Locally Informed Global Search for Sums of Combinatorial Games

Martin Müller and Zhichao Li
University of Alberta
Edmonton, Canada
Presented by Xiaozhen Niu

Overview

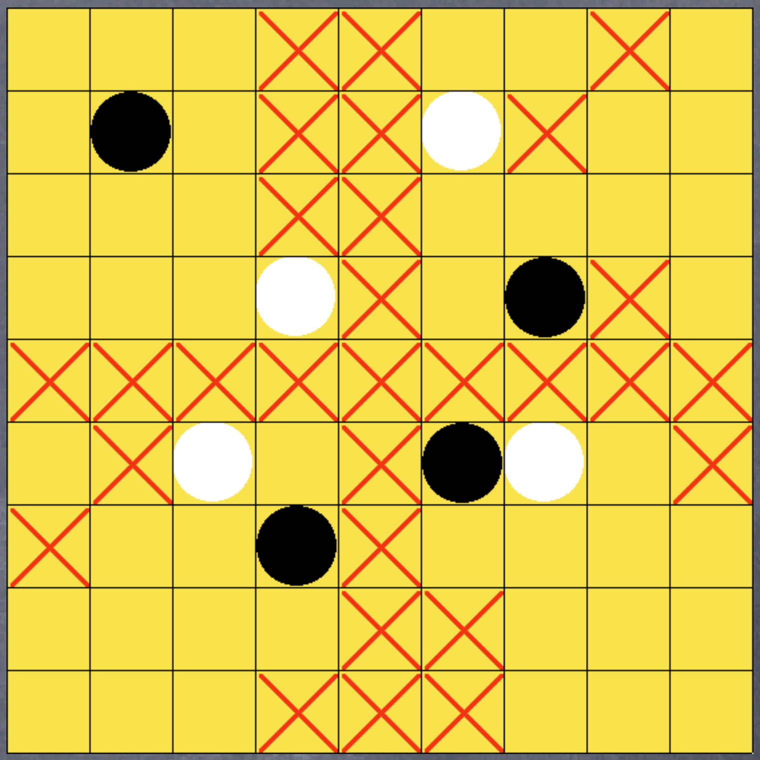
- Sums of games
- Global and Local search
- Locally informed global search
- Experiments

