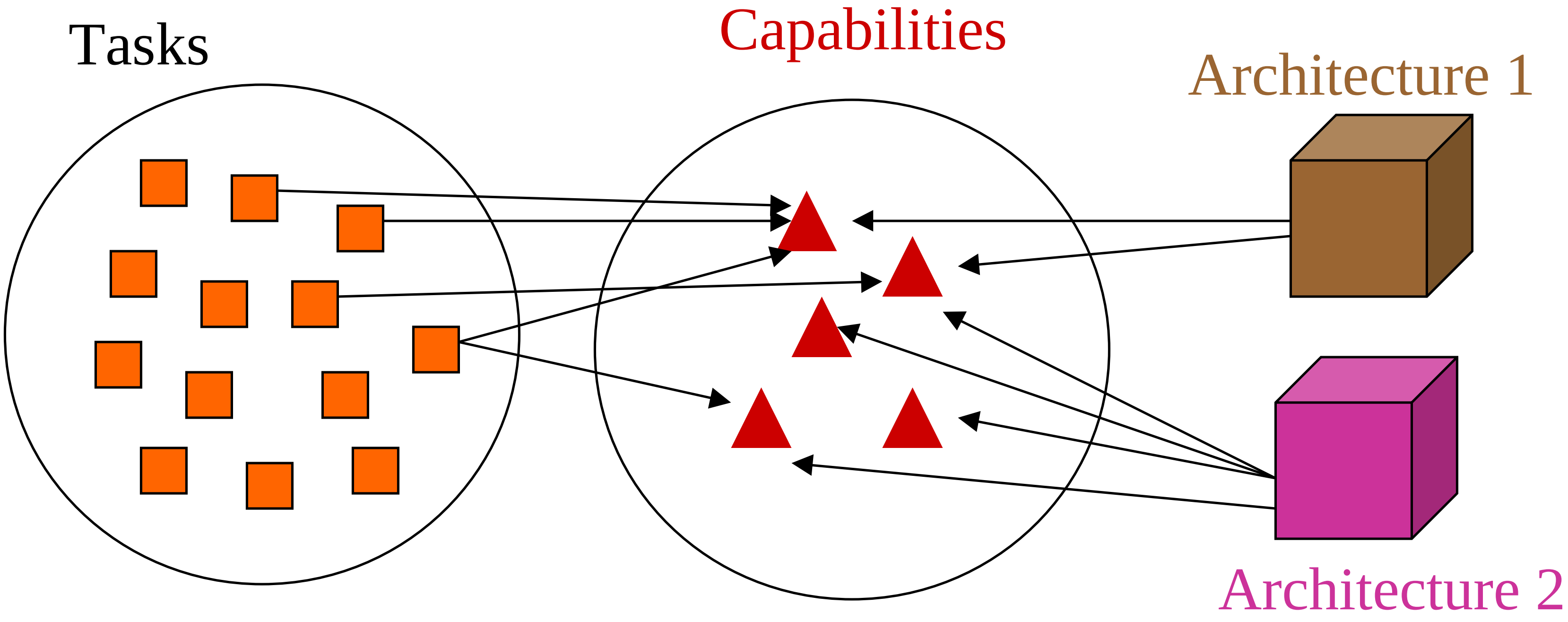
Toward a Methodology for AI Architecture Evaluation: Comparing Soar and CLIPS

