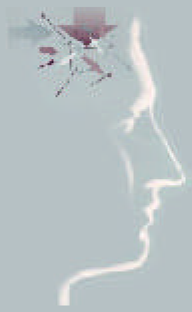
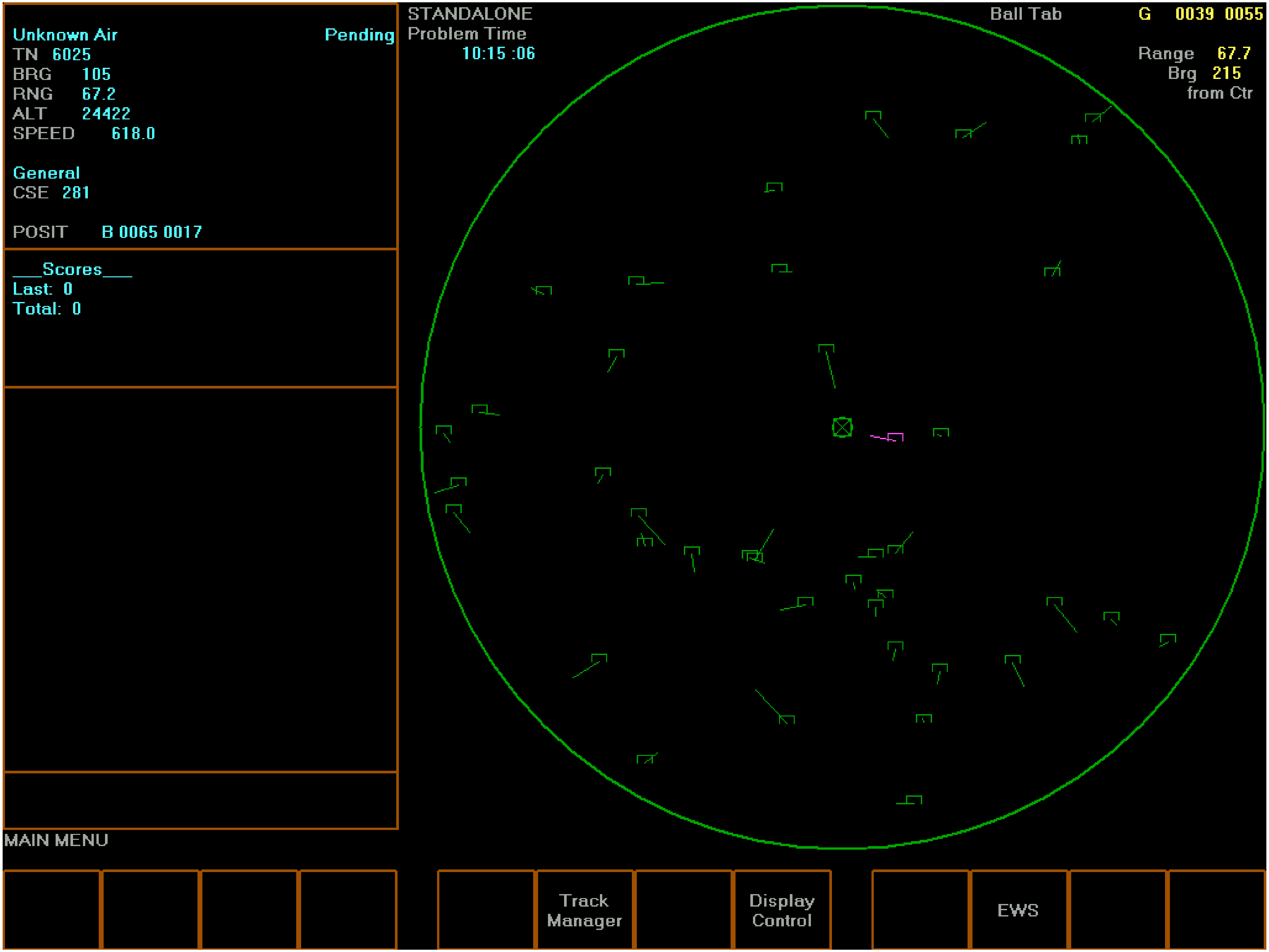
CMU-ASP: ACT-R Learns to be an Anti-air Warfare Coordinator

John Anderson, Dan Bothell, Scott Douglass, Craig Haimson and Myeong-Ho Sohn



Why It’s Neat

1. It is complex -- not a simple protected laboratory task.
2. It looks at all the the buffers interacting in a time-pressure situation.
3. It involves all of the forms of learning in ACT-R.
4. Its ONR goal is to produce a system that can evaluate over-the-shoulder instruction.



