Procedural learning in the control of a dynamic system

Danilo Fum and Andrea Stocco

Laboratorio di Sistemi Cognitivi
Dipartimento di Psicologia
Università di Trieste

Overview

- Learning in Sugar Factory
- Computational models
- The experiments
- A new model
- Conclusions

Sugar Factory

- The Drosophyla of learning in dynamic systems
- People have to keep the production *P* of a simulated sugar factory on a target value by allocating an appropriate number of workers *W* to the job
- Discrete number of states [1..12] for both *P* and *W*, and discrete computational steps
- The system dynamics is controlled by the relation

  $$P_t = 2W_t - P_{t-1} + ?$$

- The task is made difficult by the existence of random noise ?, uniformly distributed with values {-1, 0, +1}.

Sugar Factory

- For a more realistic interpretation, the values of *W* are multiplied by 100 (hundreds of workers), and the values of *P* by 1000 (tons of sugar)

- Resulting values of *P* less than 1000 are simply set to 1000, and values exceeding 12000 are set to 12000

- Participants are given the goal to produce a target value of 9000 tons of sugar on each trial.

Sugar Factory

- Typical phenomenon: dissociation between task performance and associated verbalizable knowledge
- Initially assumed as a case for the existence of a separate implicit learning system
- The phenomenon could be explained by assuming that people rely on memorized records (instances) of their interactions with the system.

Computational models

Two instance-based models have been developed to explain the behavior of participants in the SF task:

- Dienes & Fahey (1995)
- Wallach and coworkers (Lebiere, Wallach & Taatgen, 1998; Taatgen & Wallach, 2002)

Both models show a good fit with data, with no model being clearly superior

Wallach's model relies on the ACT-R architecture and requires fewer additional assumptions.

D&F in a nutshell

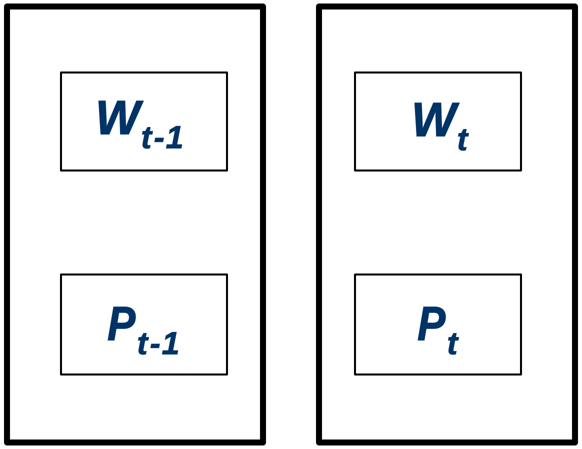
- Whenever, starting from a situation $<W_{t-1}, P_{t-1}>$, an action W_t leads to a sugar production P_t that is correct (within the limits of ?), both the action and the situation are stored in memory

- More particularly, two records (instances) are created:
  a. the first storing the link between the current sugar production and the action that lead to it: $<P_{t-1}, W_t>$
  b. the second storing the link between the previous workforce and the action: $<W_{t-1}, W_t>$

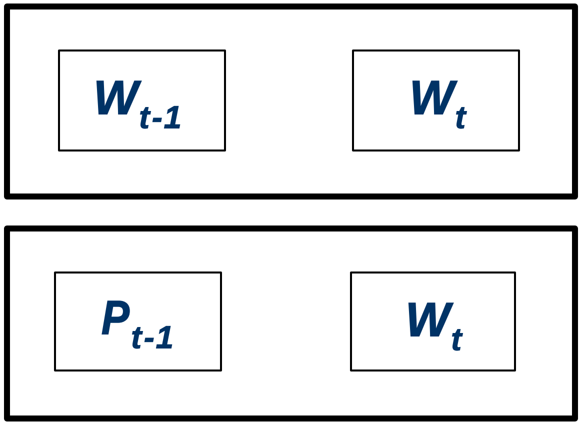
- Only instances referring to successful interactions are stored (this is a critical assumption!)

D&F in a nutshell

What you see



What you get



D&F (cont.)

- On any given trial, a random selection between the instances that match the current situation is performed, and the associated action is executed

  For instance, let us suppose that

  $W_{t-1} = 600$

  $P_{t-1} = 8000$

  Among all the instances matching the patterns:

  $<600, W_t>$

  $<8000, W_t>$

  one is randomly picked out, and the W_t associated with the selected instance is chosen as the workforce for the trial.

D&F (cont.)

D&F noted that 86% of the first ten input values could be explained by assuming the following behavior:

- **if *P* is above/below target, then set *W* to a value that is different from the previous one by {0, ±100, ±200}**
- **if *P* is on the target, then set *W* to a value that is different from the previous one by {-100, 0, +100}**
- **for the very first trial, start with a *W* in the range [700..900].**

D&F: Some assumptions

- To replicate this behavior, D&F had to stuff into the model a number *N* of instances covering each of the three cases (*N* is a critical parameter of the model!)

- D&f assume the storage of only successful instances

- D&F use a “loose” criterion of correctness by considering as successful a situation in which *P* was within ± 1000 tons from the target value.

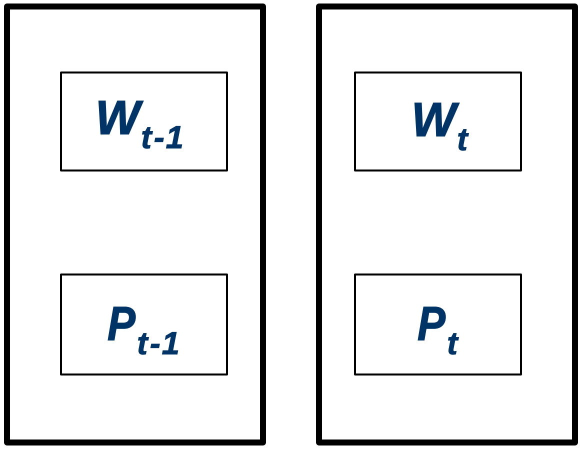
Wallach’s model

- Grounded on the ACT-R architecture
- Encodes every interaction episode, irrespective of the result, e.g.:

```
(transition1239
   ISA          transition
   state        3000      ; $P_{t-1}$ the old production
   worker       8         ; $W_t$
   production   12000)    ; $P_t$ the current production
```

Wallach’s model

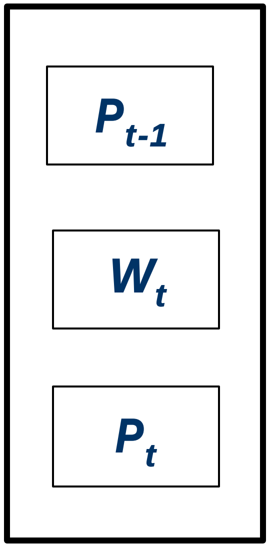
What you see



W_{t-1} W_t

P_{t-1} P_t

What you get



P_{t-1}

W_t

P_t

Wallach’s model

- The participants’ performance is explained by assuming a match between the present situation and the encoding of instances experienced in the past

- On each trial, a memory search is initiated, based on the current situation and the target value of 9000 tons, in order to retrieve an appropriate workforce value

- Instances that only partially match the retrieval pattern are penalized by lowering their activation proportionally to the degree of mismatch.

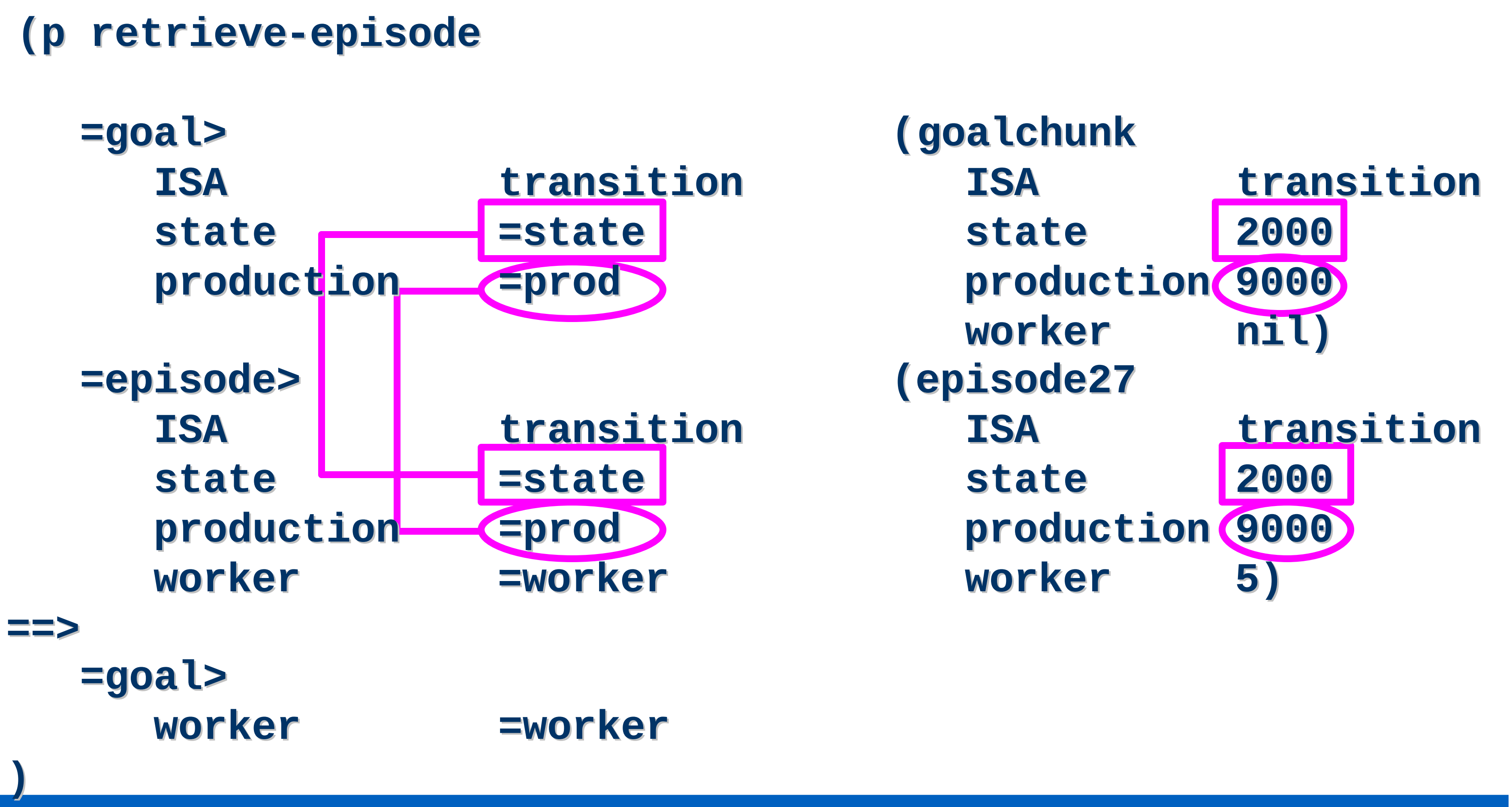
Wallach’s model

The fundamental production:

```
(p retrieve-episode

   =goal>
      ISA              transition
      state            =state
      production       =prod

   =episode>
      ISA              transition
      state            =state
      production       =prod
      worker           =worker
==>
   =goal>
      worker           =worker
)
```

Wallach's model

A perfect match:

