

Preemptive Strategies for Overcoming the Forgetting of Goals

To be presented at AAAI 2013

Justin Li John Laird

Computer Science and Engineering
University of Michigan
justinnh, laird@umich.edu

2013-06-05

Forgetting in ACT-R and Soar

- ▶ Memory elements are removed architecturally
- ▶ Soar
 - ▶ working memory elements decay via base-level activation
- ▶ ACT-R
 - ▶ limited working-memory size require memory elements be over-written
 - ▶ long-term memory elements decay via base-level activation

Forgetting in ACT-R and Soar

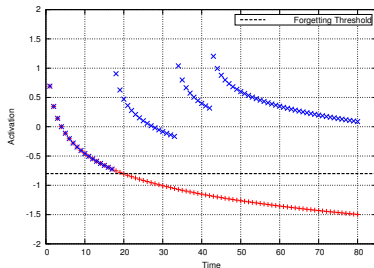
- ▶ Memory elements are removed architecturally
- ▶ Soar
 - ▶ working memory elements decay via base-level activation
- ▶ ACT-R
 - ▶ limited working-memory size require memory elements be over-written
 - ▶ long-term memory elements decay via base-level activation
- ▶ To understand preemptive strategies, need to first understand base-level activation

Base-Level Activation

- ▶ Memory elements are forgotten if their *activation* falls below a threshold

$$A = \ln\left(\sum_{j=1}^n t_j^{-d}\right)$$

- ▶ d — the decay rate (an architectural parameter)
- ▶ t_j — time since access j
- ▶ Activation increases with recency and frequency of access



Preemptive Actions

- ▶ Two *actions*
 - ▶ *rehearsal*: boost the activation of an existing goal
 - ▶ *retrieval*: bring back a forgotten goal

- ▶ When should the agent perform these actions?

Preemptive Triggers

When do goals come to mind in human behavior?

- ▶ For *time-based* targets, periodically
- ▶ For *event-based* targets, during context switches

Preemptive Triggers

When do goals come to mind in human behavior?

- ▶ For *time-based* targets, periodically
- ▶ For *event-based* targets, during context switches

Two *triggers*:

- ▶ *timing*: periodically, over some time span
- ▶ *context*: at context-switches, which precede large changes in perception

Preemptive Strategies

This gives four variations:

- ▶ Two actions
 - ▶ rehearsal: boost the activation of an existing goal
 - ▶ retrieval: bring back a forgotten goal
- ▶ Two triggers:
 - ▶ timing: periodically, over some time span
 - ▶ context: at context-switches, which precede large changes in perception

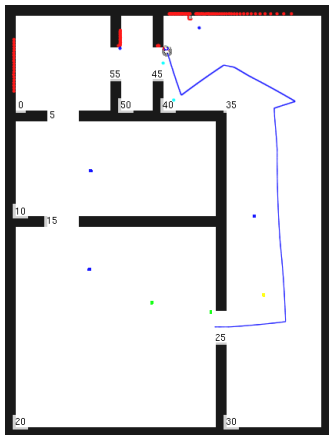
Preemptive Strategies

This gives four variations:

- ▶ Two actions
 - ▶ rehearsal: boost the activation of an existing goal
 - ▶ retrieval: bring back a forgotten goal
- ▶ Two triggers:
 - ▶ timing: periodically, over some time span
 - ▶ context: at context-switches, which precede large changes in perception
- ▶ Two questions:
 1. How do preemptive strategies scale to large numbers of intentions?
 2. How successful are these strategies?

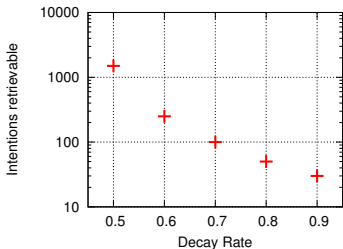
Mobile Robot Domain

- ▶ Strategies implemented in simulated indoor mobile robot
- ▶ Robot must pick up and deliver objects while keeping to a patrol
- ▶ Robot does not know the location of objects
- ▶ Measure deliveries completed in various conditions



