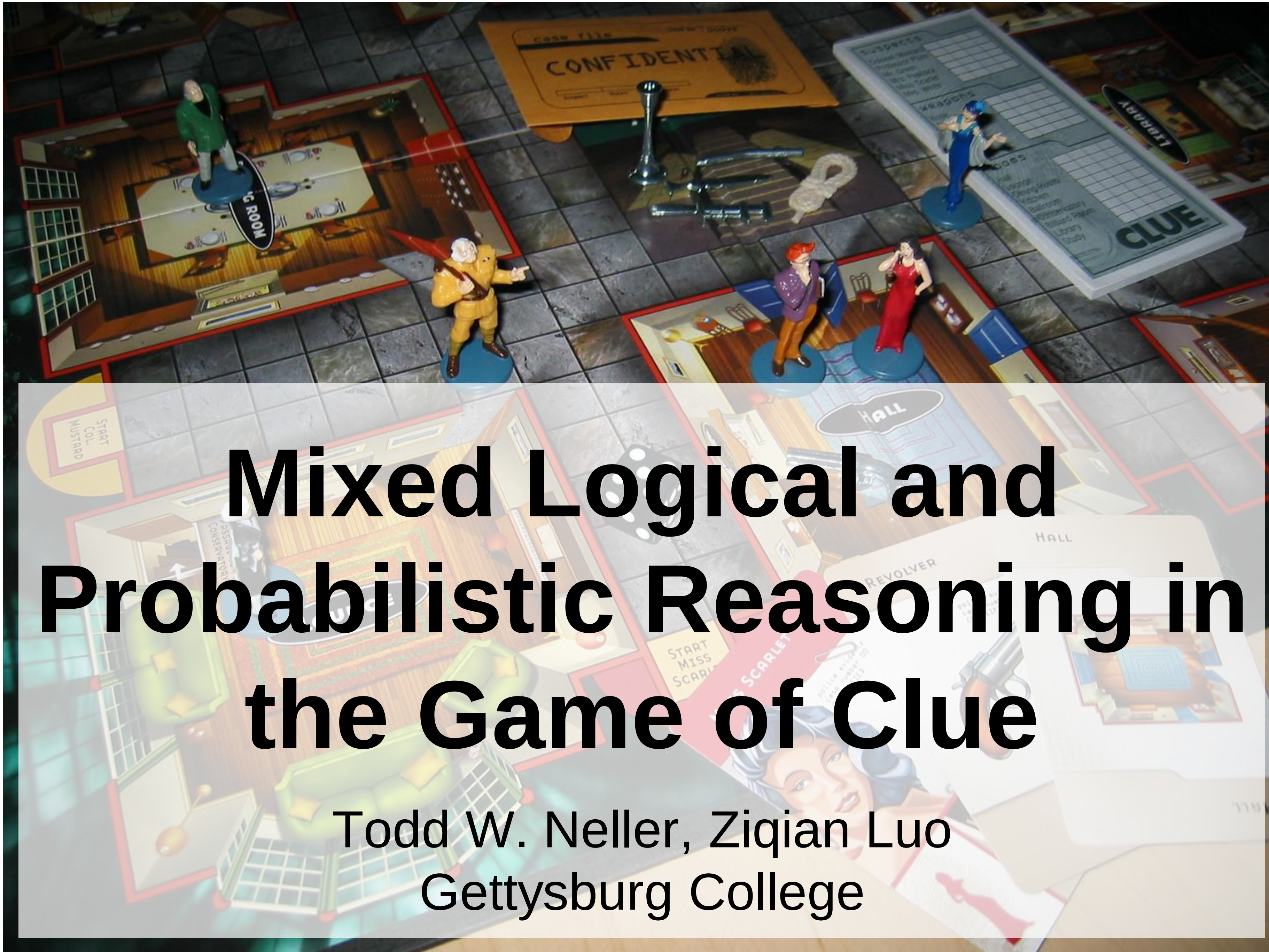


Mixed Logical and Probabilistic Reasoning in the Game of Clue

Todd W. Neller, Ziqian Luo
Gettysburg College

Overview

- Clue/Cluedo Rules
- Basic Propositional Game Knowledge
- At-Least Constraints for Logical Reasoning
- Probabilistic Estimation through WalkSAT-like Sampling
- Algorithmic Variations and Experimental Results
- Conclusions and Future Work

case file
CONFIDENTI
LIBRARY
CLUE
START COL. MUSTARD
HALL
LOUNGE
CONSERVATORY
START MISS SCARL
MISS SCARLET
REVOLVER
HALL