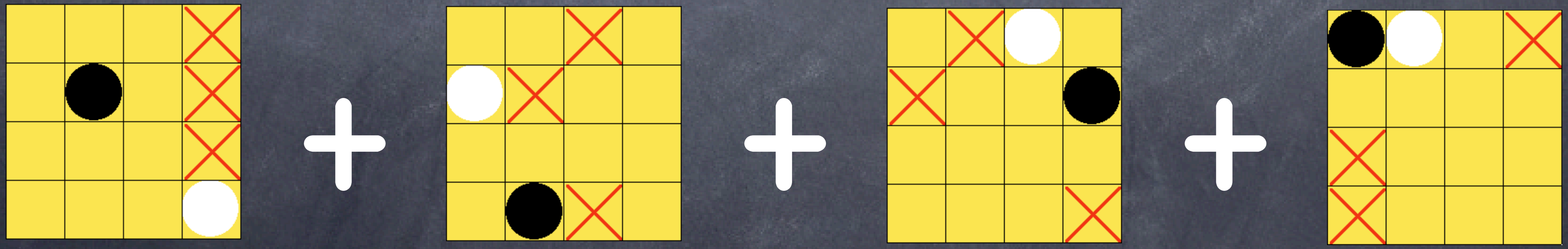
Sums of Games

- Given a game
- Split it into sum
- Result: independent subgames

Example - Amazons

- X = burnt-off square
- Wall of X's divides board into independent subgames





Abstract Games

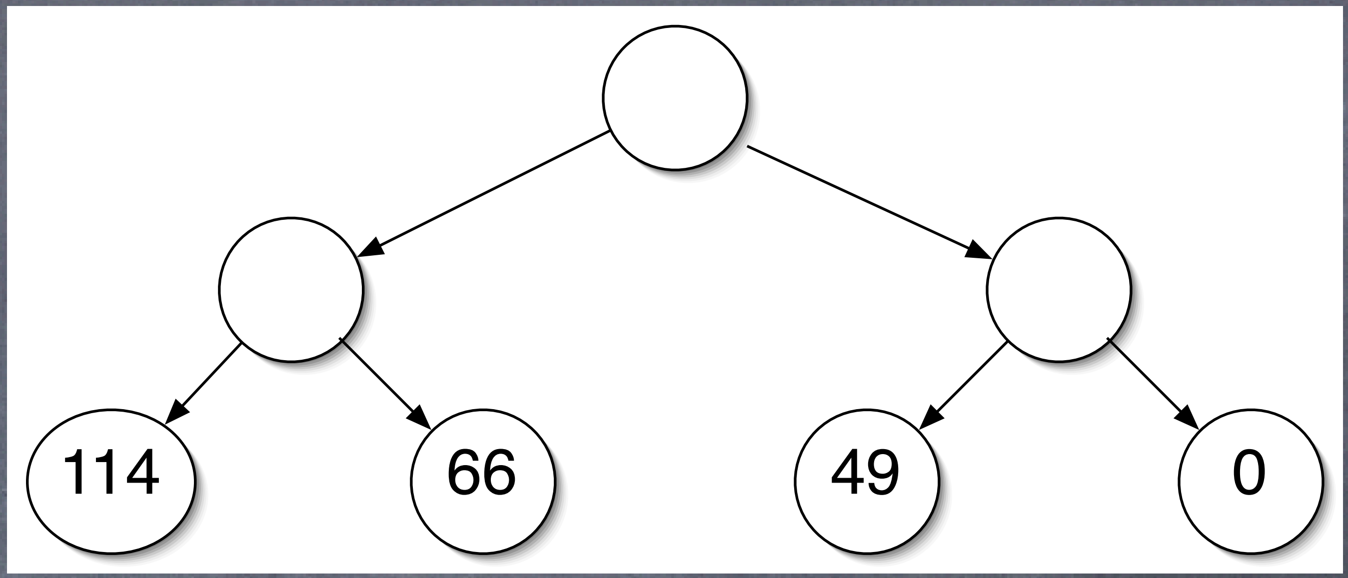
- Play for numeric payoffs
- Game consists of Left and Right options
- $G = G^L|G^R$
- Recursive, until game is integer
- $G = G^L|G^R$, $G^L = 200|150$, $G^R = 100|50$
- Shorter: $G = 200|150 \;||\; 100|50$

Random Combinatorial Games

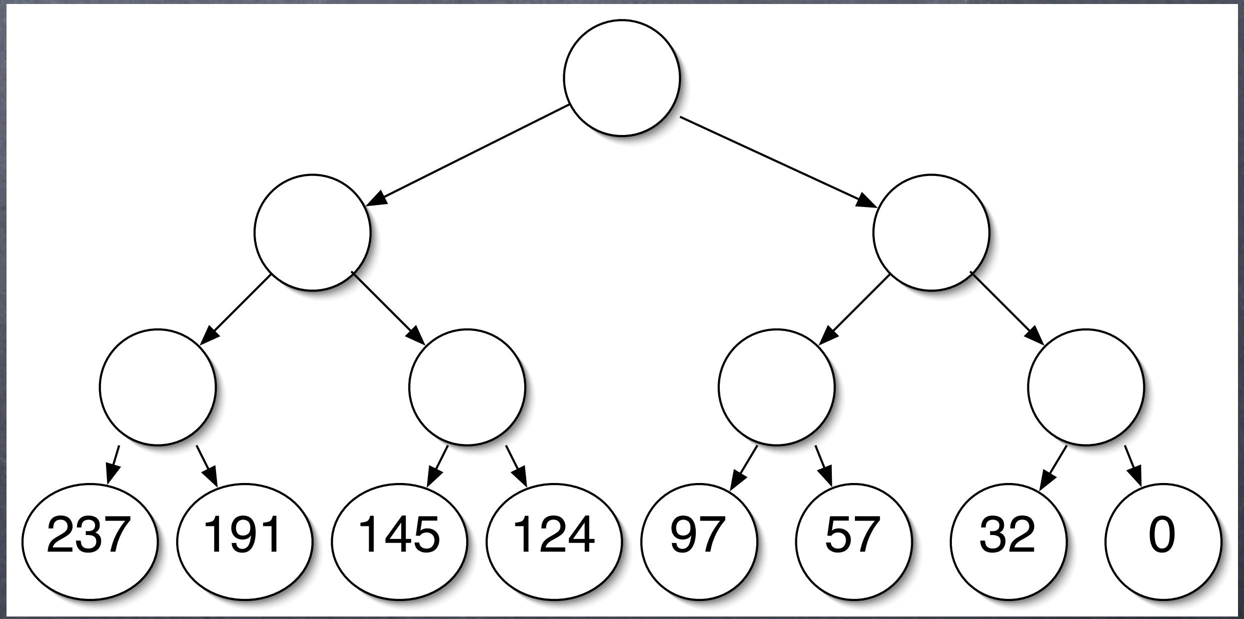
- Model similar to (Cazenave 2002)
- Build binary tree, k levels deep
- Assign random values to leaves, right-to-left
- $v_1 = 0$, $v_{i+1} = v_i + \text{random}(n)$

