COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 36 (2012) 333–358

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/j.1551-6709.2011.01222.x

When, What, and How Much to Reward in Reinforcement Learning-Based Models of Cognition

Christian P. Janssen,[a,b,c] Wayne D. Gray[b]

[a]*UCL Interaction Centre, University College London*
[b]*Cognitive Science Department, Rensselaer Polytechnic Institute*
[c]*Department of Artificial Intelligence, University of Groningen*

Received 5 August 2009; received in revised form 30 May 2011; accepted 19 June 2011

Abstract

Reinforcement learning approaches to cognitive modeling represent task acquisition as learning to choose the sequence of steps that accomplishes the task while maximizing a reward. However, an apparently unrecognized problem for modelers is choosing when, what, and how much to reward; that is, when (the moment: end of trial, subtask, or some other interval of task performance), what (the objective function: e.g., performance time or performance accuracy), and how much (the magnitude: with binary, categorical, or continuous values). In this article, we explore the problem space of these three parameters in the context of a task whose completion entails some combination of 36 state–action pairs, where all intermediate states (i.e., after the initial state and prior to the end state) represent progressive but partial completion of the task. Different choices produce profoundly different learning paths and outcomes, with the strongest effect for moment. Unfortunately, there is little discussion in the literature of the effect of such choices. This absence is disappointing, as the choice of *when, what, and how much* needs to be made by a modeler for every learning model.

Keywords: Reinforcement learning; Choice; Strategy selection; Adaptive behavior; Expected utility; Expected value; Cognitive architecture; Skill acquisition and learning

1. Introduction

1.1. Four parameter sets for learning models

When agents maneuver in a novel environment, they can learn an appropriate plan for acting in that environment based on their experience with the environment and the rewards

Correspondence should be send to Christian P. Janssen, UCL Interaction Centre, University College London, Gower Street, London, WC1E 6BT, UK. E-mail: c.janssen@ucl.ac.uk

that they achieve after executing actions. This is the typical approach adopted in reinforcement learning models of cognition (Sutton & Barto, 1998), which are being applied increasingly in cognitive science research (e.g., Daw & Frank, 2009). Developing a reinforcement learning agent requires a modeler to use four sets of parameters related to (a) the general cognitive architecture, (b) the environment, (c) the actor, and (d) the critic. First, the general cognitive architecture defines the characteristics of the agent (e.g., its perception, action, and memory components). Second, the task being performed and task environment within which performance takes place imposes a hard constraint (Gray & Boehm-Davis, 2000) on the possible behaviors and strategies that are available to an agent (cf. Howes, Lewis, & Vera, 2009). Third, if given these constraints there are multiple ways in which a goal can be achieved, then an actor needs to choose which action to perform (in our model ''action to perform'' will be interpreted as ''strategy to be executed''). Fourth, a critic component assesses the contribution of the action in achieving the goal so as to inform future choices (Barto, Sutton, & Anderson, 1983).

Our focus in this article is on characteristics and parameters of the fourth set, the critic. Utility of actions can be calculated by relating actions to the rewards that are experienced and internalized after the execution of an action (e.g., Cohen, 2008; Davis, Staddon, Machado, & Palmer, 1993; Fu & Anderson, 2004, 2006; Gray, Schoelles, & Sims, 2005; Lovett, 1998; Montague, Dayan, & Sejnowski, 1996; Nason & Laird, 2005; Rieskamp & Otto, 2006; Schultz, Dayan, & Montague, 1997; Sun, Merrill, & Peterson, 2001; Sutton & Barto, 1998). We will investigate three parameters that influence the calculation of internal utility. *Moment* determines when the critic is applied. *Objective function* determines the type of reward (''what''). *Magnitude* determines how much reward is given. Table 1 gives a sample of different combinations of parameter settings that have been used in the literature.

Although other parameters of reinforcement learning have been investigated systematically in previous research (e.g., Ahn, Busemeyer, Wagenmakers, & Stout, 2008; Sutton & Barto, 1998; Yechiam & Busemeyer, 2005), relatively less attention has been given to the reward parameters of the critic (though some exploration is reported in Singh, Lewis, & Barto, 2009). This is surprising as the rationale behind ''when, what, and how much'' to reward is as important as the rationale behind other parameters. Rewards reflect the success

Table 1
Sample of studies that applied different moments, objective functions and magnitudes of reward

Moment	Objective Function	Magnitude	Example Studies
End of trial	Monetary gains	Continuous	Ahn et al. (2008), Erev & Barron (2005), Rieskamp and Otto (2006), Yechiam and Busemeyer (2005)
End of trial	Accuracy	Binary	Gray et al. (2005)
End of subtask	Distance traveled toward a target	Continuous	Sun et al. (2001)
End of subtask	Task completion time	Continuous	Gray et al. (2006)
End of subtask	Resource conservation and consumption	Categorical	Nason and Laird (2005)

of a model in achieving its goals. By changing when, what, and how much is rewarded, essentially the model's reflection on its performance is affected. This article is the start of a more principled investigation of the effects of different settings for these parameters for human cognition. As a case study, we will use an ACT-R model of the Blocks World task (Ballard, Hayhoe, & Pelz, 1995; Ballard, Hayhoe, Pook, & Rao, 1997; Gray et al., 2005; Gray, Sims, Fu, & Schoelles, 2006). We will provide more information on this task in Section 1.3.

1.2. Understanding and constraining the choices for reward parameters

Out of all the possible settings for moment, objective function, and magnitude of reward, architectural-based approaches tend to constrain the modeler's choices to those that are derived from the principles of the architecture. In versions 1 through 5 of ACT-R (Anderson, 1993; Anderson & Lebiere, 1998; Anderson et al., 2004), for example, the moment was after the completion of a goal on the goal stack, the objective function was task success or failure weighed by completion time, and the magnitude was binary (Lovett, 1998).

However, further experience with ACT-R's utility mechanism revealed serious limitations (Fu & Anderson, 2004; Gray et al., 2005). For example, the magnitude of the rewards was binary, whereas it is clear that in the real-world rewards can have different magnitudes to which people are sensitive. In response to these issues, ACT-R adopted a reinforcement learning-based utility learning mechanism, in which magnitudes are not limited to binary values (Anderson, 2007).

Still, for this framework and other reinforcement learning-based approaches, questions of cognitive fidelity persist. Modelers have an infinite set of reward magnitudes at their disposal, an unconstrained set of possible objective functions, and freedom to place rewards after any one of many different (sub)steps during task performance. Although it may be the case that eventually cognitive neuroscience will provide constraints for these factors, at least for models of human behavior (e.g., Cohen, 2008; Holroyd & Coles, 2002; Schultz, 2006; Schultz et al., 1997), at present such data are not available for real-world tasks. As this parameter set is currently unconstrained yet of importance to all reinforcement learning approaches of cognition, this article provides a case study of why these settings matter, and how they lead to different model behavior.

The current model interacts with a simple task environment to perform a simple interactive task. Between conditions the time costs of interacting with the task environment varies. In the original research (Gray et al., 2006), the focus was on the adaptation of the selected strategies to the time costs of interactive behavior. That model, and the one used here, required general assumptions about the time cost of cognitive, perceptual, and motor resources. It did not require detailed assumptions about the inner working of such processes. For example, the most widely used parameter was based on Fitts' Law for the movement time of a mouse (Fitts, 1954; see also Soukoreff & MacKenzie, 2004). The most important parameter was based on Anderson's rational analysis of memory (Anderson & Schooler, 1991) to predict retrieval time and forgetting. Although more detailed models exist for

motor movement and the inner workings of memory, to our knowledge, estimates of the parameters of interest here (movement time, retrieval time, and forgetting) do not vary greatly from those we use here (see Sims & Gray, 2004 for a comparison of memory models). Hence, although we use the ACT-R software for our parameter estimates, the assumptions we derive from that architecture are general assumptions and are not the subject of current research controversies. The specific parameter settings of our model will be discussed in more detail in Section 4. For a more general discussion of these parameters, we refer to work by Anderson (2007) and Anderson et al. (2004).