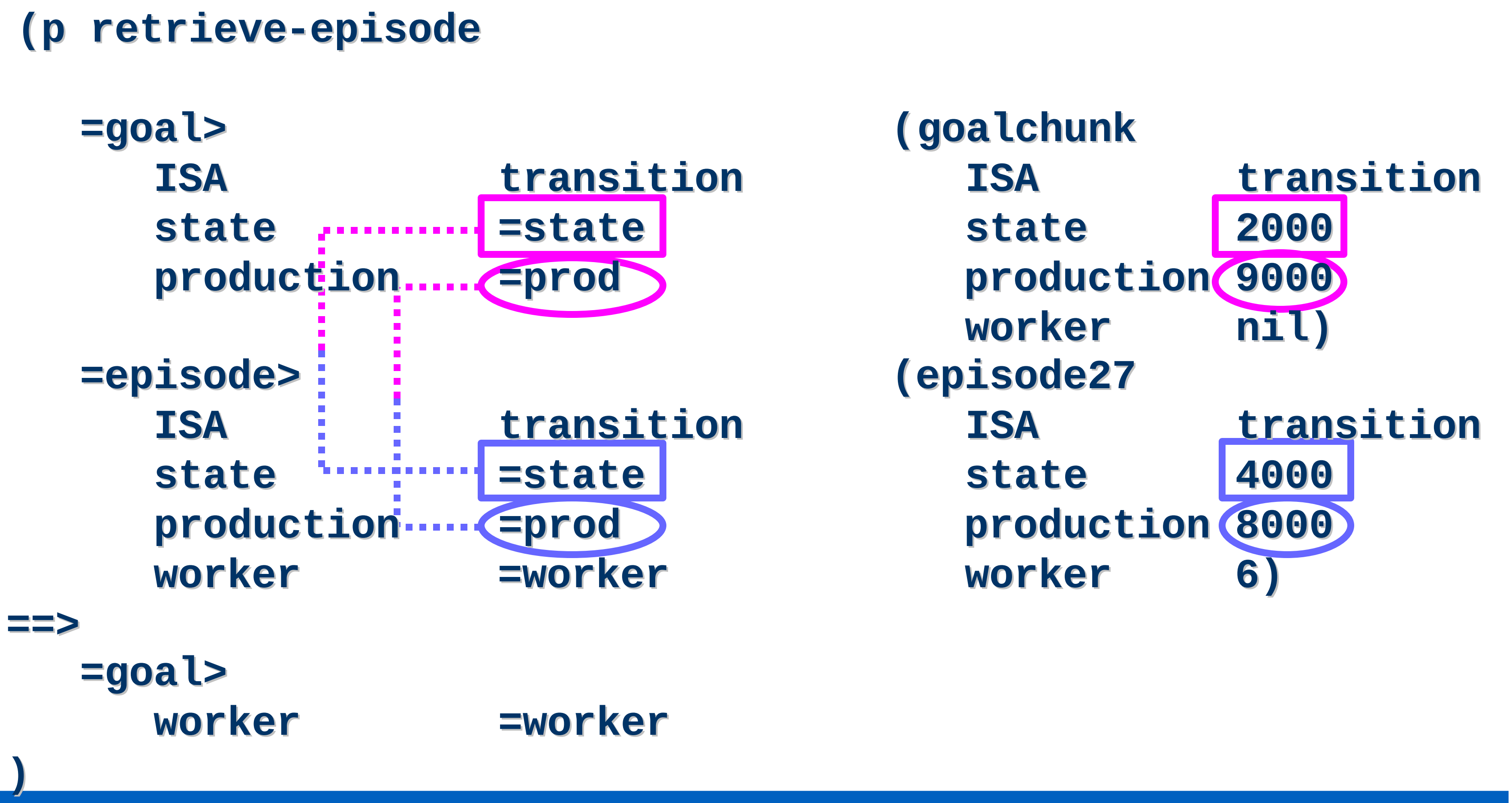


```
(p retrieve-episode

   =goal>
      ISA              transition
      state            =state
      production       =prod

   =episode>
      ISA              transition
      state            =state
      production       =prod
      worker           =worker
==>
   =goal>
      worker           =worker
)
```

```
(goalchunk
   ISA          transition
   state        2000
   production   9000
   worker       nil)
(episode27
   ISA          transition
   state        2000
   production   9000
   worker       5)
```

Wallach’s model

A partial match:



```
(p retrieve-episode

   =goal>                                   (goalchunk
      ISA            transition                ISA         transition
      state          =state                    state       2000
      production     =prod                     production  9000
                                               worker      nil)
   =episode>                                (episode27
      ISA            transition                ISA         transition
      state          =state                    state       4000
      production     =prod                     production  8000
      worker         =worker                   worker      6)
==>
   =goal>
      worker         =worker
)
```

Wallach’s model

In case of partial match, a penalty is computed according to the formula:

$$penalty \; ? \; MP \, ?_{s} (1 \, ? \, sim(required_{s}, actual_{s}))$$

where:

MP is the mismatch penalty parameter, and

s is each slot in the matched chunk.

Wallach's model

To calculate the similarity of two numbers *a* and *b* representing the sugar productions in respective instance chunks, the following function (Lebiere, 1999) is used:

$$sim(a,b) \text{ ? } \frac{min(a,b)}{max(a,b,1)}$$

Wallach’s model

The ugly duck production:

```
(p worker-guess-rule
    =goal>
       ISA    transition
       state  =state
    ==>
    !eval!   (setf *worker-guess*
                  (+ (signum
                       (-  9000 (get-number-value  =state)))
                     (1- (random 3))
                     *worker*))
 )
```

D&F (cont.)

D&F noted that 86% of the first ten input values could be explained by assuming the following behavior:

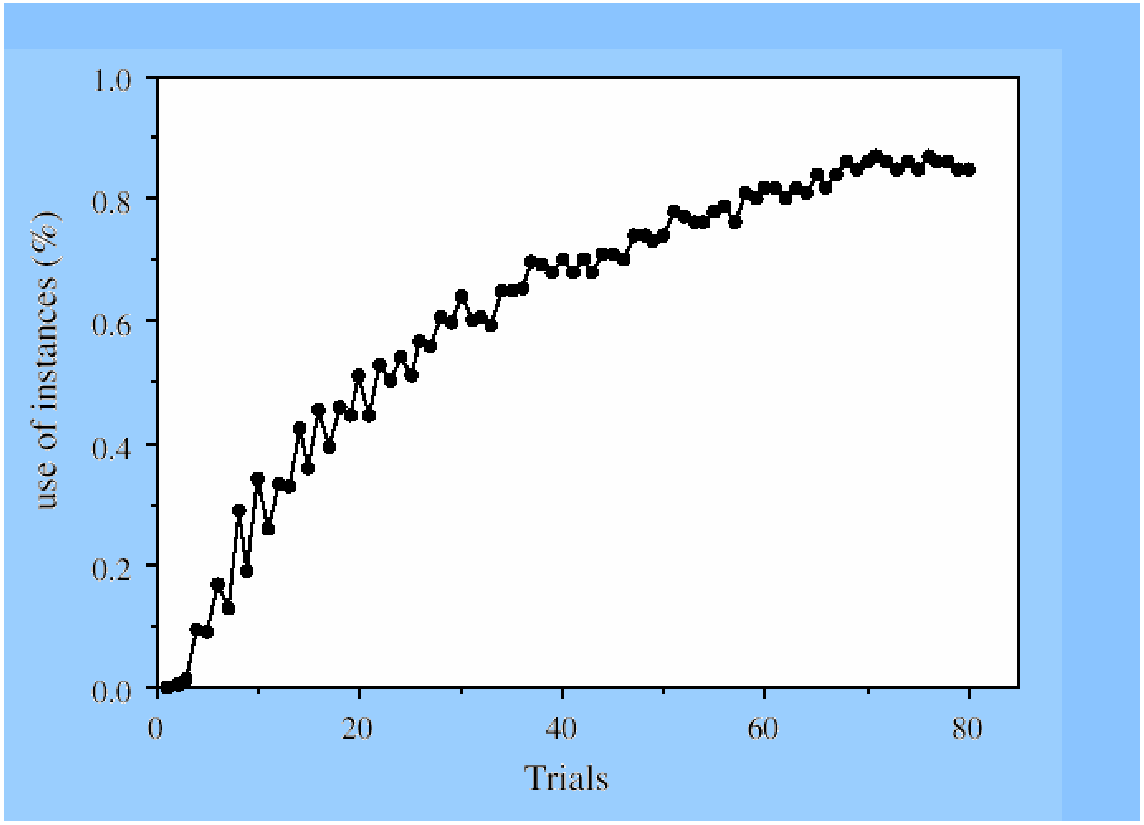
- if *P* is above/below target, then set *W* to a value that is different from the previous one by {0, ±100, ±200}

- if *P* is on the target, then set *W* to a value that is different from the previous one by {-100, 0, +100}

- for the very first trial, start with a *W* in the range [700..900].

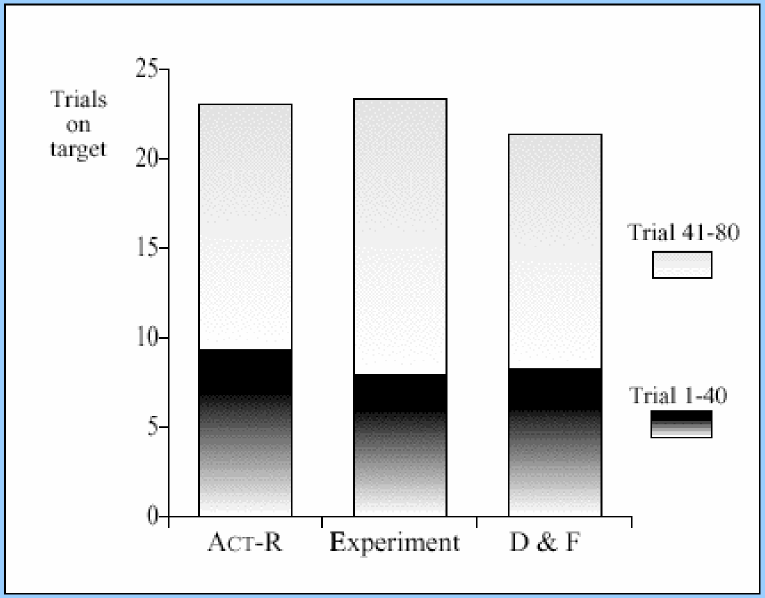
Wallach's model

The use of instances increases over time (from Lebiere, Wallach, & Taatgen, 1998)



Wallach vs. D&F

Both models show a pretty good fit with data (from: Lebiere, Wallach, & Taatgen, 1998)



Overview

- Learning in Sugar Factory
- Computational models ← You are here
- The experiments
- A new model
- Conclusions

Playing the science game

We tried to falsify Wallach's model by testing two of its main assumptions:

- the interaction episode as the basic knowledge unit
- the declarativeness of the acquired knowledge.

Pilot A

Two blocks of 40 trials each.