Scott Wallace & John Laird

May, 1999

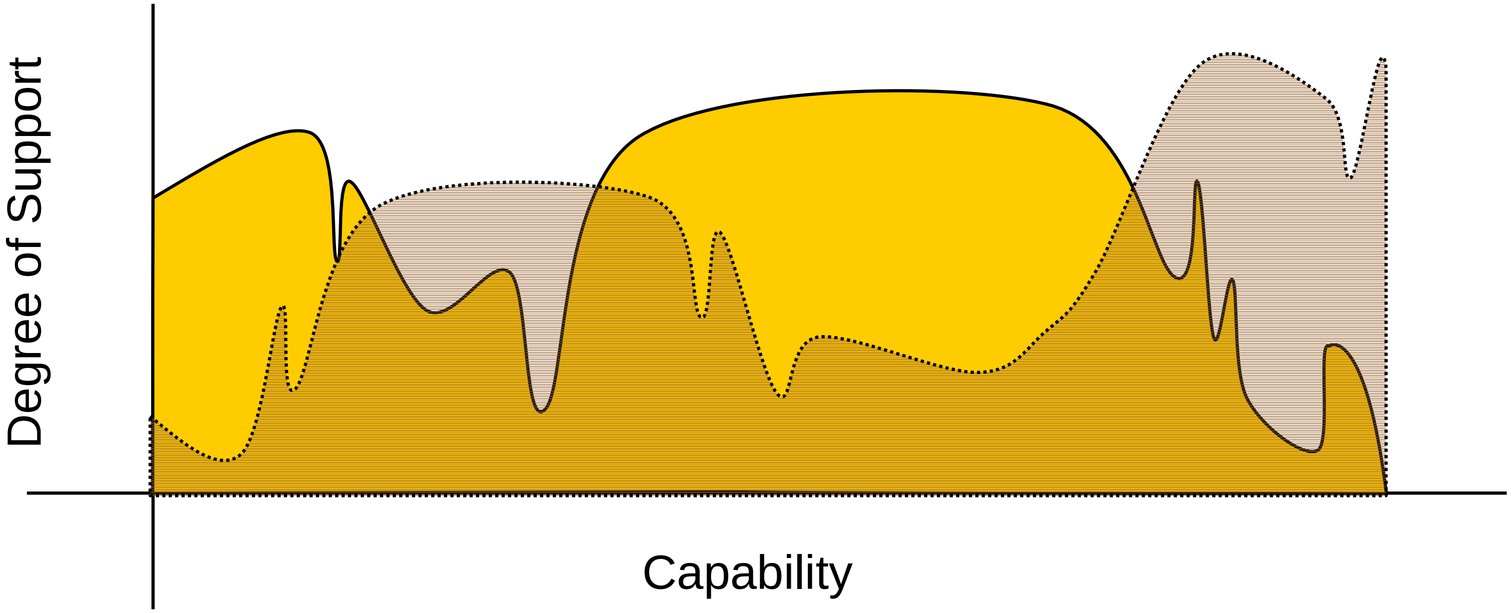
Architectures, Tasks and Capabilities

- What capabilities are required by the task?
- What capabilities are supported by the architecture?



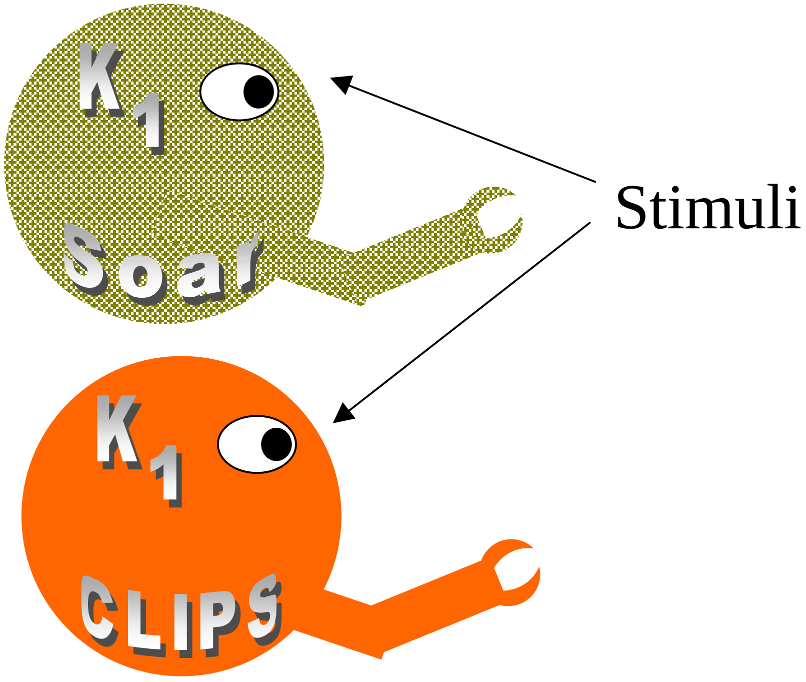
Result

- Performance profiles of architectures and tasks



Methodology

- Degree of support:
  - Runtime performance
  - Number of rules
- Create comparable agents to evaluate a range of capabilities
- Maintain knowledge consistency
  - Detailed specification of agent's internal and external behavior
- Ensure agents are exposed to same stimuli
- Examine multiple architectural mechanisms
  - Create an array of agents which exhibit identical external behavior, but differ internally



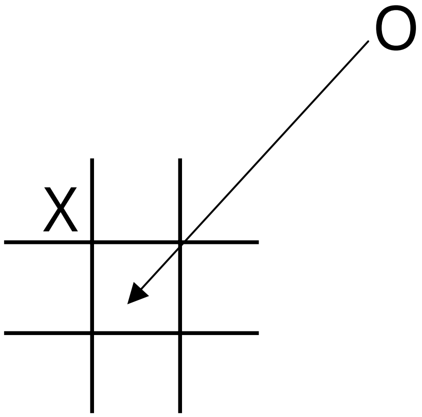
Examining Decision Making Capabilities

- Mutually Exclusive Actions
- Segregation of Control Knowledge
- Two Phase Decision Process
- Three Phase Decision Process
- Goal Constrained Selection
- Utility Based Decision Process
- Analogy Based Decision Process
- ...