1.3. Case study: Blocks World

As a case study, we use the Blocks World task. This task has been investigated substantially in previous research (e.g., Ballard et al., 1995, 1997; Gray et al., 2005, 2006; Morgan, Patrick, Waldron, King, & Patrick, 2009; Waldron, Patrick, Morgan, & King, 2007). The consistent finding across all Blocks World studies is that performance adapts to (i.e., is *learned* as a function of) changes in the interface that manipulate the ease with which information can be gathered. That learning depends on characteristics of the environment is important, as it shows that in this context all parameters of reinforcement learning are involved (namely, those for cognitive architecture, environment, actor, and critic).

Another reason for choosing this task is that our work can build on two previous modeling efforts that investigated the role of learning in this task (Gray et al., 2005, 2006). One effort used the reinforcement learning technique of Q-learning (Sutton & Barto, 1998; Watkins & Dayan, 1992). Although successful in predicting optimal performance, this approach (Gray et al., 2006) required 100,000 training trials and, as such, provided an ideal performer model of this task, not a cognitively plausible one. In another effort a more cognitively plausible model was developed using the ACT-R cognitive architecture (Gray et al., 2005). Interestingly, this model could not capture the trend of the data without the incorporation of scalar rewards. At the time, this required modification of the architecture. However, in the version of ACT-R that we use here (Anderson, 2007), reinforcement learning and the use of scalar rewards are the default options.

1.4. The goals of this article and limitations of the approach

Our core model combines and extends the models that were used in the two previous modeling efforts, and that contain all necessary parameters of a reinforcement learning model. It has one set of parameters for cognitive functions, one set of strategies that is appropriate given the task interface, and one type of reinforcement learning (i.e., one actor and critic mechanism). Within the fourth component, the critic, we explore the space defined by alternative moments, objective functions, and magnitudes of rewards. A self-imposed constraint is that all models are exposed to the same number of trials as human participants (in contrast to Gray et al., 2006). Although not the main focus of this article, we will also briefly compare model results to human performance.

2. Calculating utility

2.1. The parameter space of reinforcement learning models

In contrast to our effort, prior work investigated settings for each of the four parameter sets of reinforcement learning models (general cognitive architecture, environment, actor, and critic). Different cognitive architectures, and architectural assumptions, have been used across studies. For example, there have been architectural-based approaches (e.g., Napoli & Fum, 2010; Nason & Laird, 2005) and non-architectural approaches, of which several examples are given by Sutton and Barto (1998).

Similarly, reinforcement learning techniques have been applied in different task environments. These range from gambling tasks such as the Iowa gambling task (e.g., Yechiam & Busemeyer, 2005), to interactive tasks (e.g., Gray et al., 2006), to tasks in which (virtual) agents have to maneuver in an environment (e.g., Ballard & Sprague, 2007; Singh et al., 2009; see also several examples in Sutton & Barto, 1998).

Different settings have also been proposed for the actor and critic components of reinforcement learning models. Yechiam and Busemeyer (2005) and Ahn et al. (2008) investigated how different choices for the actor and critic component influence performance in the context of the Iowa Gambling task. The success of the different models in predicting human behavior depended on the modeler's objective: Some models are better at predicting short-term performance, while others are better at predicting long-term performance.

The critic components that Yechiam and Busemeyer (2005) and Ahn et al. (2008) investigated differ from the parameters that are the focus of this article. The aforementioned studies investigated the *algorithms* by which utility is calculated. An overview of the options for these algorithms is given in Sutton and Barto (1998). To mention a few, a modeler needs to decide which actions get their utility value updated (e.g., all actions preceding a reward, or only the last n actions), how strongly a reward impacts utility value (e.g., is the influence of a reward on an action's utility value linearly or exponentially ''discounted'' with the number of time steps between an action and a reward), and how long the agent learns (e.g., during its entire life span, or only during a designated learning phase). We will discuss our settings in Section 2.2.

Modeling work on the effect of different settings for rewards is scattered across the literature. Different models have used different settings for moment, objective function, and magnitude of reward (see Table 1). However, systematic investigations of the different combinations for these parameters are scarce. Indeed, the only extensive study of the effect of different reward settings which we found is reported by Singh et al. (2009). In two simple simulated agent environments, they explored model behavior for, respectively, 3,240 and 54,000 alternative reward values. These different settings were achieved by systematically varying the magnitude of rewards from a distribution of continuous values within the range [−1.0, 1.0]. The value of the rewards was not motivated by a rationale for a specific objective function or magnitude. Rather, the researchers investigated whether the rewards that provided the agent with the best performance score also matched with an understanding of optimizing a specific objective function.

Similar to Singh and colleagues, we also systematically explore the effect of alternative reward types on performance of a reinforcement learning model. However, in our work the alternative reward types will be motivated by a rationale for objective function and magnitude. In addition, we will also explore the effect of the moment at which the reward is given. In the work by Singh and colleagues this parameter is not varied; rewards are given after every time step (Singh et al., 2009). We will also investigate model performance over a shorter time frame (48 trials), similar to the duration of the human experiment. This contrasts with the focus on evolutionary aspects of rewards by Singh and colleagues (where models had at least 20,000 reward updates), which was without a comparison to human performance.

2.2. Actor and critic in ACT-R

Our choice for using the ACT-R cognitive architecture (Anderson, 2007) as a framework provides us with parameter settings for the general cognitive architecture, the actor, and the critic. As the model interacts with a task interface, this combination also provides parameter settings for the task environment. The settings for the general cognitive architecture and environment will be discussed when we flesh out the model in Section 4. In the current section, we will outline the parameter settings for the actor and critic. We will also show how the parameter settings for moment, objective function, and magnitude of reward influence the performance of the critic.

ACT-R is a production rule system, which means that behavior is modulated by a sequence of production rules (condition–action pairs). Each production rule has a utility value associated with it. Higher utility values reflect successes in the past and predict successes in the future.

The functionality of a critic is achieved by updating the utility value of production rules that are associated with an experienced reward, using a utility function. ACT-R's utility function is a special case (Anderson, 2007; Fu & Anderson, 2004, 2006) of the temporal difference learning algorithm from reinforcement learning (Sutton & Barto, 1998). At the moment when a reward is given, all production rules i that preceded the reward (and postceded the previous reward) get their utility value U updated as follows:

$$U_i(n) \leftarrow U_i(n-1) + \alpha(R_i - U_i(n-1)) \tag{1}$$

In this equation, $U_i(n)$ is the estimated utility of production rule i after its nth usage, R_i is the estimated reward and α is the learning rate. The estimated utility at the current time (nth usage) is based on the previous estimate of the utility ($U_i(n-1)$) plus an *error term* ($R_i - U_i(n-1)$) that reflects the difference between the estimated reward and the previous estimated utility. By scaling this error term with the learning rate α (ranging between zero and one), the impact of recent experience on the estimated utility is limited, and learning is gradual.

The interesting parameter in this equation for our study is the estimated reward, R_i, which is based on a *behavioral reward* and a *temporal difference value*. The *behavioral reward*, r_j in Eq. 2, represents the reward that is experienced in the environment. Its value is determined by the objective function and its magnitude.

The second component of R_i is the *temporal difference value*. For each production rule, this value estimates how much a specific production rule i contributed to the magnitude of the eventual behavioral reward. In ACT-R, the temporal difference value is calculated as a linear difference between the time at which the production rule fired and the time at which the behavioral reward was experienced. In the end, if production rule i was used at time t_i, and a behavioral reward r_j is given at time t_j, then the estimated reward of production rule i is (Anderson, 2007):[1]

$$R_i = r_j - (t_j - t_i) \tag{2}$$

The magnitude of R_i decreases linearly with a delay of the moment at which the reward is given (i.e., with an increase in the difference between t_j and t_i). This captures the intuition that actions (or production rules) that are used closer to a behavioral reward contribute more to the magnitude of that reward than actions from the more distant past. In this sense, the *moment* when the reward is given influences the estimated reward and the estimated utility of production rules.[2]

Utility values can be used for action selection in choice situations (i.e., for the actor component in an actor–critic system). As common in production rule systems, ACT-R models initially select a subset of available production rules that can be executed given the current state of the world and the model (i.e., the contents of the diverse buffers in ACT-R). If multiple alternatives are available, the model selects the production rule with the highest utility value using the soft-max action selection rule, which is widely applied in reinforcement learning models (e.g., Anderson, 2007; Sutton & Barto, 1998).