First block:

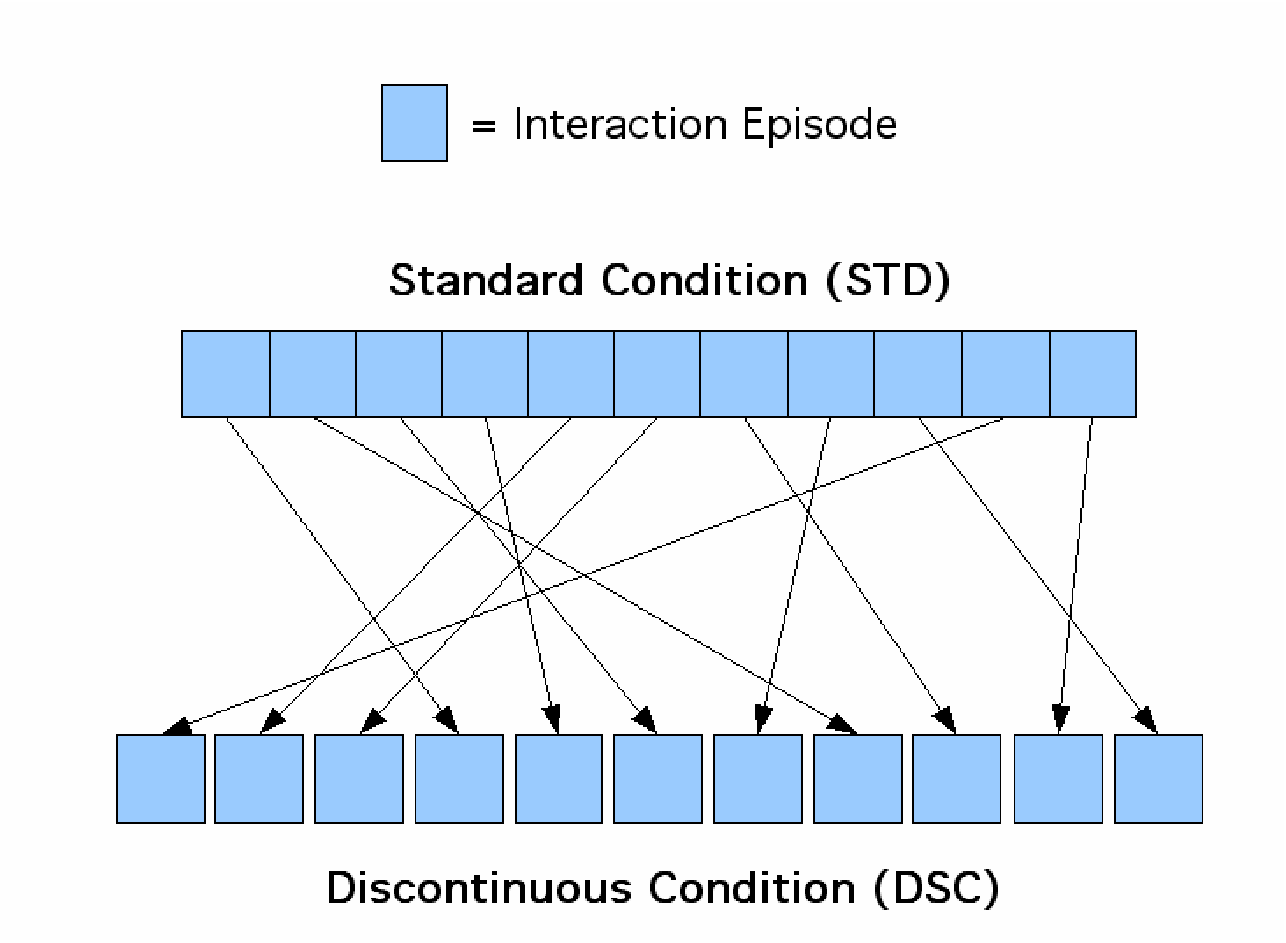
- STD (standard): the output of each interaction episode constitutes the input for the following episode

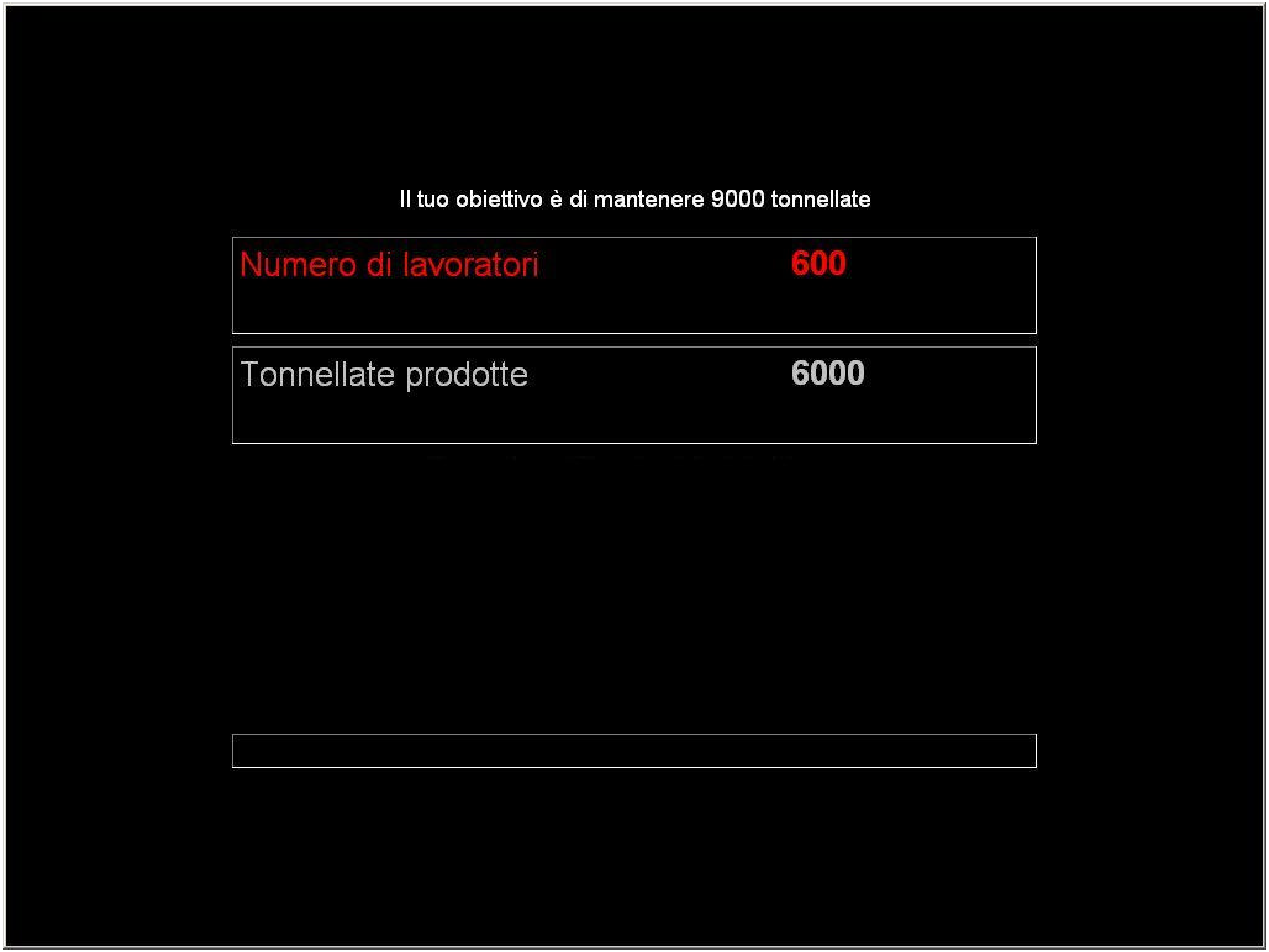
- DSC (discontinuous): every interaction episode is discrete, but the participants experience the same situations of the STD group.

Second block:

- STD

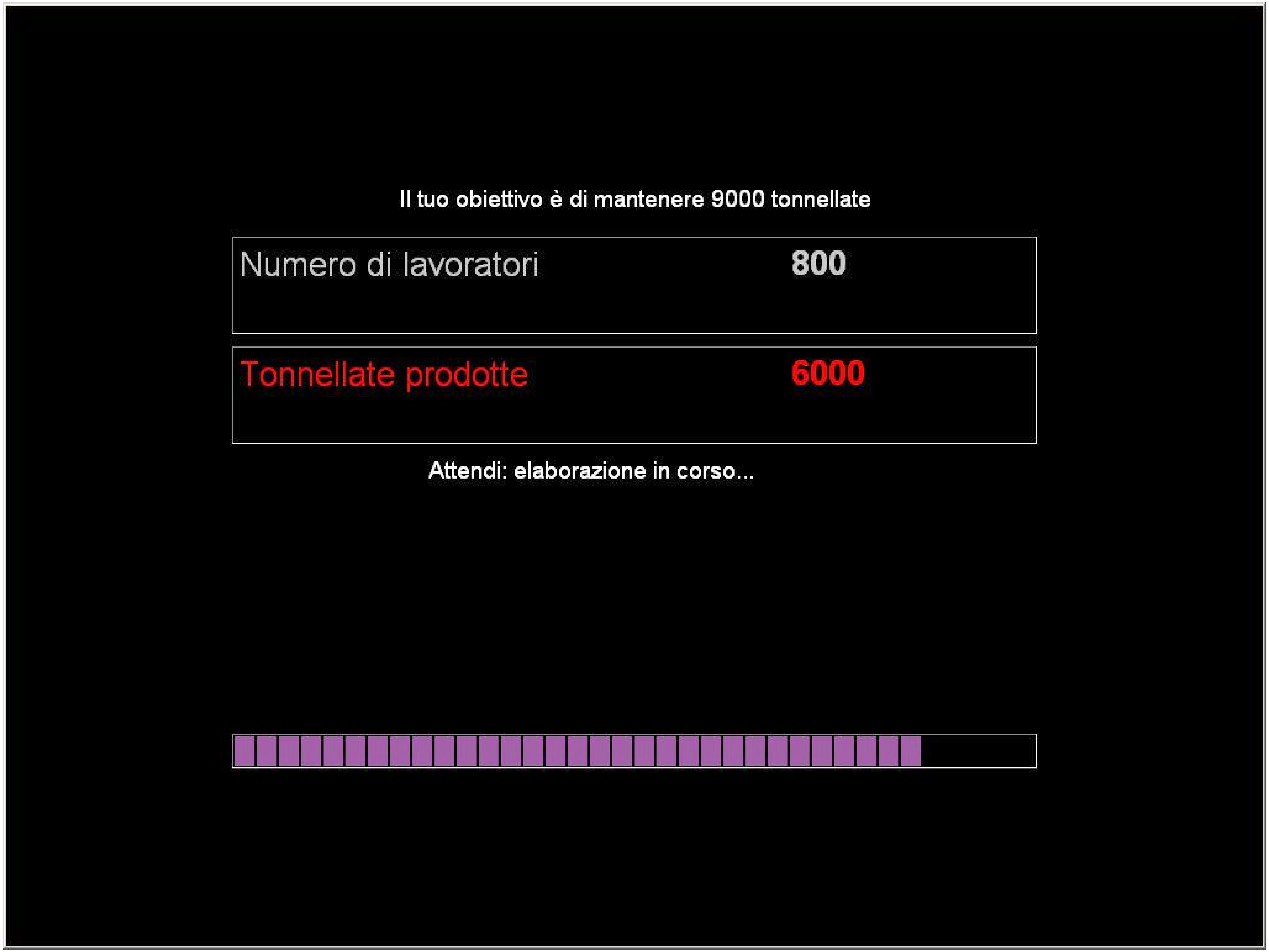
Pilot A


= Interaction Episode
Standard Condition (STD)
Discontinuous Condition (DSC)



