



EFFICIENTLY IMPLEMENTING EPISODIC MEMORY IN SOAR

Nate Derbinsky

Advised by John E. Laird

WHAT IS EPISODIC MEMORY?



- A long-term, contextualized store of specific events
- Comparable to CBR with a large, highly generalized, case database

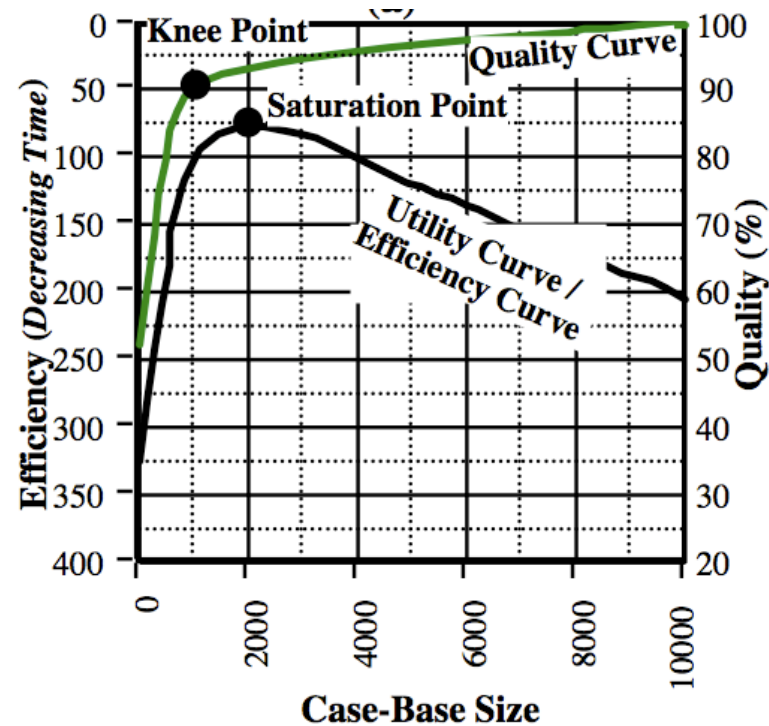
Tulving, E. (1983). Elements of Episodic Memory.

NUXOLL: AGENTS WITH EPISODIC MEMORY

- Explored cognitive capabilities gained by virtue of an episodic memory
 - Action modeling, virtual sensors, ...
- Developed variables defining the space of episodic memory implementations
 - Encoding
 - Storage
 - Retrieval

UTILITY OF EPISODIC MEMORY

- Episodic memory can be crucial to an agent's efficacy
- The functional specification may lead to a monotonically increasing store



Smyth, B. and Cunningham, P. (1996).
The Utility Problem Analysed: A Cased-Based Reasoning Perspective.

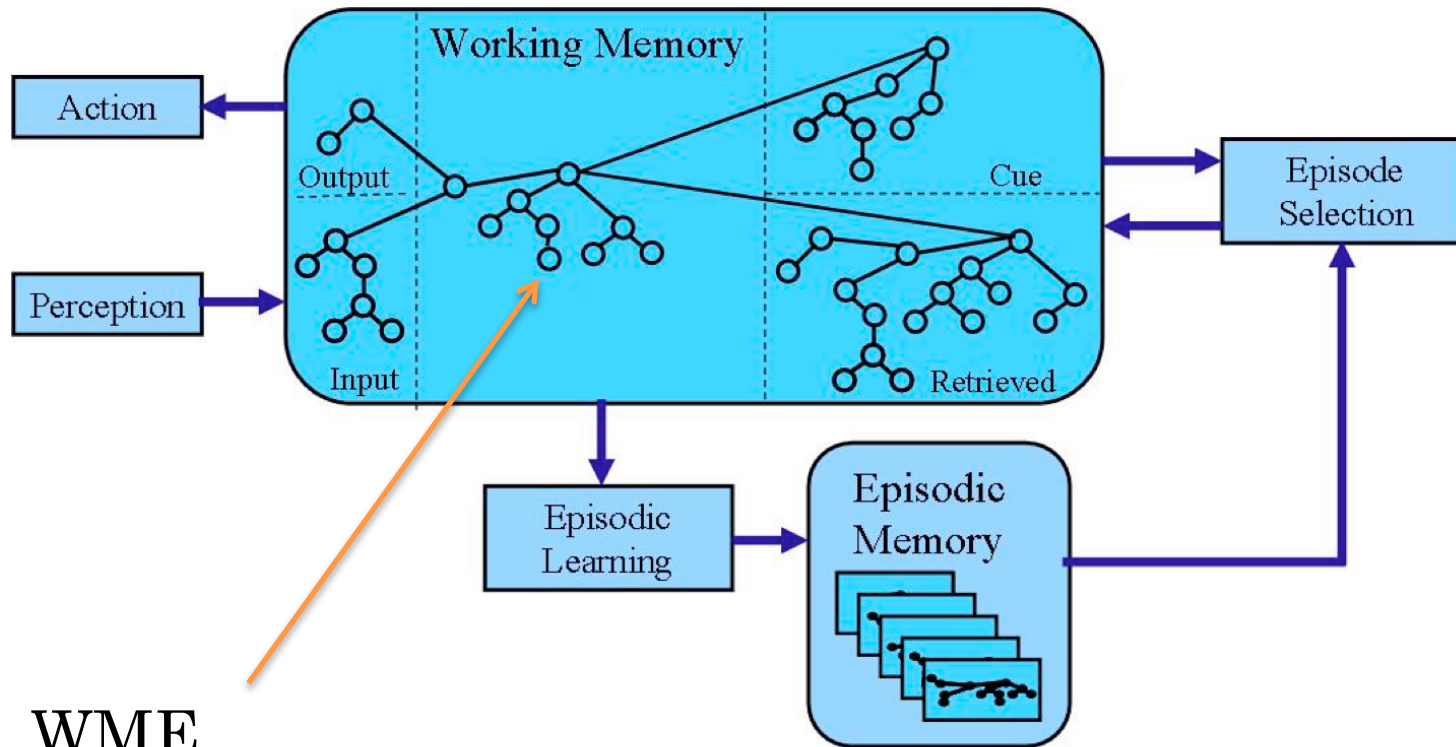
PROBLEM STATEMENT

- How to efficiently implement task-independent episodic storage and retrieval over long agent lifetimes in the Soar cognitive architecture

OUTLINE

- Soar-EpMem
- Research Process
 - Methodology
 - Iterative Process
 - Final Results
- Discussion
 - Future Work

INTEGRATING EPISODIC MEMORY IN SOAR



WME

(id ^attrib value)

Nuxoll, A. and Laird, J. (2007). Extending Cognitive Architecture with Episodic Memory.

CORE EPISODIC OPERATIONS

○ Storage

- Automatically triggered
- WM => Store
- Indexing Method: associate WMEs with temporal id

○ Cue-Based Retrieval

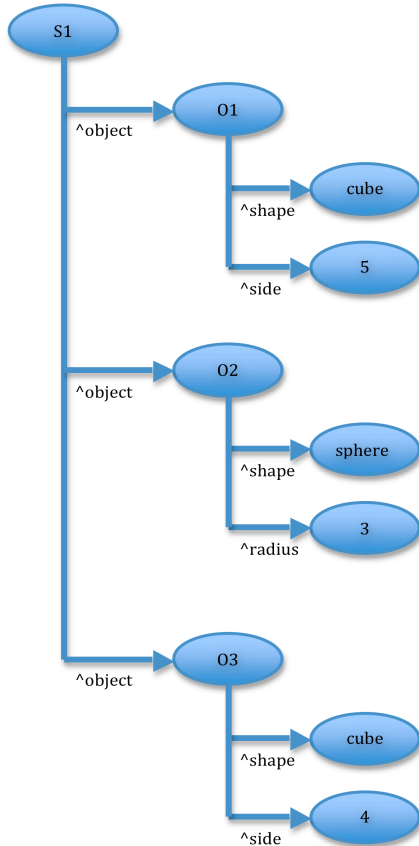
- Agent deliberately constructs a retrieval cue (potentially with optional modifiers)
- Search store for candidates, rank with respect to cue
- Reconstruct best match => WM

○ Non-Cue-Based (NCB) Retrieval

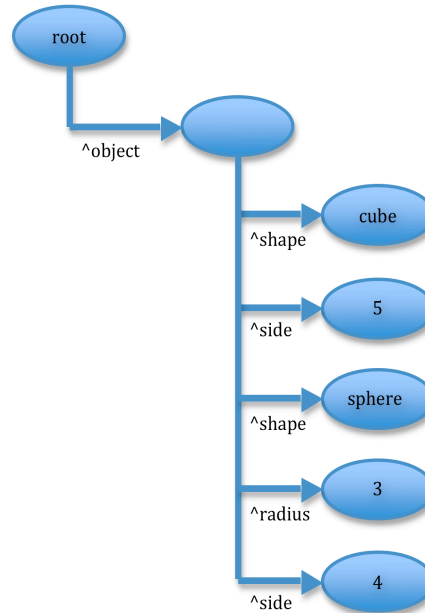
- Agent deliberately requests episode by temporal id
- Reconstruct episode => WM