To moderate the strict reliance on exact utility value, and to reflect uncertainty in action selection, the soft-max selection rule has a built-in temperature component that applies some noise to each utility value during an action selection round. This is useful in situations where two actions are so close in utility value as to be practically, though not statistically, indistinguishable. The temperature component then insures that the less ranked action is occasionally chosen. Similarly, this occasional selection of less-high ranked actions can help in exploring the (perhaps changed) value of alternative actions.

Importantly, modelers are free to choose the settings of the moment, objective function, and magnitude of reward. In the next section we will introduce the Blocks World task, which will be our test-bed for investigating these parameters. This will be followed by a description of the model and its parameter settings.

3. Blocks World

3.1. Task

The Blocks World task requires participants to replicate a pattern of eight-colored items[3] as shown in a target window, by dragging items from a resource window to the correct position in a workspace window. The layout of these three windows is illustrated in Fig. 1.

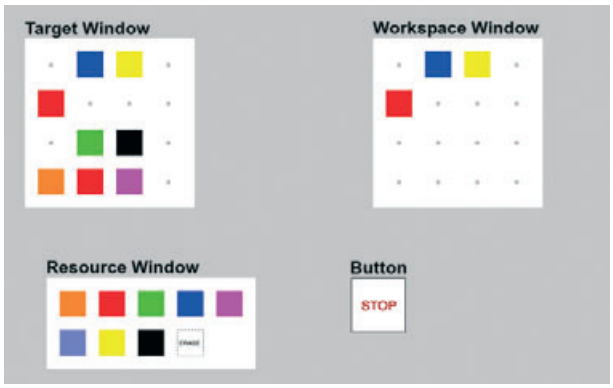


Fig. 1. Layout of the three windows in the Blocks World task, as used by the model.

As shown in Fig. 1, the target window contains a four by four grid in which eight items are placed randomly for each trial. Each item has one out of eight possible colors, and each color occurs at most twice in the pattern. In the experiment the three windows (target, workspace, and resource) are covered by gray rectangles. Each rectangle can be uncovered by moving the mouse into the corresponding window area. For the workspace and resource window the information is revealed immediately. For the target window, participants first have to wait for a lockout time (0, 400, or 3,200 ms depending on condition), before the information is revealed. At any one time, at most one window can be open and the window closes again as soon as the mouse cursor leaves the window area.

The task can be solved using different strategies (Gray et al., 2006). Strategies differ in the required amount of interaction with the environment (cf. Gray & Fu, 2004; Walsh & Anderson, 2009). Interaction intensive strategies rely on the task environment to frequently look up information on the color and position of blocks, which takes time (particularly in the high lockout condition). In contrast, memory intensive strategies reduce interaction with the environment (which can save time in the higher lockout conditions) by memorizing the position and color of multiple blocks during a single visit to the target window, at the risk of forgetting some information. To solve the task, one needs to learn the right strategy, or policy. This involves carefully trading-off the speed with which the task is performed versus the accuracy with which it is performed.

3.2. Human data

In the study modeled here (Gray et al., 2005), participants were randomly assigned to one of three lockout conditions (0, 400, or 3,200 ms lockout). Each participant performed 48 trials. Our main analysis will concentrate on events surrounding the first visit to the target window. At the start of this visit, participants have no information about the color or location of any item, whereas on subsequent visits participants may have partial information for some items. Hence, the number of items placed following the first visit is assumed to be the most sensitive measure of performance (cf. Gray et al., 2005).

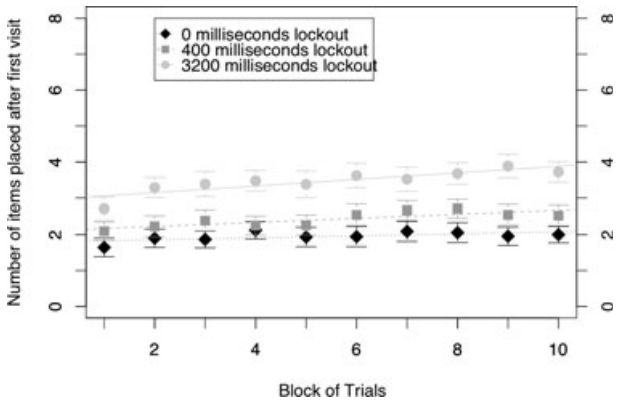


Fig. 2. Mean number of items placed after the first visit to the target window, averaged per block of five trials for the human participants. Error bars show standard error.

To highlight the learning process that participants go through, we averaged the data across blocks of five trials, for trials 1–48 (hence, the last block is an average over three trials). Fig. 2 shows the average number of items (with standard errors) that was placed after the first visit to the target window (and before a second visit) for each condition per block.[4] For each lockout condition we fitted a linear trend line to highlight whether the number of items increased over blocks of trials.

Across all three conditions, participants placed around two items during the first block of trials. With experience the number of items placed slightly increased, as did the difference between lockout conditions. In the 0 ms lockout condition, there was at most a slight increase in number of items placed across trials, whereas the 3,200 ms lockout condition had the strongest increase (i.e., the regression line had the steepest slope). During the last block, the mean number of items placed in the three lockout conditions (0, 400, and 3,200 ms) was, respectively, 2.1, 2.5, and 3.9 items (with standard errors of 0.22, 0.30, and 0.28 items).

An analysis of variance on the mean number of items placed in the three lockout conditions across the 10 blocks (i.e., estimating changes over time) confirmed these results. There was a main effect of lockout condition, $F(2, 51) = 39.84$, $p < .001$, a main effect of block, $F(1, 51) = 36.78$, $p < .001$, and a significant interaction between lockout condition and block, $F(2, 51) = 4.06$, $p = .02$. Follow-up tests on the effect of block for each lockout condition confirmed the results that are visible in the trend lines in Fig. 2. There was no effect for the 0 ms condition, $F(1, 17) = 2.07$, $p = .17$, but a significant effect for the 400 ms condition, $F(1, 18) = 14.22$, $p = .001$, and for the 3,200 ms condition, $F(1, 16) = 12.75$, $p < .001$.

4. Modeling the Blocks World task

We used ACT-R 6.0 (Anderson, 2007) to explore the use of temporal difference learning as a utility learning mechanism. Our model followed the same general structure as two

previous modeling efforts (a non-reinforcement learning-based approach within ACT-R in Gray et al., 2005; a pure reinforcement learning-based approach in Gray et al., 2006). This allowed us to focus our work on the exploration of the effects of the three parameters of interest on model behavior and to keep other parameters at default values. In separate sections we will outline the general model structure, credit assignment, and the parameter manipulations for each of the three parameters of interest (moment, objective function, and magnitude).

4.1. General model structure

The basic structure of the model is illustrated in Fig. 3. Gray rectangles indicate the crucial steps that the model goes through while executing the task, with the first step at the top of the figure, and subsequent steps beneath it. Note that in the model each of these steps