The Game of Clue (a.k.a. Cluedo)

- 21 cards: 6 suspects, 6 weapons, 9 rooms
- Case file has unknown, random suspect, weapon, and room (SWR)
- Remaining cards dealt to players
- Player suggests SWR, first player clockwise that can refute, must show card
- Each player can make 1 SWR accusation
- Correct → win; incorrect → lose (& refute)

Clue Knowledge Representation

- Basic Clue reasoning is constraint satisfaction.
- One formulation: Boolean variables c_p denoting “Card *c* is in place *p*.”
- Given CNF representation of Boolean constraints, one can reason with SAT solver refutations.
- However, not all game knowledge can be expressed in SAT efficiently...

Basic Propositional Game Knowledge

- Initial knowledge
    - Each card is in exactly one place.
    - Exactly one card of each category is in the case file.
    - You know your hand of cards.
    - You know how many cards have been dealt to each player.
- Play knowledge
    - A player cannot refute a suggestion.
    - A player refutes your suggestion by showing you a card.
    - A player refutes another player's suggestion by showing them a card privately.
    - A player makes an accusation and shares whether or not the accusation was correct.

Which of these leads to the largest number of SAT clauses in a CNF representation?

Basic Propositional Game Knowledge

- Initial knowledge
  - Each card is in exactly one place.
  - Exactly one card of each category is in the case file.
  - You know your hand of cards.
  - You know how many cards have been dealt to each player.
- Play knowledge
  - A player cannot refute a suggestion.
  - A player refutes your suggestion by showing you a card.
  - A player refutes another player's suggestion by showing them a card privately.
  - A player makes an accusation and shares whether or not the accusation was correct.

At-Least Constraints for Logical Reasoning

- Too simple: SAT clauses
  - 6-player game where each player has exactly 3 of the 21 cards.
  - At most 3 cards are held by a player *p*: **35,910 CNF SAT clauses**
  - At least 3 cards are held by a player *p*: **1,260 CNF SAT clauses**
- Too complex/expressive: linear psuedo-Boolean constraints
  - $\sum_i a_i \cdot l_i \ \# \ b$ where a_i, b are integers, l_i are 0/1 literals, and # is an (in)equality.
- Just right: at-least constraints (a.k.a. cardinality constraints, extended clauses)
  - $\sum_i l_i \ \geq b$ where b is an integer and l_i are 0/1 literals.
  - **"At least $\boldsymbol{b}$ of literals $\{\boldsymbol{l_1}, \ldots, \boldsymbol{l_n}\}$ are true."**
  - 6-player game where each player has exactly 3 of the 21 cards → only **12 at-least clauses** needed.
  - We have extended Donald Knuth's Dancing Links (DLX) algorithm for more general constraint satisfaction with at-least constraints.

Probabilistic Estimation through Sampling

- Probabilities for unknown card positions can be *exactly computed* with model counting.
  - Model counting is combinatorially infeasible for all but endgame scenarios with few models.
- Probabilities for unknown card positions can be *approximately computed* with model sampling.
- *WalkSAT step*:
  - Pick a random unsatisfied constraint clause.
    - Flip a variable chosen at random from among those that would cause the fewest clauses to become unsatisfied.
- *Tabu metaheuristic*: A tabu tenure is the number of steps that must pass before a variable may be flipped again.