Mutex Actions

- Actions are only considered when they when should be invoked

Proposal & Selection



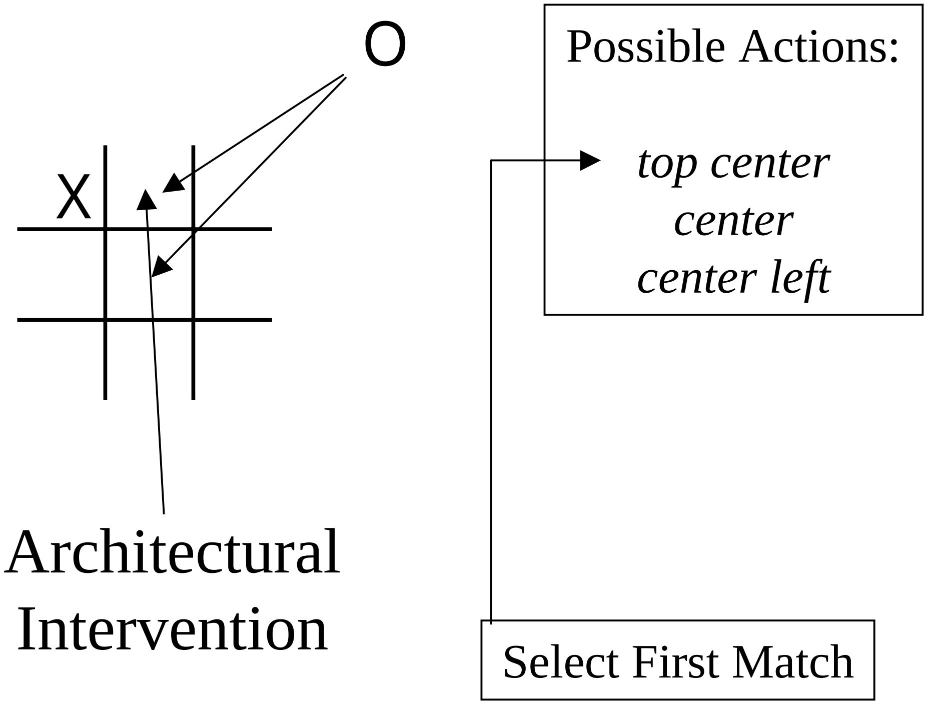
Preconditions:
X in upper left all others empty

Effects:
Put O in center

Segregated Control

- Actions may be considered even when they are not invoked, but an architectural mechanism ensures the proper action is always taken

Proposal & Selection



O

X

Architectural Intervention

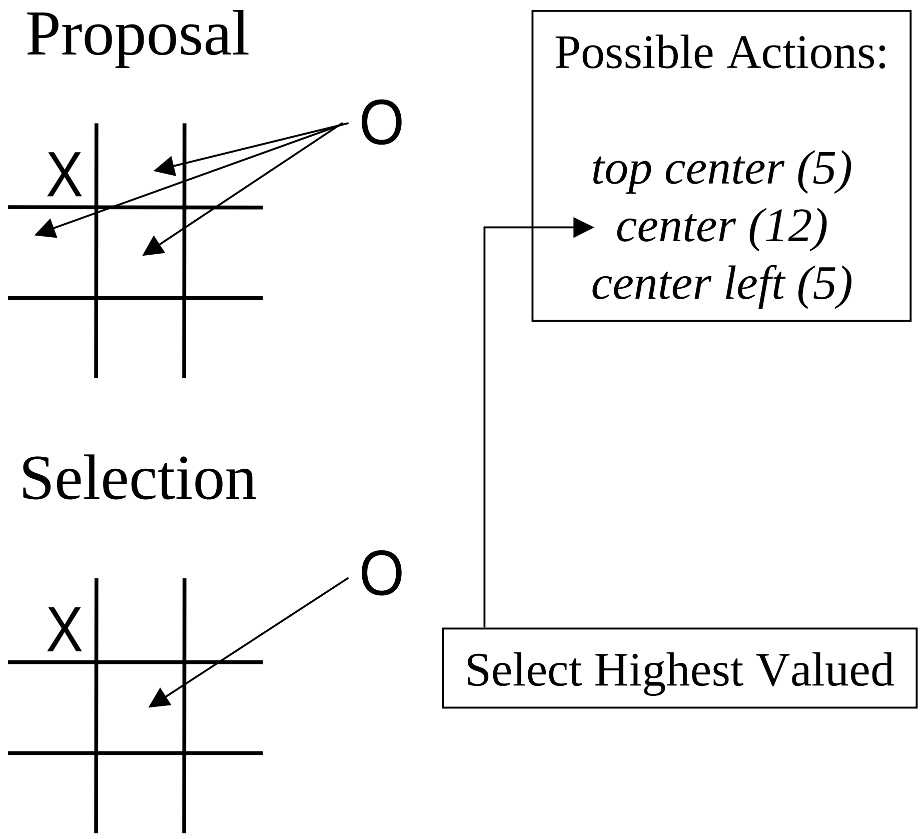
Possible Actions:

top center
center
center left

Select First Match

Two Phase Process

- Proposal and Selection take place asynchronously.
- Proposal involves creating symbols represent each action
- Selection pursues one action



Proposal

Possible Actions:

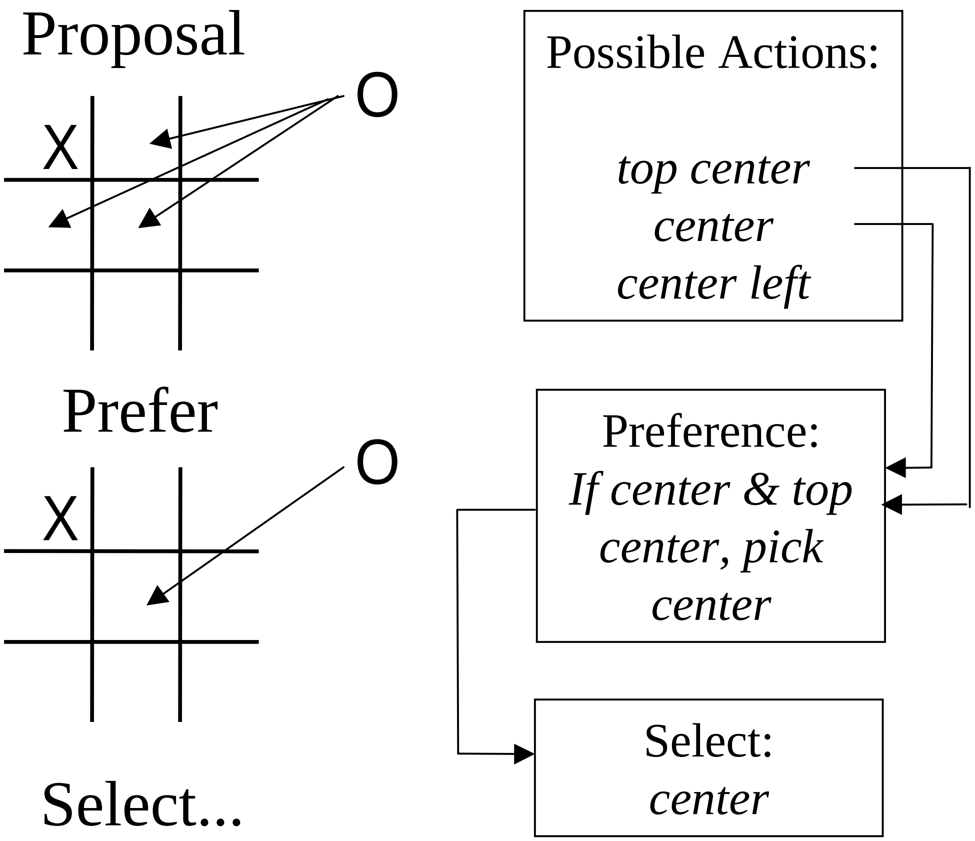
top center (5)
center (12)
center left (5)

Selection

Select Highest Valued

Three Phase Process

- Three distinct phases occur
  - Proposal
  - Preference
  - Selection
- Preferences can be stated in terms of other actions under consideration



Possible Actions:

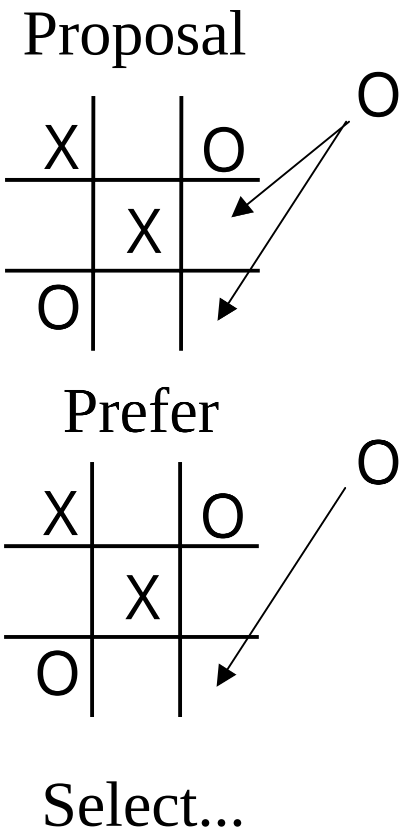
top center
center
center left

Preference:
If center & top center, pick center

Select:
center

Goal Constrained

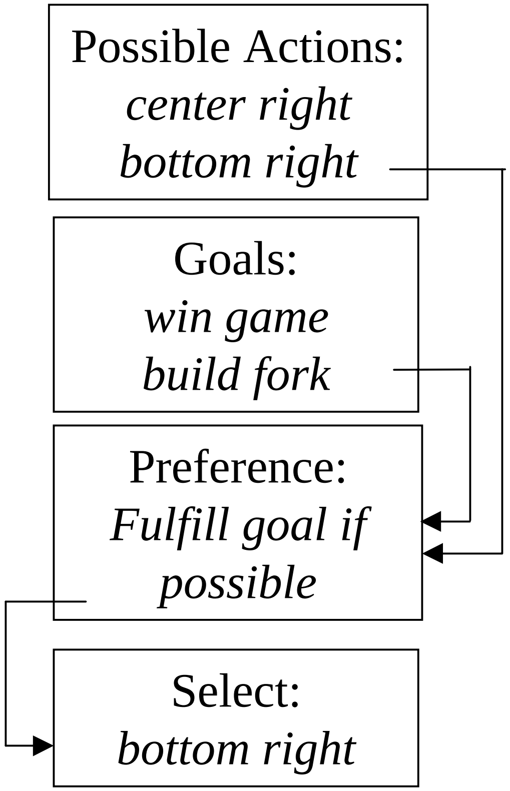
- Consider actions even if they cannot be invoked in the current situation
- Currently proposed actions as well as selected goals can be used for preferences
- Build goal list of selected actions



Proposal

Prefer

Select...