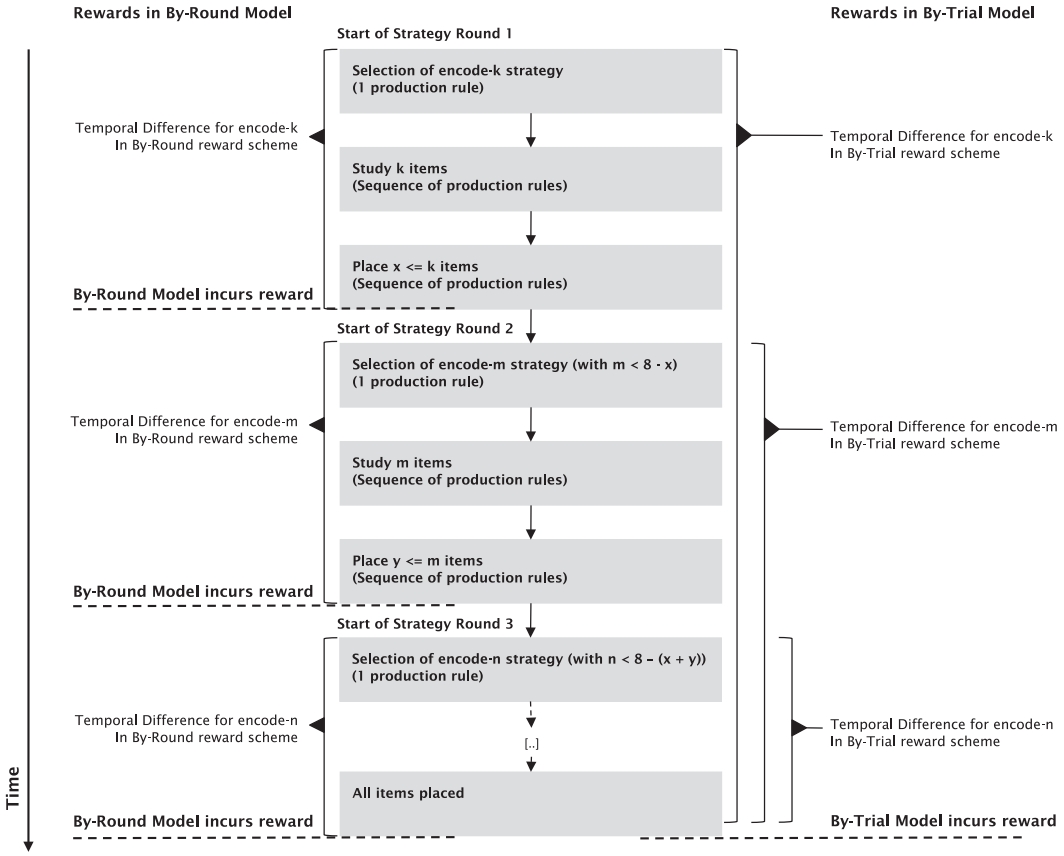


Fig. 3. Each trial consists of a series of rounds. This figure shows a three-round trial. Rectangles indicate (series of) production rules. The three dashed lines on the left show where rewards are incurred in by-round models and the one on the right shows the reward incurred in by-trial models. Braces highlight the temporal difference between rewards and strategy selections.

was represented by one or more production rules (as indicated in the squares), and that the figure is not to scale with the actual time it took the model to execute each step.

The model was built around a set of eight encode-strategies, named encode-1 through encode-8 that determined the number of items that the model encoded during a visit to the target window (cf. Gray et al., 2005, 2006). Each strategy was incorporated as a production rule, and learning consisted of acquiring utility values so that an optimal set of encode strategies was adopted for each of the three lockout conditions.

For each trial, the model went through a series of one or multiple strategy rounds. Fig. 3 shows a situation with three strategy rounds. Each strategy round began with the model choosing one encode strategy based on its utility value (captured by the ''selection of encode strategy'' rectangle in Fig. 3). The model then moved the mouse to the target window, waited for the gray box to vanish and the window to open, and encoded the number of items as specified by the strategy (these steps are captured by the ''study items'' rectangle in Fig. 3). The model then visited the resource window and made an attempt to retrieve information about an encoded but not yet placed item from memory. If the retrieval was successful, the model clicked on an item of the appropriate color and moved it to the correct position in the workspace window. The model continued to place items in this way until it had placed all the encoded items, or until it no longer successfully retrieved information about an item from memory (all these steps are captured by the ''place items'' rectangle in Fig. 3). This was where a strategy round ended. If the model had not yet placed all eight items in the workspace window, a new strategy round commenced.

Although the model had eight encode-strategies available, not all strategies could be applied on each strategy round due to environmental constraints. Take the example of a model that used an encode-6 strategy on the first strategy round, encoded six items, but only placed three. On the second strategy round only five items remained to be placed, and hence only the encode-1 through encode-5 strategies could be selected. Prior modeling work (Gray et al., 2005) also conformed to these environmental constraints (i.e., that the model could only select encode-strategies that encode no more than the total to-be-placed items); hence, the number of strategies a model had available decreased per strategy round as the model progressed in a trial.

To model the retrieval time and the probability of a retrieval success, we used the rational activation theory of memory (Anderson & Schooler, 1991). This theory of memory includes activation increases across multiple encodings or rehearsals of the same item, decay in activation after encoding, the notion of a retrieval threshold above which an item's activation suffices for it to be retrieved and below which it cannot be retrieved, and a probabilistic fluctuation in current activation such that two successive retrieval attempts of the same item might have different results (Anderson, 2007; Anderson et al., 2004). Built into the model was a constant number of encodings for each item studied on each round and a maximum of two retrieval attempts per item such that if the first failed to retrieve the item a second attempt would be made.

The characteristics of the rational activation theory of memory made it unlikely for our model to be able to place all eight items during one strategy round. Particularly, as within each strategy round items were encoded roughly at the same time (namely, during the visit

to the target window), the activation value of those items from then on only decayed. Items that were placed later in a sequence (e.g., the third item compared to the second item) had a relatively longer time interval between encoding and retrieval, and therefore a lower activation value. This decreased the probability of successful retrieval. Our model therefore had to learn the optimal number of items to encode by experience, to avoid forgetting.

With three exceptions, all ACT-R parameters were left at their default values (Bothell, 2008). As there is no established parameter for noise of the activation value of the chunks of declarative memory, we set this to a value representative of the range within which this parameter is typically set (with s set to 0.25). Second, Sims and Gray (2004) questioned the typical ACT-R practice of using a default option for computing the base-level activation of memory chunks that minimizes calculation time and argued for using the full equations so as to be truer to the rational activation theory of memory (Anderson & Schooler, 1991). We adopted their suggestion here (cf. Gray et al., 2005, 2006). Finally, as there is no established parameter for temperature in the utility learning mechanism, that was set to 1.5. In pilot runs of the model, this value turned out to offer a good balance between exploration and exploitation of the models' behavior.

4.2. Credit assignment

Using the model structure in Fig. 3, credit assignment can be illustrated. As was explained in Section 2.2, ACT-R models assign credit to every production rule that postceded the last reward and preceded a new reward. The crucial production rules in our model were the encode strategies that executed at the beginning of a strategy round. The strategy that was chosen here determined how many items were encoded and how many items could be maximally placed. So although other production rules also were assigned credit, these values did not matter for the strategy selection. This assumption was also made in previous modeling efforts (Gray et al., 2005, 2006).[5]

4.3. The moment of reward—when?

We used our basic model to manipulate the three parameters related to reward: moment, objective function, and magnitude. Conceptually, moment reflects the ''moment'' when feedback on performance is given or internally experienced. In many tasks one can distinguish between situations where feedback is given after the entire task is completed (at the end of a trial), and situations where feedback is given after part of the task has been completed. In the Blocks World task feedback was provided at the end of a trial, when participants clicked on the Stop button (which indicated trial completion). If all items were placed correctly, participants received feedback on correctness and trial time, and then continued to the next trial. If an error had been made participants had to identify and correct the error(s) first.

Feedback was also implicitly available during the trial, after subtask completion. After placing each item, participants received visual feedback on the assumed state of the task (i.e., assuming that the items were placed correctly). They also received feedback at the end of

each *strategy round* when their memory for items to place had been exhausted and they had to return to the target window to determine the color and position of the remaining items.

We therefore incorporated two conceptualizations of feedback moment in different model runs. In the *by-trial*-rewarded models, rewards were given at the end of a trial (these map to the ''once-rewarded models'' in Gray et al., 2005). In the *by-rounds*-rewarded models, rewards were given after each strategy round (mapping to ''each-rewarded'' models in Gray et al., 2005). Hence, by-trial-rewarded models rewarded performance for correctly placing all eight items, whereas by-round-rewarded models rewarded the completion of one strategy round regardless of whether the items were correctly placed.[6]

In Fig. 3 dashed lines show when By-Round (left side) and By-Trial models (right side) incurred a behavioral reward r_j. As explained in Section 2.2, the temporal distance between the time at which a behavioral reward r_j was incurred and the time at which a strategy was chosen influenced the experienced reward R_i, and the utility of the encode-*x* strategy. In Fig. 3, braces illustrate the temporal difference between the behavioral reward r_j and the moment the encode-*x* strategy production rule fired for our By-Round (left side) and By-Trial (right side) models. Note that both model versions only experienced the same temporal difference for the final encode strategy. In all other cases, the distances were always larger in the By-Trial rewarded models.