WORKING MEMORY TREE

Working Memory

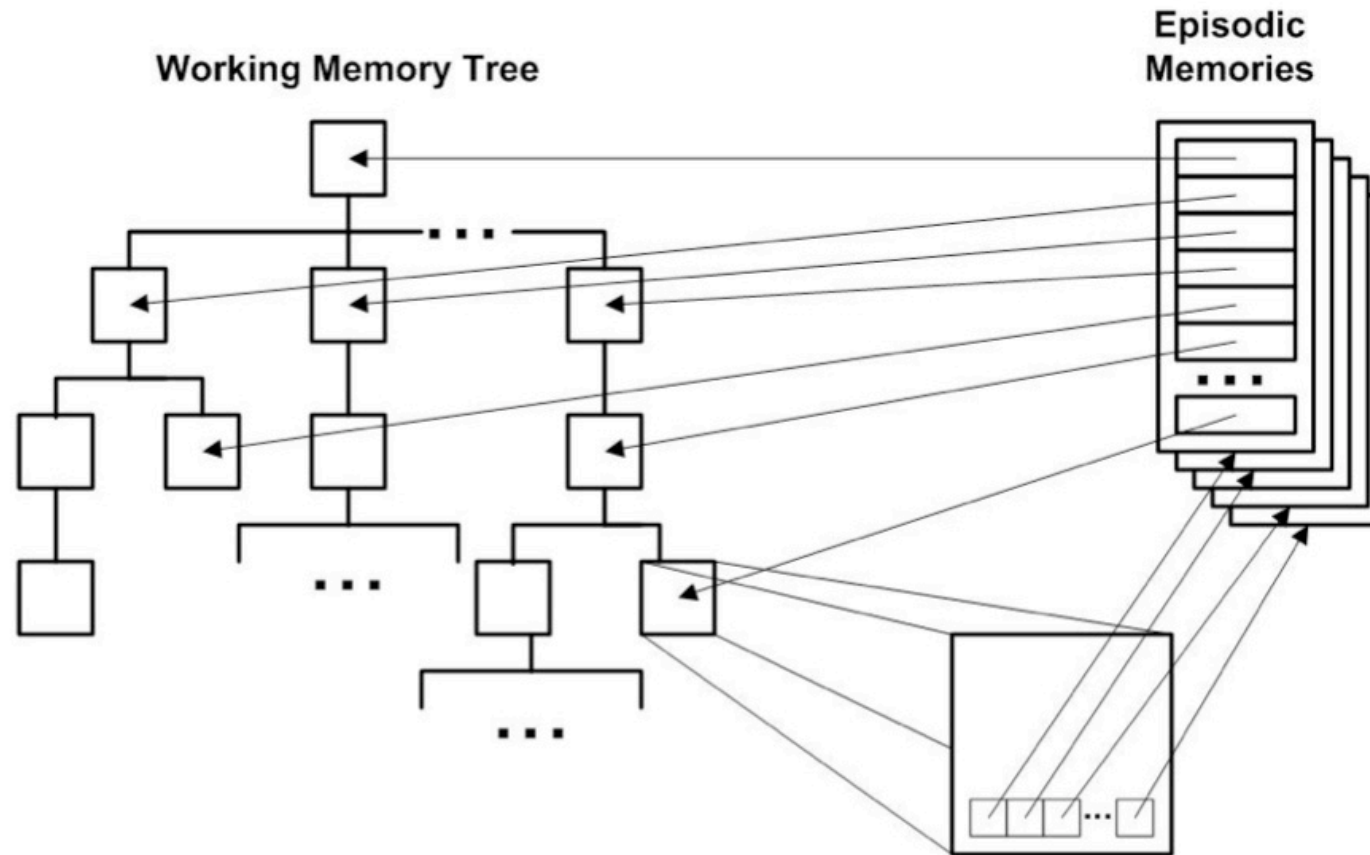


Working Memory Tree



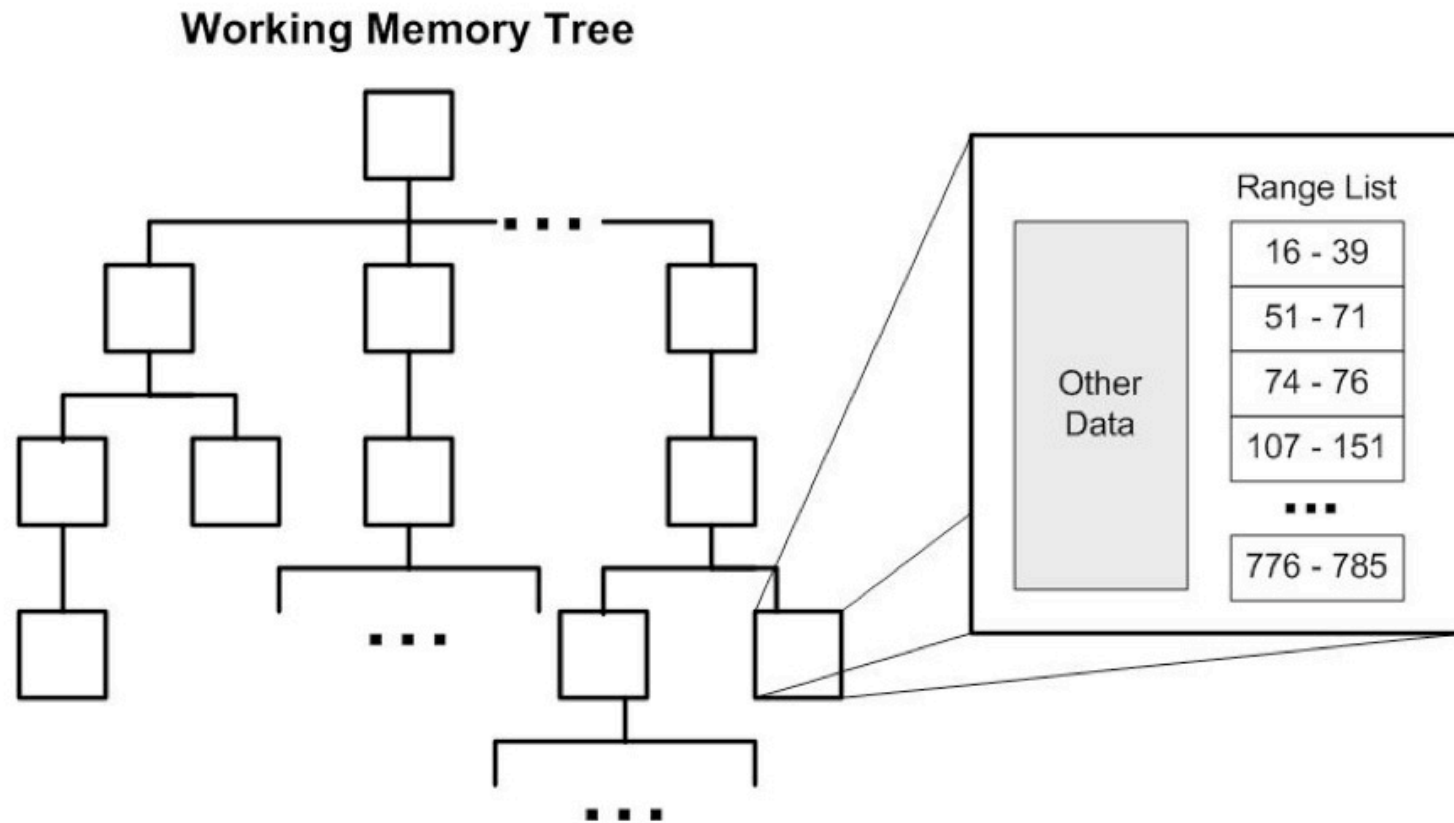
Indexing Method. A scheme for associating nodes of the Working Memory Tree (WMT) with a temporal id.

INSTANCE INDEXING METHOD



Nuxoll, A. and Laird, J. (2007). Extending Cognitive Architecture with Episodic Memory.

RANGE INDEXING METHOD

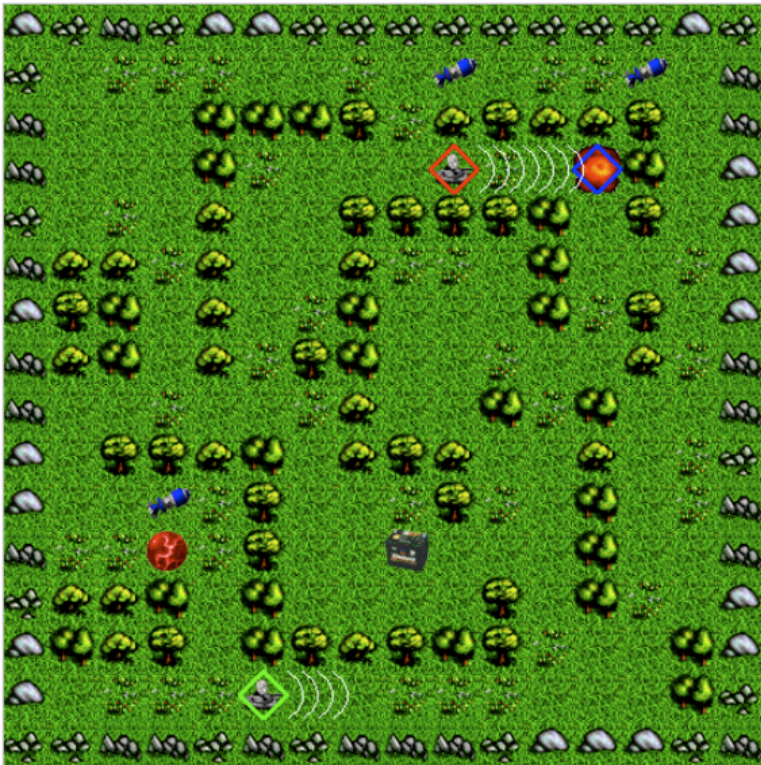


Nuxoll, A. and Laird, J. (2007). Extending Cognitive Architecture with Episodic Memory.

RESEARCH METHODOLOGY

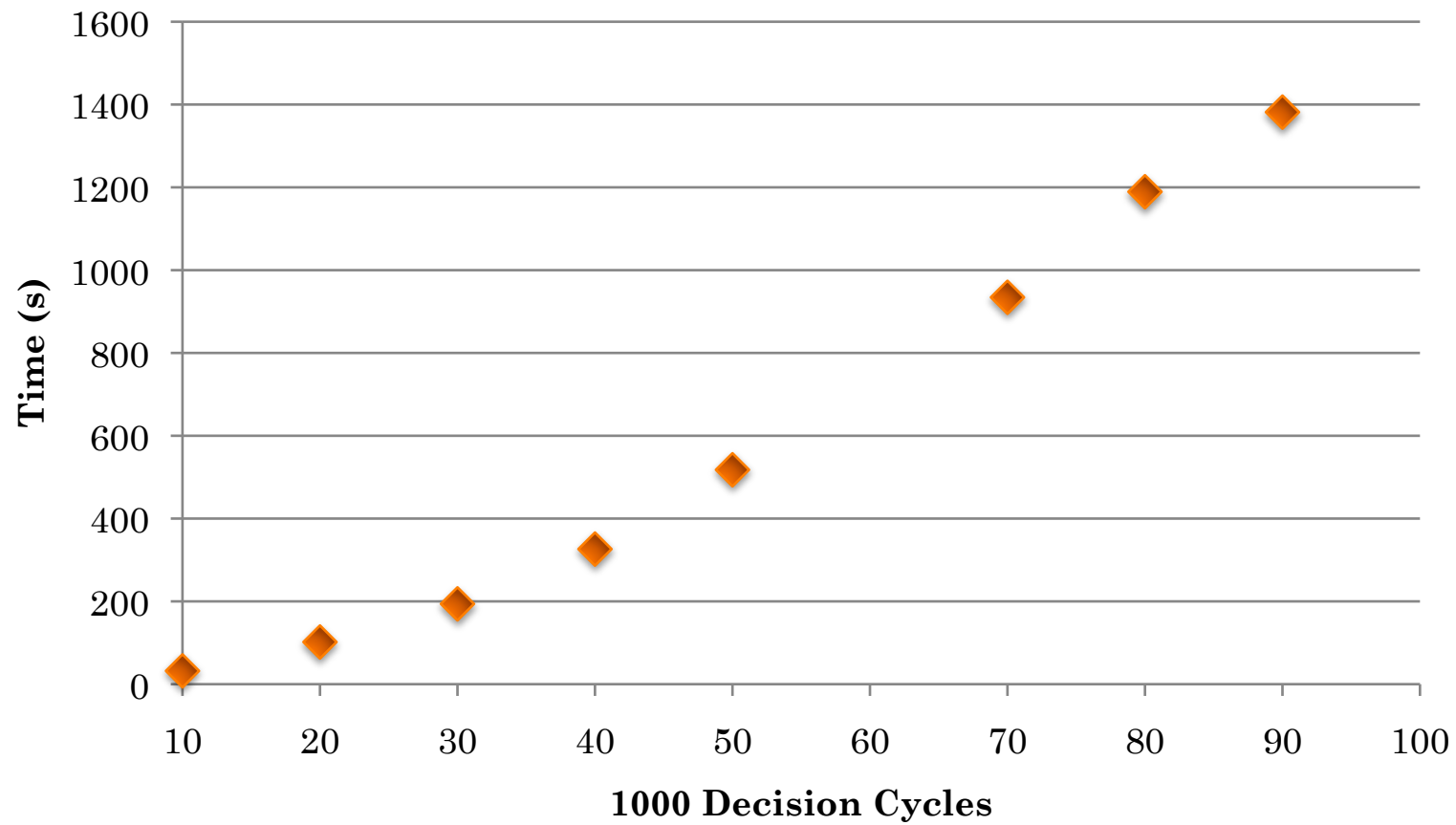
- “Efficiency” unknown
- Iterative, data-driven development and evaluation
 - Observation
 - Hypothesis
 - Contribution
 - Results
 - rinse and repeat...

EXPERIMENTATION DOMAIN: TANKSOAR



- Single agent, fixed number of decisions
- Typically 100 WMEs
 - 70-90% change/dc
 - 300 attribute/value pairs

