Learning and Using Hand Abstraction Values for Parameterized Poker Squares

Colin Messinger & Zuozhi Yang

Todd W. Neller

Gettysburg College

Structure of the AI Player

Static Evaluation

Key Techniques:

- Partial Hand Abstractions
- Monte Carlo Methods
- ϵ-Greedy Strategy

Naïve Partial Hand Abstraction

- Why naïve? – Assume that each row/column is independent
- Key Features:
  - number of cards
  - row (“-”) or column (“|”)?
  - rank counts and how many cards undealt in the rank
  - flush (“f”) is achievable?
  - straight (“s") is achievable?
  - royal flush (“r") is achievable?
- For example, “14|1(3)1(2)1(2)f3(8)s” represents a column hand abstraction after the 14th move.

Monte Carlo and ϵ-greedy

- Play a lot of games during 5 min of training
- Exploration versus Exploitation
  - Exploration: random plays
  - Exploitation: greedy plays
- ϵ is the proportion of random plays
- On-policy learning
  - constantly update the policy that is used to make decisions

Real-Time Search

- Colin’s goal: Utilize George’s heuristic evaluation of PPS game states to design a limited real-time search that approximates optimal play.
- Two main design decisions:
  - Time Management
  - Search Algorithms

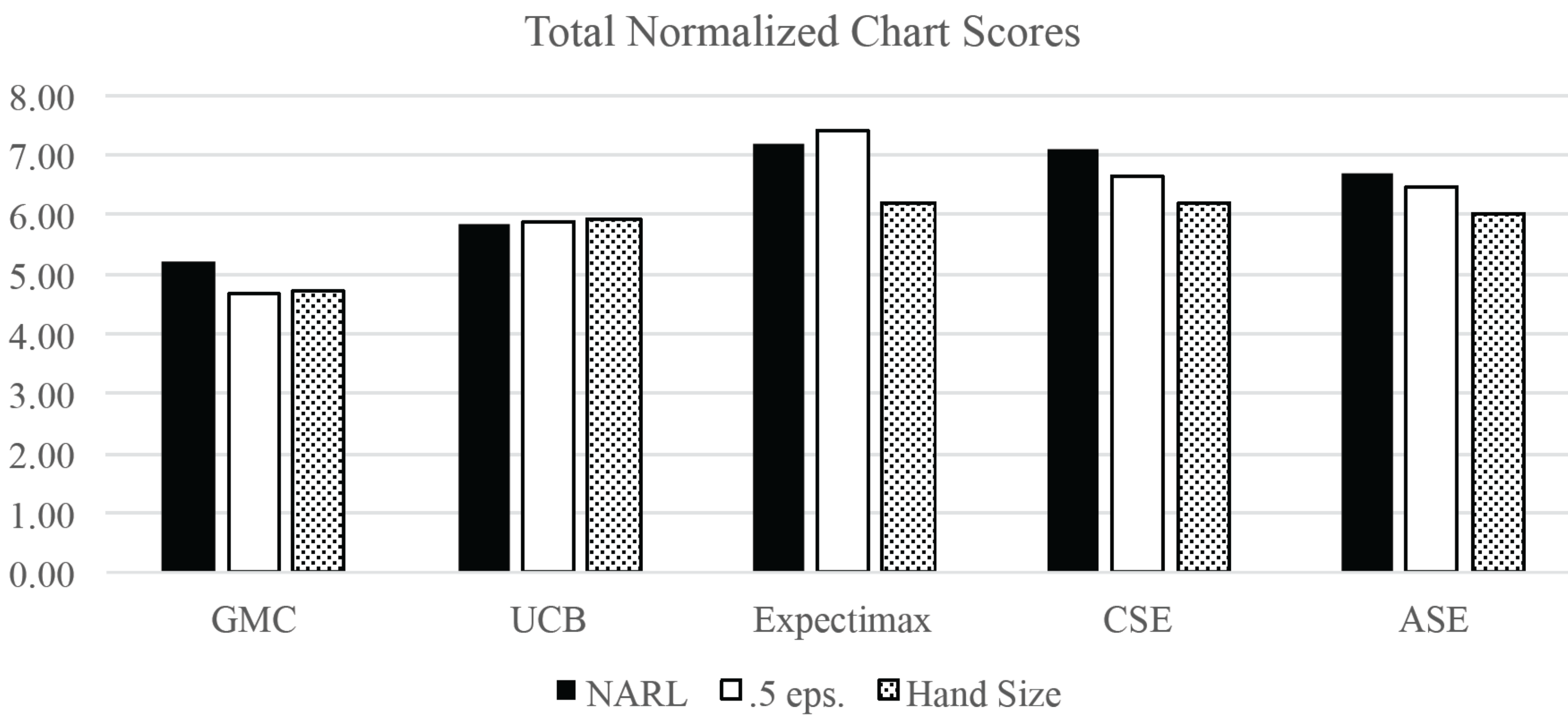
Time Management

- Sometimes, it is beneficial to distribute reasoning time non-uniformly in real-time play
- Comparison:
  - Uniform Time Management (UTM)
  - Gaussian Time Management (GTM)
    - more middle-game time concentration
    - 4 sets of means and standard deviations
- UTM dominated GTMs for almost all point systems