Unknown Air Pending
TN 6025
BRG 105
RNG 67.2
ALT 24422
SPEED 618.0

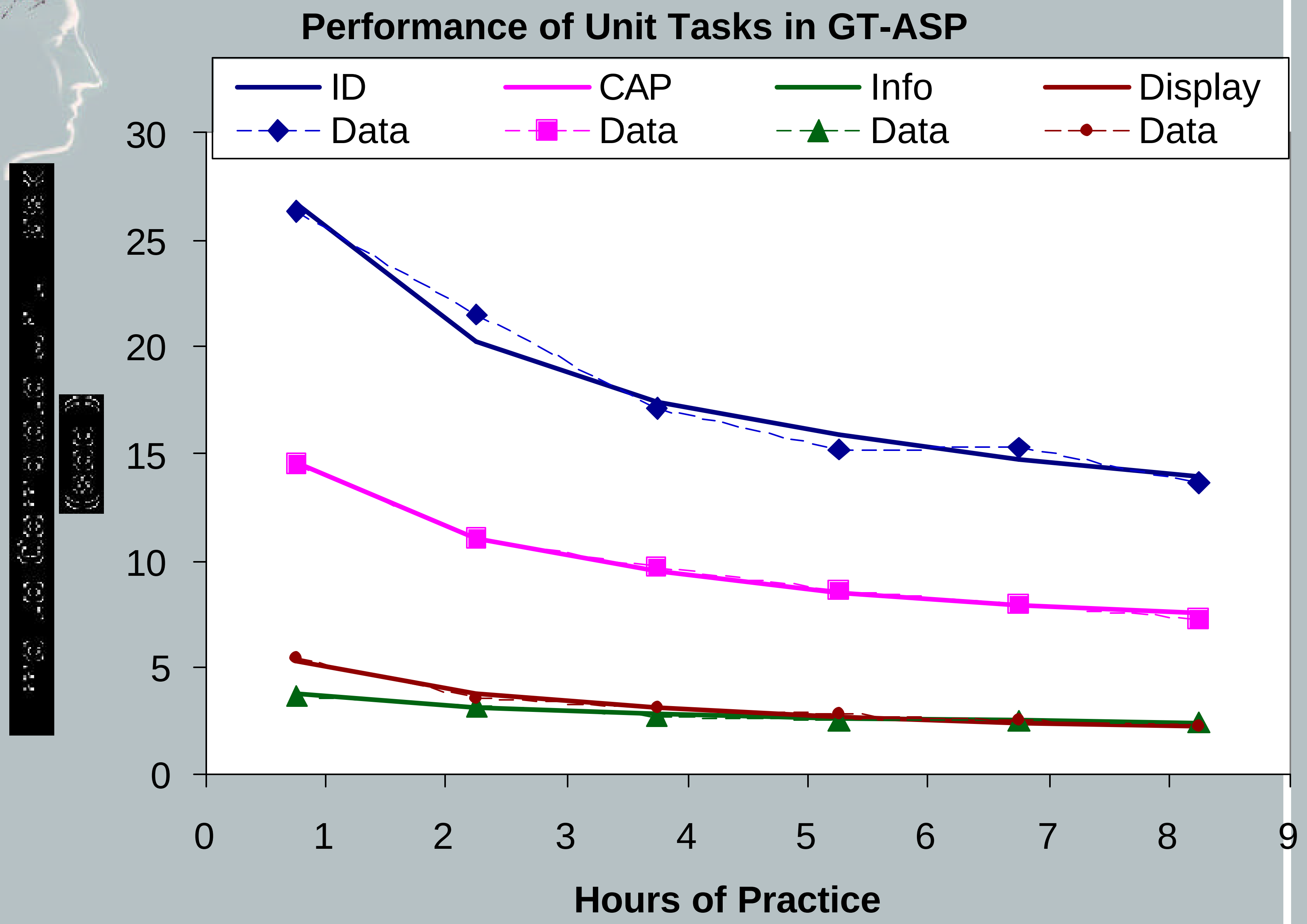
General
CSE 281

POSIT B 0065 0017

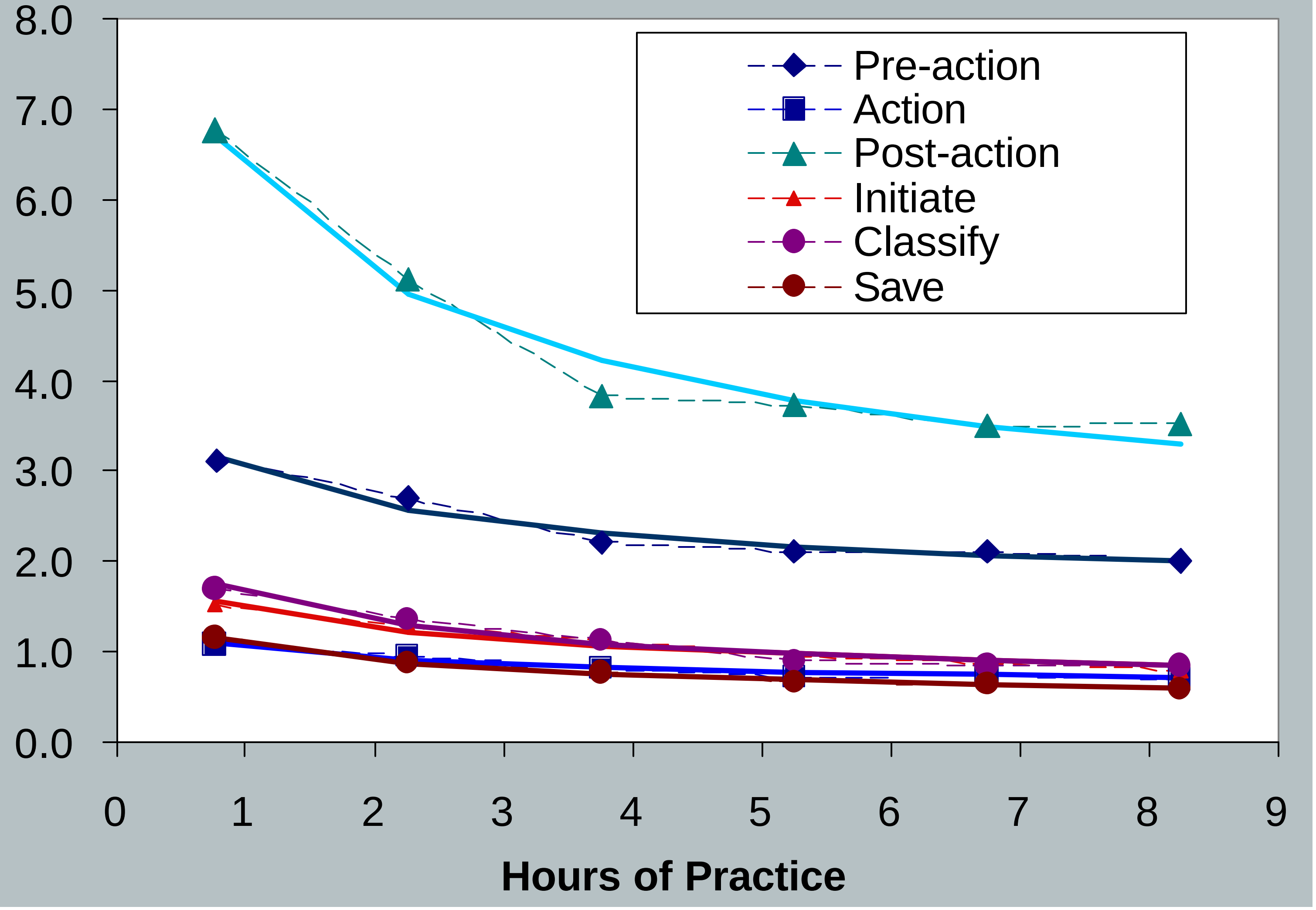
___Scores___
Last: 0
Total: 0

STANDALONE
Problem Time
10:15 :06

Ball Tab
G 0039 0055
Range 67.7
Brg 215
from Ctr


Performance of Unit Tasks in GT-ASP
ID
CAP
Info
Display
Data
Data
Data
Data
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
Hours of Practice

Subgoals of ID Unit Task

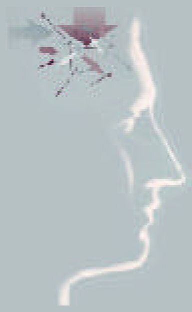


State of Work on CMU-ASP

1. Studied acquisition of task in large Georgia Tech data base and CMU students
2. Implemented a version of the task that ACT-R can interact with and which delivers just-in-time over-the-shoulder instructions.
3. Spawned off task subsets that we have subjected to intensive experimental study

Version for Demo

1. A simplified version of the task, only do identification and none of the supportive actions nor any military actions.
2. No rules of Engagement given, but scored based on compliance with simple ones
3. Instructed on how to do the task but not how score is determined
4. Tutor running to teach the rule (May or may not hear the tutor)
5. Demo



Types of Learning Involved

1. **Instructions --** Encode initial and over-the-shoulder instructions.
2. **Declarative Practice** – Perfect the the initial declarative information.
3. **Perceptual** – Learn where the critical information is on the screen.
4. **Procedural** – Learn new procedures that no longer require retrieving the instructions for the task.
5. **Strategic –** Learn with experience the relative value of the various information sources and the most efficient methods for achieving goals.
6. **Reflective** – Reason about performance and (for instance, vectoring caps) and adjust behavior. This probably depends heavily on prior knowledge.

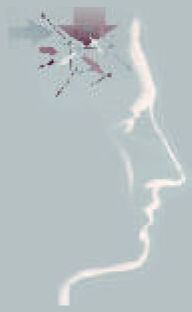