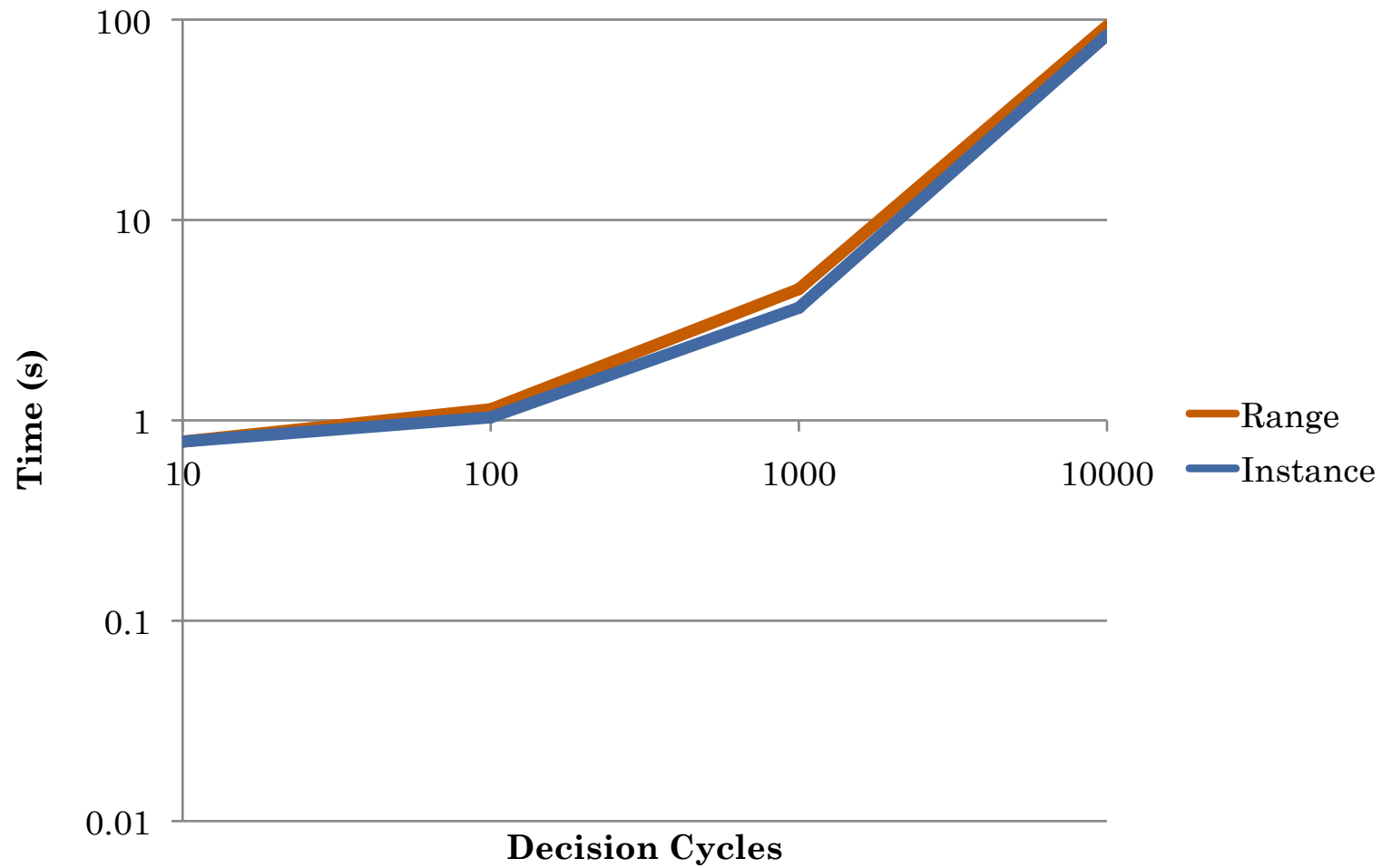
NUXOLL BASELINE: STORAGE



INITIAL IMPLEMENTATION

- Nuxoll Baseline
 - Slow, unreliable
 - Range indexing method not implemented
 - No efficient memory-disk policy
- Mapped to RDBMS
 - Instance and Range
 - SQLite: in-process, standardized, ACID*, memory-disk
 - B+-Tree indexes

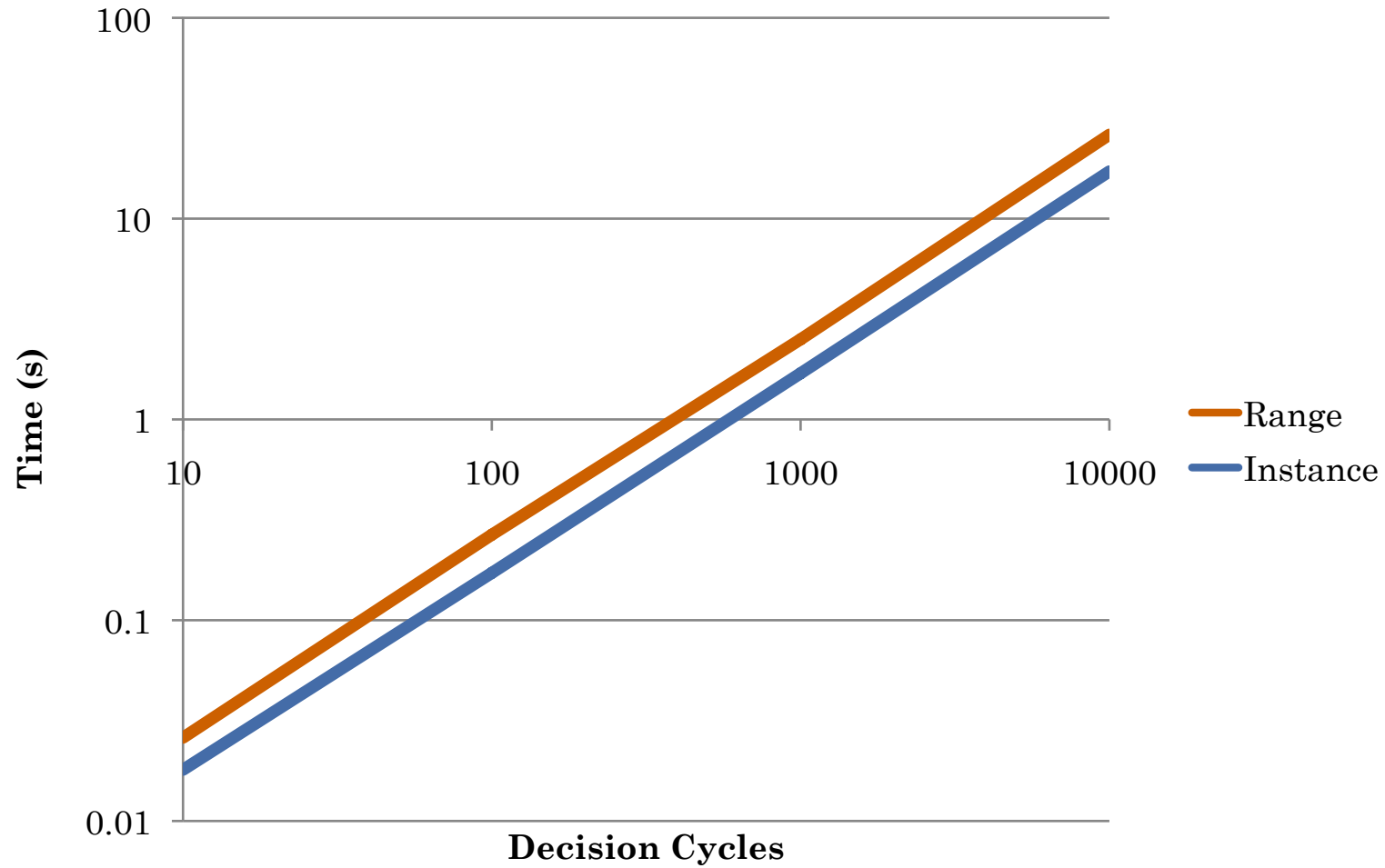
INITIAL IMPLEMENTATION: STORAGE



PROBLEM: WEAK HASH FUNCTION

- Nuxoll's hash function for was weak for WMEs with numerical values
- Hypothesis
 - High collision rate => b-tree rebalancing
 - Profiling: 26% process time
- Contribution
 - Improved hash function

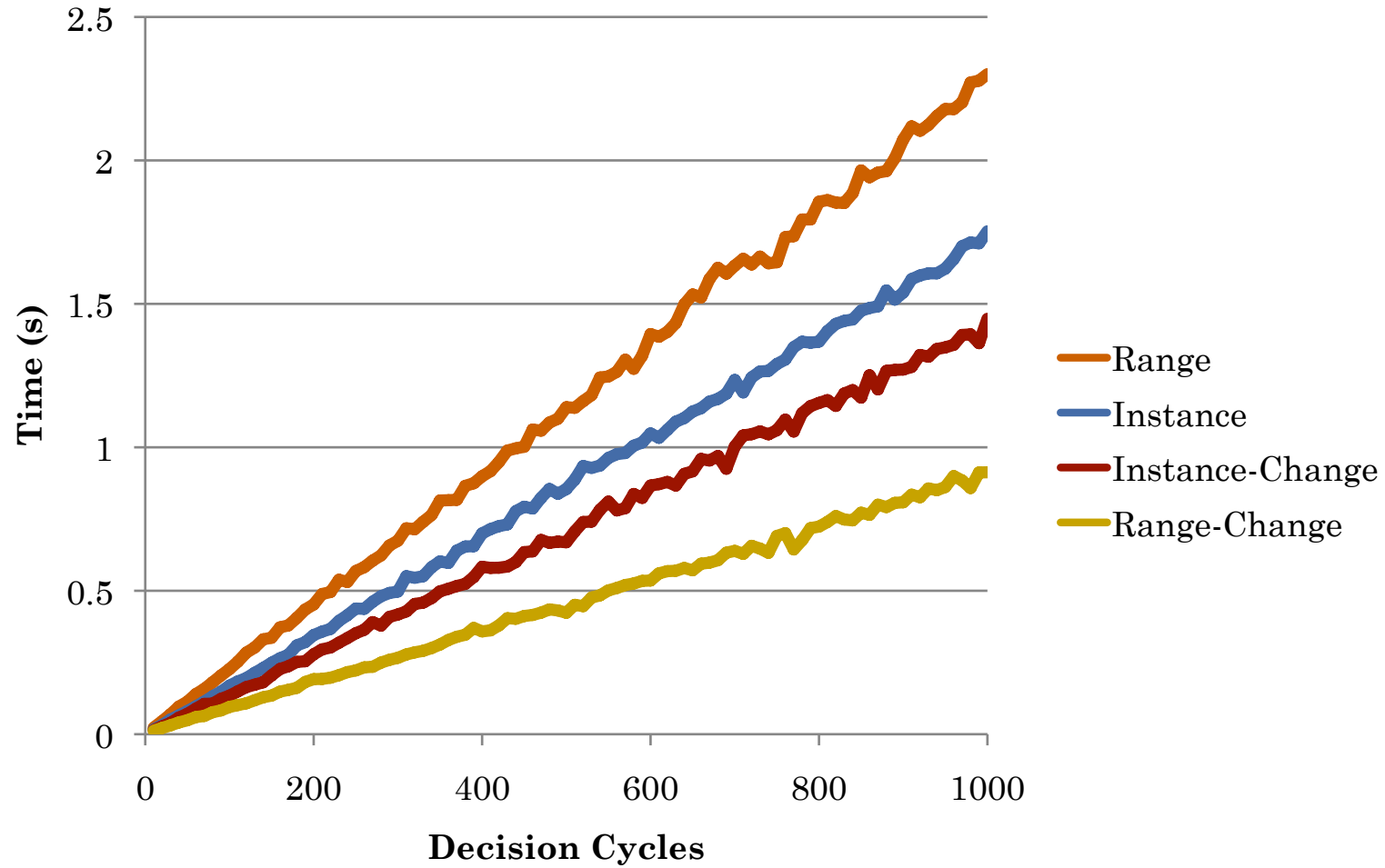
STRONG HASH FUNCTION: STORAGE



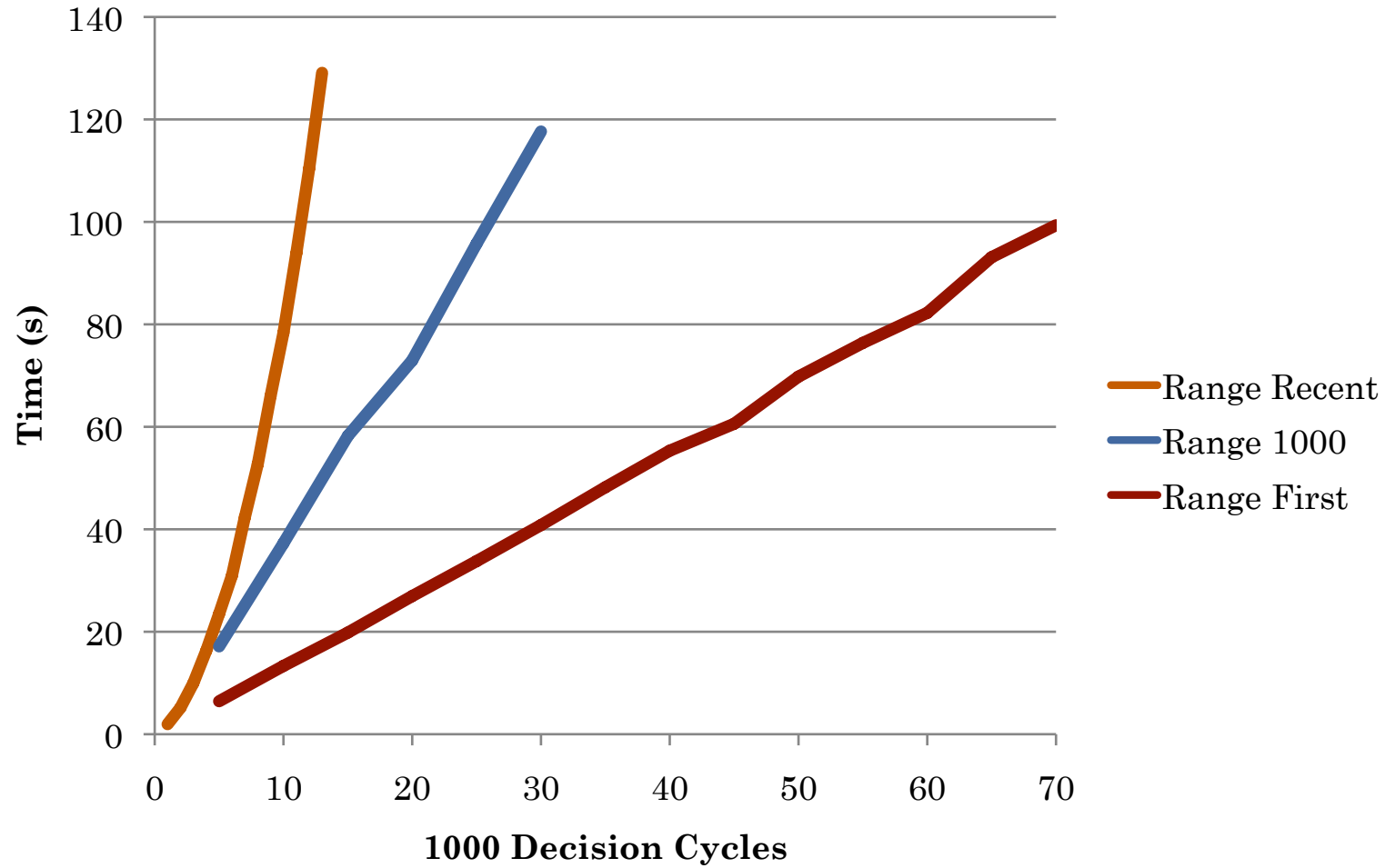
PROBLEM: INEFFICIENT RANGE UPDATES

- Ranges for WMEs that had not been removed from working memory were being updated each episode
- Hypothesis
 - Range storage time was incurring unnecessary cost from range updates and associated b-tree update/rebalancing
- Contribution
 - Update ranges only on WME removal