Examples

- 2-level game, n=50
- 114 | 66 || 49 | 0



- 3-level game, n=50
- 237 | 191 || 145 | 124 ||| 97 | 57 || 32 | 0



Playing Sum Games

- Given sum game $G = G_1 + G_2 + \ldots + G_n$
- Play well (or optimal)
- Use local analysis in G_i as much as possible
- Minimize amount of global-level search

Mean and Temperature

- Mean: ``average" value of a game
  - Example: 5|-5 mean = 0
- Temperature: ``urgency" of a move
  - Example 5|-5 temperature = 5

Previous Work

- Exact algorithm: minimax search, alpha-beta pruning
- Heuristic algorithms: hotstrat, thermostrat, sentestrat

This Study

- Enhance minimax search by using local information
  - Move ordering by temperature
  - Move pruning by incentives
- Test quality of searches with limited depth, or with temperature bound
- Compare with standard approaches

Exact Algorithm

- Alpha-beta minimax search
- Search until end of the game
- Plays optimally

Heuristic Search Algorithms

- Limit search
  - Depth limit
  - temperature bound
- Use heuristic evaluation in leaf nodes
  - Sum-of-means of local games
  - Hotstrat rollouts

Experiments

Experiment 1
Move Ordering

- Exact search
- Tried four move ordering schemes
  - BEST-PREV: best move from iterative deepening
  - TEMP: Sort by temperature, hottest first
  - Both