Production Compilation(Taatgen & Anderson): The Basic Idea

IF reading the word for a paired-associate test
and a word is being attended
THEN retrieve the associate of the word
and note trying to recall

Recall Vanilla-7

IF recalling for a paired-associate test
and an associate has been retrieved with response N
THEN type N
and note the answer is being typed

Results in:
IF reading the word for a paired-associate test
and “vanilla” is being attended
THEN type “7”
and note the answer is being typed



The Basic Plan for Learning from Instruction

- Instructions are encoded as declarative structures characterizing the sequence of goals that must be achieved.
- There are a set of production rules that will interpret any such sequence of instructions.
- Production compilation will convert this into task specific procedures.
- As an aside we solve the mystery of task instructions that has haunted Experimental Psychology.

Instructions for CMU-ASP

1. At the beginning set the radius to 128 nm and begin identifying
2. In identifying select the closest track.
3. To select a track you move your mouse to it and click.
4. First try identifying a track by checking whether it fits the altitude and speed profile of a commercial aircraft and if it does classify it as such.
5. Second, try identifying a track by requesting an EWS signal and if it gives one classify it accordingly.
6. A commercial aircraft is classified with a primary intent of “assumed friendly” and a air ID of “non-military”.
7. To classify a plane with primary intent X and air ID Y, select the following sequence of menu options: “track”,”update”,”primary”, X, “air”, Y, “save”.
8. Commercial airlines fly between 20,000 and 40,000 feet
9. Commercial airlines fly between 350 and 500 nm.
10. To change the radius to X select the following sequence of menu options: “display”, “radius”, X, “execute”.
11. To select a menu option hit the f-key that corresponds to it.