Algorithm 1 Search and Sample — a WalkSAT-like sampling algorithm

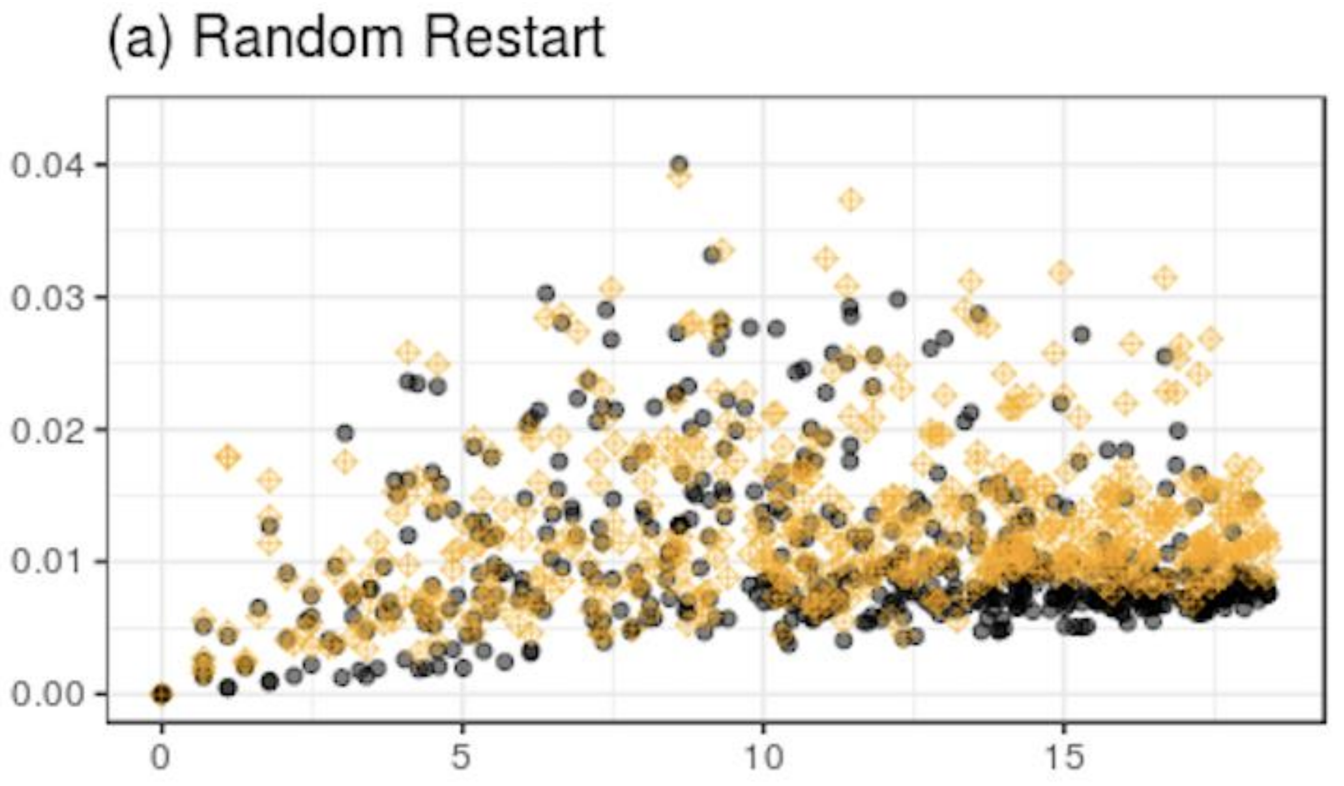
```
Input: at-least clauses of the game state, fixed variable assignments, number of search iterations
Output: probability estimate of each card being at each place

function SearchAndSample(clauses, fixedVarAssignments, numIter):
RandomRestart(); {randomly assign all non-fixed variables}
if there are no unsatisfied clauses then
    RecordSample(); {tally true non-fixed variables, increment sample count}
end if
tabuTenure = 10
lastSampleStep = 0
for step = {1, 2, 3, ..., numIter} do
    tabuCutoffStep = max(1, step − tabuTenure, lastSampleStep)
    if all clauses are satisfied then
        Choose a random non-fixed variable.
    else
        Choose a random unsatisfied clause.
        Choose a variable from that clause according to the WalkSAT heuristic.
    end if
    Flip(); {make chosen variable flip assignment, update data structures}
    if there are no unsatisfied clauses then
        RecordSample();
        lastSampleStep = step
    end if
```

Algorithmic Variations

- Our testing revealed two problems that cause probabilistic approximation bias:
  - In opening game states:
    - Too high a tabu tenure results in too few samples.
    - Too low a tabu tenure results in too many returns to the same sample, and too few unique samples.
  - In endgame states:
    - Even when all solutions can be sampled, WalkSAT-like sampling is still non-uniform and biased.