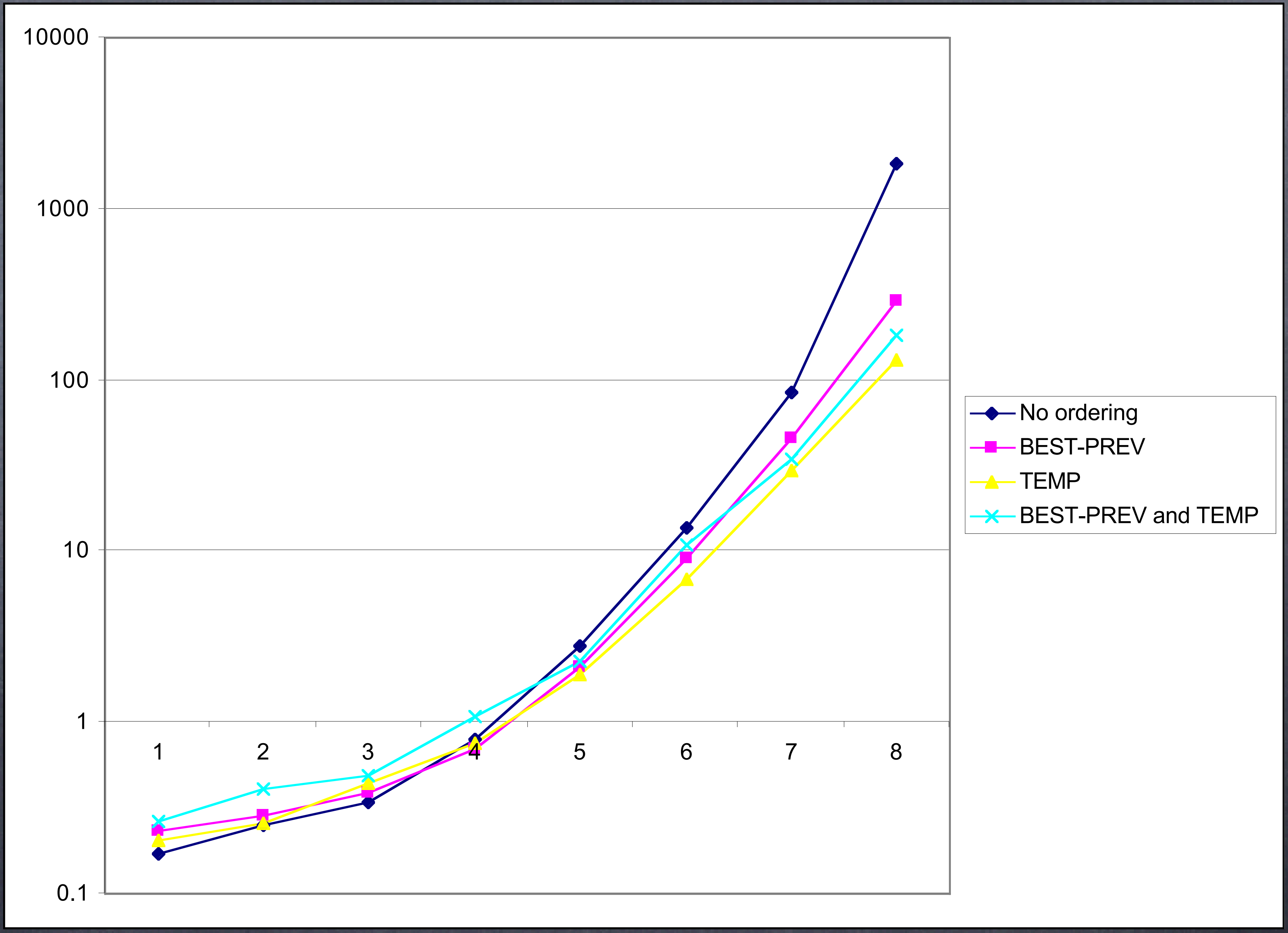
Move Ordering

2-level games

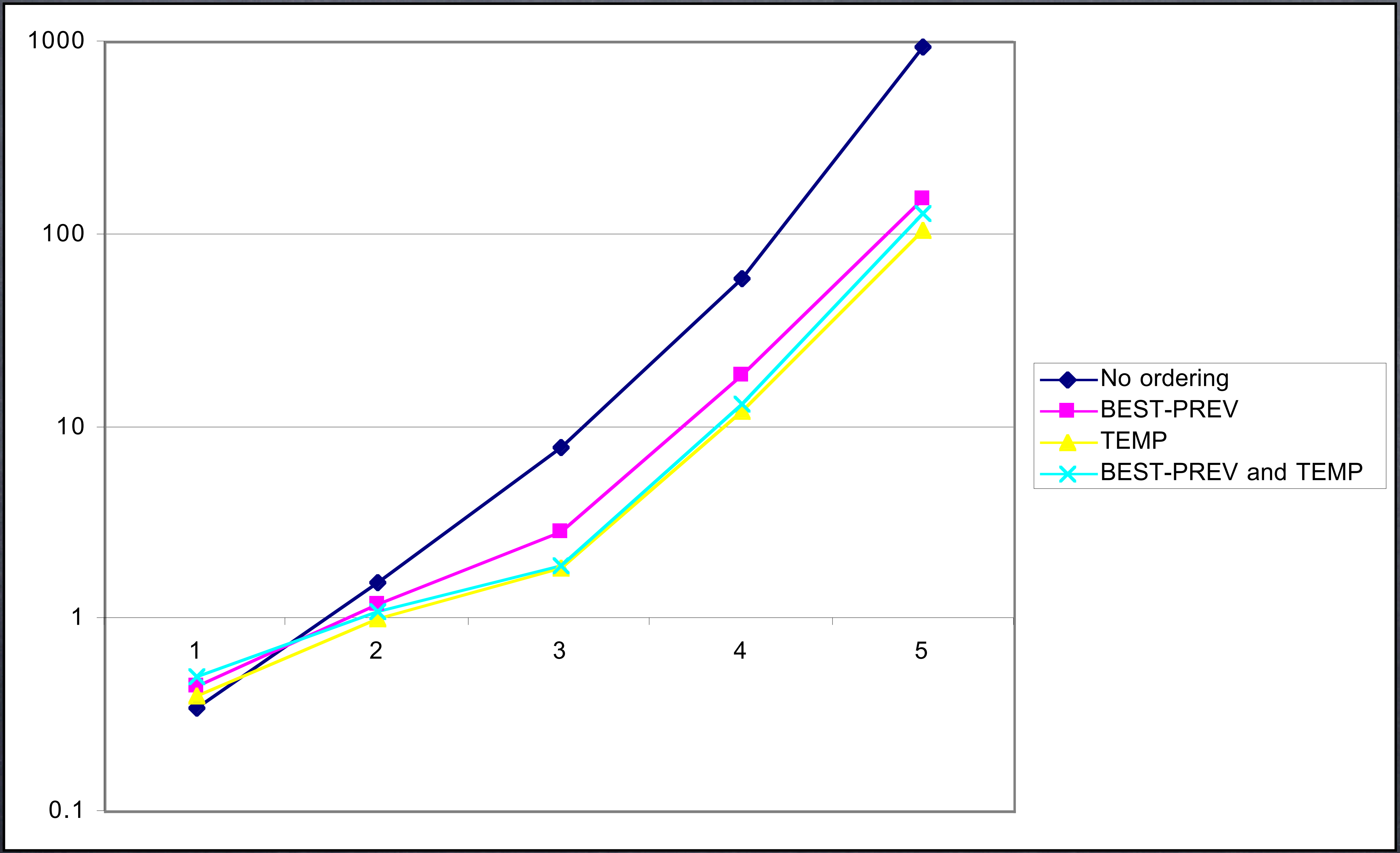
horizontal: number of subgames

vertical: time (log-scale)

TEMP is best!