Etc.

Production Compilation: Applied to CMU-ASP

IF trying to retrieve a rule to achieve a goal
and rule for achieving that goal has been retrieved
THEN retrieve the first step of that rule
and note trying to recall the first step

The first step in classifying a plane is to select “track”.

IF trying to retrieve the first step of a goal
and a step has been retrieved involving a subgoal
THEN change goal to trying to achieve that subgoal
and try to retrieve a rule to achieve that subgoal

Results in:

IF trying to retrieve a rule for classifying a plane
THEN set a subgoal to select “track”
and try to retrieve rule for selecting “track”.

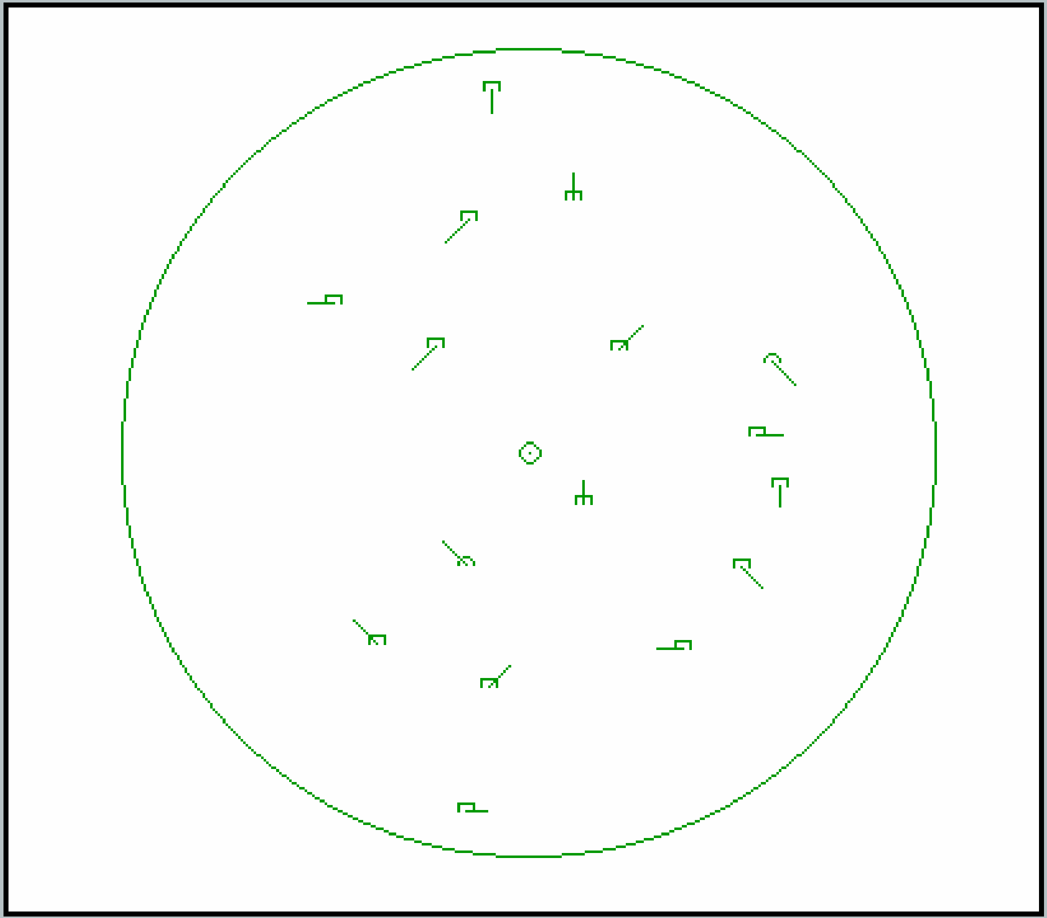
Eventually production rules are learned like:

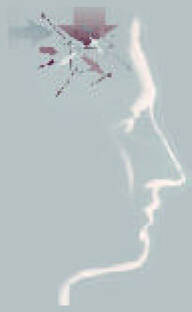
```
IF trying to retrieve a rule for classifying a plane
THEN set a hit F1
     and set a subgoal to select “update”
```

The model moves from taking over 100 seconds to classify a plane to less than 10 seconds. Part of the learning depends on production compilation and part of it depends on location learning.

It sort of learns like participants but does their learning really correspond in detail to the predictions of production compilation?

Pick the Closest CAP to Home Ship





Craig Haimson’s Experiment: Lets look at the Simplest Possible Subtask: Hooking the Closest Relevant Plane

72 participants, 384 trials	1 Close 1 Far	4 Close 4 Far	4 Close 12 Far
Target Close To Home Ship	821 msec.	1033 msec.	1060 msec.
Target Far From Home Ship	995 msec.	2028 msec.	2671 msec.

Note: ACT-R already has a theory of visual search and motor movement and so except for the learning aspects it is a zero-parameter model for this task that would make a-priori predictions.