4.4. Objective function—what?

Our second parameter is the objective function, which captures what aspect of performance is rewarded. Two objective functions have been identified for the Blocks World task before, and were further investigated here. With the objective function of *accuracy* (Gray et al., 2005) the model is reinforced to optimize the accuracy of the number of items encoded and placed during each strategy round. This function gained points for the number of items encoded on each round but lost points if not all those encoded items were placed; that is, if some of them were forgotten (the exact algorithm was manipulated as part of the magnitude settings—see the next section). The objective function of *time* was reinforced to minimize the amount of time spent on a trial, or a round (Gray et al., 2006). Together, these two objective functions (accuracy and time) incorporated a very common aspect of psychological experiments and of everyday decision making: the trade-off between speed and accuracy of performance (Edwards, 1965).

4.5. Magnitude—how much?

The third parameter related to rewards is the magnitude of the reward, which is dependent on the objective function. Objective function determines what is rewarded, and magnitude determines how much is rewarded, allowing the model to distinguish between different levels of success and failure. In our models, we used a reward scheme (i.e., a function) to calculate the magnitude of each reward. For each of the two objective functions, we identified two reward schemes, which are summarized in Table 2.

Table 2
Value type and value range for the different reward magnitudes

Objective Function	Magnitude	Value Type	Value Range
Accuracy	Success rewarded	Categorical	[0, 8]
	Success–failure rewarded	Categorical	[−8, 8]
Time	Total time rewarded	Continuous	[−70, −9] (estimate)
	Lockout time rewarded	Continuous	[−25.6, 0]

In the two models with the objective function of accuracy, rewards had categorical values. The success-rewarded model took only successful actions into account, by calculating reward magnitude as the number of items that the model had placed correctly in the target window between two rewards. This reward could take discrete values between zero (no items placed) and eight (all items placed).

In the success–failure-rewarded model, both successes and failures were taken into account. Here, rewards took the magnitude of the number of items that had been placed correctly minus the number of items that were encoded but forgotten between two rewards. The upper bound of this reward was eight (all items encoded and placed), and the lower bound was minus eight (all items encoded, but none placed).

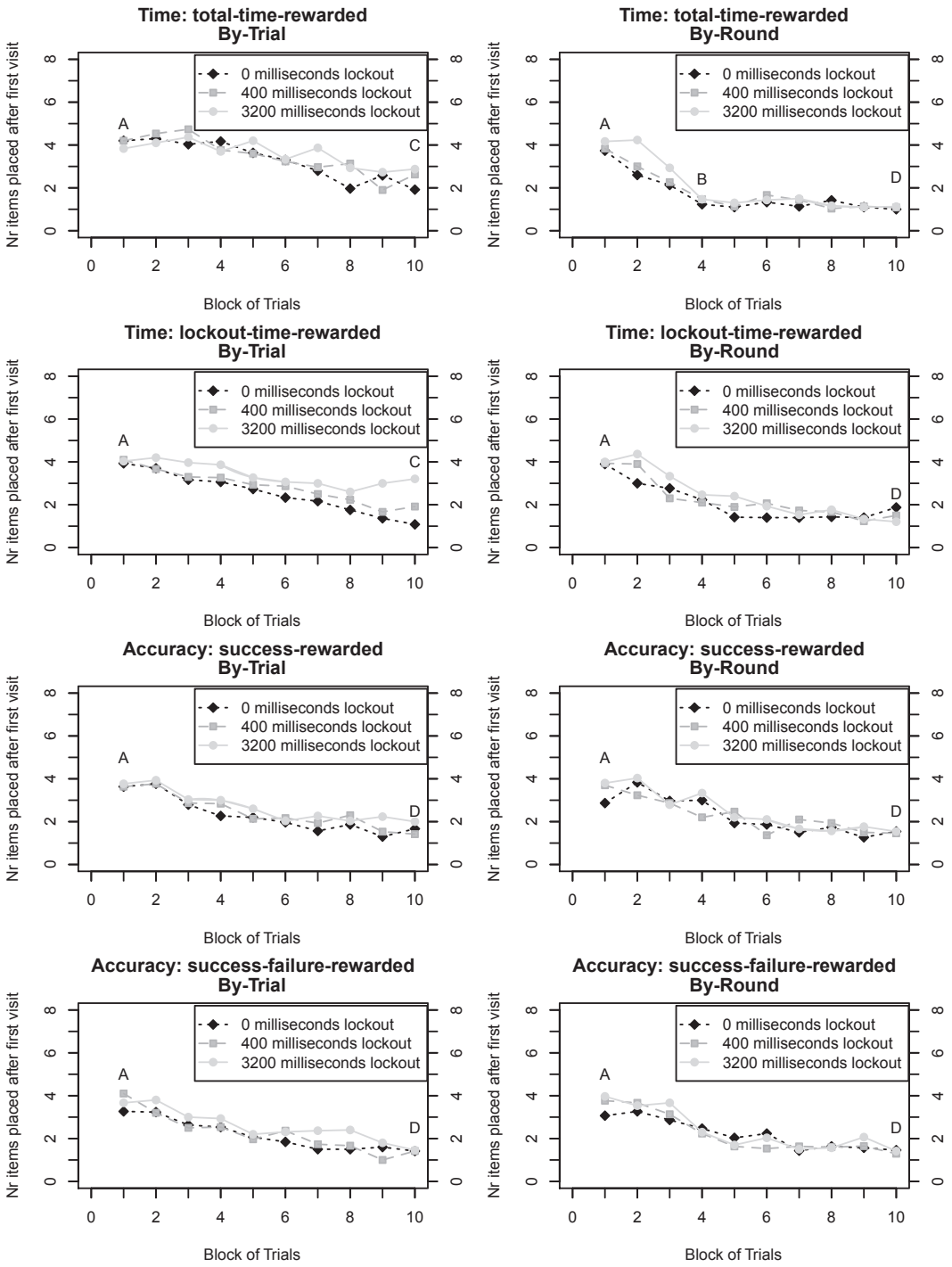
For the two models with the objective function of time, rewards had continuous values. In the total time-rewarded model, the reward magnitude was calculated as the total time lag between two rewards in seconds (with typical values ranging between 9 and 70).

In a lockout time-rewarded model the reward magnitude was calculated as the total amount of lockout time that was experienced between two rewards (ranging between 0, for no lockout, and 25.6, for eight lockouts of 3,200 ms). The rationale behind this reward was that if the lockout time of a condition increased, so did the amount of time spent on the task. To converge the amount of time spent toward a minimum, the rewards in both the total time-rewarded model and the lockout time-rewarded model were multiplied with minus one (i.e., they formed a penalty; see Table 2).

4.6. Summary of the different model manipulations

We have developed one model of the Blocks World task, with one set of parameters for cognitive functions, one set of strategies that is appropriate given the task interface, and one type of reinforcement learning (i.e., one actor). Within the fourth component, the critic, we

Fig. 4. Mean number of items placed after the first visit to the target window (and before the second visit) per block of trials (average over five trials per block). The different plots show performance of models with different moments of rewards (columns) and different objective functions and magnitudes (rows). Some special points in the graphs are highlighted with a letter. A: All models start by placing an average of four blocks per visit (due to averaging of all explored strategies). B: The total time by-round-rewarded model stabilizes the fastest on a strategy. C: The by-trial rewarded models with the objective function of time are the only models for which the strategies in the different lockout conditions diverge after learning, a qualitative pattern found in the human data. D: The other models than the one mentioned in C, mostly converge toward a single strategy for all lockout conditions, in contrast to the human data.