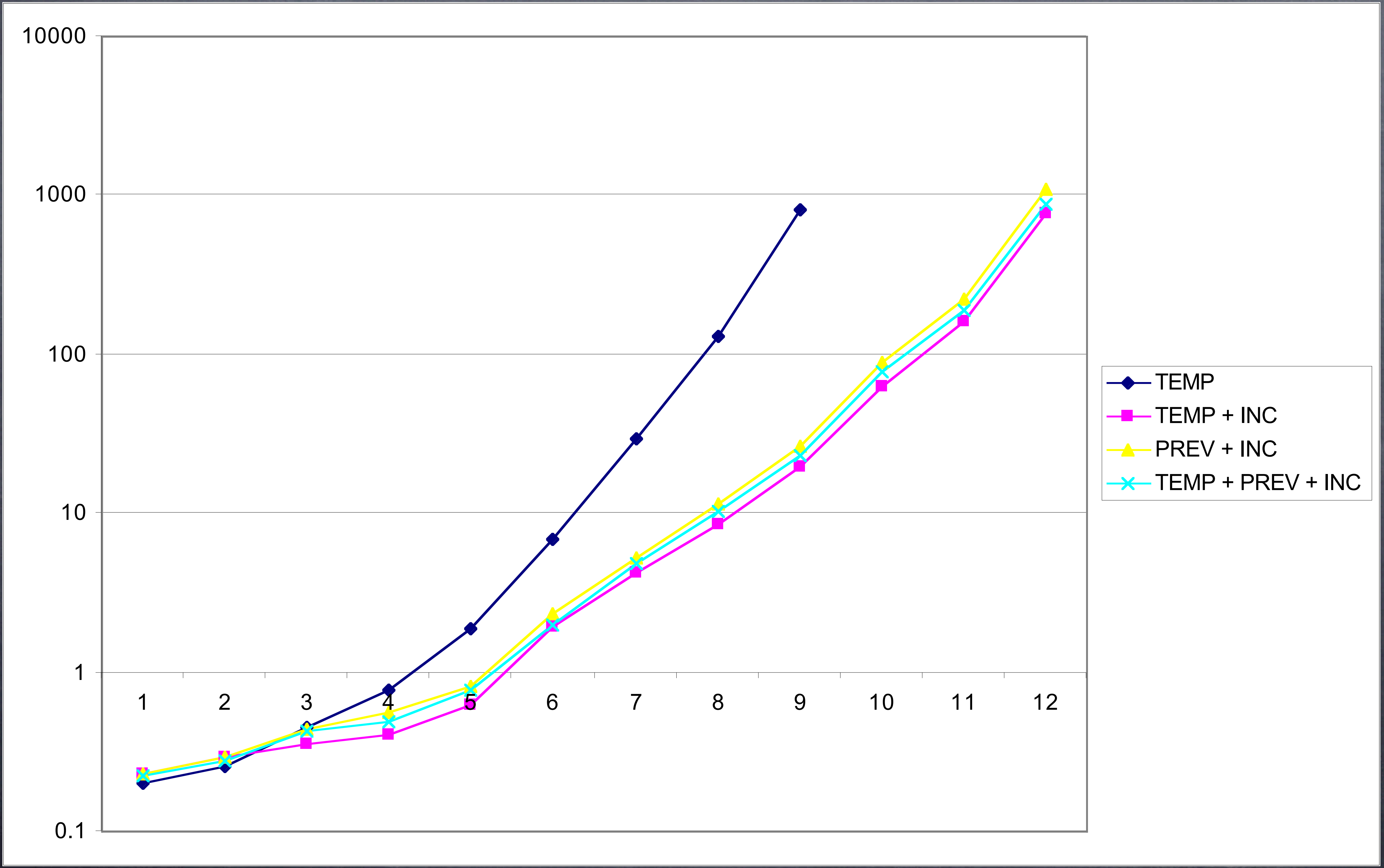
3-level Games



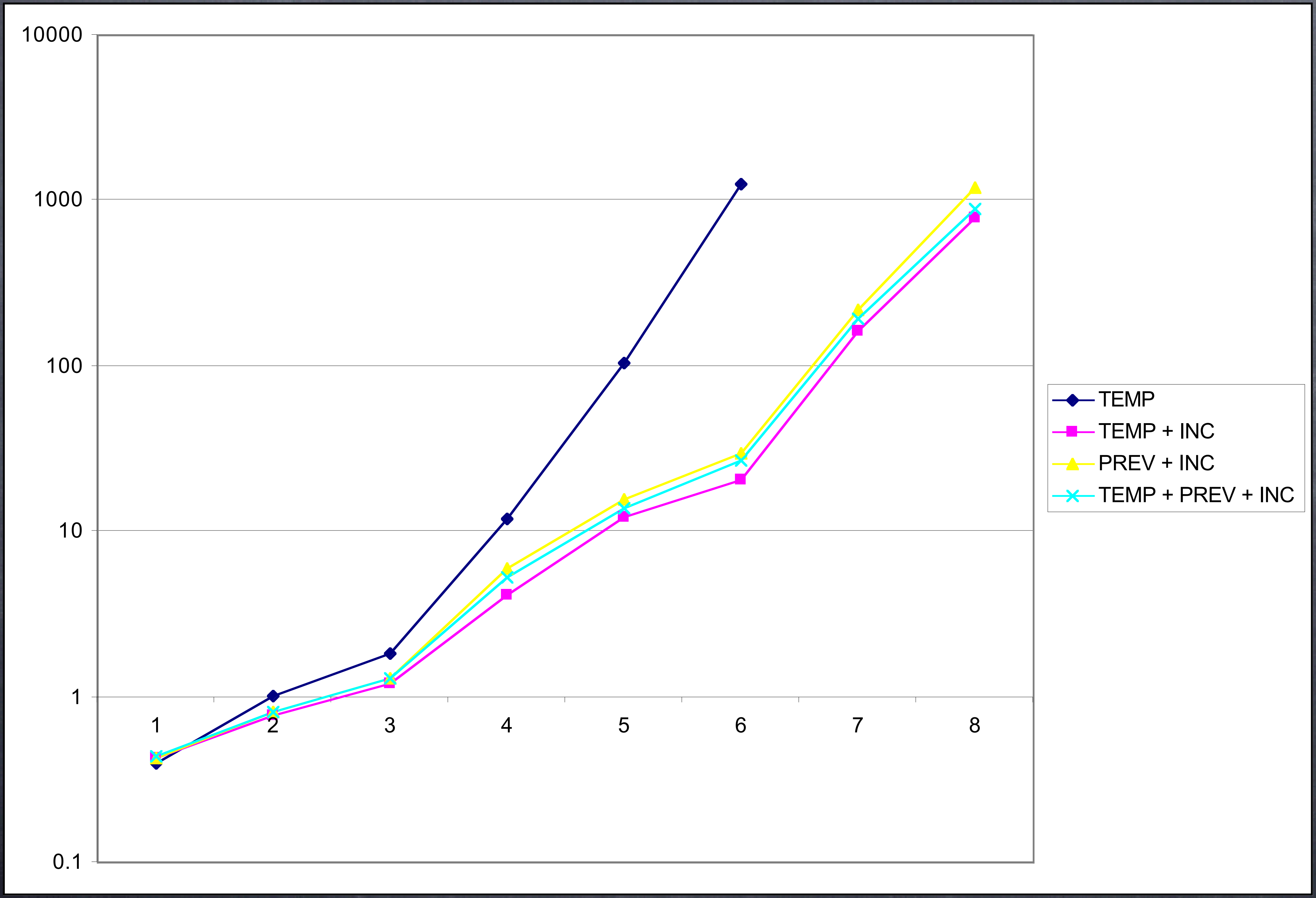
Experiment 2

- Exact search
- Move pruning
  - Compute incentives of moves
    - Can be computed locally!
  - Prune moves with dominated incentive
    - Pruning on global level