Il tuo obiettivo è di mantenere 9000 tonnellate

Numero di lavoratori	**600**
Tonnellate prodotte	**6000**

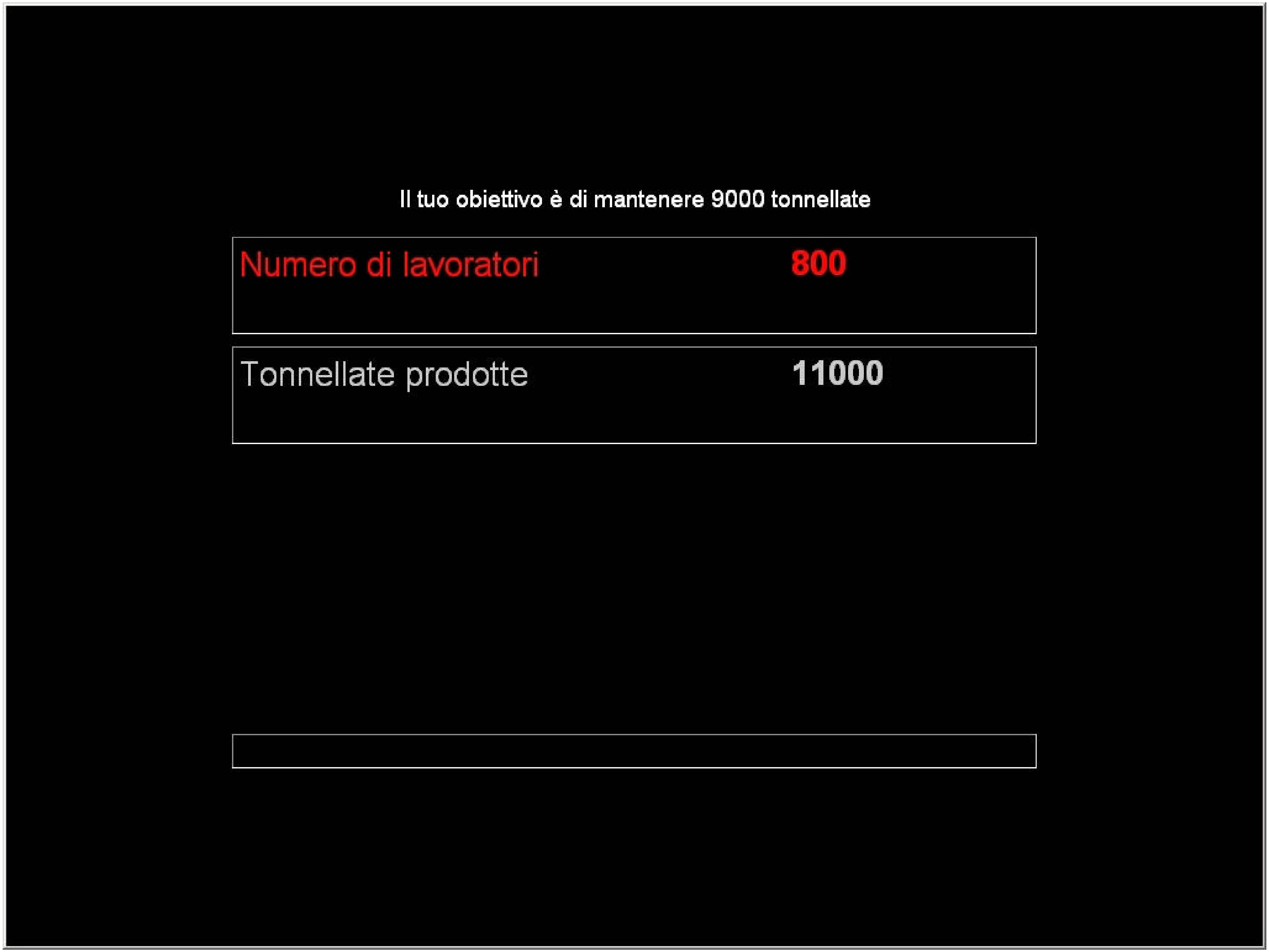


Il tuo obiettivo è di mantenere 9000 tonnellate

Numero di lavoratori 800

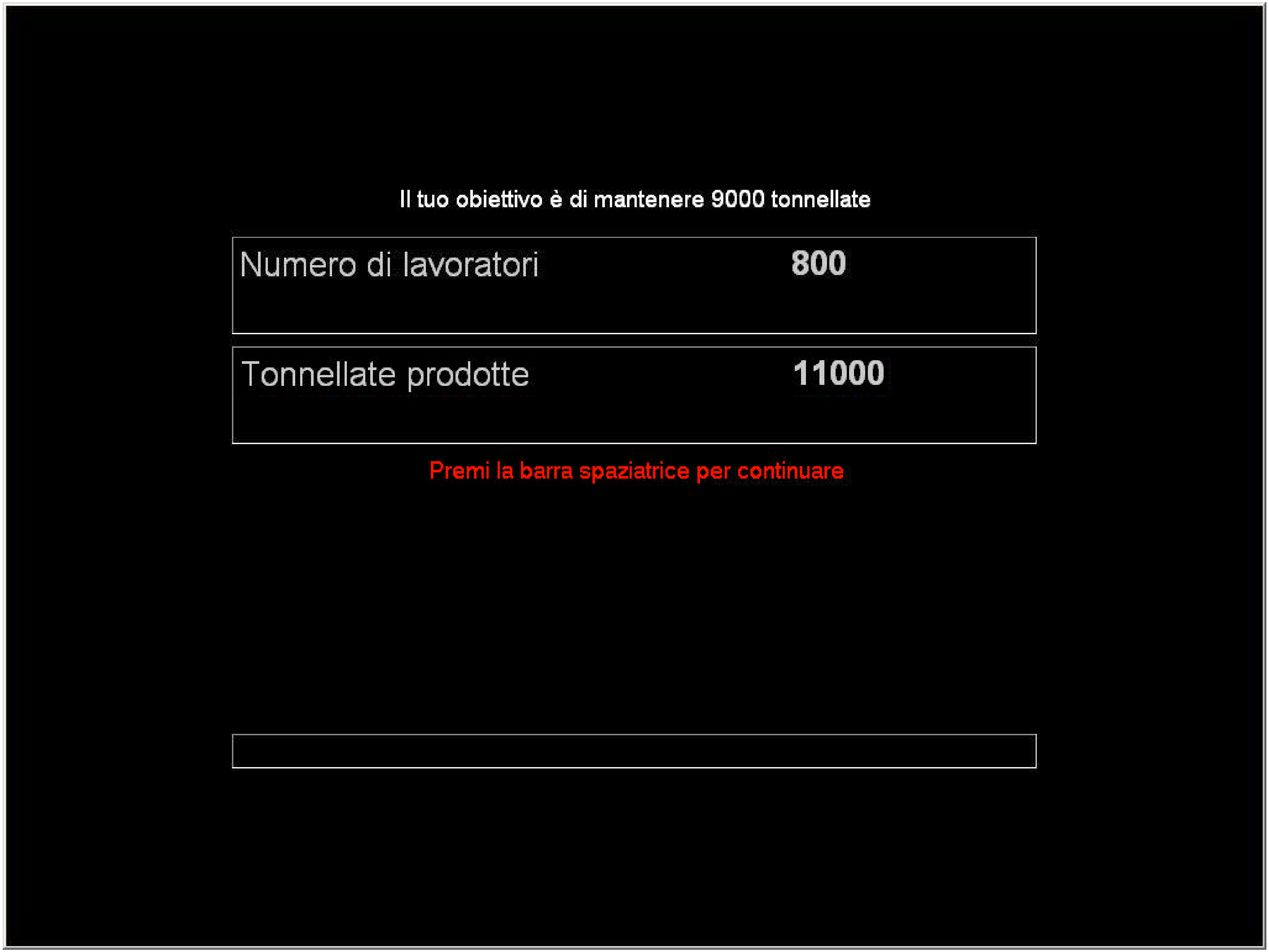
Tonnellate prodotte 6000

Attendi: elaborazione in corso...



Il tuo obiettivo è di mantenere 9000 tonnellate

Numero di lavoratori	800
Tonnellate prodotte	11000



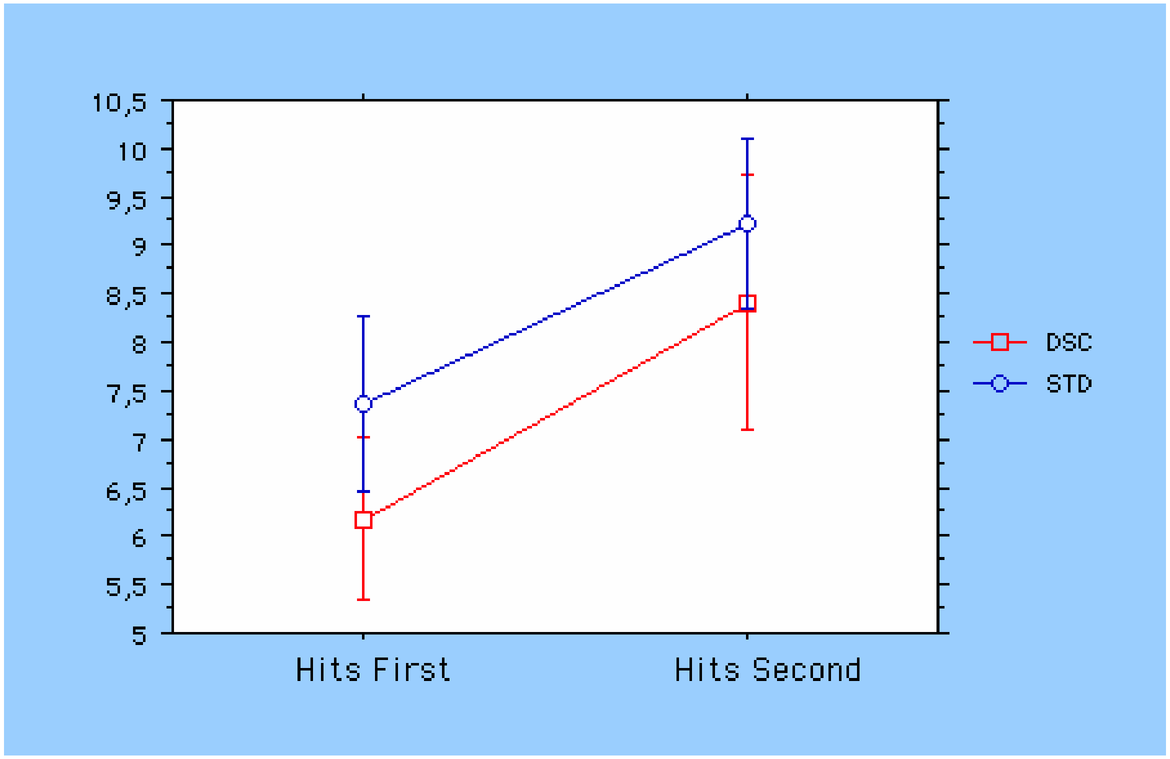
Il tuo obiettivo è di mantenere 9000 tonnellate

Numero di lavoratori **800**

Tonnellate prodotte **11000**

Premi la barra spaziatrice per continuare

Pilot A: results



Pilot B

Two blocks of 40 trials each.

First block:

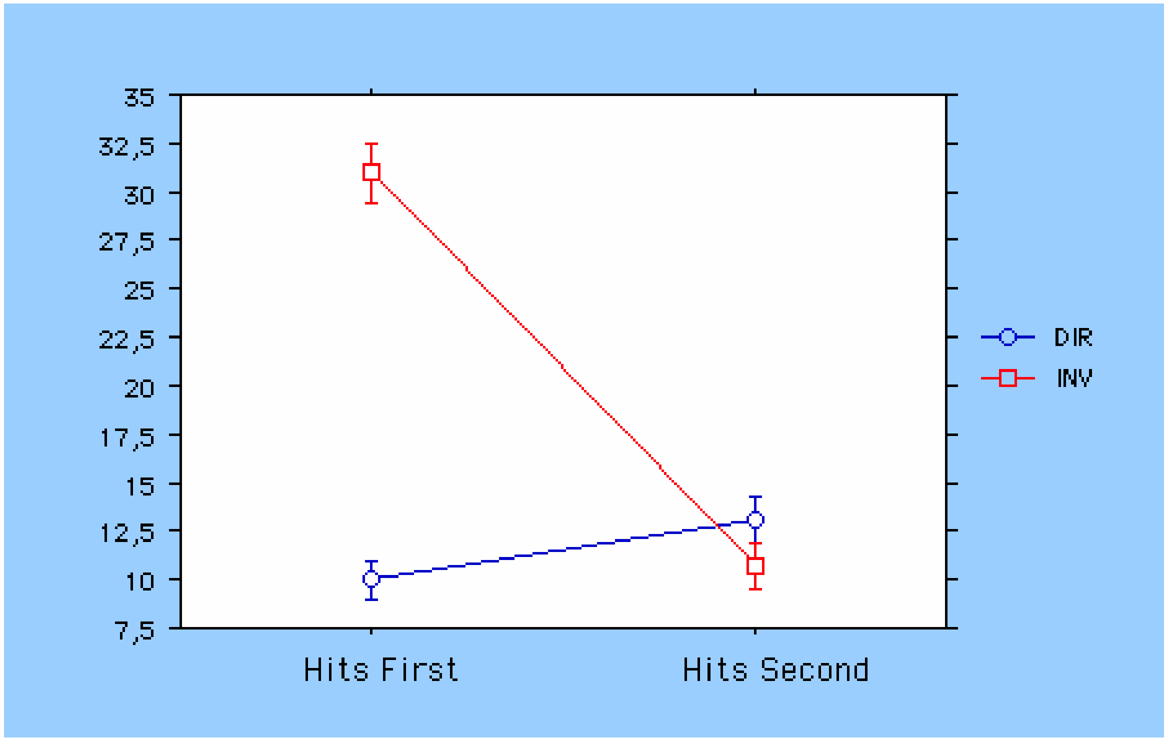
- DIR (direct): set *W* to control *P*, with a target value *P* = 3000
- INV (inverse): set *P* to control *W*, with a target value *W* = 200

P target value has been changed in order to equate the success probability in the two conditions.