IMPROVED RANGE UPDATES: STORAGE



RANGE: NON-CUE-BASED (NCB) RETRIEVAL

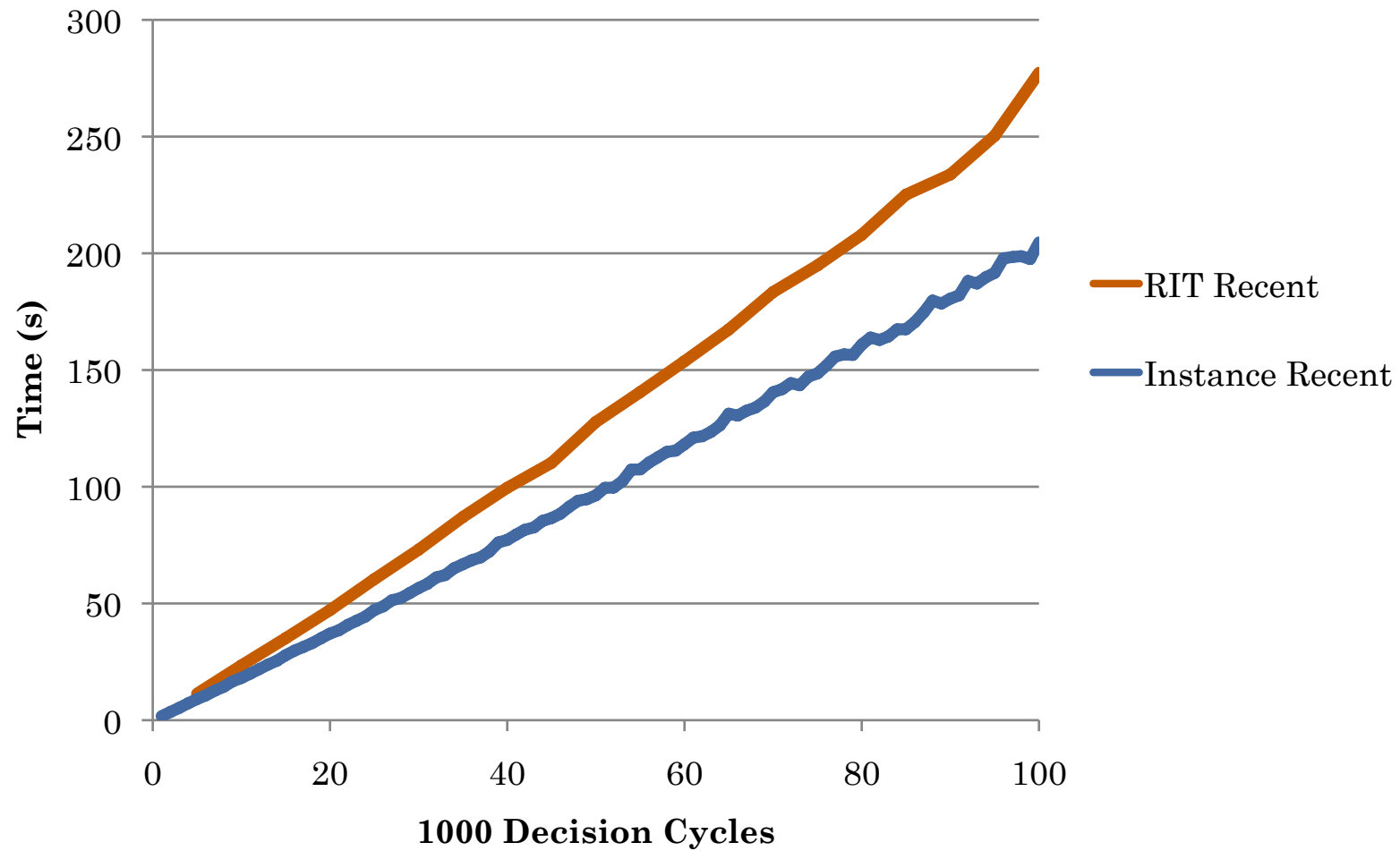


PROBLEM: INTERVAL INTERSECTION SEARCH

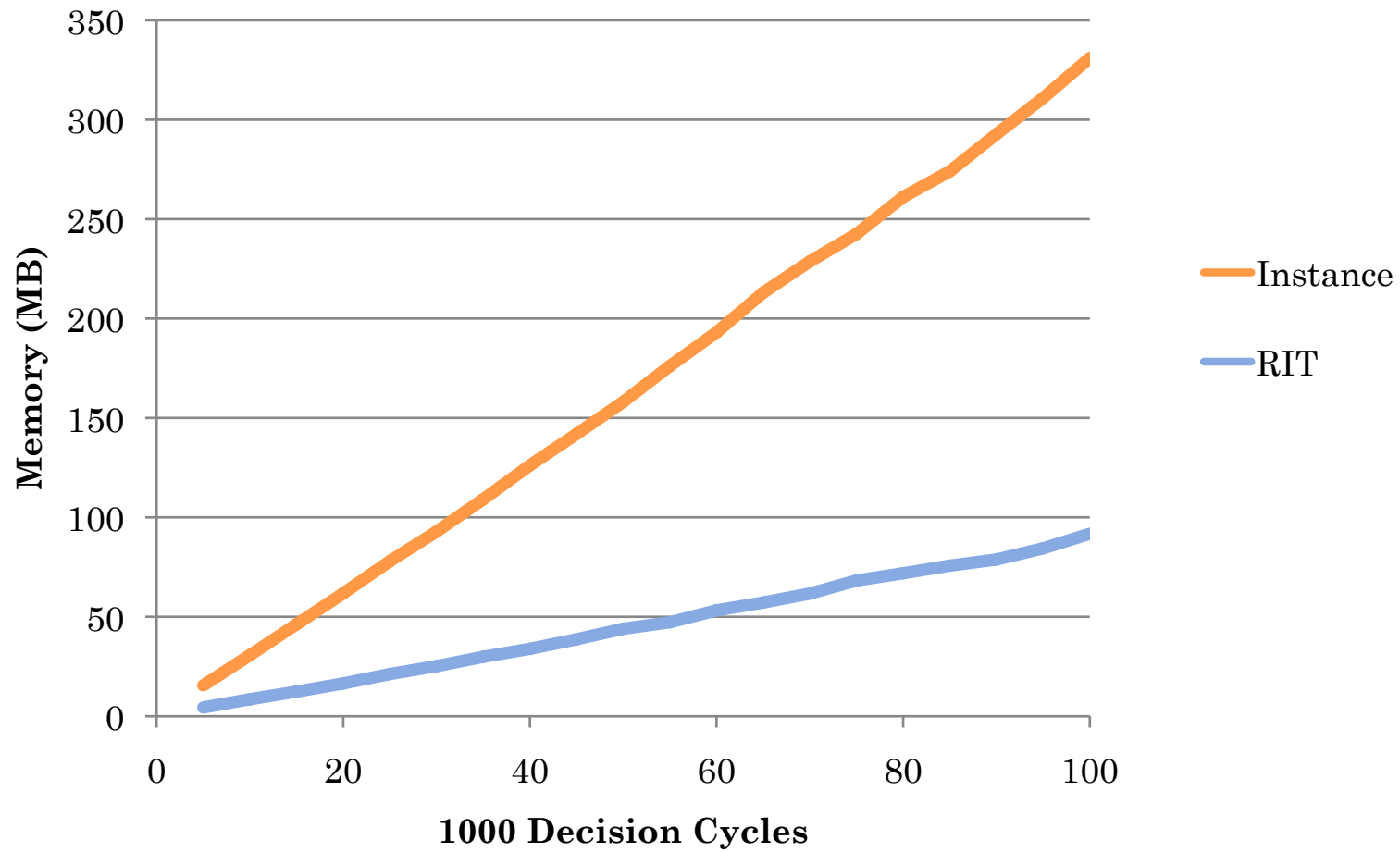
- Episode reconstruction in the Range method is an interval intersection query
- Range episode b-tree index was keyed on (WME, range start, range end)
 - This order induces linear traversal over end ranges of previous episodes
- Hypothesis
 - An interval tree would reduce NCB retrieval time growth to $\log(\text{episodic store size})$
- Contribution
 - Implemented Relational Interval Tree (RIT)

Kriegel, H., Potke, M., Seidl, T. (2000).
Managing Intervals Efficiently in Object-Relational Databases.