2-level Games



3-level Games

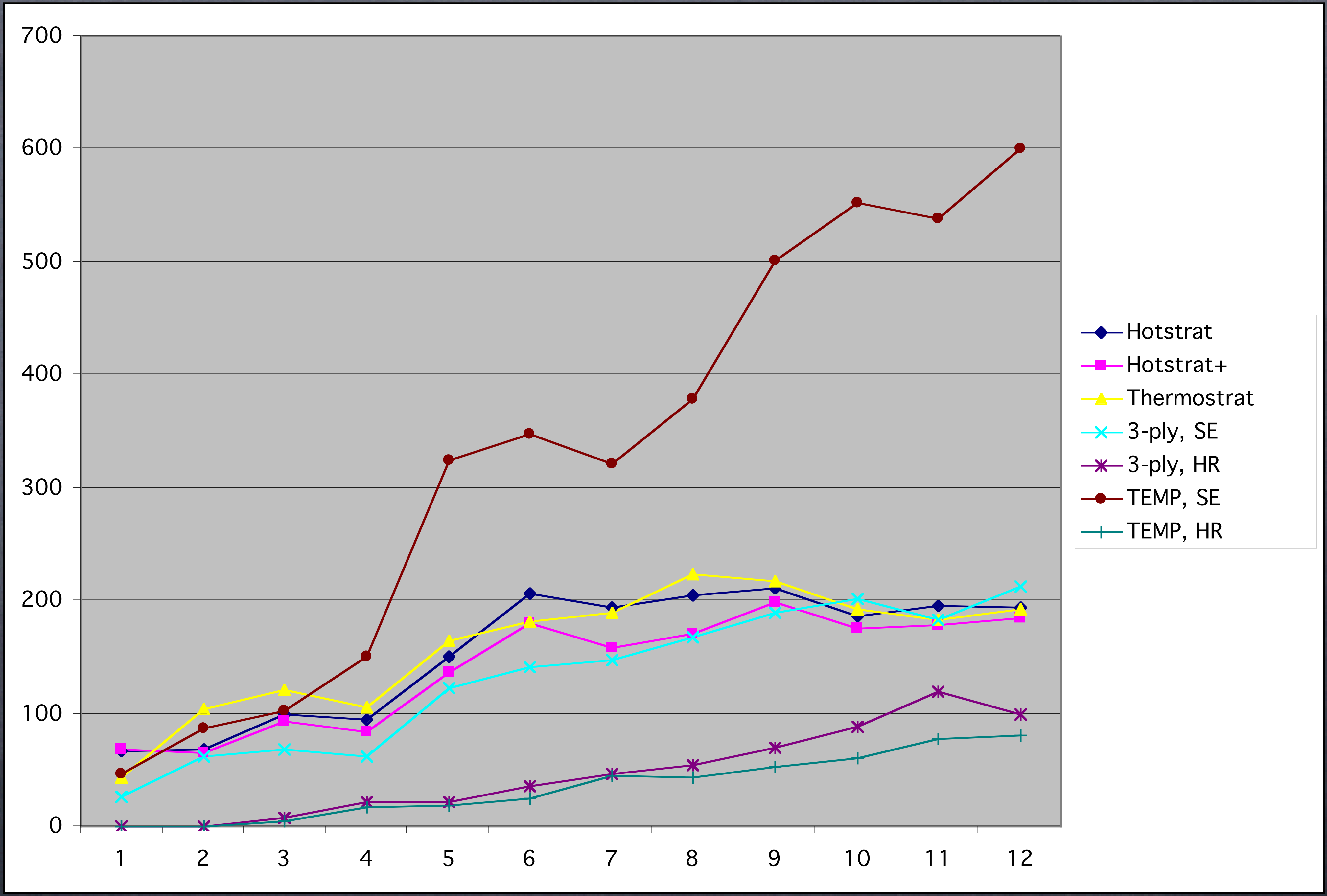


Experiment 3

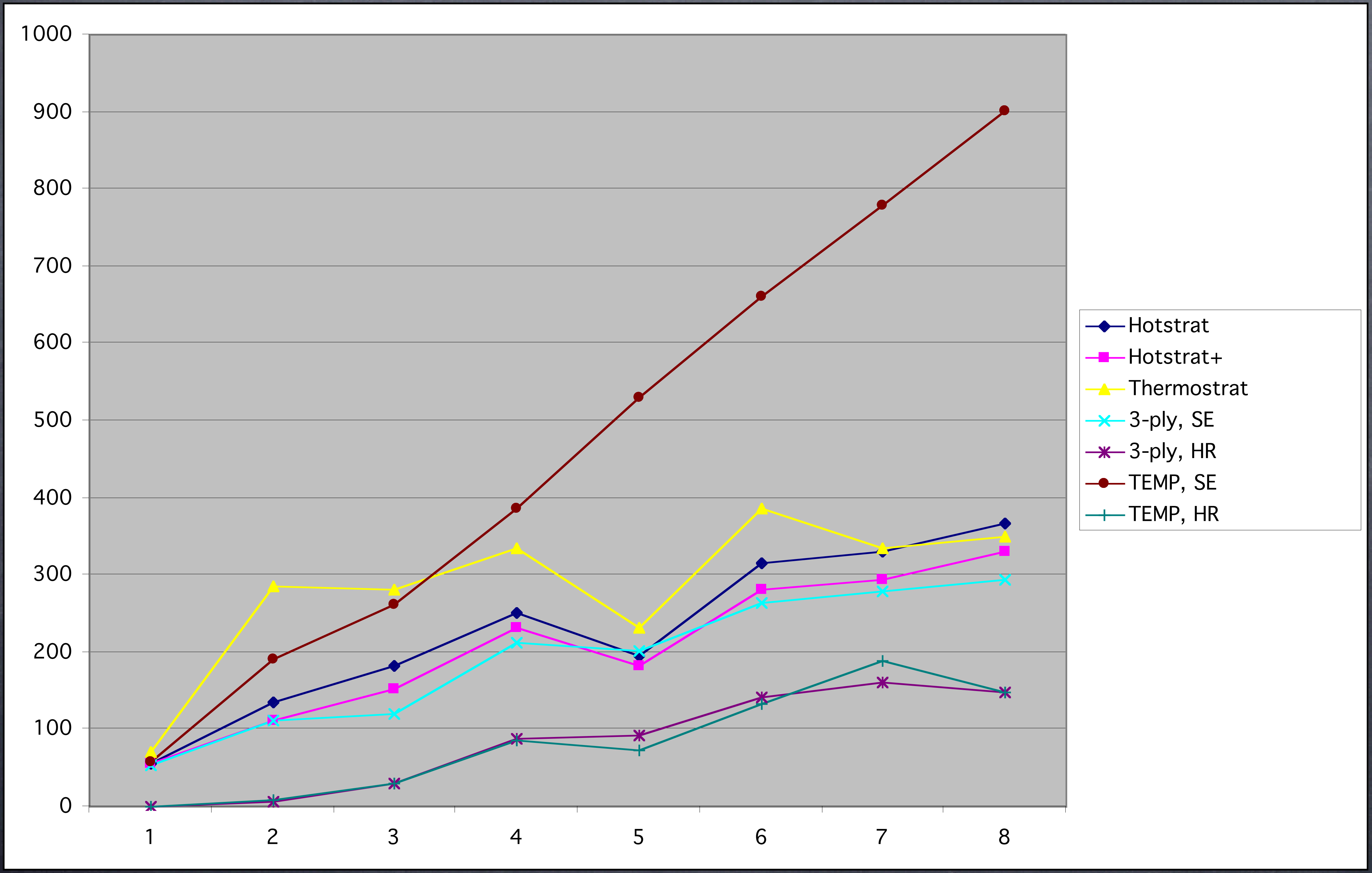
- Heuristic search
- Tried two resource-limited searches
  - depth limit (d=3 here)
  - temperature limit (t = 0.8 * tmax)
- Two evaluation functions
  - Sum-of-means
  - Hotstrat rollouts

2-level Games

Error



3-level Games



Experiment 4

- Similar to Experiment 3
- Measure the error relative to time used
- Result: simple is best!
  - Depth-bounded search
  - Sum-of-means evaluation

Conclusions

- Developed and tested search methods for sums of hot games
- Move ordering by temperature and pruning using incentives are very effective
- Heuristic search: hotstrat rollouts reduce the error, but are expensive
- Best time-error tradeoff: depth-bounded search, sum-of-mean evaluation
- Much room for further research