Algorithmic Variations

Random Restart: After finding and recording a sample solution, perform a random restart, reinitializing all non-fixed variable to random values.

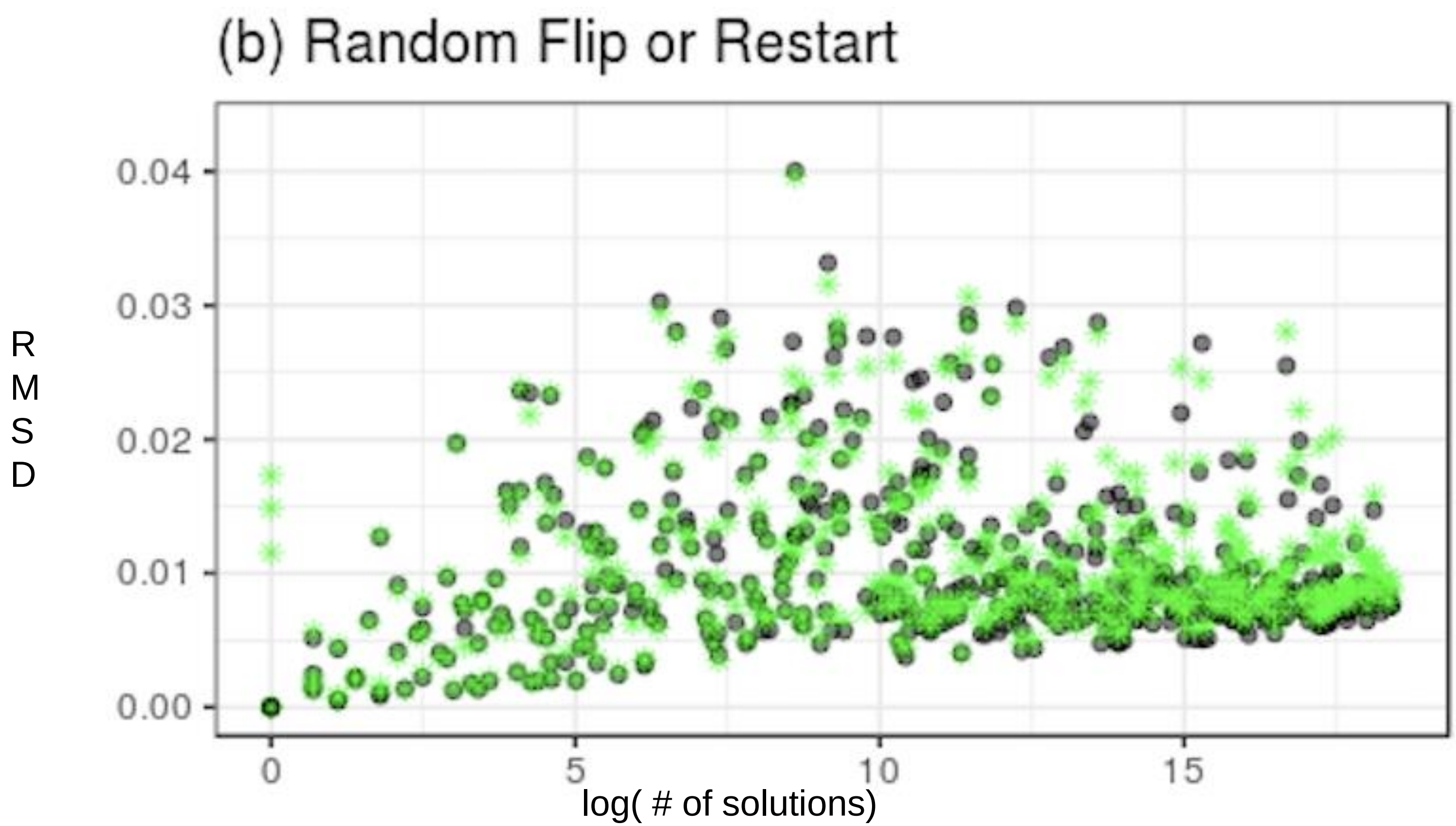


R
M
S
D

log(# of solutions)