Instructions for Haimson-ASP

1. At the beginning find home ship, click it, and find target.
2. To find the target, attend to the unattended item closest to home, and click it if it is of the correct shape.
3. If the item is not of the correct shape repeat 2.

Eventually learn productions like:

IF looking for the target track
and the currently attended object is not of the correct shape
THEN shift attention to the closest unattended location

This production will compete with its more primitive parents according to its experienced utility in solving the problem. The critical parameters are its rate of learning its utility and the noise in utility estimation. The parameters are set so that it will take roughly 50 trials before the better production comes to reliably dominate.

Utility = PG - C + Noise where

P is the probability of success, always 1 here

G is the value of goal, doesn’t matter since P = 1

C is the cost of production measured in seconds

Noise is a parameter set with ? = 1

$$C\ ?\ \frac{W * \mathrm{Pr}ior\,\mathrm{cos}t\ ?\ N * Average\mathrm{cos}t}{W\ ?\ N}$$

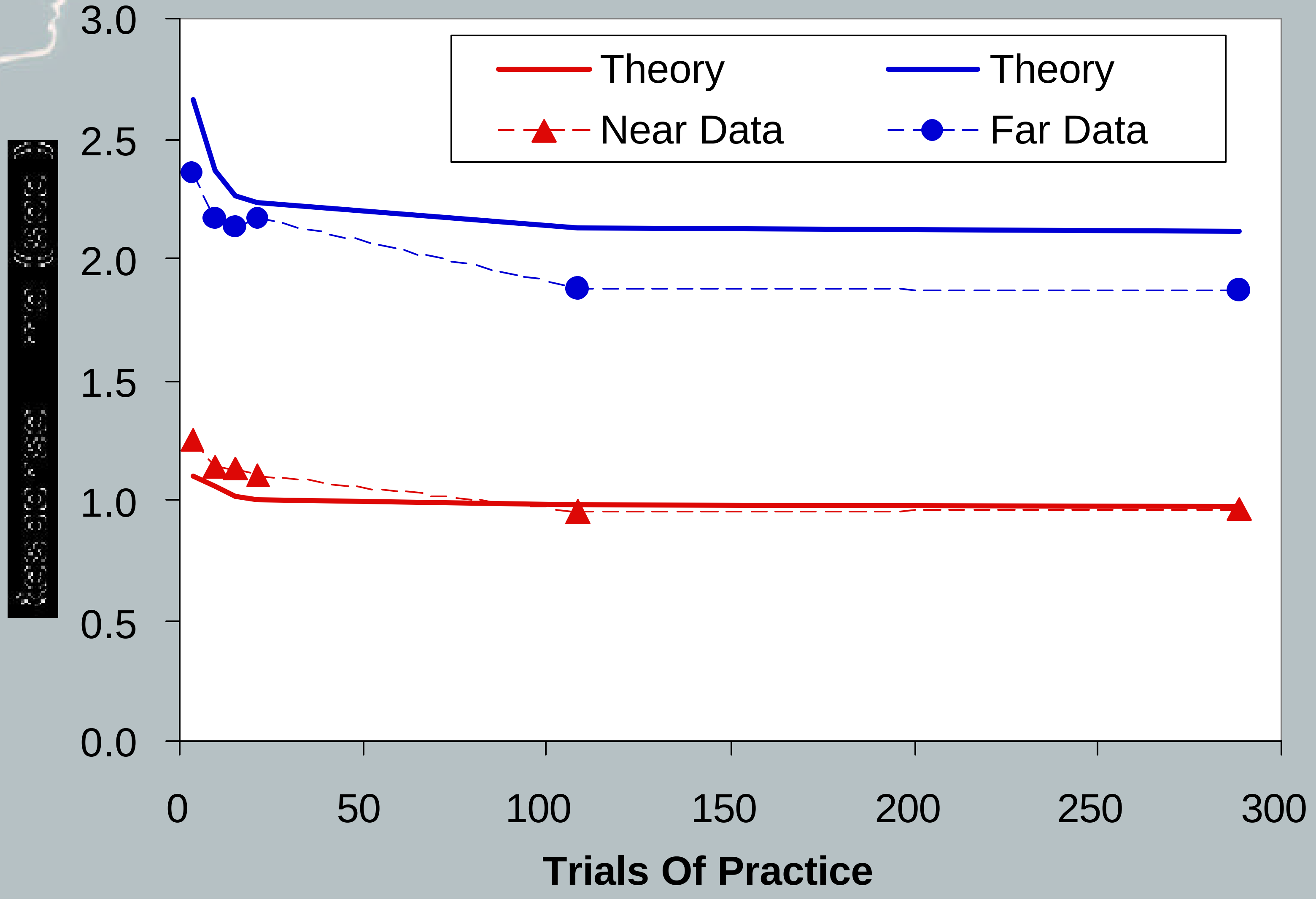
Priorcost is the cost of first production in pair, >= 4.0.