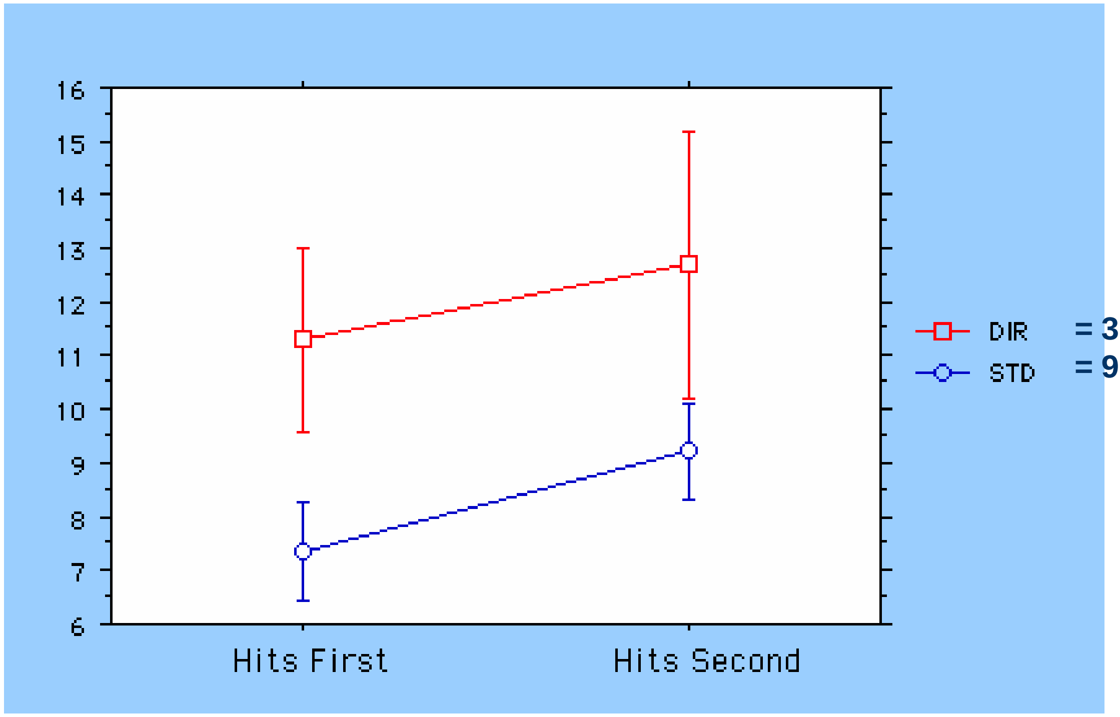
Second block:

- DIR

Pilot B: results



A new effect



= 3000
= 9000

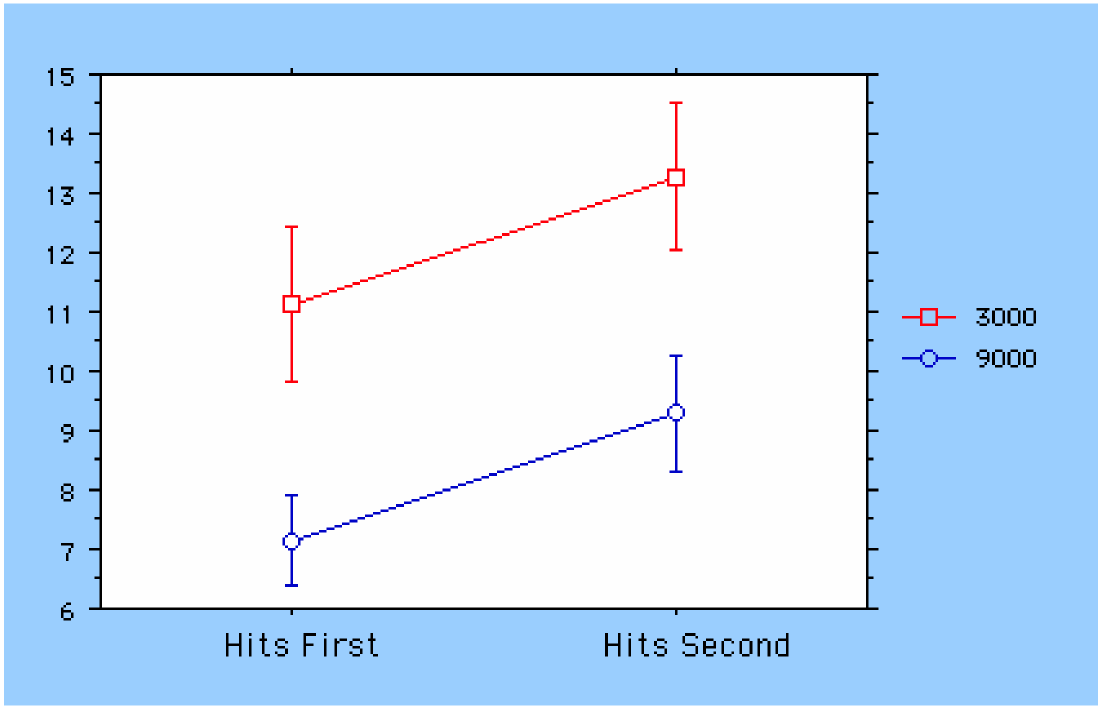
Experiment #1

Two conditions:

- target *P* = 3000
- target *P* = 9000

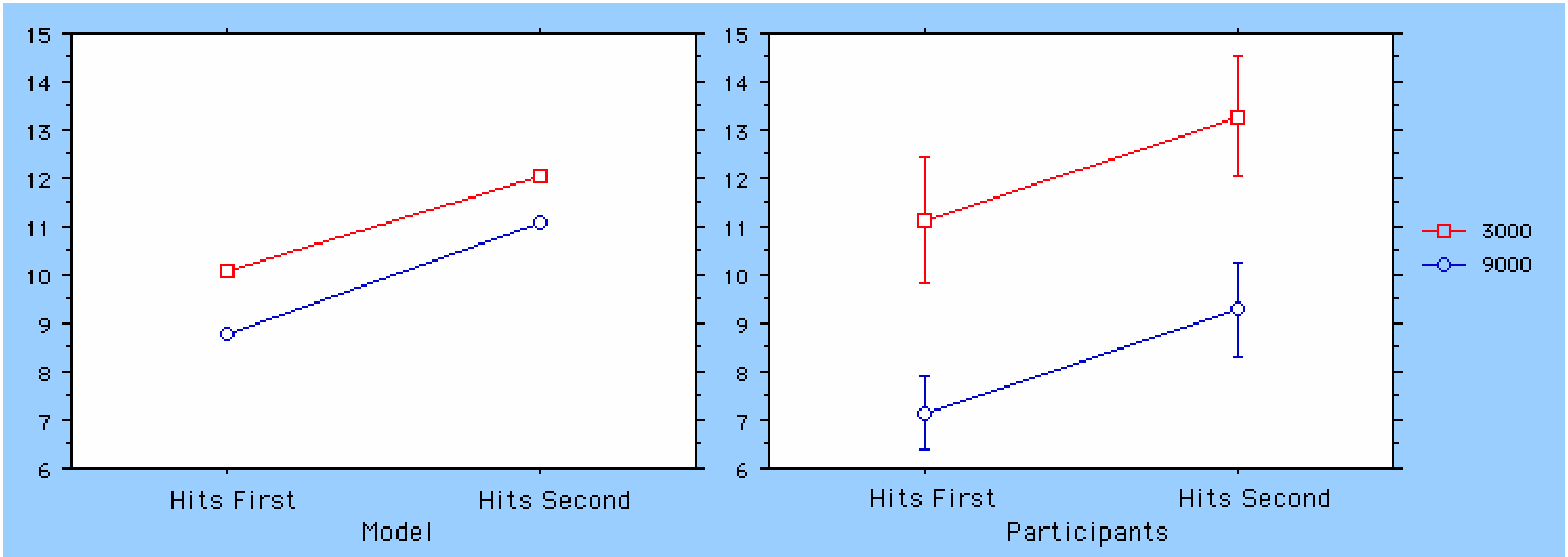
to test the new effect.

Experiment #1



Surprise!

The new effect is predicted by Wallach’s model (but not by the D&F’s)!



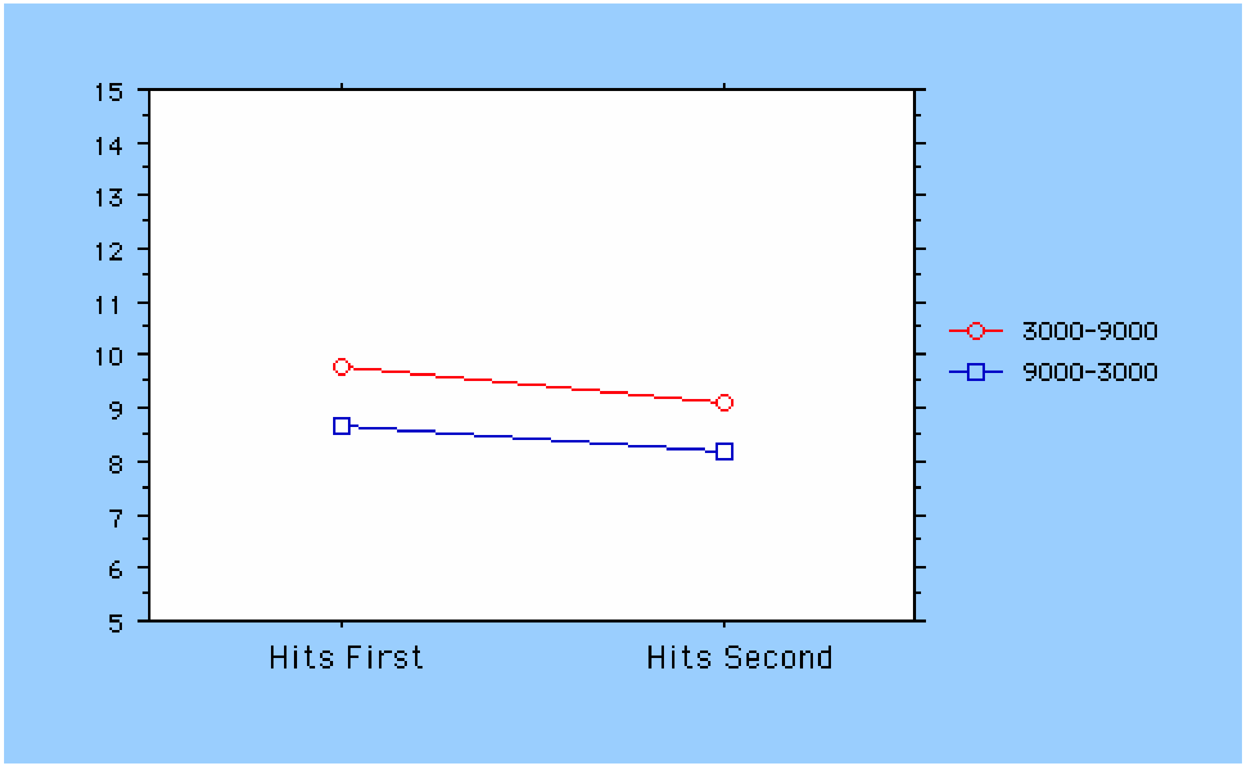
Experiment #2

What happens if we switch the target between the first and the second phase?