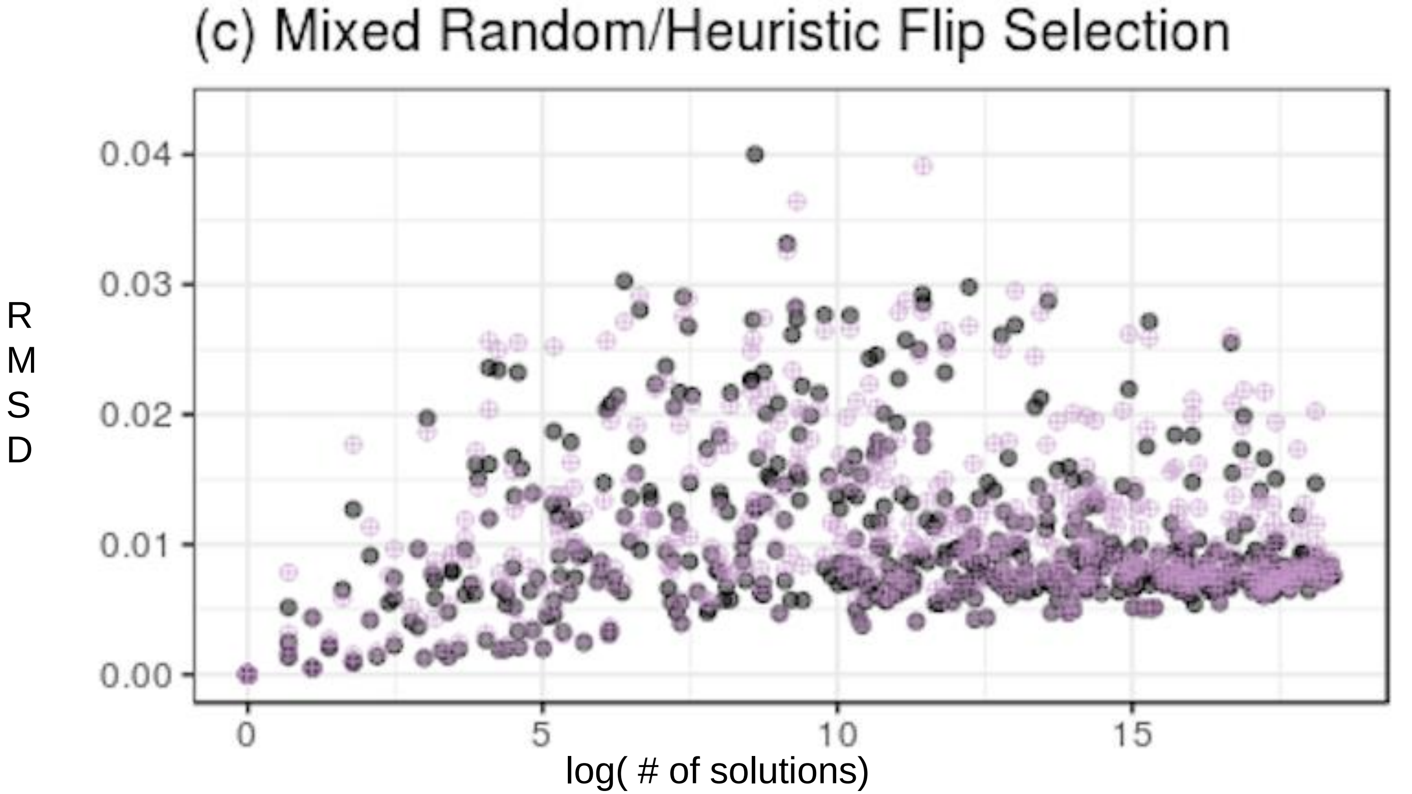
Algorithmic Variations

Random Flip or Restart: After finding and recording a sample solution, perform a random variable flip with probability 0.2. Otherwise, perform a random restart.