RIT: NCB RETRIEVAL



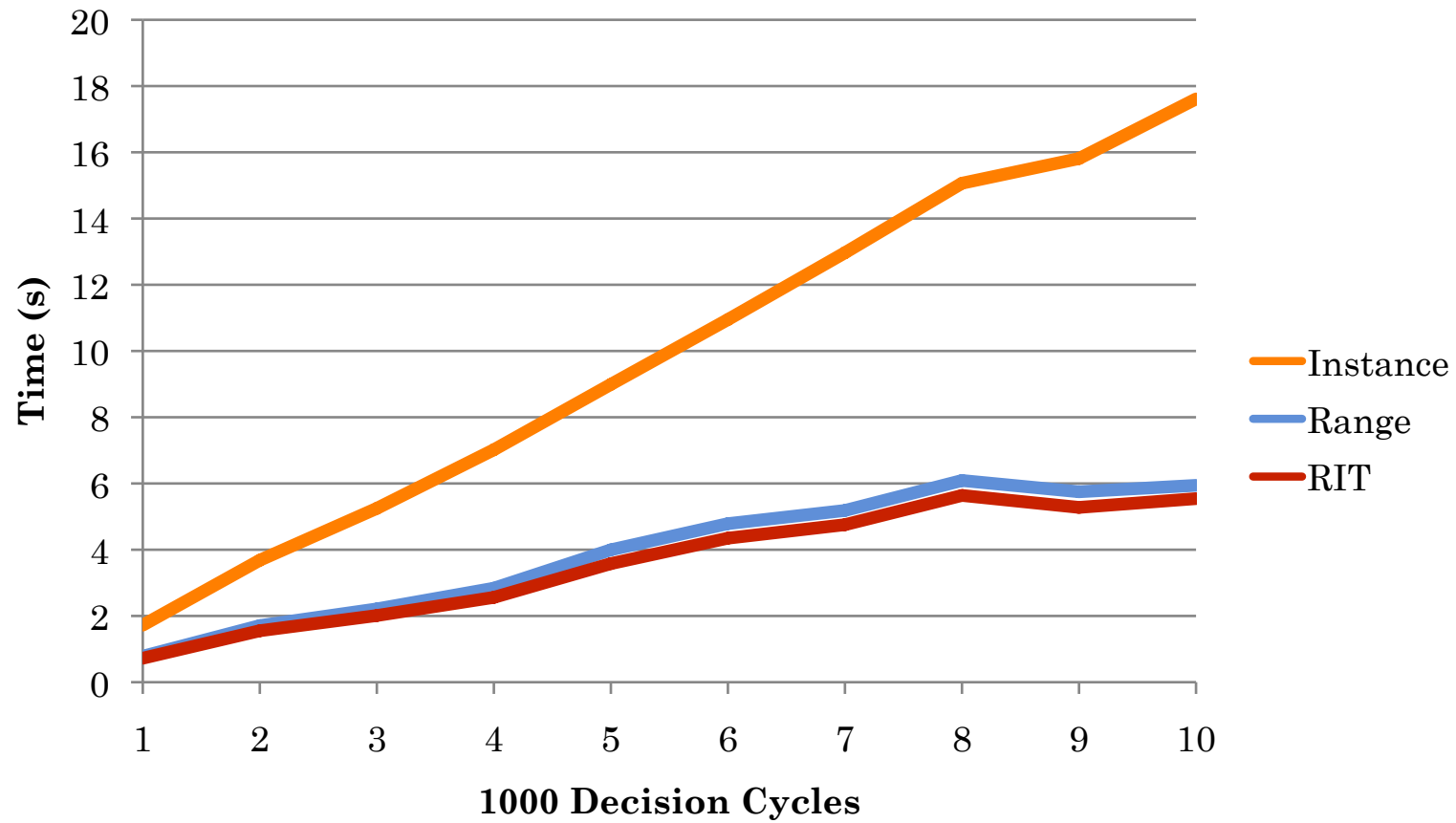
RIT: MEMORY CONSUMPTION



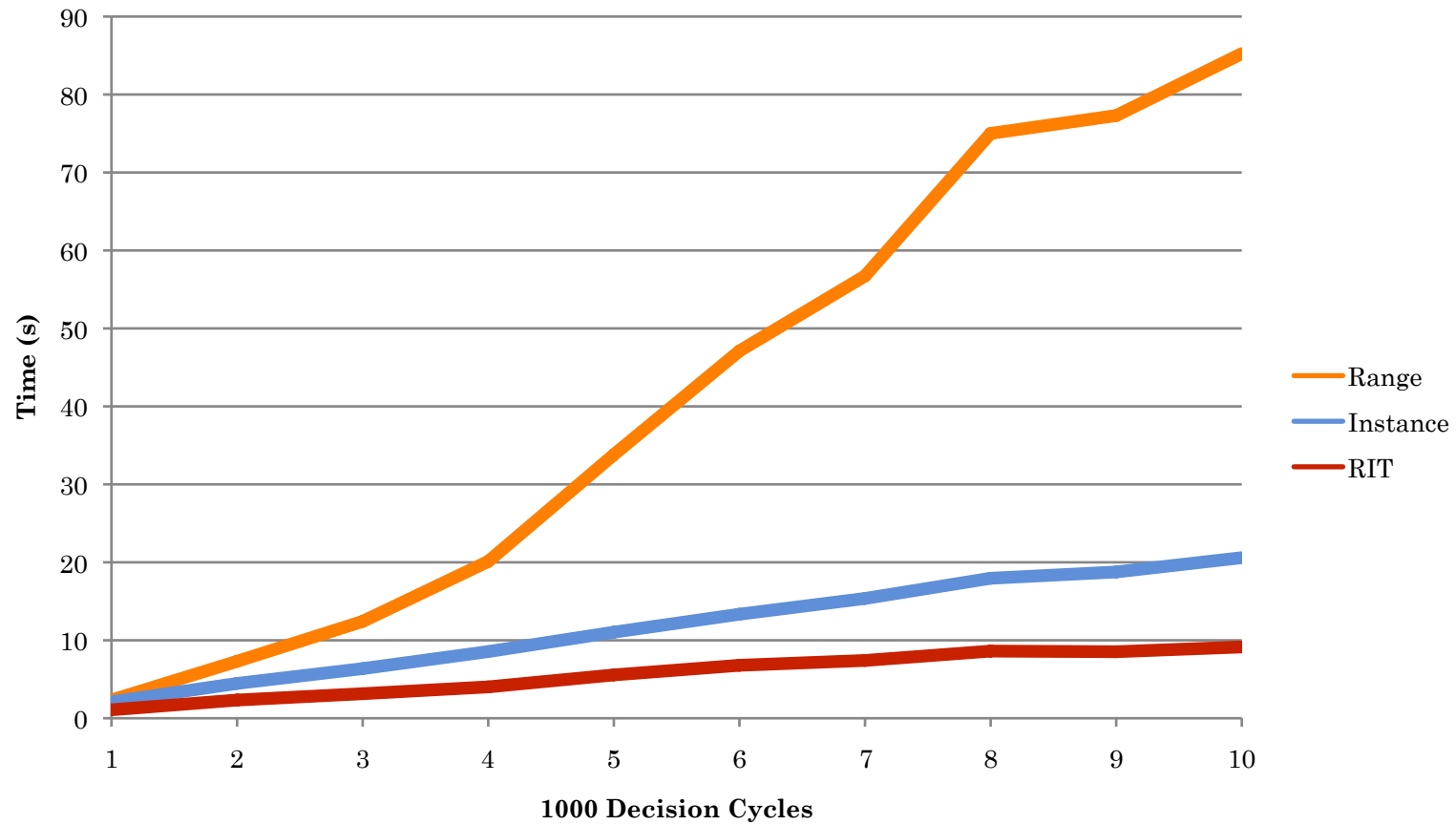
PROBLEM: RIT “NOW” RANGES

- RIT algorithm called for insertion of “now” ranges into RIT with a special “now” end value, with later removal and re-insertion
- Hypothesis
 - Computational cost of calculating RIT placement, as well as b-tree maintenance, was a significant cost
- Contribution
 - Implemented “now” relation to temporarily store “now” ranges