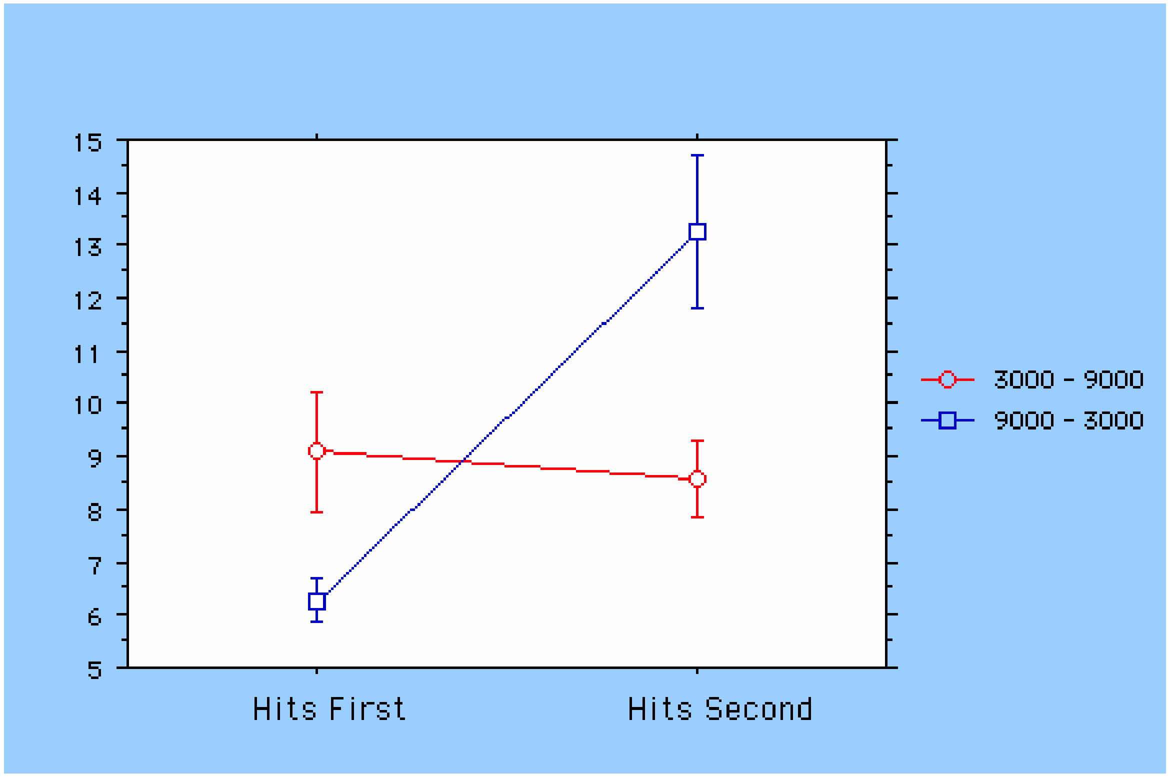
Two conditions:

- 3000 - 9000
- 9000 - 3000

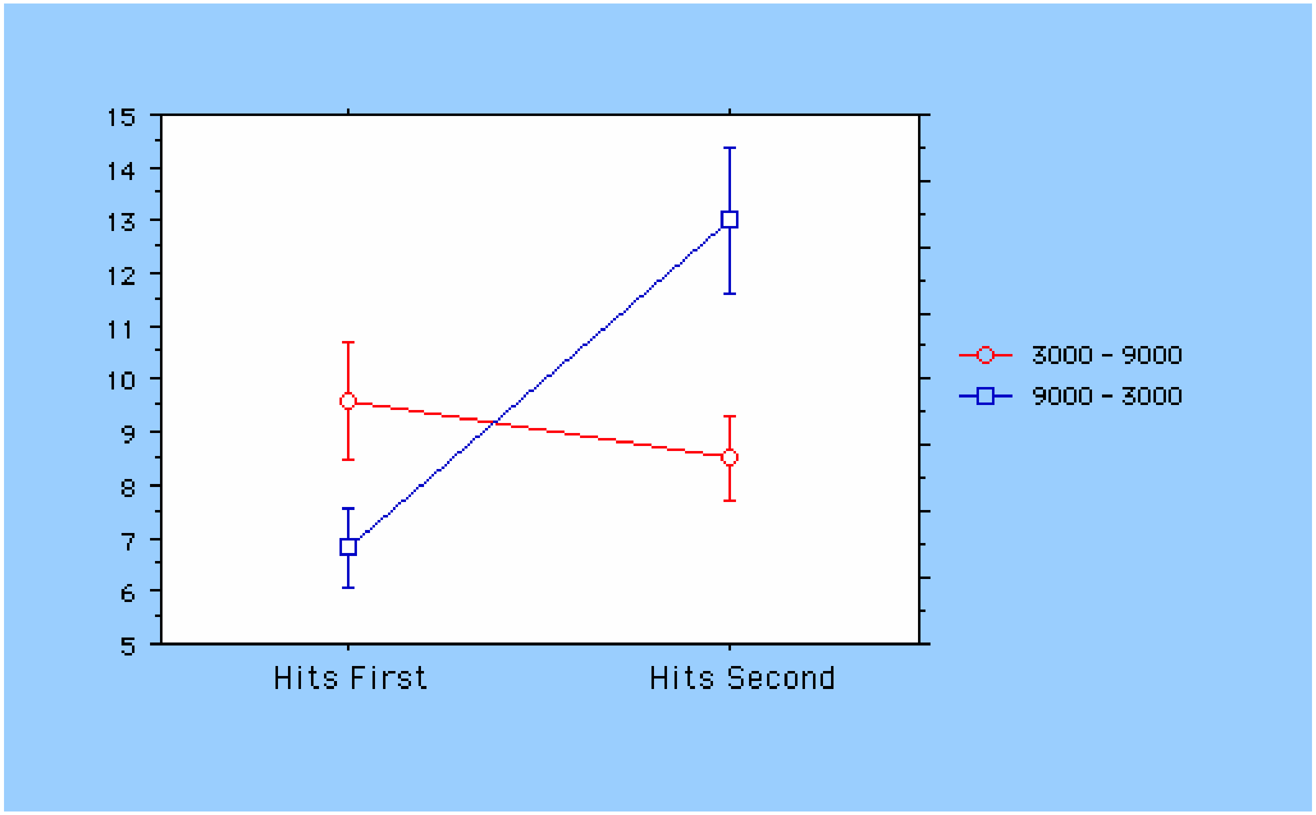
Wallach’s predictions



The results


15
14
13
12
11
10
9
8
7
6
5
Hits First
Hits Second
3000 - 9000
9000 - 3000

The replication



A new model

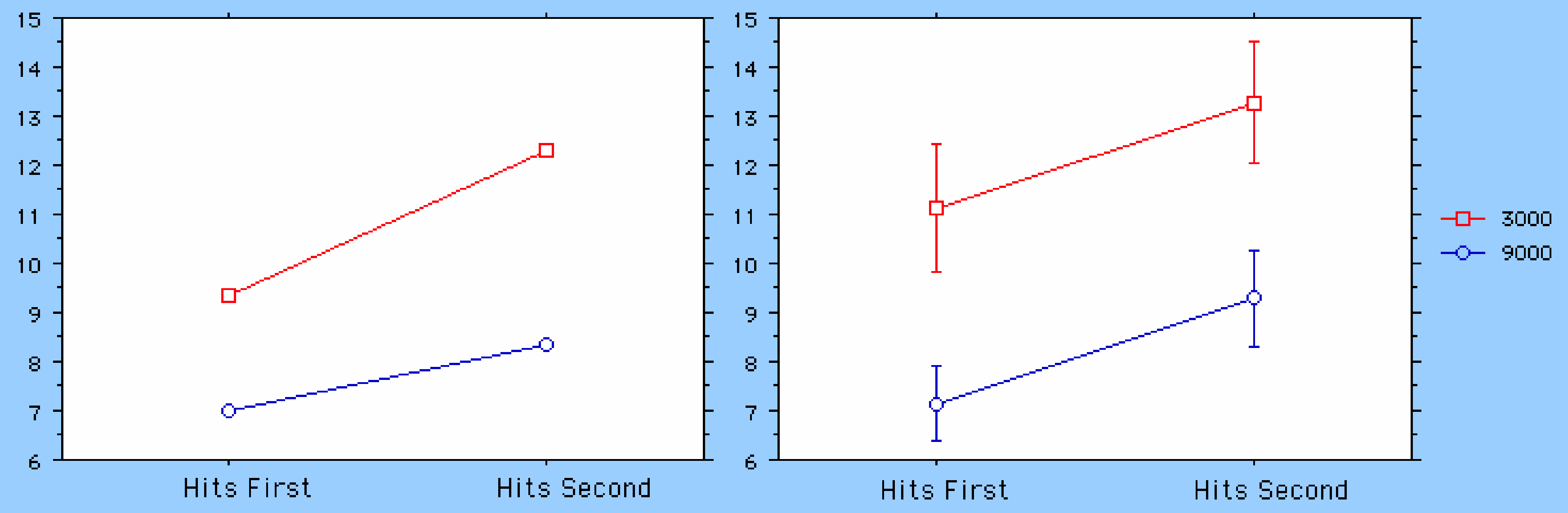
Six productions compete according to a pure ACT-R learning scheme:

- **choose-random**: choose a random value between 1 and 12
- **repeat-choice**: repeat the previous *W* value
- **stay-on-hit**: if you hit the target, keep the same *W* value
- **pivot-around-target**: choose as *W* the value of the target (plus noise)
- **jump-up**: if your production *P* is below the target increase the value of *W*
- **jump-down**: if your production *P* is above the target decrease the value of *W*.