Time: total-time-rewarded
By-Trial
Time: total-time-rewarded
By-Round
Time: lockout-time-rewarded
By-Trial
Time: lockout-time-rewarded
By-Round
Accuracy: success-rewarded
By-Trial
Accuracy: success-rewarded
By-Round
Accuracy: success-failure-rewarded
By-Trial
Accuracy: success-failure-rewarded
By-Round
0 milliseconds lockout
400 milliseconds lockout
3200 milliseconds lockout
Nr items placed after first visit
Block of Trials
A
B
C
D

varied the moment at which the model was internally rewarded (with two levels), using two objective functions, each with two reward magnitudes. This resulted in eight different combinations of model types (i.e., all four combinations in Table 2 had a by-trial and a by-rounds version).

Each model type was run six times in three experimental conditions (lockouts of 0, 400, and 3,200 msec). Therefore, across the eight model types, and the three conditions, we ran the model 144 times ($6 \times 8 \times 3$). As the model interacted with the same interface as our human participants, each model run took approximately as long as running one participant. Hence, our rationale for running the model six times on each variation rather than more comes down to a simple judgment of what constitutes a reasonable test of the model variations and a reasonable expenditure of experimenter effort. In each model run, the model started learning as soon as the first trial started, and the model continued to update its utility values until the end of the experiment.

4.7. Results

We first describe how performance of the model changed as a result of the parameter settings for the rewards. Afterward these results are compared with human performance.

4.7.1. Different conceptions of rewards lead to different model performance

To highlight how performance of the model changed with different settings for the rewards, Fig. 4 shows eight plots for the different combinations of moment (columns), objective function, and magnitude of reward (rows). Each plot displays the number of items that the model placed after its first visit to the target window (and before the second visit), averaged across blocks of five trials. Within each plot separate lines are shown for the three lockout conditions.

Each model started the experiment without a bias to favor any particular strategy. The model therefore explored performance of all alternative strategies (encode-1–encode-8), which gave an average value of around four items placed during the first block of trials (Fig. 4: point A in all plots). With higher encode strategies, the model was more likely to forget at least one item. As a result, the mean number of items *placed* over the first block of trials was lower than would be expected of an agent that executed all eight strategies equally successfully (i.e., the value is lower than 4.5 items).

After the first block of trials, the patterns diverged between the different models due to the differences in the reward schemes. The models that were rewarded with the objective function of accuracy had relatively similar performance (Fig. 4: rows 3 and 4). These models placed about the same number of items in each lockout condition, with an eventual strategy that fluctuated around one to two items per visit (Fig. 4: rows 3 and 4, point D). Hence, we concluded that these models adapted successfully to their objective function: By keeping the number of (encoded and) placed blocks low, the models avoided situations where information of encoded items was lost due to failed memory retrievals.

For the accuracy conditions, the eventually chosen strategy (Fig. 4: rows 3 and 4, point D) did not differ depending on the magnitude of the reward (success or success–failure

rewarded), despite that the success–failure-rewarded model (Fig. 4: rows 4) was penalized for making errors. We had expected that as a result of this penalty the model would be more conservative in terms of the number of items encoded. This was not the case.

For the models that were rewarded based on the objective function of time (Fig. 4: rows 1 and 2), the difference between the by-trial and by-round models was striking. The by-round models (Fig. 4: rows 1 and 2, column 2) showed a rapid drop in the number of items placed (especially the total time-rewarded model) over the first four blocks of trials (Fig. 4: row 1, column 2, point B). Eventually these models placed around one to two items per visit (Fig. 4: rows 1 and 2, column 2, point D).

This end result (Fig. 4: rows 1 and 2, column 2, point D) differed from what one would perhaps expect for models that optimize time. If more than two items were placed, overall performance would have been faster. This discrepancy between our expectations and the result of the by-round models with the objective function of time can be explained by taking a closer look at the credit assignment in these models. The by-round total time-rewarded model received a penalty based on the amount of time the model spent on a strategy round. Hence, the faster one round was completed, the smaller the penalty was. This is a case where the model knew better than the modeler: It minimized the duration of each strategy round by minimizing the number of items it encoded and placed.

For the lockout time-rewarded by-round model (Fig. 4: row 2, column 2), the penalty was the experienced lockout time during a visit. As this value was constant for every strategy within each lockout condition, the temporal difference between strategy selection and reward experience was the critical factor in distinguishing the utility of different strategies (see also Section 2.2). Again, the shorter the strategy round, the smaller the temporal difference. Hence, lower encode strategies were preferred.

The by-trial rewarded models with the objective function of time (Fig. 4: rows 1 and 2, column 1) differed from their by-round (rows 1 and 2, column 2) rewarded counterparts. The by-trial rewarded models slowly adjusted the number of items that were placed per visit over the different blocks of trials. Interestingly, these were the only models that had a clear divergence in the strategy applied across different lockout conditions. This divergence in strategies is consistent with the idea that placing more items per round would speed up performance mostly in the high lockout conditions by reducing lockout costs. However, attempting to place more items per round requires that more items be encoded per round. This increase in number of items encoded raises the risk of increased forgetting, where each item encoded but forgotten counts as time wasted during initial encoding and time that must be re-incurred in a subsequent round of encoding. Both versions of the by-trial rewarded models with the objective function of time placed a small number of items (1–3) in the 0-ms and 400-ms lockout condition. In the 3,200-ms lockout condition slightly more items were placed (i.e., the mean lies above that of other conditions, but is still below five items). In this way, the models avoided situations where information was forgotten or took long time to retrieve. This point is highlighted with a C in Fig. 4 (rows 1 and 2, column 1).

Something that is not directly visible from the figures is that the performance in all by-trial models (Fig. 4: column 1) is degraded as a by-effect of the environmental constraints

that are posed on the application of strategies. Only low-encode strategies (e.g., encode-1) can be used toward the end of a trial, once a number of items has already been placed. Strategies that are used more toward the end of a trial are closer to the behavioral reward, have a smaller temporal difference value (i.e., $t_j - t_i$ in Eq. 2), and (given similar behavioral rewards) a higher expected reward and expected utility. Due to this high utility value, they are more likely to be selected in future choice situations. In effect these choices decrease the average number of items placed after a first visit. We will discuss ways of overcoming this problem in the general discussion.

4.7.2. Comparing model and human behavior

The previous analyses demonstrated the core contribution of this article: Model performance changed depending on the settings for moment, objective function, and magnitude of reward. But how did the models perform compared to human performance? To this end, we compared model behavior (Fig. 4) with human behavior (Fig. 2) on two aspects: learning trajectory, and eventual learned strategies. Two aspects characterized the learning trajectory: the speed with which a relatively stable strategy was adopted, and the direction in which the learning occurred. As can be seen in Fig. 2, human performance changed the most (depending on lockout condition) during the first two blocks of trials.

Of all the reward variations, the total time-rewarded by-round model (Fig. 4: row 1, column 2) stabilized the fastest, requiring roughly four blocks (i.e., 20 trials). All other variants took longer, sometimes requiring up to eight blocks to learn (e.g., as in the total time-rewarded, by-trial model, Fig. 4: row 1, column 1). This striking difference between model and human learning speed was implicitly recognized in prior studies as the prior ACT-R model (Gray et al., 2005) only considered performance after 25 trials (five blocks of five trials per block) and the Q-learning model (Gray et al., 2006) only considered performance after 100,000 model trials (i.e., 20,000 blocks).

The second aspect that characterized the learning process was the direction in which learning occurred. As can be seen in Fig. 2, human participants started off placing a relatively small set of items. Across blocks, the number of items that was placed per visit slowly increased—especially in the higher lockout condition.

In contrast, our models started with placing an average of four items per visit (Fig. 4: point A in all plots), and then slowly decreased this number. This was a result of the model's broad exploration of performance of all strategies—the model had no past experience with the imperfect nature of memory retrieval (unlike the human participants). At the beginning of the experiment, all strategies were explored, with an average number of items placed around four. As encoding a high number of items during one visit comes at the risk of forgetting those items (Anderson & Schooler, 1991), the model gradually learned to avoid high encode strategies. This is reflected in the results in a drop of the number of items placed.

The final aspect of comparison was on the strategies adopted by the end of the 48 trials. For human participants the number of items placed increased with an increase in lockout time. In contrast, the only models that complied with this pattern were the by-trial models rewarded using the objective function of time (Fig. 4: rows 1 and 2, column 1). There were

two model variants of this: the lockout time-rewarded (row 2, column 1) and the total time-rewarded (row 1, column 1) models.