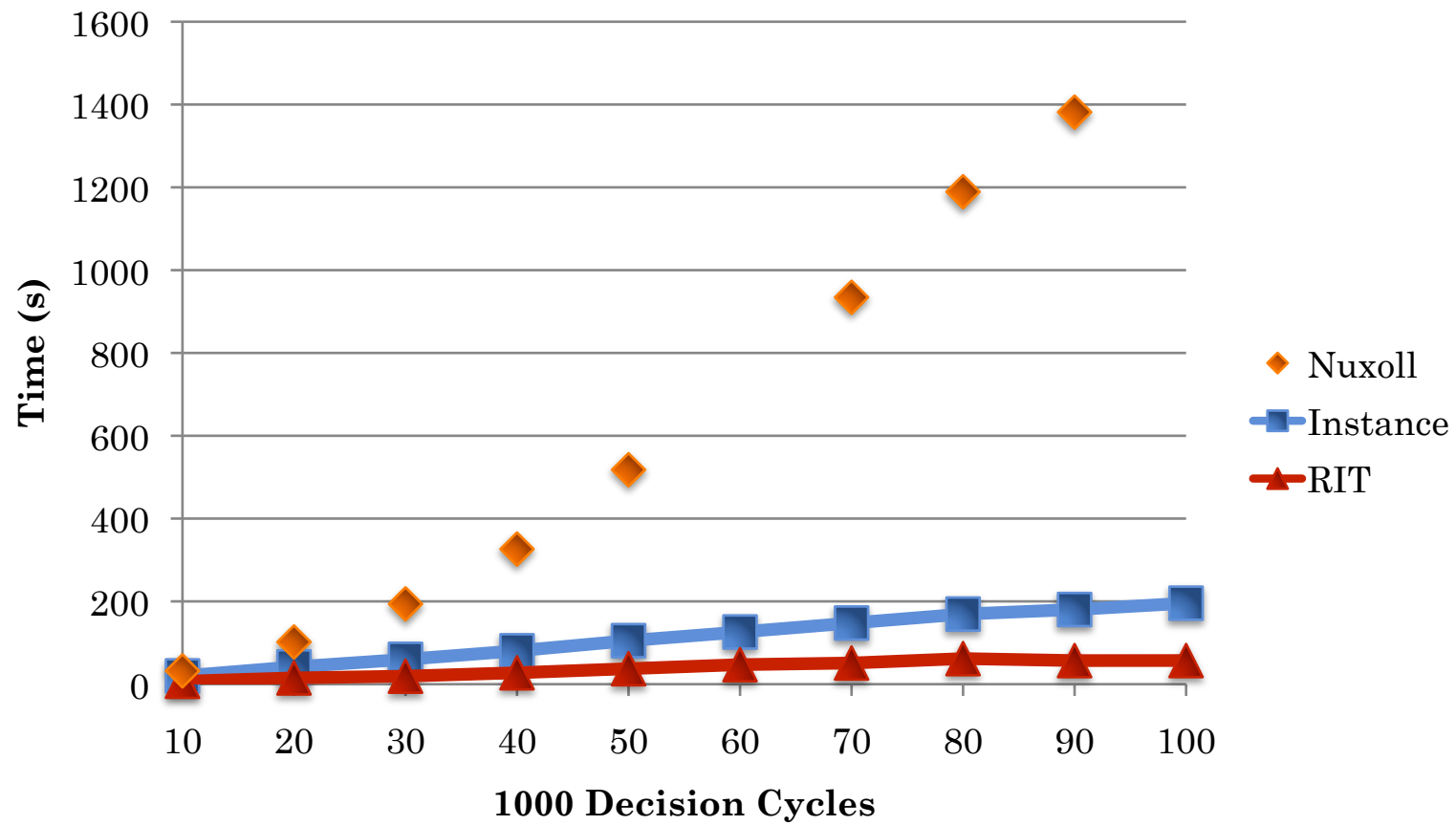
IMPROVED RIT: STORAGE



IMPROVED RIT: NCB RETRIEVAL



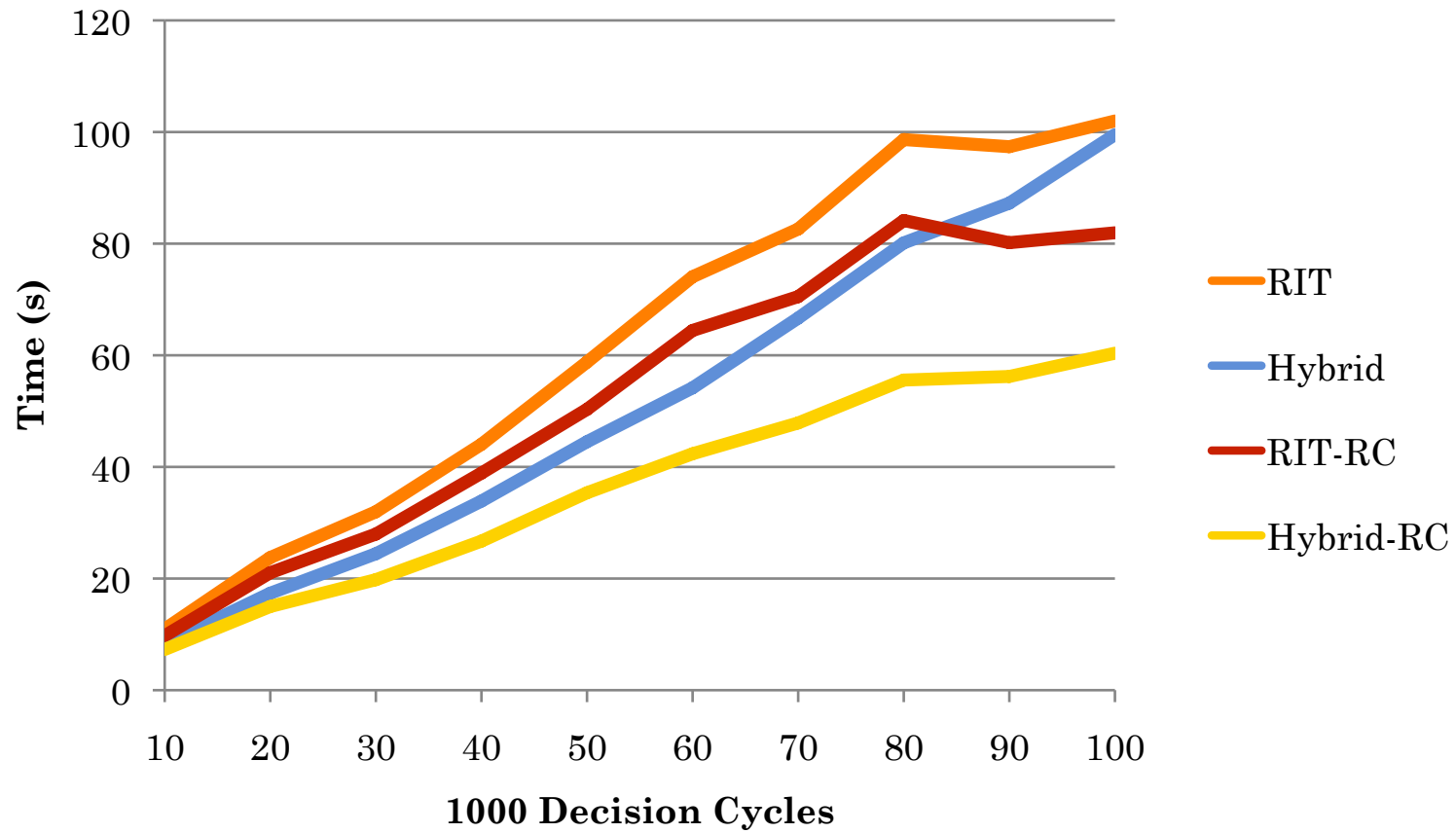
NUXOLL COMPARISON: STORAGE



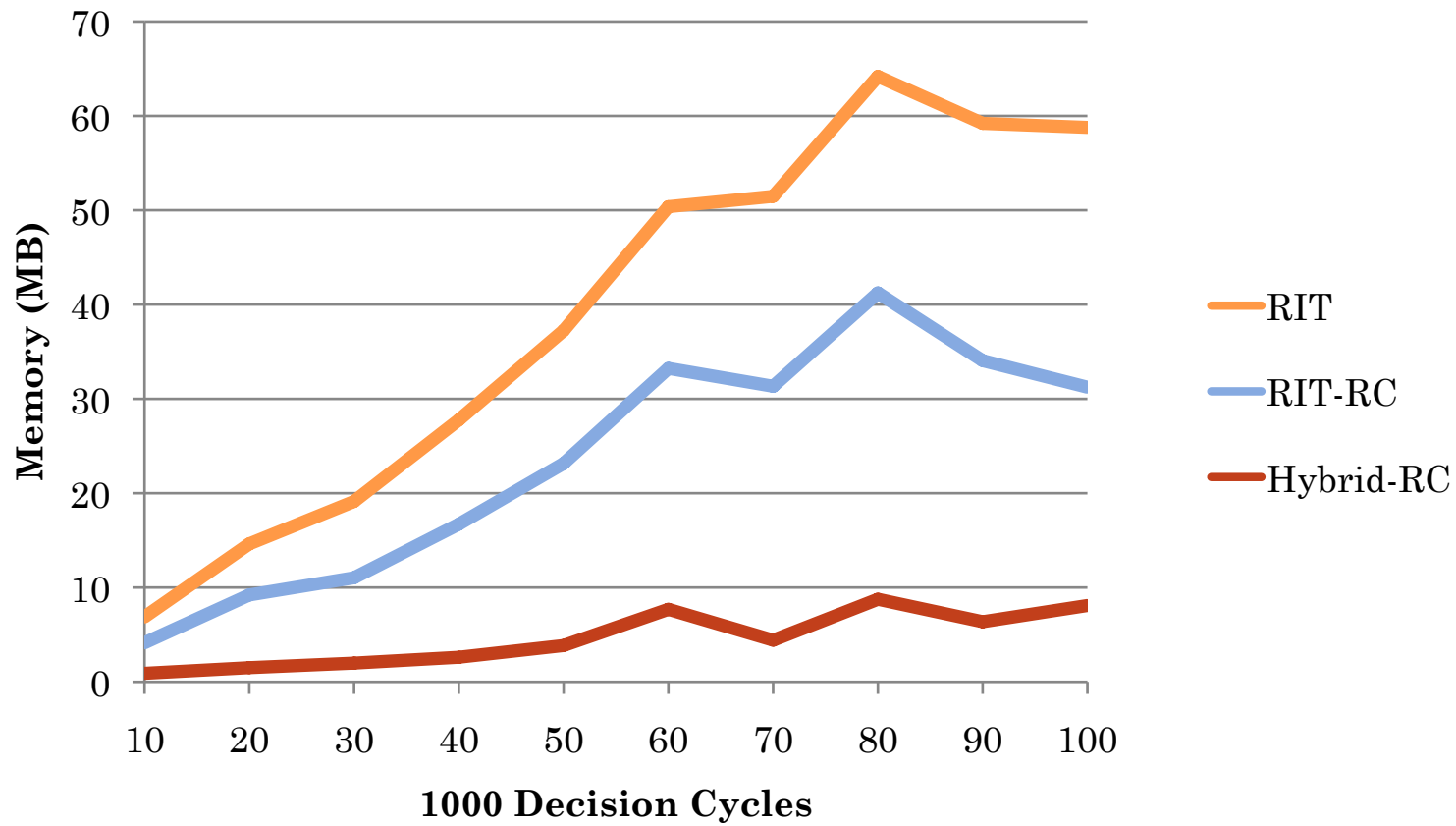
A NEW HYBRID APPROACH

- Observation
 - RIT data structure was used only for NCB retrievals
- Hypothesis
 - A bit-vector “instance” representation would reduce computational overhead of RIT while maintaining efficient NCB retrieval
- Contribution
 - Developed bit-vector episode representation combined with “lean” range relation
 - Afforded run-time ability to exclude WMEs from episodic storage*

HYBRID: NCB RETRIEVAL



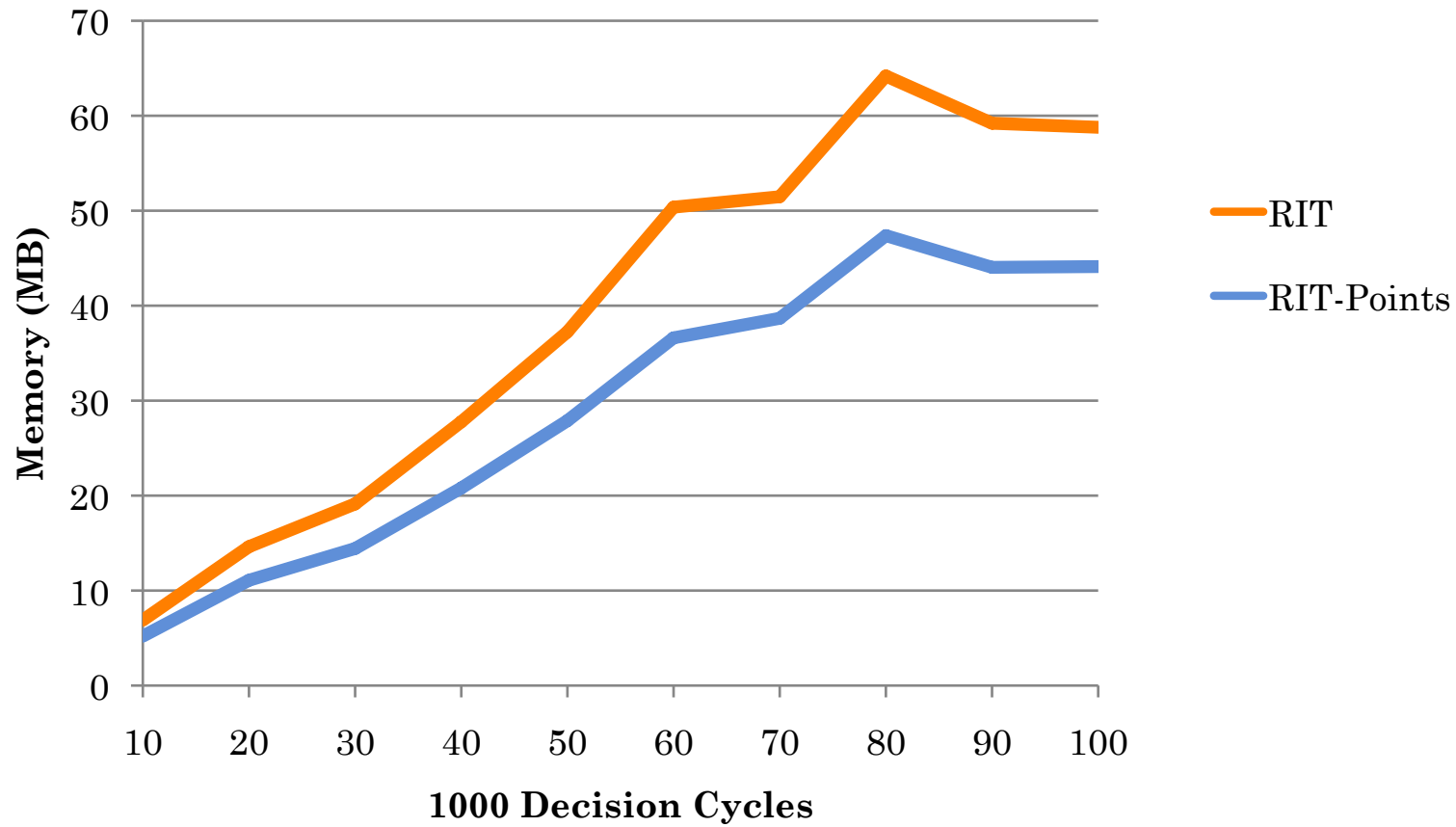
HYBRID: MEMORY CONSUMPTION



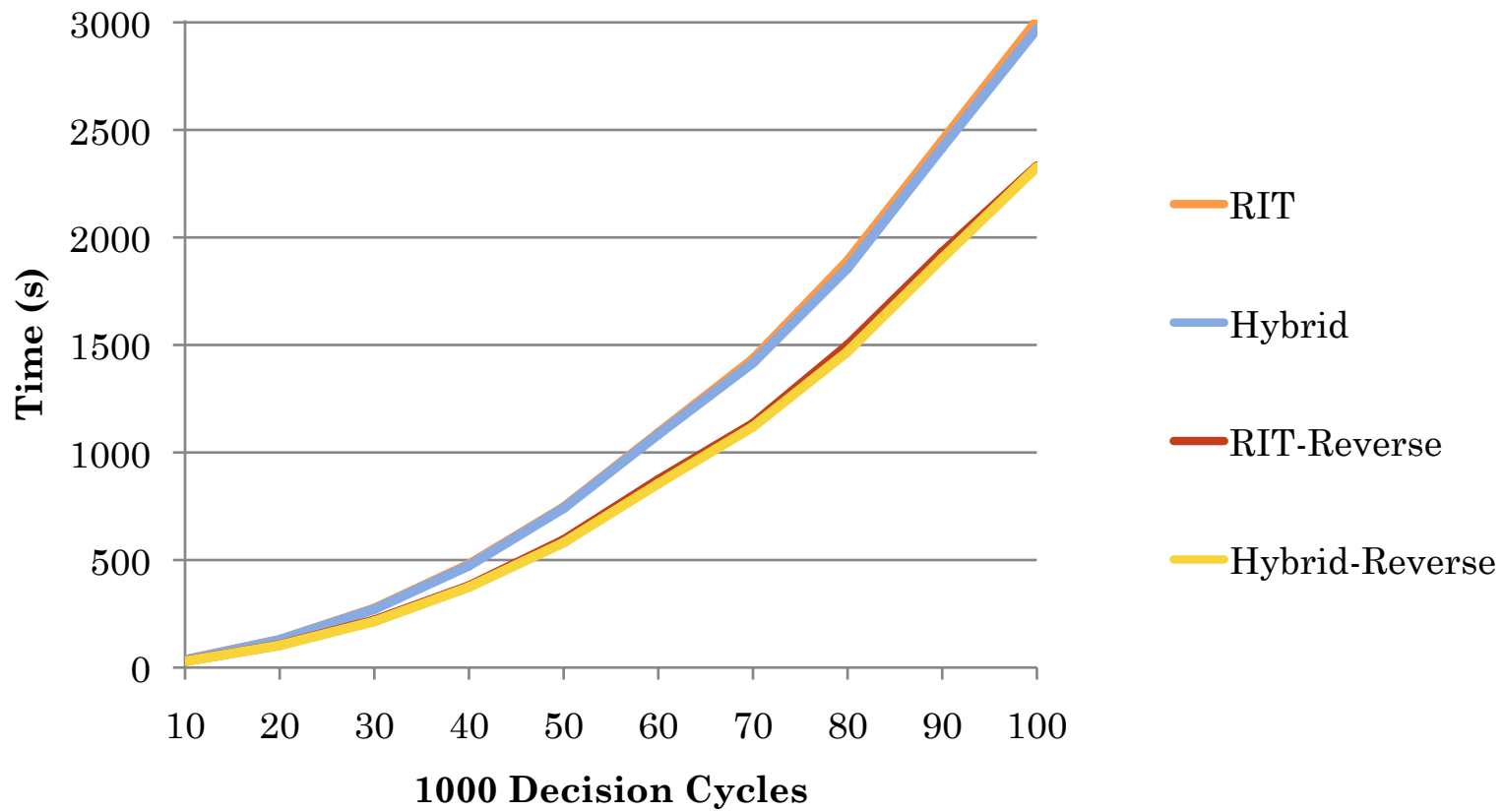
FINAL IMPROVEMENTS

- A large proportion of WMEs in our testing domain were present for only a single episode
 - Represented “point” ranges in a separate relation to reduce memory consumption
- Our cue-based retrieval algorithm biased towards recency, but searched in ascending temporal order
 - Reversed direction of query algorithm to reduce time requirement in where retrieved episodes were relatively recent