Experiment #1

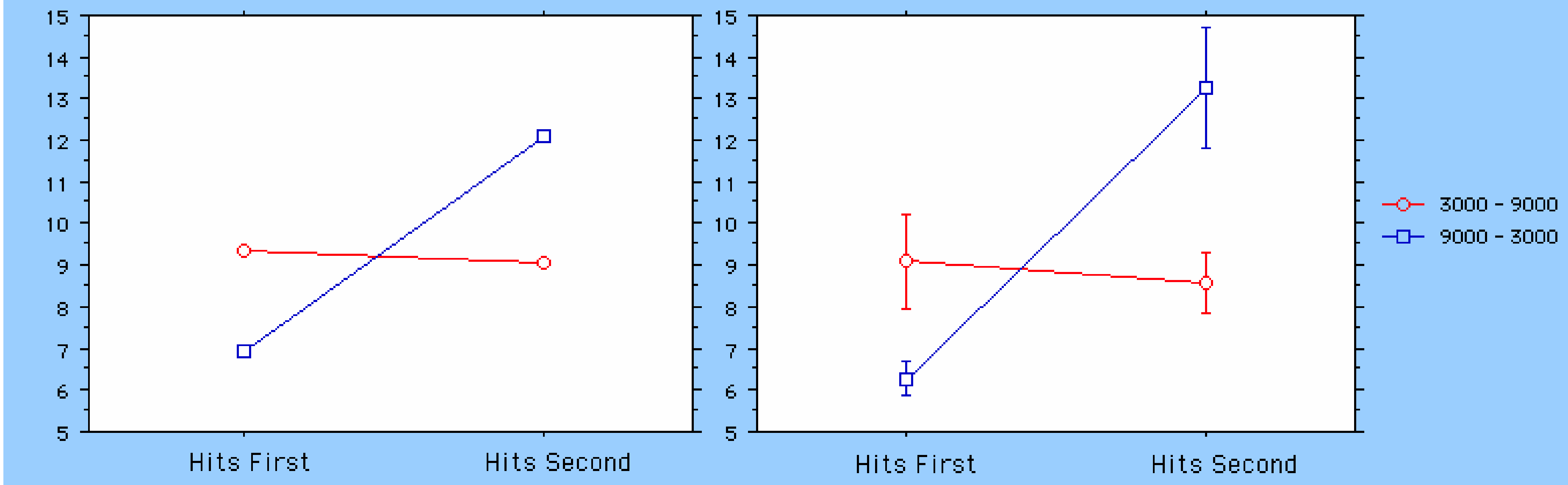
Model

Data



Experiment #2

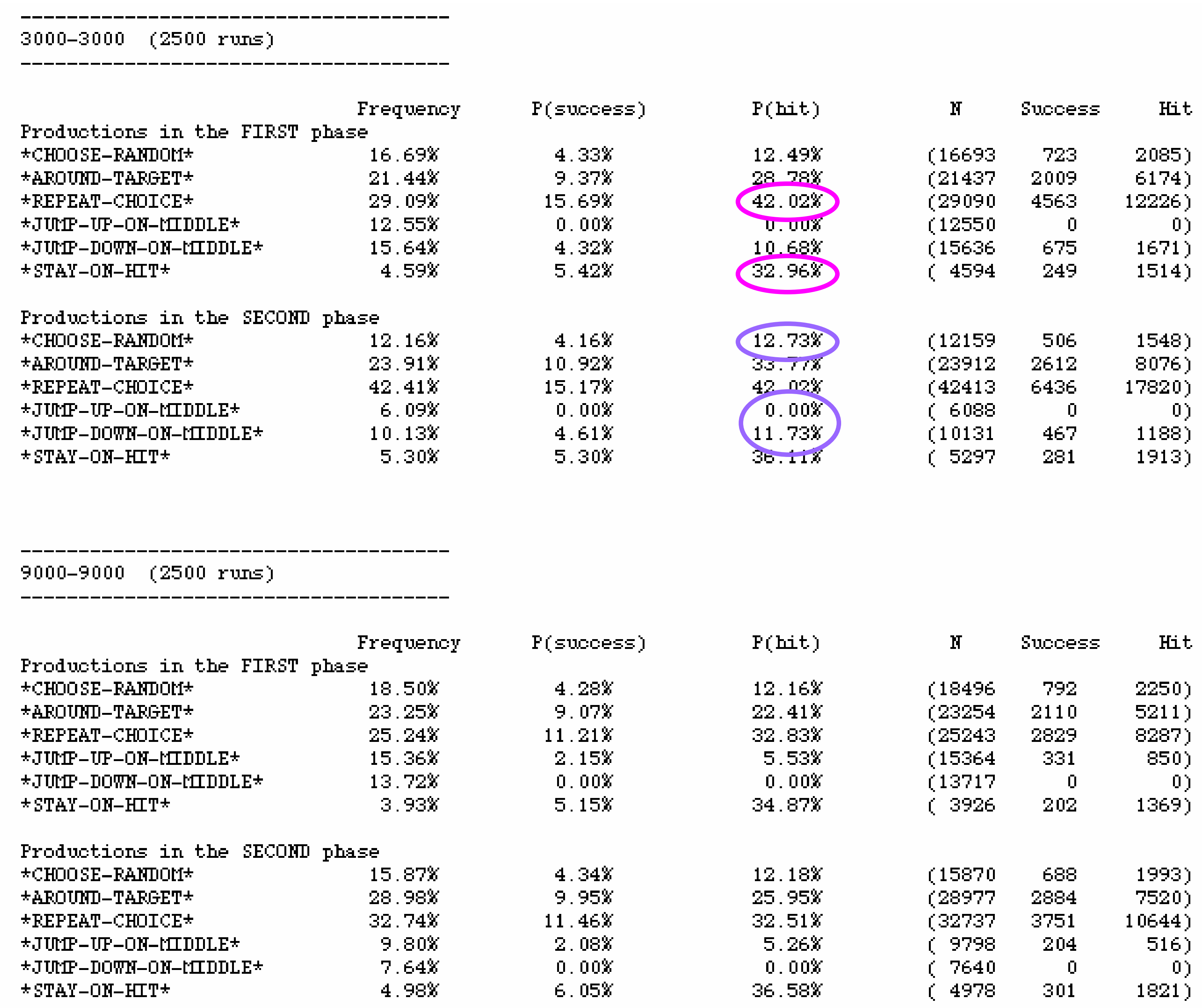
Model **Data**



Why is it so?

Two key concepts:

- good (i.e., repeat-choice, stay-on-hit) vs. bad (i.e., jump, choose-random) productions

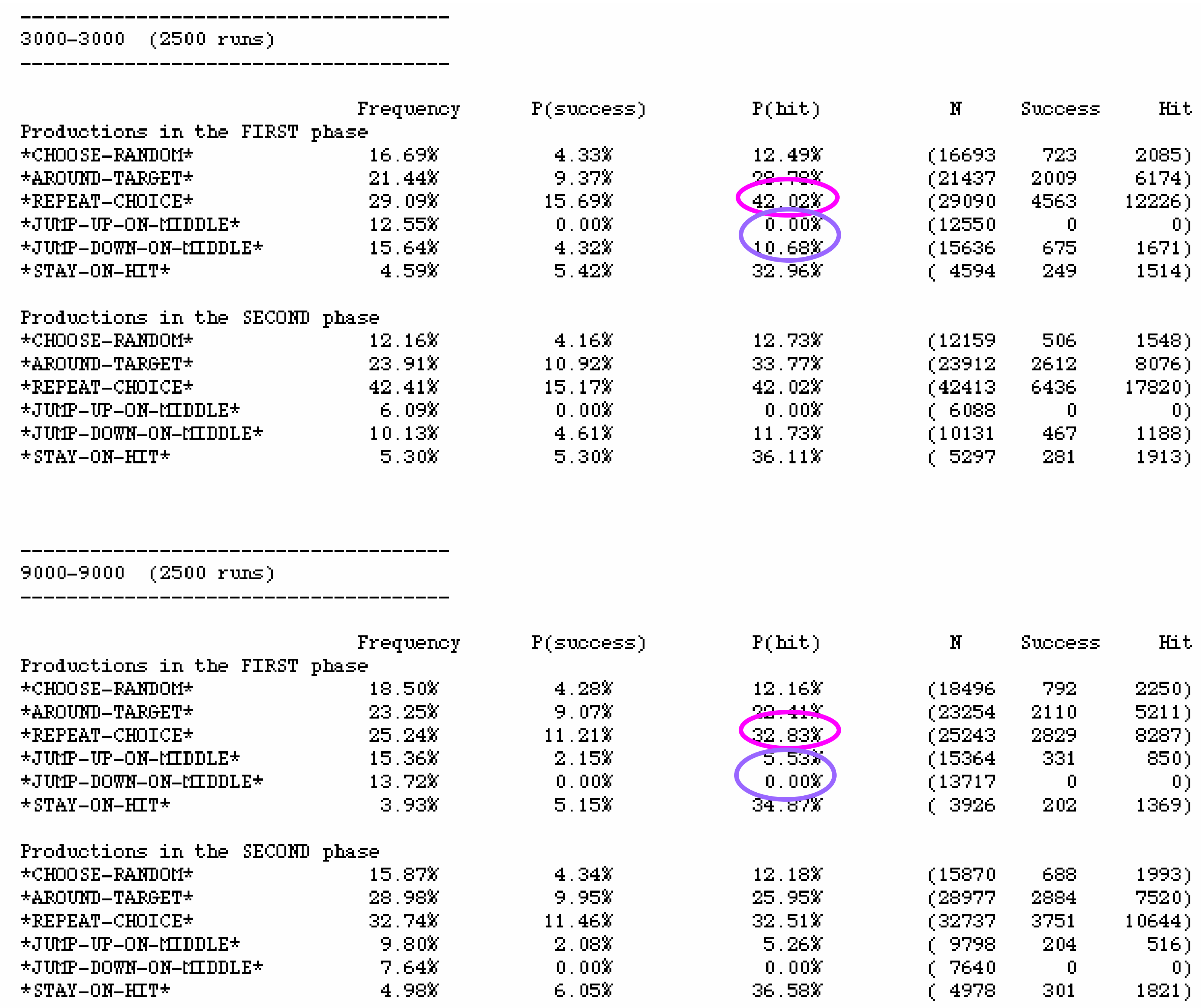


```
-------------------------------------
3000-3000  (2500 runs)
-------------------------------------

                            Frequency      P(success)        P(hit)           N      Success      Hit
Productions in the FIRST phase
*CHOOSE-RANDOM*               16.69%          4.33%          12.49%       (16693      723       2085)
*AROUND-TARGET*               21.44%          9.37%          28.78%       (21437     2009       6174)
*REPEAT-CHOICE*               29.09%         15.69%          42.02%       (29090     4563      12226)
*JUMP-UP-ON-MIDDLE*           12.55%          0.00%           0.00%       (12550        0          0)
*JUMP-DOWN-ON-MIDDLE*         15.64%          4.32%          10.68%       (15636      675       1671)
*STAY-ON-HIT*                  4.59%          5.42%          32.96%       ( 4594      249       1514)

Productions in the SECOND phase
*CHOOSE-RANDOM*               12.16%          4.16%          12.73%       (12159      506       1548)
*AROUND-TARGET*               23.91%         10.92%          33.77%       (23912     2612       8076)
*REPEAT-CHOICE*               42.41%         15.17%          42.02%       (42413     6436      17820)
*JUMP-UP-ON-MIDDLE*            6.09%          0.00%           0.00%       ( 6088        0          0)
*JUMP-DOWN-ON-MIDDLE*         10.13%          4.61%          11.73%       (10131      467       1188)
*STAY-ON-HIT*                  5.30%          5.30%          36.11%       ( 5297      281       1913)



-------------------------------------
9000-9000  (2500 runs)
-------------------------------------

                            Frequency      P(success)        P(hit)           N      Success      Hit
Productions in the FIRST phase
*CHOOSE-RANDOM*               18.50%          4.28%          12.16%       (18496      792       2250)
*AROUND-TARGET*               23.25%          9.07%          22.41%       (23254     2110       5211)
*REPEAT-CHOICE*               25.24%         11.21%          32.83%       (25243     2829       8287)
*JUMP-UP-ON-MIDDLE*           15.36%          2.15%           5.53%       (15364      331        850)
*JUMP-DOWN-ON-MIDDLE*         13.72%          0.00%           0.00%       (13717        0          0)
*STAY-ON-HIT*                  3.93%          5.15%          34.87%       ( 3926      202       1369)

Productions in the SECOND phase
*CHOOSE-RANDOM*               15.87%          4.34%          12.18%       (15870      688       1993)
*AROUND-TARGET*               28.98%          9.95%          25.95%       (28977     2884       7520)
*REPEAT-CHOICE*               32.74%         11.46%          32.51%       (32737     3751      10644)
*JUMP-UP-ON-MIDDLE*            9.80%          2.08%           5.26%       ( 9798      204        516)
*JUMP-DOWN-ON-MIDDLE*          7.64%          0.00%           0.00%       ( 7640        0          0)
*STAY-ON-HIT*                  4.98%          6.05%          36.58%       ( 4978      301       1821)
```

Why is it so?

