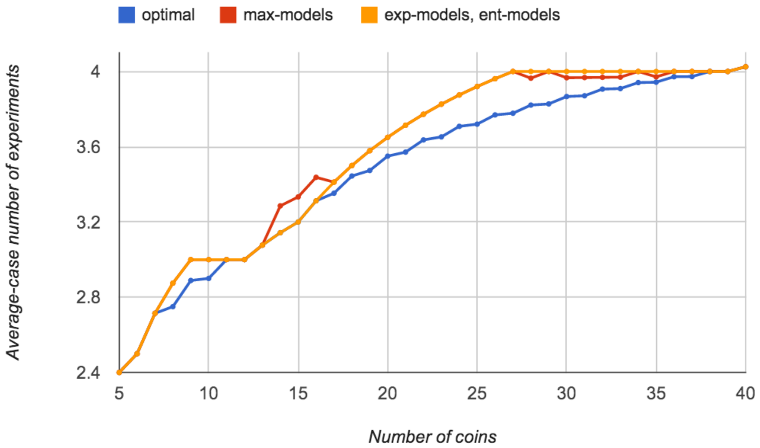
```
n = 4
xvars = ["x1", "x2", "x3", "x4"]
VARIABLES(xvars + ["y"])
CONSTRAINT("Exactly-1(%s)" % ",".join(xvars))
ALPHABET(xvars)
MAPPING("X", xvars)

for m in range(1, n//2 + 1):
  EXPERIMENT("weighing" + str(m), 2*m)
  PARAMS_DISTINCT(range(1, 2*m + 1))
  OUTCOME("lighter", "((%s) & !y) | ((%s) & y)" ...
  OUTCOME("heavier", "((%s) & y) | ((%s) & !y)" ...
  OUTCOME("same", "!(%s)" % params(1, 2*m))
```

Implementation – the Cobra tool

- Command-line tool written in C++
- Modes of operation
  - Overview
  - Simulation
  - Strategy analysis
  - Optimal strategy synthesis
- Uses SAT solvers (Minisat, Picosat)
- Uses graph canonization tool (Bliss)

Experimental results



Conclusions

- Challenges
  - ✓ Formal model based on propositional logic
  - ✓ Strategies for experiment selection
  - ✓ Symmetry detection based on graph isomorphism
  - ✓ Algorithms for strategy evaluation and synthesis
  - ✓ Computer language built on top of Python
  - ✓ Implementation in the Cobra tool

Conclusions

- Challenges
  - ✓ Formal model based on propositional logic
  - ✓ Strategies for experiment selection
  - ✓ Symmetry detection based on graph isomorphism
  - ✓ Algorithms for strategy evaluation and synthesis
  - ✓ Computer language built on top of Python
  - ✓ Implementation in the Cobra tool

- Applications
  - easily reproduce existing results
  - evaluate new strategies
  - analyse other code-breaking games