Possible Actions:
center right
bottom right

Goals:
win game
build fork

Preference:
Fulfill goal if possible

Select:
bottom right

Architectural Properties

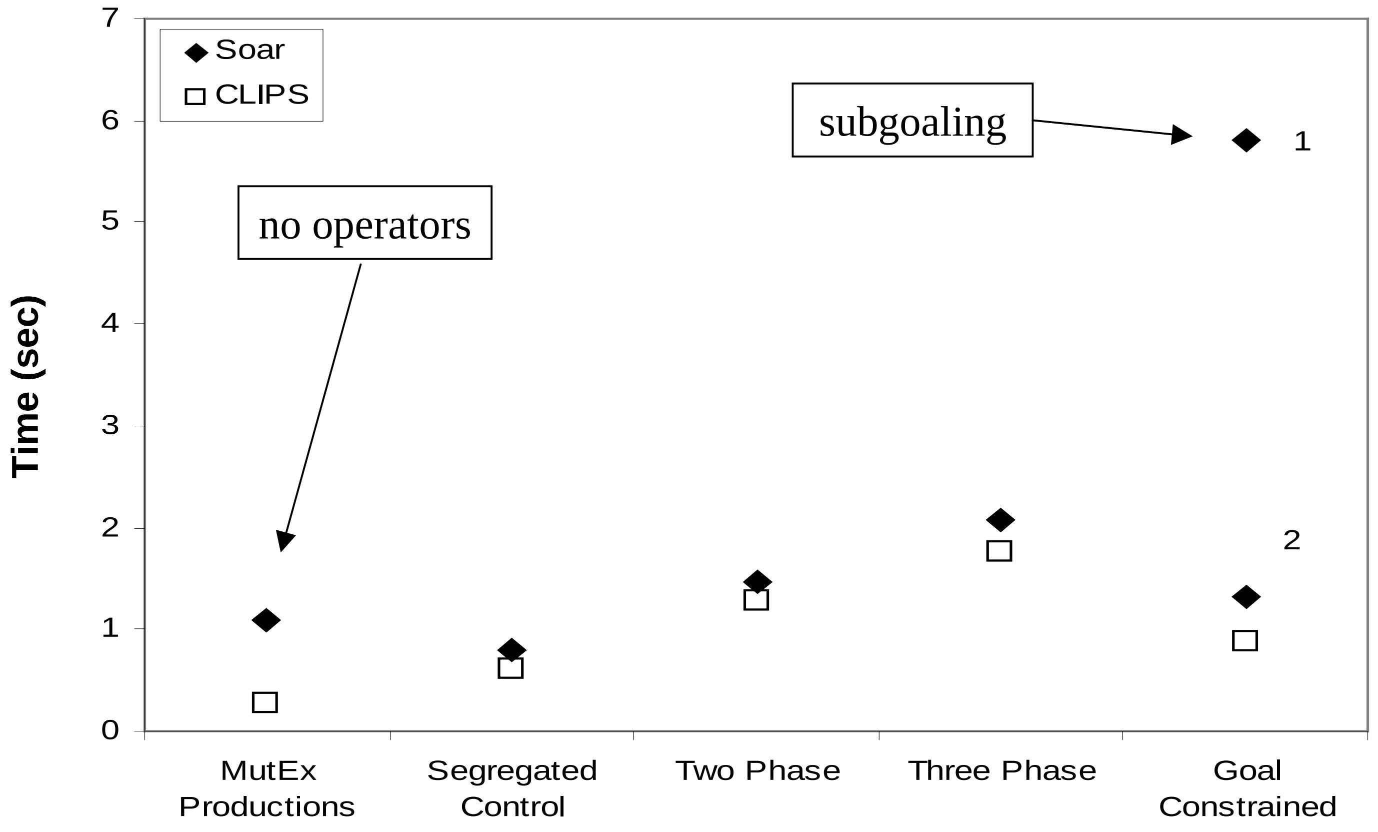
CLIPS

- Short term memory -- a list of facts
- Rules are fired serially, matches recalculated after each firing
- Flow control:
  - Rule matching
  - Salience
  - Search strategy

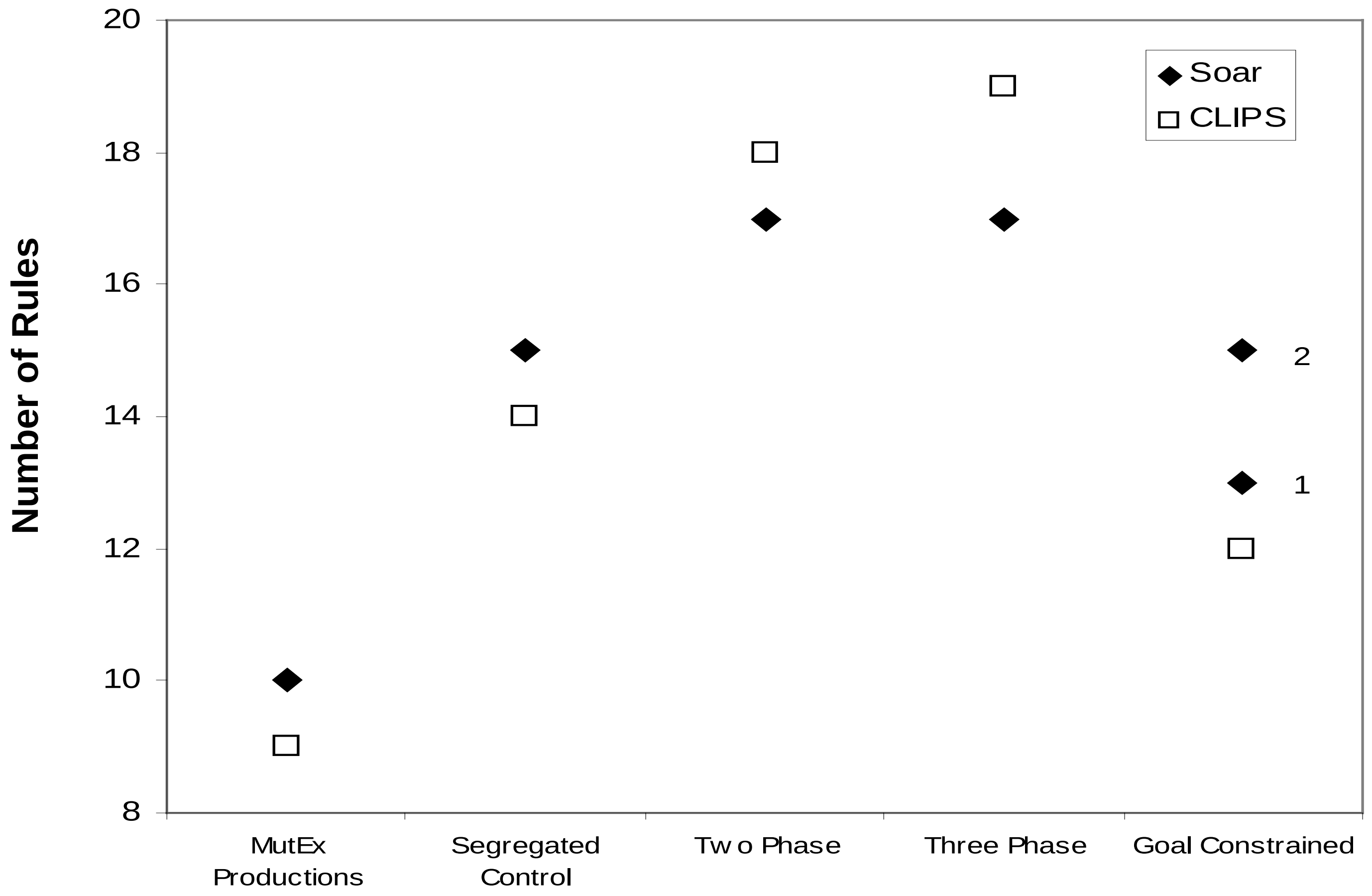
Soar

- Short term memory -- directed graph
- Rules are fired in parallel, matches recalculated after each elaboration cycle
- Flow control:
  - operators
  - preferences
  - 3 phase decision cycle