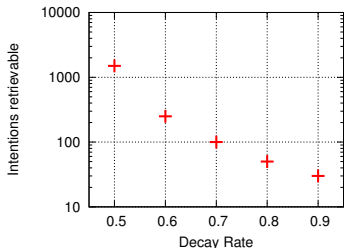
Scalability Results

- Soar: Number of intentions retrievable until the first is forgotten

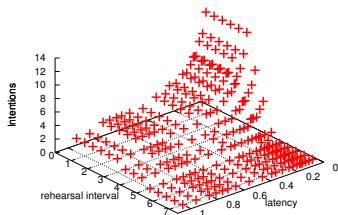


Scalability Results

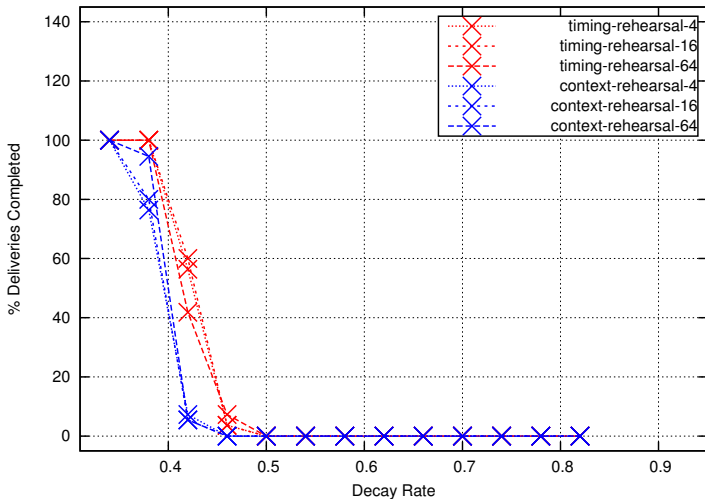
- ▶ Soar: Number of intentions retrievable until the first is forgotten



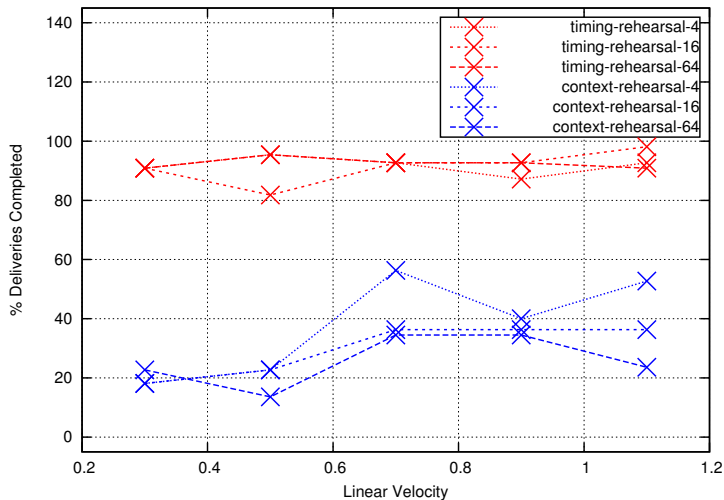
- ▶ ACT-R: Number of intentions retrievable after 20 sec, $d = 0.5$



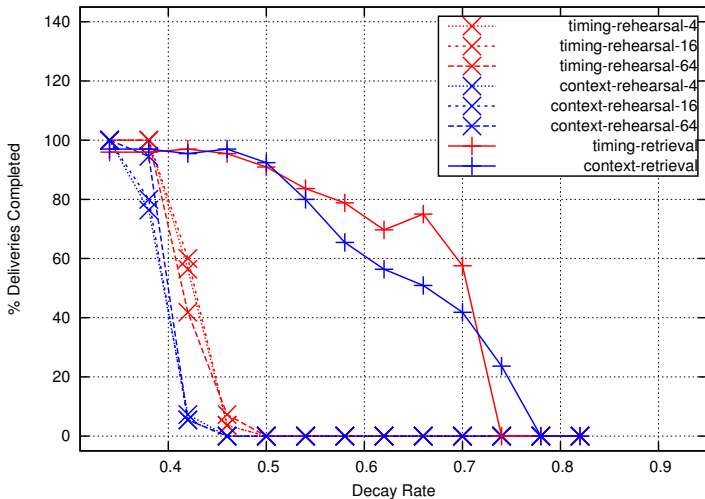
Strategy Comparison Results



Strategy Comparison Results



Strategy Comparison Results



Nuggets and Coal

Nuggets

- ▶ Preemptive strategies are a viable solution to prospective memory
- ▶ Functional Soar agent implementing preemptive strategies
- ▶ Delivery domain for prospective memory evaluation
- ▶ Understanding of effects of domain parameters on preemptive strategies

Coal

- ▶ Preemptive strategies are domain dependent
 - ▶ rehearsal strategies require agent knowledge of decay rate and temporal dynamics
 - ▶ timing triggers also require the same
 - ▶ context-switch triggers require domain knowledge
- ▶ Mixture of strategies were not explored

Trigger Comparison Results

- ▶ Timing triggers are expensive
 - ▶ for 64 rehearsals, take up 9% of agent decisions
 - ▶ for retrievals, require 50% more retrievals than context-switches

- ▶ Context-switches are more predictive of opportunity