POINTS: MEMORY CONSUMPTION



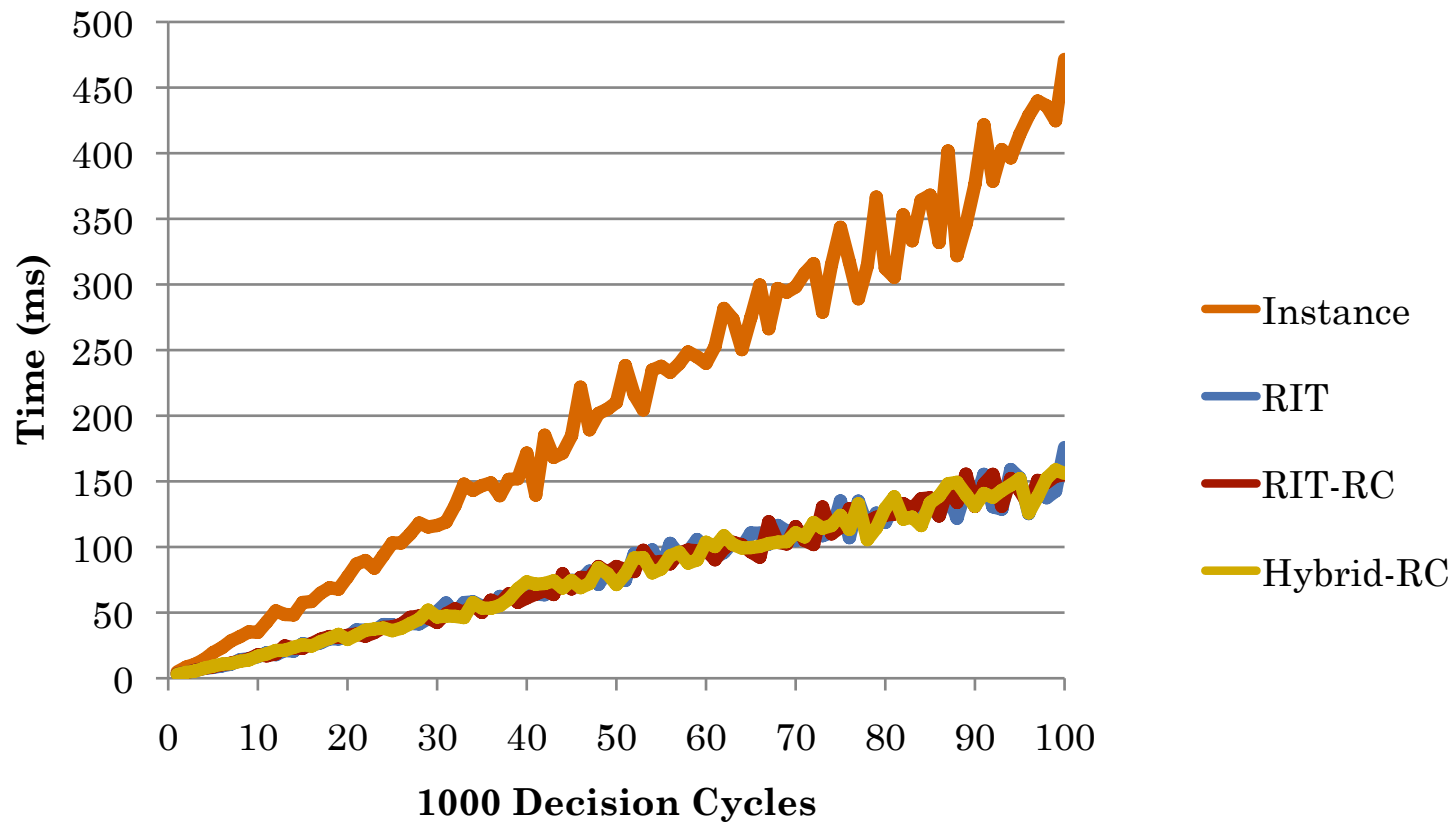
REVERSE: CUE-BASED RETRIEVAL



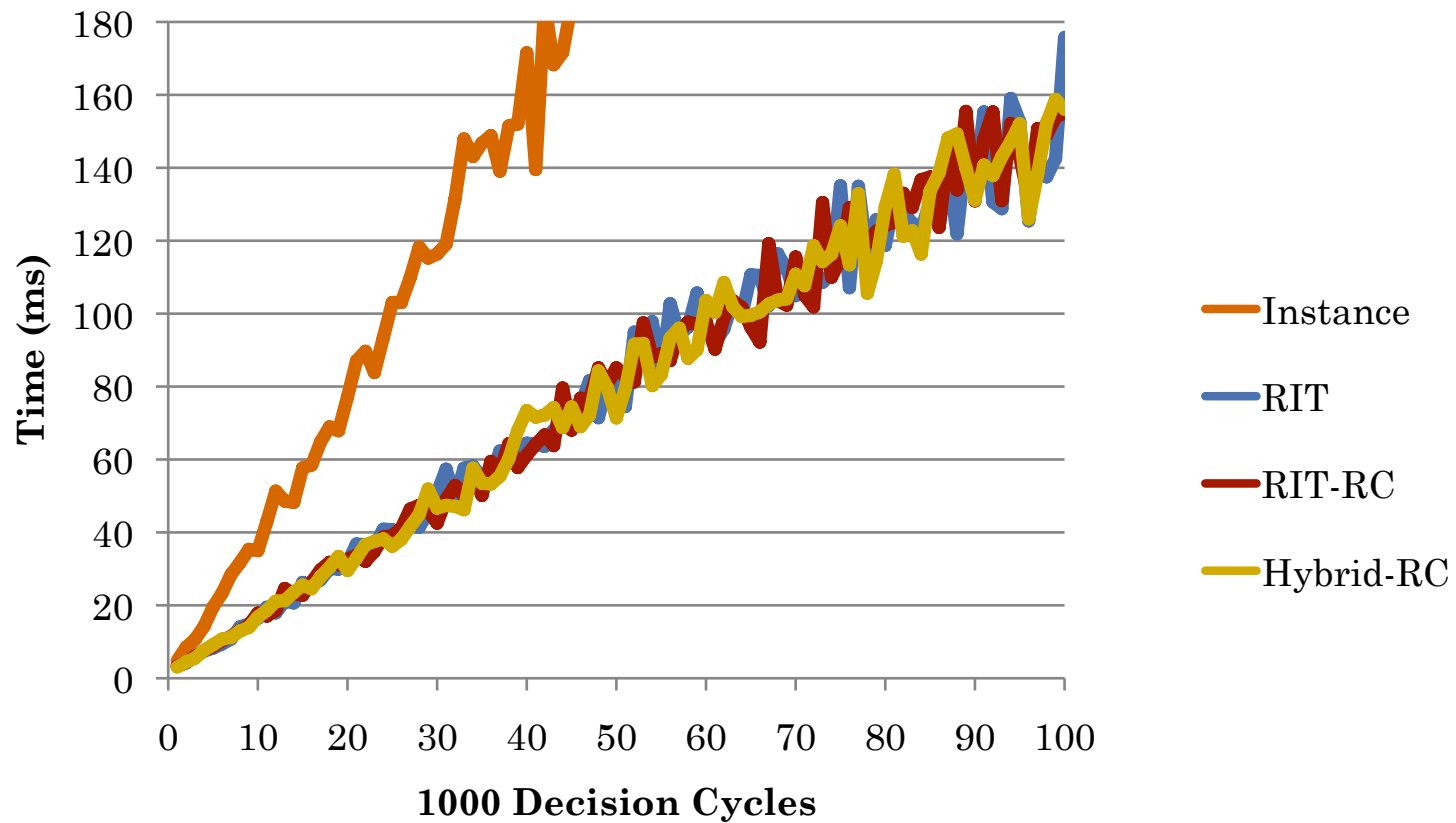
FINAL RESULTS

- Cue-Based Retrieval
 - Cue: current (X, Y) and direction
 - Optimal result is most recent episode
- 5 trials/random seeds, 3 episodes each
- 2.8GHz Core 2 Duo, 4GB RAM, Mac
 - Soar 9.1.0-beta

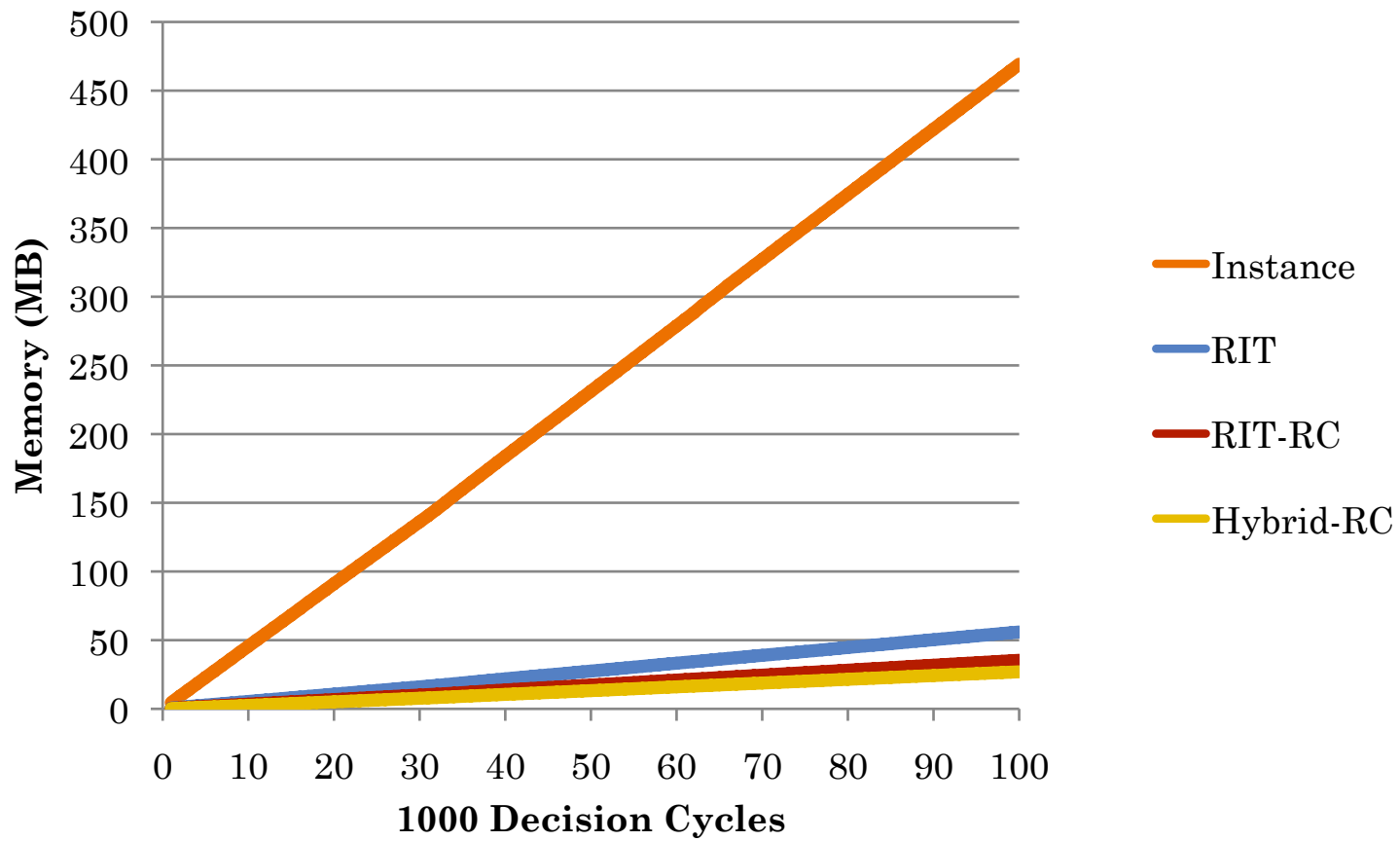
FINAL: CUE-BASED RETRIEVAL (TIME/CYCLE)