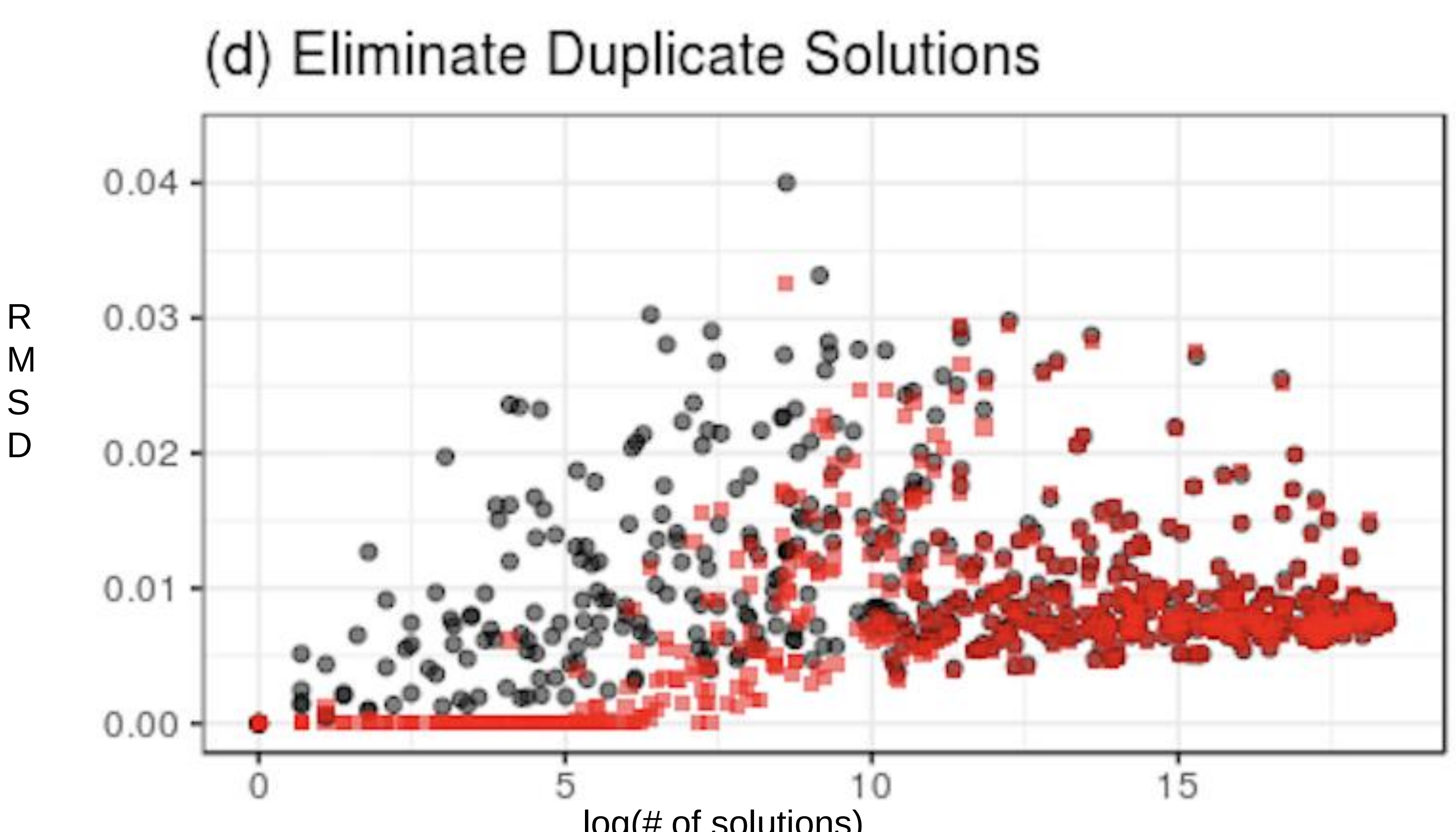
Algorithmic Variations

Mixed Random/Heuristic Flip Selection: After having chosen a random unsatisfied clause, with probability 0.2, flip a random variable of that clause. Otherwise, flip a random variable among those that minimize the number of clauses that will become false as a result.