Two key concepts:

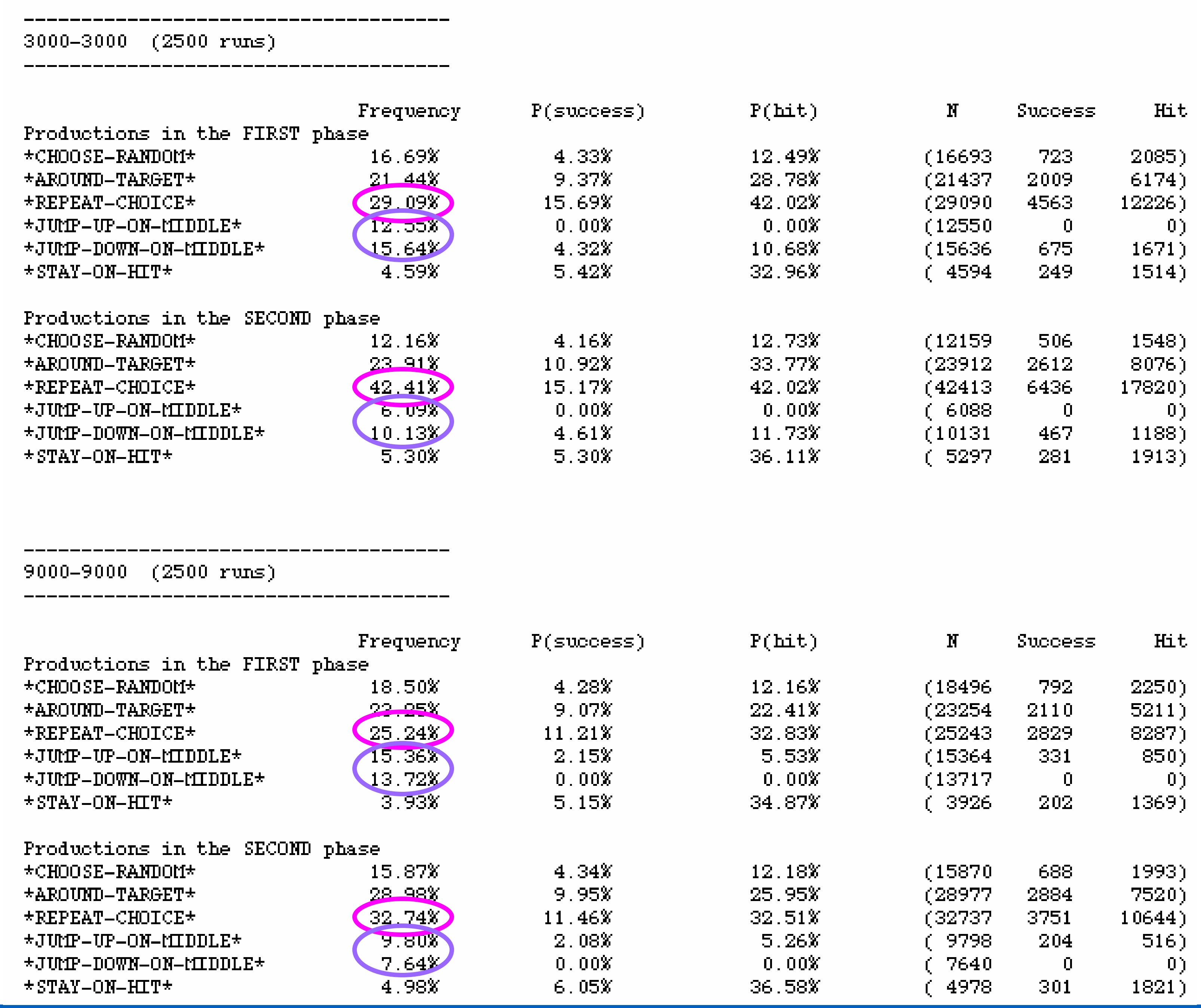
- good (i.e., repeat-choice, stay-on-hit) vs. bad (i.e., jump, choose-random) productions
- different hit probability for each production in the separate target conditions.



```
----------------------------------------
3000-3000  (2500 runs)
----------------------------------------

                              Frequency      P(success)       P(hit)          N     Success      Hit
Productions in the FIRST phase
*CHOOSE-RANDOM*                 16.69%          4.33%         12.49%      (16693      723       2085)
*AROUND-TARGET*                 21.44%          9.37%         28.78%      (21437     2009       6174)
*REPEAT-CHOICE*                 29.09%         15.69%         42.02%      (29090     4563      12226)
*JUMP-UP-ON-MIDDLE*             12.55%          0.00%          0.00%      (12550        0          0)
*JUMP-DOWN-ON-MIDDLE*           15.64%          4.32%         10.68%      (15636      675       1671)
*STAY-ON-HIT*                    4.59%          5.42%         32.96%      ( 4594      249       1514)

Productions in the SECOND phase
*CHOOSE-RANDOM*                 12.16%          4.16%         12.73%      (12159      506       1548)
*AROUND-TARGET*                 23.91%         10.92%         33.77%      (23912     2612       8076)
*REPEAT-CHOICE*                 42.41%         15.17%         42.02%      (42413     6436      17820)
*JUMP-UP-ON-MIDDLE*              6.09%          0.00%          0.00%      ( 6088        0          0)
*JUMP-DOWN-ON-MIDDLE*           10.13%          4.61%         11.73%      (10131      467       1188)
*STAY-ON-HIT*                    5.30%          5.30%         36.11%      ( 5297      281       1913)


----------------------------------------
9000-9000  (2500 runs)
----------------------------------------

                              Frequency      P(success)       P(hit)          N     Success      Hit
Productions in the FIRST phase
*CHOOSE-RANDOM*                 18.50%          4.28%         12.16%      (18496      792       2250)
*AROUND-TARGET*                 23.25%          9.07%         22.41%      (23254     2110       5211)
*REPEAT-CHOICE*                 25.24%         11.21%         32.83%      (25243     2829       8287)
*JUMP-UP-ON-MIDDLE*             15.36%          2.15%          5.53%      (15364      331        850)
*JUMP-DOWN-ON-MIDDLE*           13.72%          0.00%          0.00%      (13717        0          0)
*STAY-ON-HIT*                    3.93%          5.15%         34.87%      ( 3926      202       1369)

Productions in the SECOND phase
*CHOOSE-RANDOM*                 15.87%          4.34%         12.18%      (15870      688       1993)
*AROUND-TARGET*                 28.98%          9.95%         25.95%      (28977     2884       7520)
*REPEAT-CHOICE*                 32.74%         11.46%         32.51%      (32737     3751      10644)
*JUMP-UP-ON-MIDDLE*              9.80%          2.08%          5.26%      ( 9798      204        516)
*JUMP-DOWN-ON-MIDDLE*            7.64%          0.00%          0.00%      ( 7640        0          0)
*STAY-ON-HIT*                    4.98%          6.05%         36.58%      ( 4978      301       1821)
```

Why is it so?

The overall learning effect is explicated by the fact that the ACT-R learning mechanism endorses and glorifies good productions.



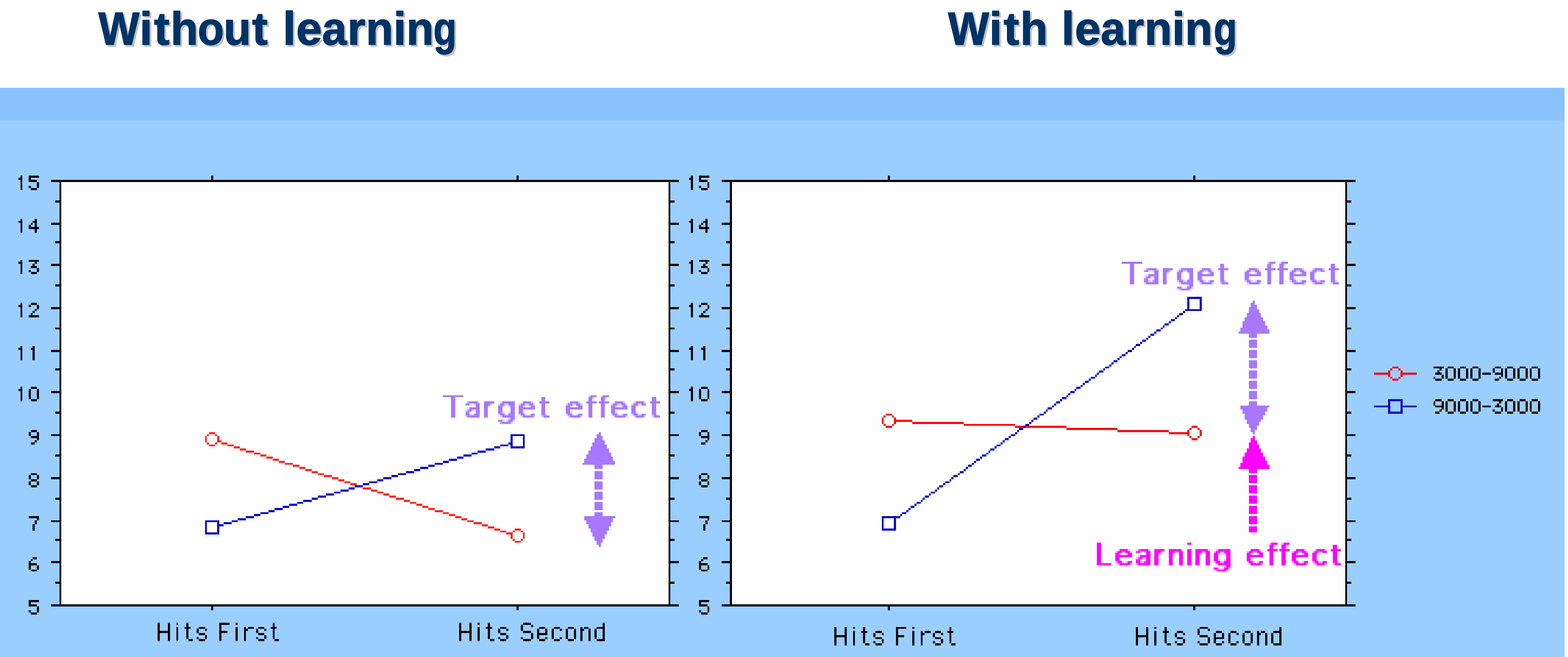
```
-------------------------------------
3000-3000  (2500 runs)
-------------------------------------

                             Frequency     P(success)         P(hit)           N     Success      Hit
Productions in the FIRST phase
*CHOOSE-RANDOM*                16.69%         4.33%          12.49%       (16693      723       2085)
*AROUND-TARGET*                21.44%         9.37%          28.78%       (21437     2009       6174)
*REPEAT-CHOICE*                29.09%        15.69%          42.02%       (29090     4563      12226)
*JUMP-UP-ON-MIDDLE*            12.55%         0.00%           0.00%       (12550        0          0)
*JUMP-DOWN-ON-MIDDLE*          15.64%         4.32%          10.68%       (15636      675       1671)
*STAY-ON-HIT*                   4.59%         5.42%          32.96%       ( 4594      249       1514)

Productions in the SECOND phase
*CHOOSE-RANDOM*                12.16%         4.16%          12.73%       (12159      506       1548)
*AROUND-TARGET*                23.91%        10.92%          33.77%       (23912     2612       8076)
*REPEAT-CHOICE*                42.41%        15.17%          42.02%       (42413     6436      17820)
*JUMP-UP-ON-MIDDLE*             6.09%         0.00%           0.00%       ( 6088        0          0)
*JUMP-DOWN-ON-MIDDLE*          10.13%         4.61%          11.73%       (10131      467       1188)
*STAY-ON-HIT*                   5.30%         5.30%          36.11%       ( 5297      281       1913)


-------------------------------------
9000-9000  (2500 runs)
-------------------------------------

                             Frequency     P(success)         P(hit)           N     Success      Hit
Productions in the FIRST phase
*CHOOSE-RANDOM*                18.50%         4.28%          12.16%       (18496      792       2250)
*AROUND-TARGET*                23.25%         9.07%          22.41%       (23254     2110       5211)
*REPEAT-CHOICE*                25.24%        11.21%          32.83%       (25243     2829       8287)
*JUMP-UP-ON-MIDDLE*            15.36%         2.15%           5.53%       (15364      331        850)
*JUMP-DOWN-ON-MIDDLE*          13.72%         0.00%           0.00%       (13717        0          0)
*STAY-ON-HIT*                   3.93%         5.15%          34.87%       ( 3926      202       1369)

Productions in the SECOND phase
*CHOOSE-RANDOM*                15.87%         4.34%          12.18%       (15870      688       1993)
*AROUND-TARGET*                28.98%         9.95%          25.95%       (28977     2884       7520)
*REPEAT-CHOICE*                32.74%        11.46%          32.51%       (32737     3751      10644)
*JUMP-UP-ON-MIDDLE*             9.80%         2.08%           5.26%       ( 9798      204        516)
*JUMP-DOWN-ON-MIDDLE*           7.64%         0.00%           0.00%       ( 7640        0          0)
*STAY-ON-HIT*                   4.98%         6.05%          36.58%       ( 4978      301       1821)
```

Why is it so?

The overall learning effect is explicated by the fact that the ACT-R learning mechanism endorses and glorifies good productions.

The target effect is explicated by the different hit probabilities of productions in different conditions.

Why is it so?



Without learning

With learning

Conclusions: What you can buy

- Two new phenomena in the SF domain
- A new model:
  - no memory
  - pure procedural parameter learning
- The model seems to do a pretty good job (BTW: it explains the results of the pilots, too)
- but ...

Caveat emptor!

- In the very long run (600 trials):
  - people are able to completely control the system
  - people are able to verbalize their knowledge
- The model predicts that by broadening the target (thus making the scoring criterion explicit) the performance should improve.