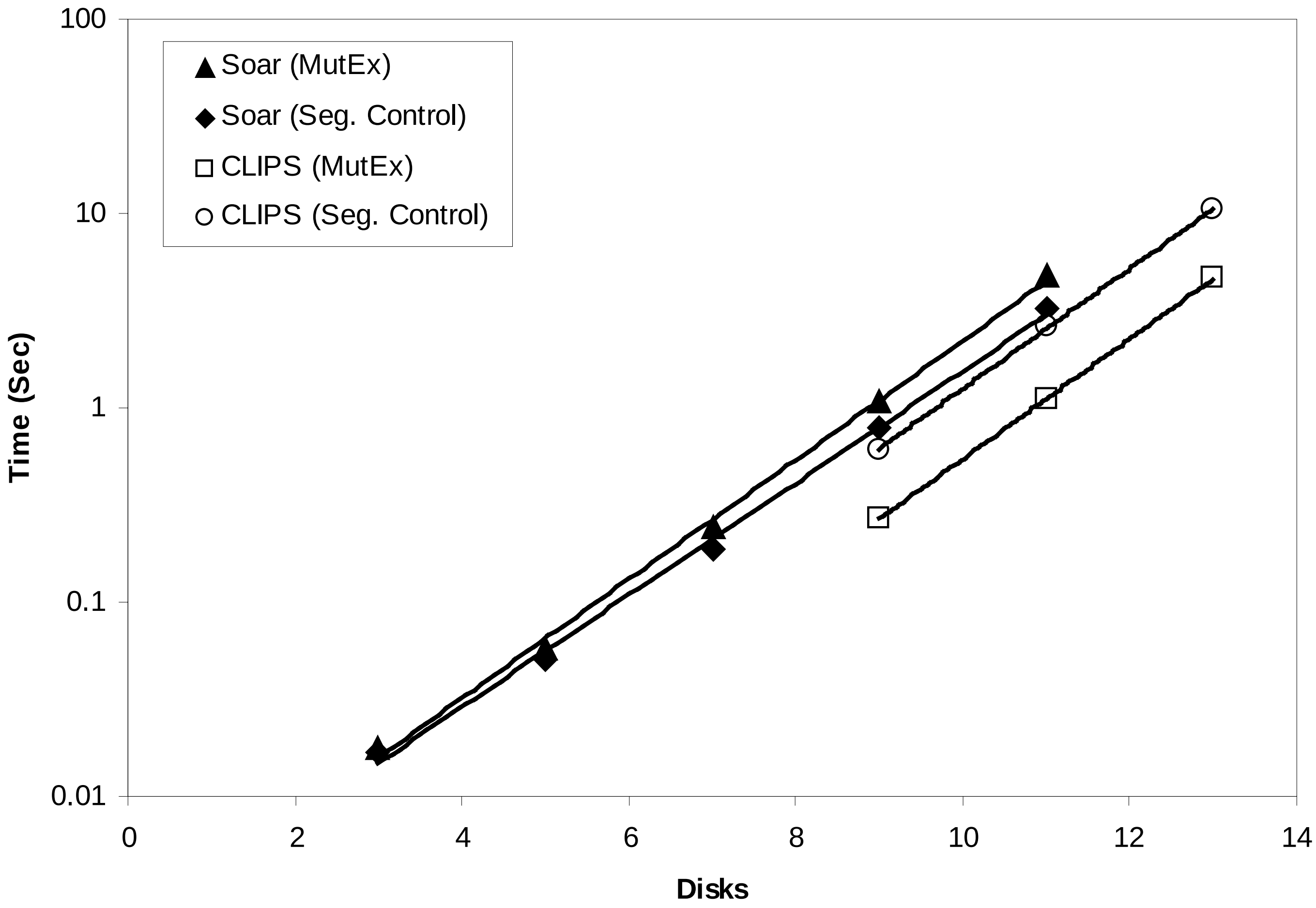
Runtime Performance in Towers of Hanoi



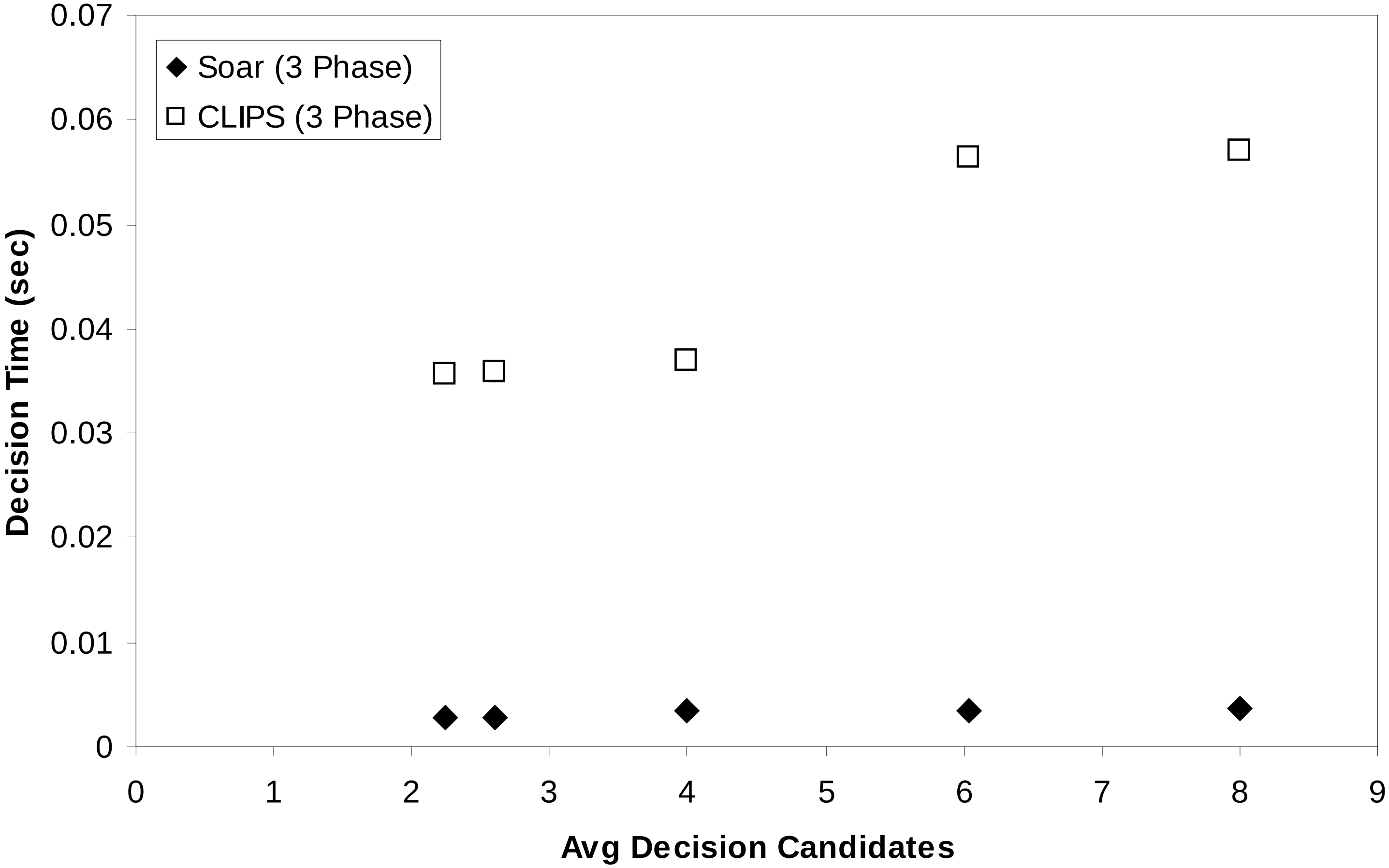
Number of Rules in Towers of Hanoi



Scaling in ToH



Eaters



0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
◆ Soar (3 Phase)
□ CLIPS (3 Phase)
Decision Time (sec)
0
1
2
3
4
5
6
7
8
9
Avg Decision Candidates

Nuggets & Coal

- Nuggets
  - Further evidence that Soar performs well compared to other architectures
  - Yielded leads for further performance improvments
- Coal
  - Difficult to prepare architectures for benchmarking
  - Hard to determine how results gained from this method apply to architectural theories
  - 5 capabilities is just a start

Next Steps & Future Work

- Why does Soar’s subgoaling mechanism seem so slow?
- What specific architectural attribute does the nonlinearity in CLIPS’s performance trace back to?
- Why does CLIPS perform so much slower than Soar in Eaters, but not in ToH?
- What other tasks can corroborate our hypothesis?
- Increase breadth of tasks
- Examine architectures which have less common elements