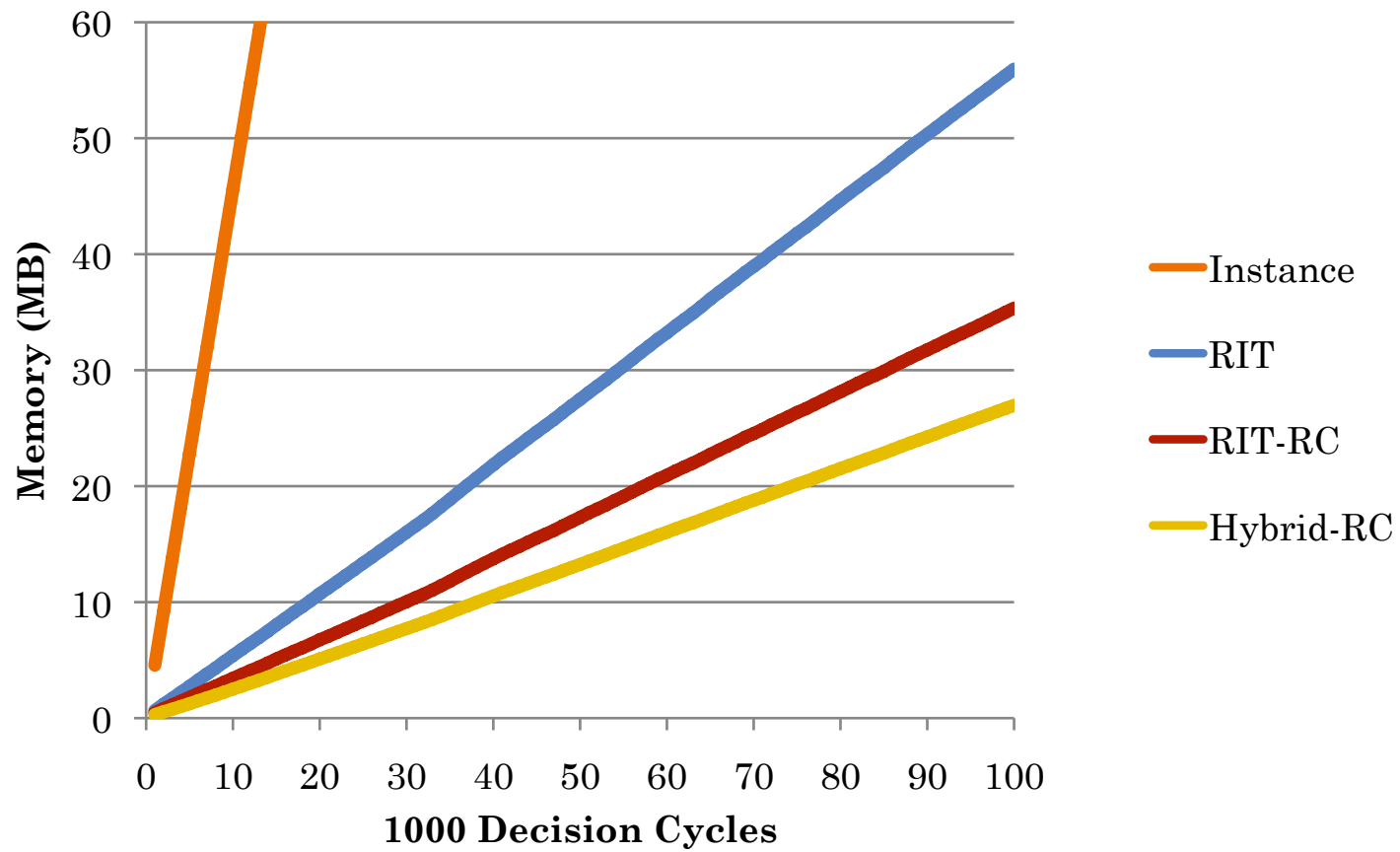
FINAL: CUE-BASED RETRIEVAL (TIME/CYCLE)



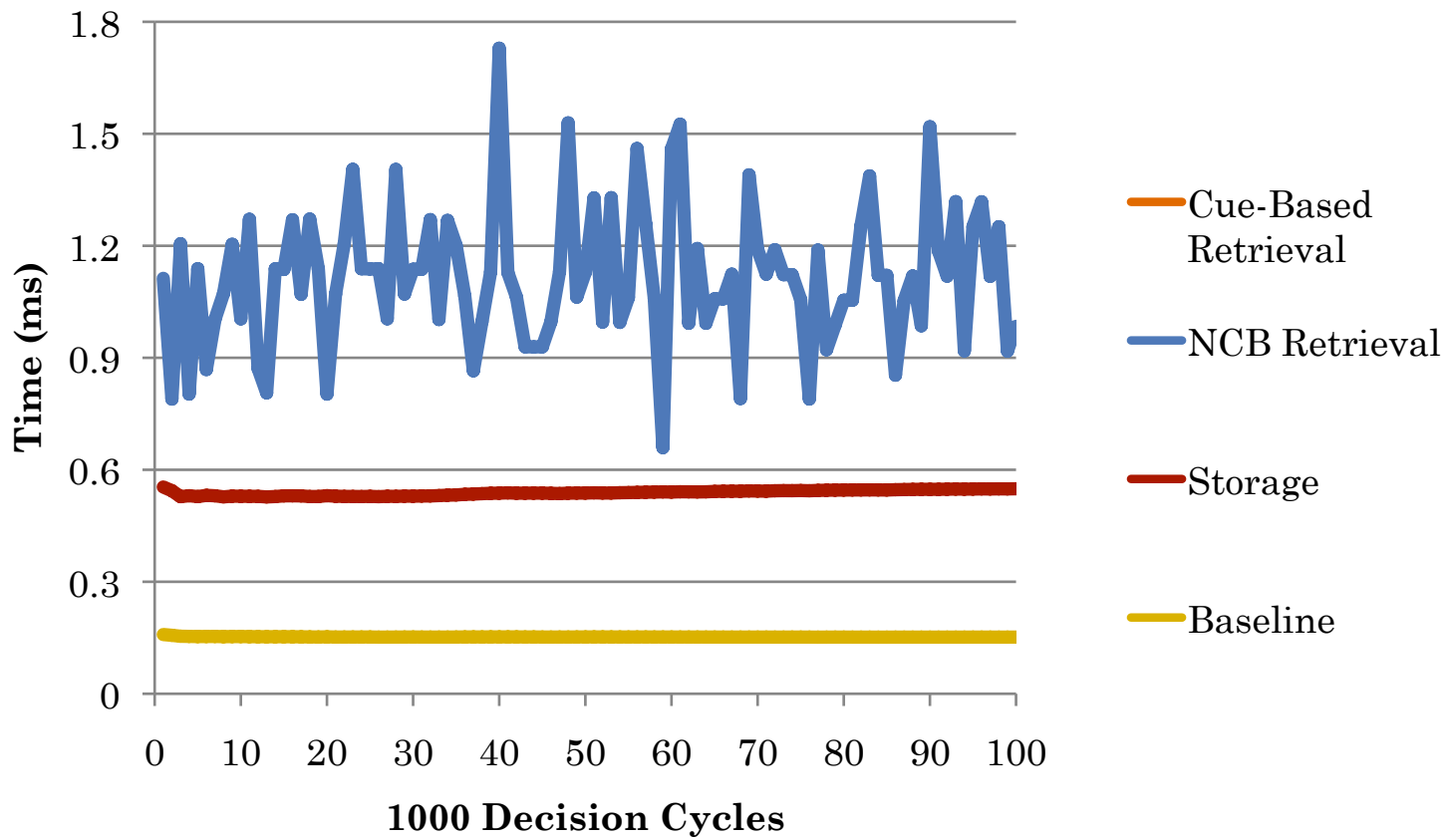
FINAL: MEMORY CONSUMPTION



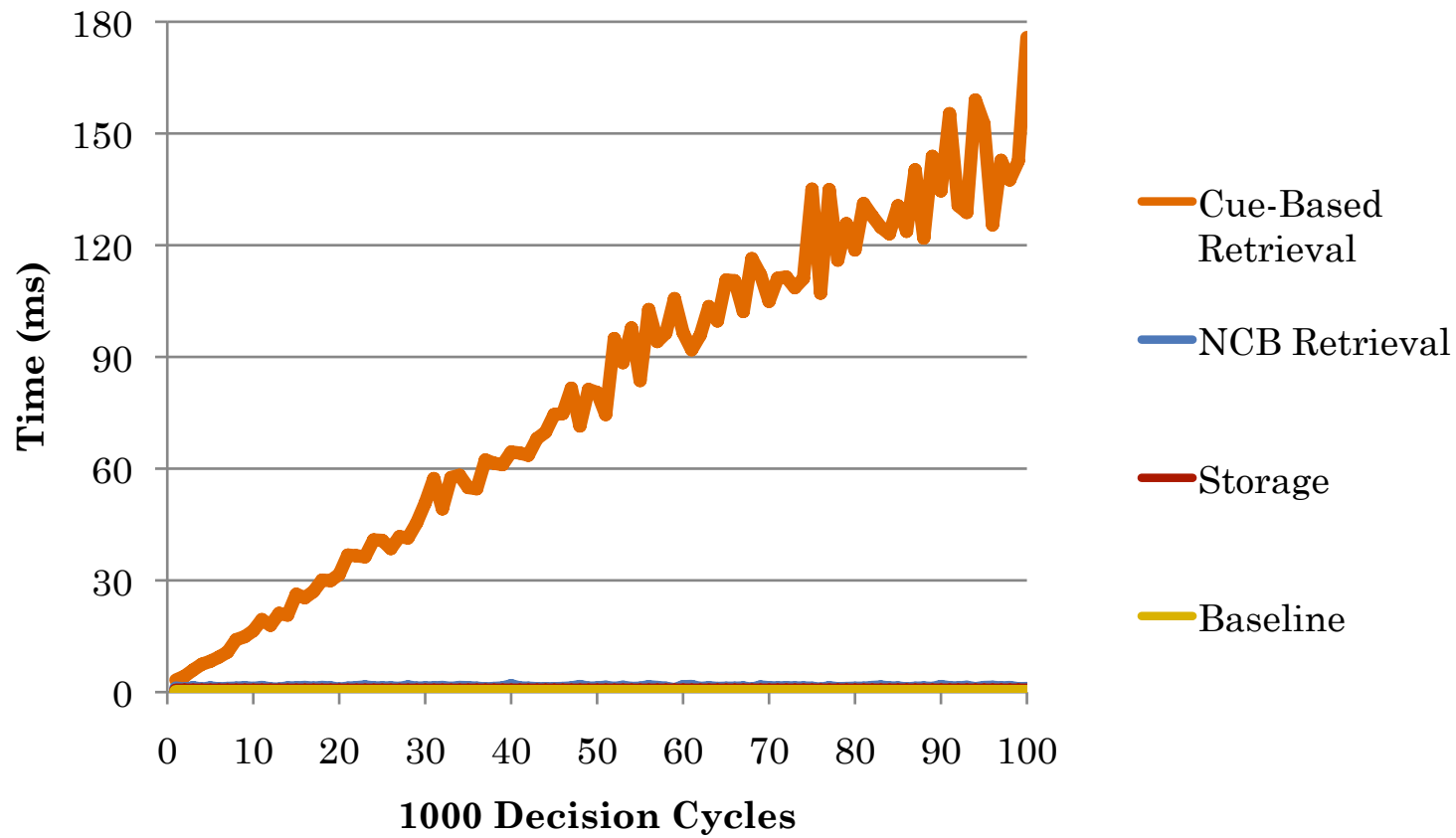
FINAL: MEMORY CONSUMPTION



FINAL: OPERATION TIME/CYCLE (RIT)



FINAL: OPERATION TIME/CYCLE (RIT)



FINAL RESULTS SUMMARY

- An order of magnitude improvement in storage operation time and memory consumption over initial implementation
- Cognitive Real-Time
 - 50 ms/decision
 - 1 episode/s
 - 250 ms/retrieval
 - 164,000 seconds = **1.9 days**
 - **95MB memory**

IMPROVING EVALUATION

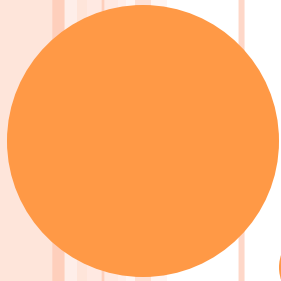
- Cue Dimensions
 - Size
 - Selectivity
- Retrieval Dimensions
 - Changes in working memory
 - Episode recency
- Impact of SQLite
 - More precise component timing
 - Low-level implementation analysis of cue-based retrieval search

DISCUSSION: CONTRIBUTION

- Current performance results represent a point in the space of factors influencing episodic memory efficiency
 - $(t,m) = e(\text{Soar}, \text{Soar-EpMem}, \textit{simple}, \textit{wander}, \textit{TankSoar})$
- Need to characterize performance as a function over design choices within this space
 - **Cognitive Architecture** - Soar, ACT-R, EPIC, LIDA, etc.
 - **EpMem** - encoding, storage, retrieval, integration
 - **Agent** - retrieval frequency, cue specificity (w.r.t. task/environment), cue elements (w.r.t. task), etc.
 - **Task** - single vs. multi-task, single vs. multi-agent, # objects, etc.
 - **Environment** - frequency of input updates, granularity of input (discrete vs. continuous), etc.

DISCUSSION: LONG-TERM GOALS

- Understanding and quantifying retrieval “quality”
=> non-optimal retrievals with confidence
 - Two-stage retrievals
 - Query optimization
 - Episode caching
- Understanding domain/task characteristics
=> online, intelligent/predictive architectural tuning
- High efficiency (and thus high utility) motivates explorations into cognitive capabilities and interactions with other learning mechanisms



THANK YOU!

Questions?