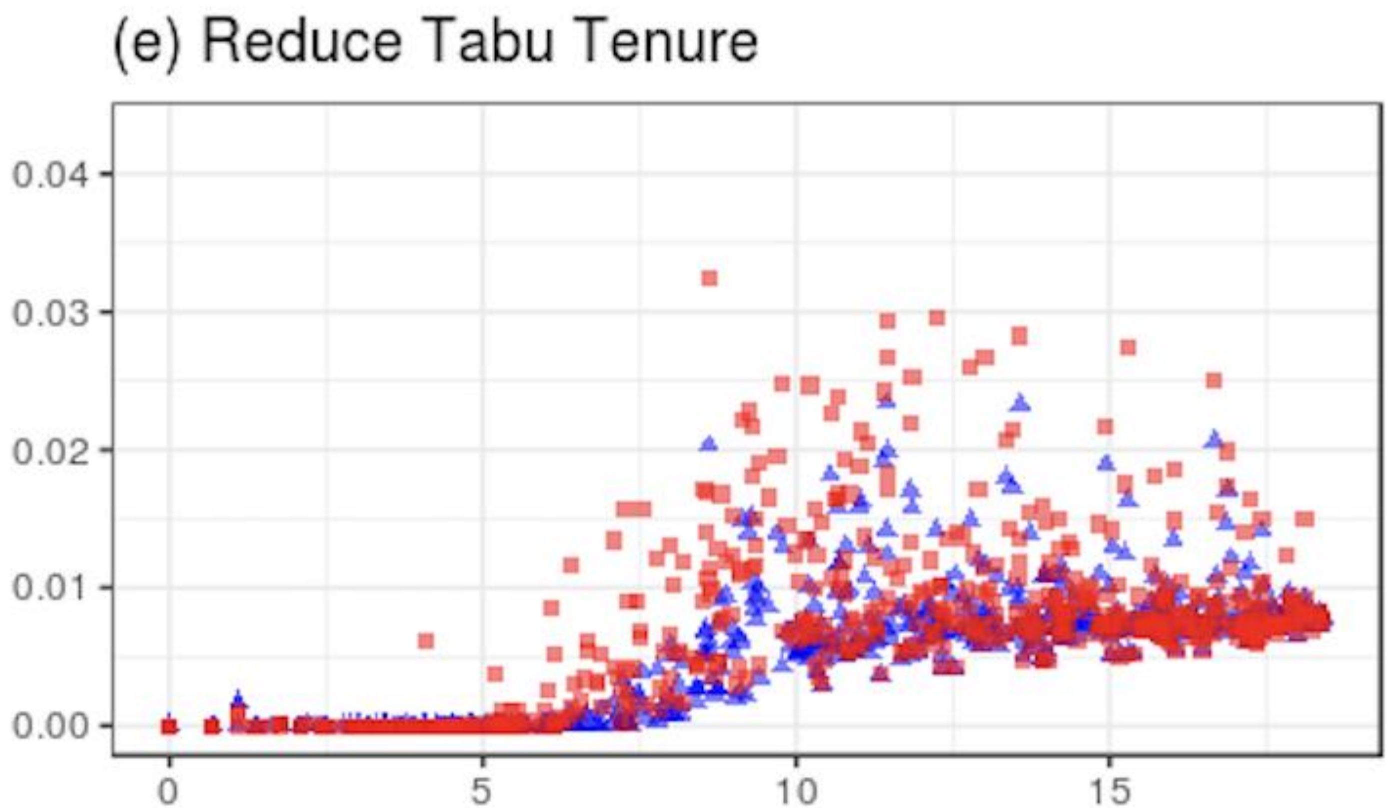
Algorithmic Variations

Eliminate Duplicate Solutions: Record only unique sample solutions.