W is the inertia, a parameter set at 50

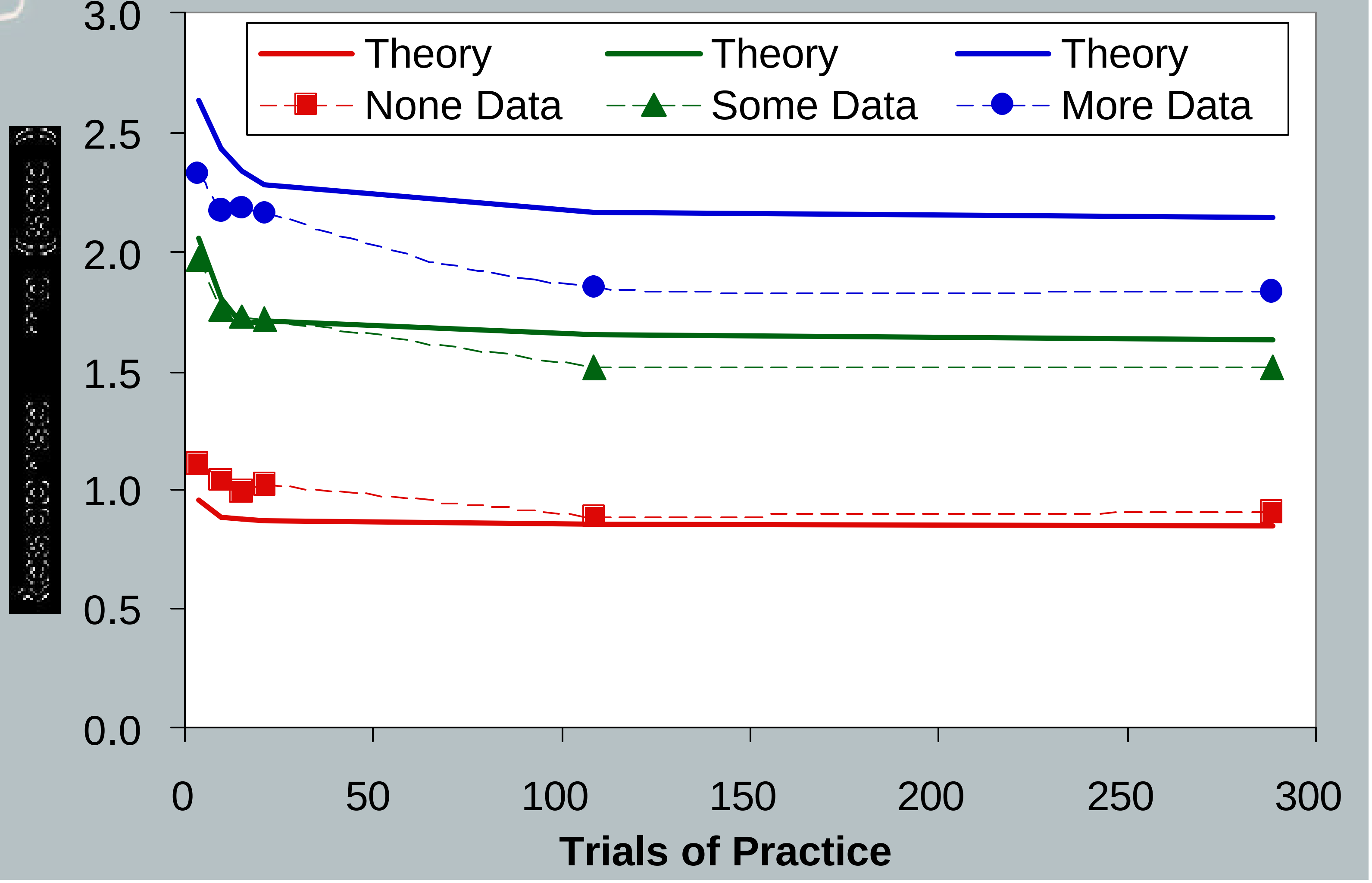
N is the amount of experience

Averagecost is the average of experience, < 2 for best.