Search Algorithm

- Three approaches tried
  - Greedy Monte Carlo (a.k.a. Flat Monte Carlo), depth limit 5
  - UCB1 Monte Carlo, depth limit 5
  - Expectimax, depth limit 2
    - Chance-Sampled
    - Action-Sampled
- Expectimax was found to have the best performance

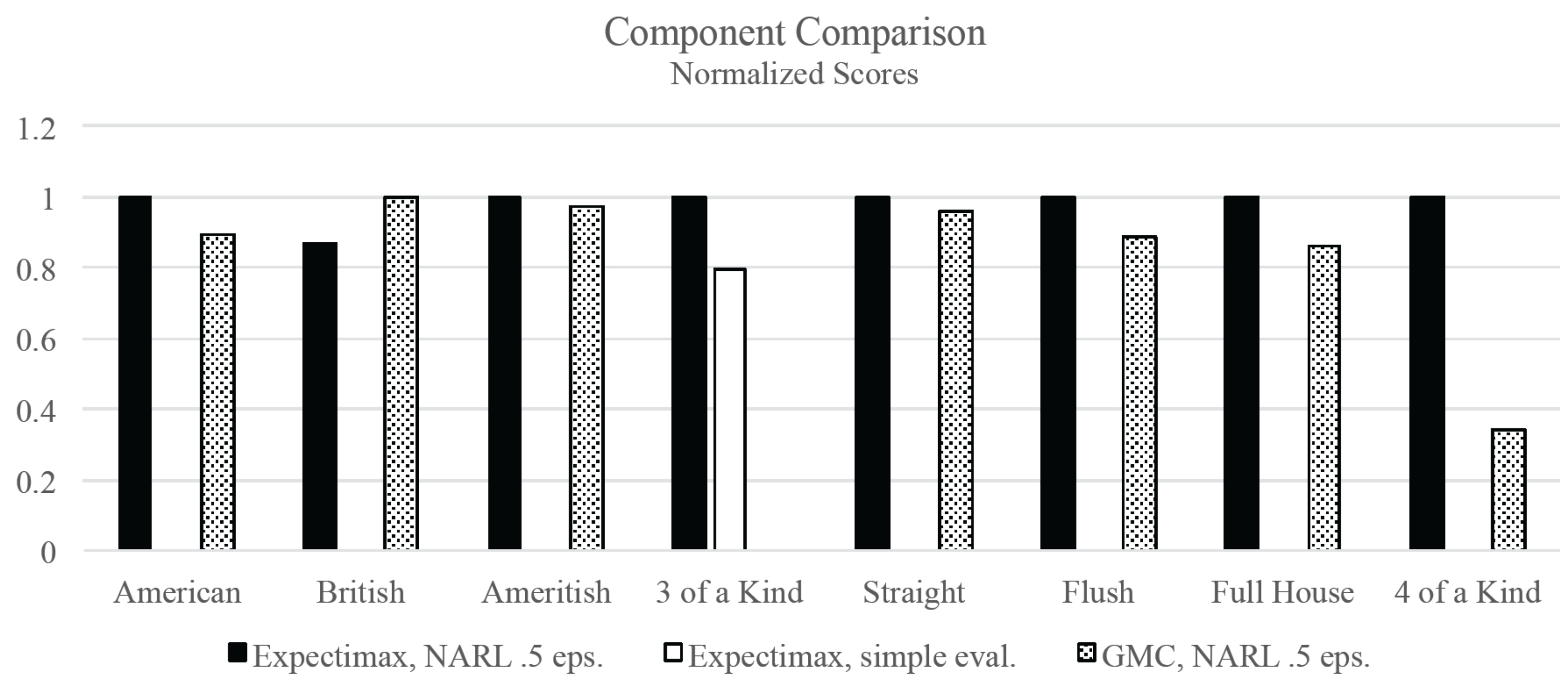
Search Algorithm Results



Comparison of Design Components

- We performed component significance testing by comparing performance of players with trivialized components:
  - Expectimax search, 0.5 initial epsilon NARL evaluation (GettysburgPlayer)
  - Expectimax search, *simple score evaluation*
  - *Greedy Monte Carlo search*, 0.5 initial epsilon NARL evaluation
- Heuristic evaluation function was most significant

Component Significance Results



Future Research

- Argument for improving heuristic evaluation:
  - It was most significant to our performance, so greater performance may come most easily here.
- Argument for improving search:
  - Expectimax with depth limit 2 only utilized 9.4% of our allotted play decision time, so we can surely leverage the unused 90.6% (e.g. Monte Carlo Tree Search with parallelization).