Algorithmic Variations

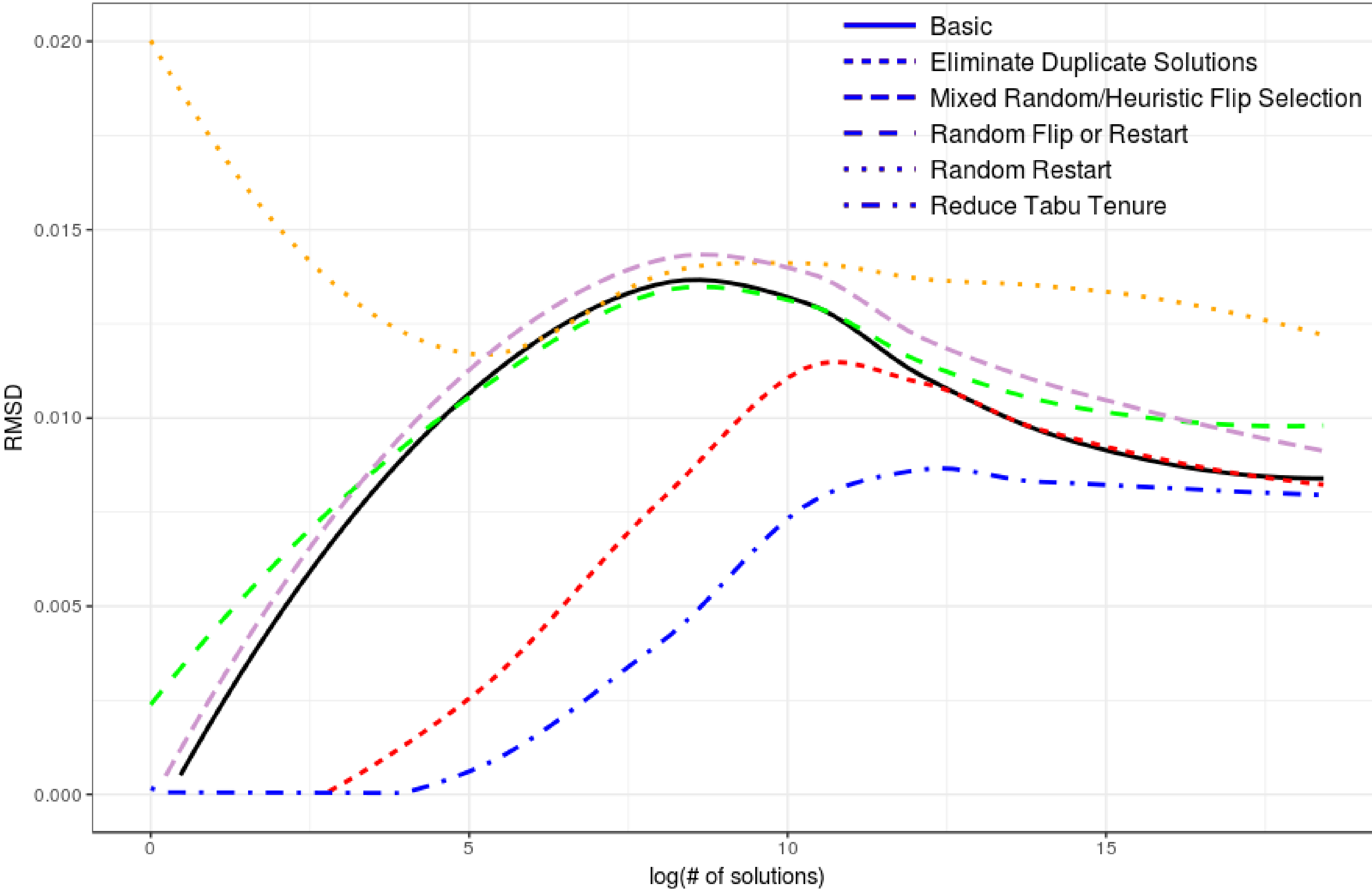
Reduce Tabu Tenure: In addition to unique samples, reduce the tabuTenure constant from 10 to 2, allowing greater frequency of individual variable flips.



RMSD

log(# of solutions)

Experimental Results



Conclusions

- The efficiency of finding and sampling solutions with a WalkSAT-like heuristic is also the cause of sampling bias.
- Two ideas resulted in a total 41% reduction of root-mean-square deviation in estimation error:
  - elimination of duplicate samples
  - reduction in tabu tenure - the tabu metaheuristic was important, yet the best tabu tenure was a short tenure for this problem domain.
- Seeking a more diverse sample through the introduction of various forms of randomness came at an even greater cost of error through much-reduced sampling.

Future Work

- This work represents initial steps to mitigate such sampling bias and compute better probabilistic estimates efficiently.
- However, we would expect that future work could improve upon this work in two important respects described in (Gomes, 2009):
    - estimation quality (i.e. through improvements such as we've found), and
    - confidence bounds on such estimations.
- Such confidence bounds are of interest in assessing the utility of making, for example, an uncertain accusation when one believes one may not get another turn to make a certain accusation in Clue.

Questions?