Distance, Averaged over Number of Distractors



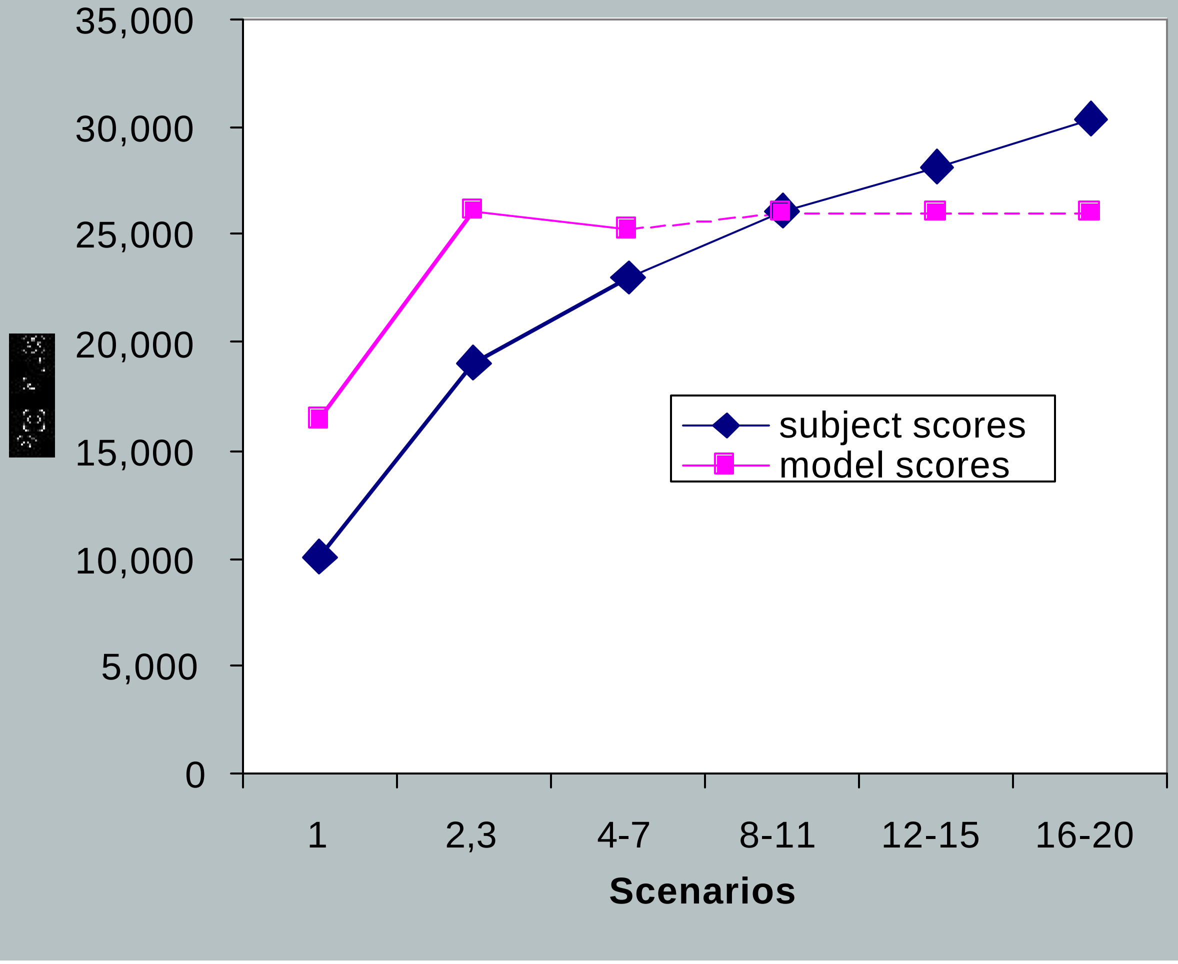
Number of Distractors, Averaged over Distance



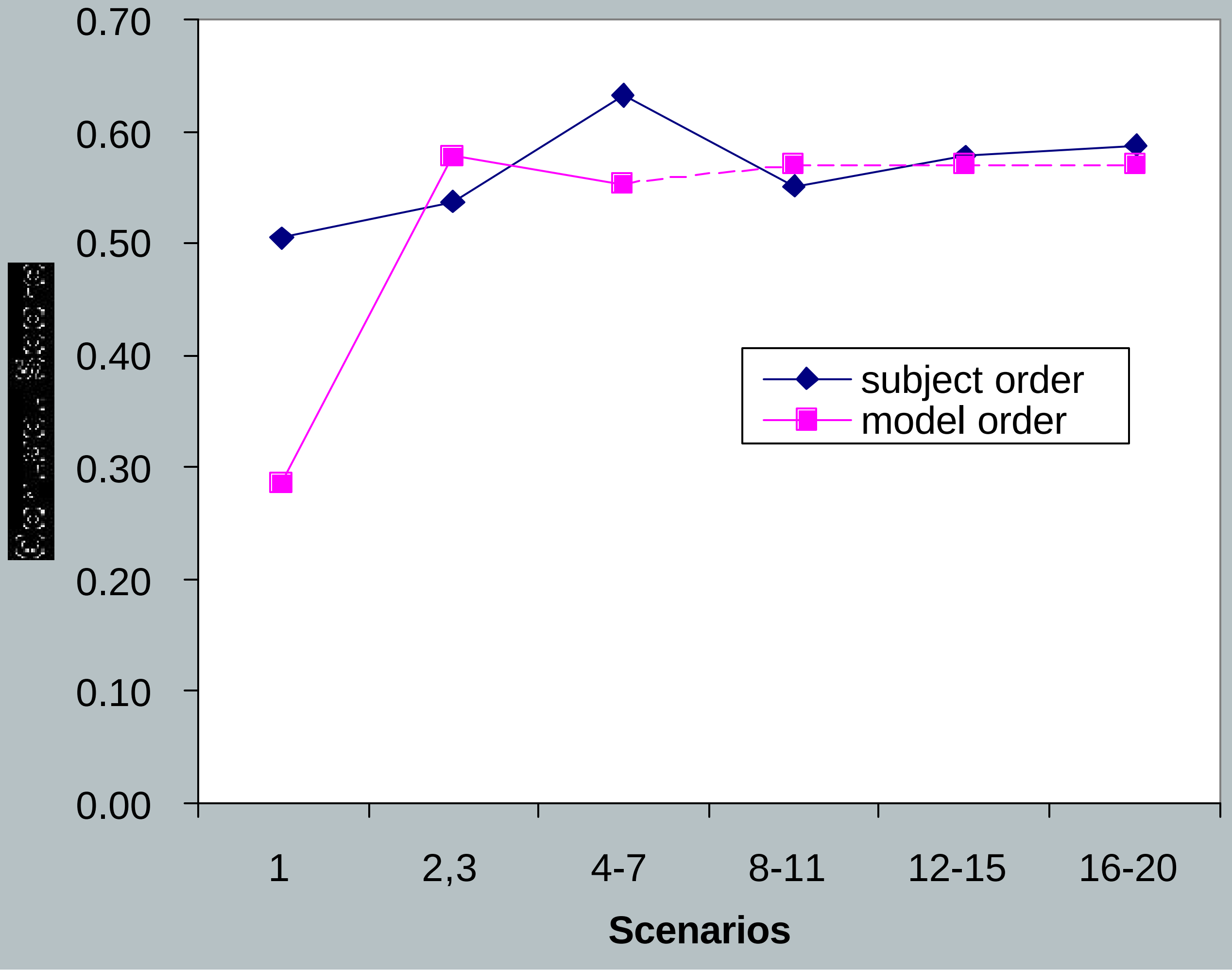
Currently Working on Simulating the Instructional Experiment