The performance of these two models overlapped with different aspects of the human data. Performance of the lockout time-rewarded model was consistent with the finding in the human data that across all blocks of trials (and not just the last blocks) the number of items that was placed increased with an increase in lockout time. In contrast, performance of the total time-rewarded model fluctuated more. However, eventually this model captured the same trend (Fig. 4, row 1, column 1, point C).

An aspect in favor of the lockout time-rewarded model over the total time-rewarded model was the observation that for this model in the last block there was a larger difference between the number of items placed in each of the lockout conditions (Fig. 4, row 1, column 1, point C). However, in disfavor of this model was that it underpredicted the number of items placed in the 0-ms lockout condition.

5. General discussion

5.1. Why ''when, what, and how much to reward'' matters

Investigations of cognition using reinforcement learning models require modelers to set parameters for the cognitive architecture, task environment, the actor, and the critic. Previous research has investigated how settings of each of these sets of parameters influence model performance (e.g., Ahn et al., 2008; Sutton & Barto, 1998; Yechiam & Busemeyer, 2005). However, the *reward* parameters for the critic have received relatively less attention (though some exploration is reported in Singh et al., 2009). This is surprising as the rationale behind ''the when, what, and how much'' of rewards is as important as the rationale behind other parameters. Rewards reflect the success of a model in achieving its goals. By changing when, what, and how much is rewarded, essentially the model's reflection on its performance is affected.

Until the moment of stronger insight from cognitive neuroscience (e.g., Cohen, 2008; Holroyd & Coles, 2002; Schultz, 2006; Schultz et al., 1997), those interested in modeling human cognitive behavior are left unconstrained in their choices for ''when, what, and how much,'' unless strong experimental control is used for these factors. In other, perhaps more naturalistic settings, an explicit reward signal might not be as clear-cut. This article is the start of a more principled investigation of the effects of different settings for these parameters for human cognition.

In Section 4.7, we demonstrated that changing only these parameters (moment, objective function, and magnitude of reward) affects model performance. These factors affect the learning path and result in different optimal strategies. Out of all reward-related parameters, the *moment* of rewarding influences performance the strongest, when judged by performance after an extensive learning period (i.e., toward the end of the trial). The best qualitative fit to human data is for models that distribute rewards at the end of a trial (i.e., by-trial). Conceptually, this matches with the understanding that local optimization of behavior

(by-round) does not always result in global optimization (e.g., Fu & Gray, 2006). Within the by-trial models, those with the objective function of time (Fig. 4: rows 1 and 2, column 1) fit the human data substantially better than models with the objective function of accuracy. Magnitude of reward (i.e., total time-rewarded or lockout time-rewarded) only leads to minor differences in the learning process and eventual preferred strategy.

Our model is developed within the cognitive architecture ACT-R (Anderson, 2007), and the model's structure is based on two previous modeling efforts (Gray et al., 2005, 2006). This provides us with previously validated and motivated theories for the cognitive architecture, the environment, the critic, and the actor thereby reducing the number of parameter choices made for this specific model. Any of these parameter choices can be argued with, and—unavoidably—changes in these parameter settings can affect the results. This critique also holds for other studies that investigated general characteristics of reinforcement learning models (e.g., Ahn et al., 2008; Sutton & Barto, 1998; Yechiam & Busemeyer, 2005). On the upside, having these choices made and motivated by previous research allowed us to focus our efforts on the parameter set for rewards.

One aspect that is relevant but has not been extensively studied in this study is the predictability of a reward. In our study, this predictability is dependent on the agent's ability to retrieve information on all encoded items successfully (as the corresponding memory equations were probabilistic; Anderson & Schooler, 1991). However, in many experiments the task environment is also probabilistic. Depending on the nature of the rewards, and the discriminability of alternative actions and their rewards, probabilistic settings with instable rewards might require a longer learning trajectory.

5.2. Generality of the findings to other settings

How do our findings translate to other tasks and settings? An investigation of ''moment'' requires a task that has multiple steps that contribute to the overall outcome of a task, and which preferably provide feedback after each step. This contrasts with tasks with (multiple) independent single decisions. Of particular interest might be to investigate whether the distinction between by-trial and by-round feedback can scale up to explain effects known as melioration of performance, where maximization of short-term gains requires different action sequences than maximization of long-term gains (e.g., Gureckis & Love, 2009; Herrnstein, 1990; Neth, Sims, & Gray, 2006; Shanks, Tunney, & McCarthy, 2002).

The magnitude of rewards can vary more easily between alternative settings and will depend on the objective function at hand. The two objective functions that we investigated here, speed and accuracy, are straightforward manipulations of traditional experimental design factors (e.g., speed-accuracy trade-off; Edwards, 1965) and are present in many experiments. Our study found that optimization of time leads to good results. The generality of this finding can be further investigated.

Alternative choices for objective function are possible, as visible in the examples in Table 1. A typical domain for studying the impact of different objective functions on performance might be multitasking situations. Here, each of the tasks can have its own reward signal associated with it, and it is up to the participant to balance off the rewards on each of the

tasks (i.e., to favor one objective function over another) to achieve good overall performance. How the rewards are traded off might change with the objective that the participant or agent has (for a broader discussion, see Janssen & Brumby, 2010).

To help a participant make a careful balance between tasks, in some multitasking experiments and models performance on all tasks has been expressed in the same currency (points, or money). Effectively this fixes the objective function, while changing the rate of rewards between tasks, which is related to the magnitude (e.g., Janssen, Brumby, Dowell, Chater, & Howes, 2011; Wang, Proctor, & Pick, 2007). This mostly allows for further investigation of the moment of reward, as even when tasks are expressed in a common currency, different tasks might trigger their rewards at different moments, which again might change performance trade-offs (e.g., Neth, Khemlani, & Gray, 2008).

5.3. Comparison with preceding modeling efforts

Two modeling efforts preceded the current exploration of performance in the Blocks World task. It is interesting to compare these efforts with the current work, as they have subtle differences in the settings for the cognitive architecture (embedded in a cognitive architecture framework or not) and the critic (probability matching or Q-learning). Note that with both models only performance after some experience with the task was investigated, and not performance during the learning process, as is done in the current article. Also, preceding work did not explore performance for all eight reward-related parameter settings as is done here.

The result in this article that a by-trial-rewarded model with the objective function of time provides a good qualitative fit to human data is in line with findings from the Q-learning model of the Blocks World task (Gray et al., 2006). However, the Q-learning model was made to reflect ideal performance and had significantly more training experience than our current ACT-R model. Upfront it was unknown whether a model would also perform this well after only limited learning experience.

The ACT-R 5 model of the Blocks World task (Gray et al., 2005) is more comparable to the current model, as it experienced the same number of trials and has similar parameter settings for cognitive architecture (ACT-R), environment, and actor. It differs in the method for the critic (a non-reinforcement learning approach of probability learning is used; Lovett, 1998). When the default binary reward function of this learning mechanism is used, the model provides a very poor fit to the data (independent of the moment at which the reward was experienced).

Gray et al. (2005) also developed ACT-R 5 models that incorporated scalar rewards, a feature that is natural to reinforcement learning-based cognitive models, but that required adaptations in the ACT-R 5 architecture. They tested several magnitudes of the rewards for models with an objective function of accuracy that were rewarded by-round. When judged after 25 trials (i.e., five blocks) of experience, their best fitting model performs about equal to our best fitting model. This is striking, as the by-round models and the accuracy rewarded models perform poorly in the study reported in the current article.

Despite these differences in which reward settings are the best, all modeling results support a general claim: Incorporating scalar rewards, which is a natural feature of reinforce-

ment learning, is essential for capturing the wide variety of rewards, and human adaptation to these rewards in tasks like Blocks World (Gray et al., 2005).