1. First trying to get performance to match up in no-instruction condition
2. Turned learning on full blast to see maximum effect
3. Working on model even as we speak.

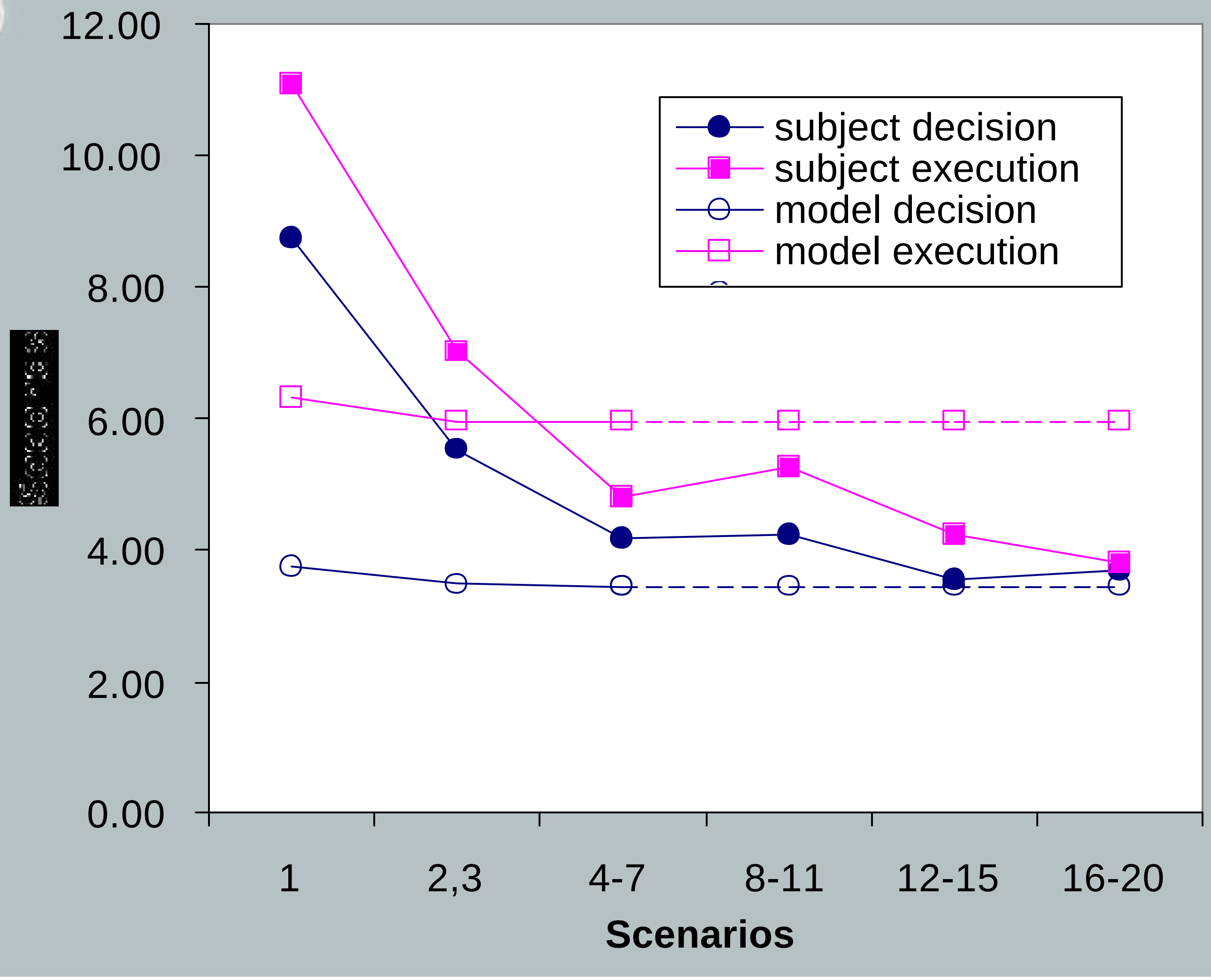
Point Performance



Choosing the Best Tracks



Action Times



Things To Do

1. More efficient/aggressive motor programming, creating new motor programs. Ask Dan
2. Processing auditory input
3. Processing over-the-shoulder instructions
4. Discoveries/deductions -- e.g., one does not check profile for fast tracks, hold hand over F9 key