5.4. Limitations and future work

Our model had two limitations. First, the trajectory in which learning occurs differs between humans and the model. Humans start the experiment by placing a small number of items, and slowly learn to *increase* this with experience. In contrast, the models start with exploring performance of all strategies, and then gradually *decrease* the number of items that is encoded and placed. One way of overcoming this is by biasing the models to prefer specific strategies (e.g., encode-1 and encode-2) at the beginning of the experiment by setting initial utility values. We decided not to do this in the current study as it reduced the learning problem and overshadowed our investigation of the three parameters of interest. Moreover, previous modeling efforts (Gray et al., 2005, 2006) also did not manipulate utility values in this way. Hence, making different assumptions would make a comparison between modeling approaches more difficult.

Another shortcoming is that the model's performance is degraded as a side effect of environmental constraints imposed on the application of strategies. As only low-encode strategies (i.e., encode-1) can be used toward the end of a trial, these are most close to the moment at which a reward is experienced in the by-trial rewarded models. Due to temporal difference mechanisms, these strategies will also receive a higher utility value, and they are more likely to be used in future situations—even at the start of a trial. In contrast, humans are more stable in their selected strategy.

One way to avoid this problem is by developing a model that differentiates strategies based on both the state of the world (how many items have been placed so far) and based on how many items are encoded. This is exactly what the Q-learning model by Gray et al. (2006) did. However, Q-learning is only guaranteed to find an optimal solution if enough trials are run. As the goal of Gray et al. (2006) was to determine optimal performance with an objective function that minimized time, they trained their model for 100,000 trials. Clearly, such a training period is implausible as a model of human learning. Alternatively, hierarchical learning approaches might be explored (Botvinick, Niv, & Barto, 2009).

We have not investigated individual differences in performance. Given that the Blocks World task allows for different conceptions of rewards, it might be that different participants focused on different reward aspects of the task. That is, some participants might focus more on time-related measures; others might focus more on accuracy. Our current setup did not allow us to test for effects of reward conception on individual performance. Any differences in performance between individuals could have arisen due to differences in the conception of rewards but also due to differences in cognitive characteristics (e.g., better memory). To analyze individual differences, one would need to know individual cognitive characteristics (cf. the methodology introduced in Howes et al., 2009).

Finally, although a broad set of combinations of moment, objective function, and magnitude has been tested, one concern can be that the set was limited. Hence, we make no claim for completeness or optimality of our selection. However, we do claim that our selection

provides a reasonable start for exploring how the space of moments, objective functions, and magnitudes influences the ability of temporal difference learning based reinforcement learning models to predict human behavior. Moreover, most of our choices are straightforward manipulations of traditional experimental design factors (e.g., speed-accuracy trade-off; Edwards, 1965). Given the strong effect of the moment of reward in the current studies, future studies can focus on a more detailed and broad investigation of the other parameters: objective function and magnitude (cf. Singh et al., 2009). As these parameters are used in all frameworks that incorporate reinforcement learning, their effect can be tested across a multitude of cognitive models and architectures, and the outcomes can further our general understanding of task acquisition and learning.

Notes

1. Note that the exact method for calculating the temporal difference can vary between different implementations of temporal difference learning (Sutton & Barto, 1998), as part of the parameter settings for the critic (see also Section 2.1 and Ahn et al., 2008; Yechiam & Busemeyer, 2005).
2. Note that in the current ACT-R architecture, a model can have positive and negative rewards. Negative rewards can be incurred in two ways: r_j can be negative, and R_i can become negative when $t_j - t_i$ is larger than r_j. As a reviewer pointed out, it might be that negative rewards are dealt with by different neural substrates, and the use of negative rewards might be questioned. In this article we do not commit to a specific range of values that r_j and R_i can have. We leave this to future research and refer the interested reader to pages 160–164 of Anderson (2007) for some of the motivations behind this architectural choice.
3. In previous studies ''items'' were referred to as ''blocks,'' hence the name ''Blocks World.'' We changed the terminology here to avoid confusion with ''blocks of trials.''
4. Note that although Gray et al. (2006) report six between-Ss conditions, we follow the lead of Gray et al. (2005) in running models on three conditions.
5. Note that in practice, this also speeded up the learning process. Instead of having to learn the utility of all possible combinations of production rules, our model could focus on the utility of different combinations of the encode-strategies.
6. Note that human participants made very few errors. Of the items that were placed directly after the first visit only 9% were placed incorrectly. For these items, either the color, position, or color plus position was incorrect.

Acknowledgments

Christian Janssen's work on this project was supported, in part, by EPSRC grant EP/G043507/1. Wayne Gray's work on this project was supported, in part, by grants

N000140710033 and N000140910402 from the Office of Naval Research, Dr. Ray Perez, Project Officer. We would like to thank the reviewers for their valuable comments on earlier versions of this manuscript. We would also like to thank Chris Sims and Duncan Brumby for feedback on previous versions of this manuscript, Michael Schoelles for his help with developing the model, and Hedderik van Rijn for feedback on Christian Janssen's M.Sc. dissertation that formed the basis of this article.

References

Ahn, W.-Y., Busemeyer, J. R., Wagenmakers, E., & Stout, J. C. (2008). Comparison of decision learning models using the generalization criterion method. *Cognitive Science*, *32*, 1376–1402.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*, 1036–1060.

Anderson, J. R., & Lebiere, C. J. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*, 396–408.

Ballard, D. H., Hayhoe, M. M., & Pelz, J. B. (1995). Memory representations in natural tasks. *Journal of Cognitive Neuroscience*, *7*, 66–80.

Ballard, D. H., Hayhoe, M. M., Pook, P. K., & Rao, R. P. N. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, *20*, 723–742.

Ballard, D. H., & Sprague, N. (2007). On the role of embodiment in modeling natural behaviors. In W. D. Gray (Ed.), *Integrated models of cognitive systems* (pp. 283–296). New York: Oxford University Press.

Barto, A. G., Sutton, R., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, *13*, 835–846.

Bothell, D. (Producer). (2008) ACT-R 6 reference manual. Accessed June 2008.

Botvinick, M., Niv, Y., & Barto, A. G. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, *113*, 262–280.

Cohen, M. X. (2008). Neurocomputational mechanisms of reinforcement-guided learning in humans: A review. *Cognitive, Affective & Behavioral Neuroscience*, *8*, 113–125.

Davis, D. G., Staddon, J. E., Machado, A., & Palmer, R. G. (1993). The process of recurrent choice. *Psychological Review*, *100*, 320–341.

Daw, N. D., & Frank, M. J. (2009). Reinforcement learning and higher level cognition: Introduction to special issue. *Cognition*, *113*, 259–261.

Edwards, W. (1965). Optimal strategies for seeking information: Models for statistics, choice reaction times, and human information processing. *Journal of Mathematical Psychology*, *2*, 312–329.

Erev, I., & Barron, G. (2005). On adaptation, maximization, and reinforcement learning among cognitive strategies. *Psychological Review*, *112*, 912–931.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*, 381–391.

Fu, W. T., & Anderson, J. R. (2004). Extending the computational abilities of the procedural learning mechanism in ACT-R. In: K. D. Forbus, D. Gentner, & T. Regier (Eds.), *Proceedings of the 26th Annual Meeting of the Cognitive Science Society* (pp. 416–421). Austin, TX: Cognitive Science Society.

Fu, W. T., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General*, *135*, 184–206.

Fu, W. T., & Gray, W. D. (2006). Suboptimal tradeoffs in information seeking. *Cognitive Psychology*, *52*, 195–242.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied*, *6*, 322–335.

Gray, W. D., & Fu, W. T. (2004). Soft constraints in interactive behavior: The case of ignoring perfect knowledge in-the-world for imperfect knowledge in-the-head. *Cognitive Science*, *28*, 359–382.

Gray, W. D., Schoelles, M. J., & Sims, C. R. (2005). Adapting to the task environment: Explorations in expected value. *Cognitive Systems Research*, *6*, 27–40.

Gray, W. D., Sims, C. R., Fu, W. T., & Schoelles, M. J. (2006). The soft constraints hypothesis: A rational analysis approach to resource allocation for interactive behavior. *Psychological Review*, *113*, 461–482.

Gureckis, T. M., & Love, B. C. (2009). Short-term gains, long-term pains: How cues about state aid learning in dynamic environments. *Cognition*, *113*, 293–313.

Herrnstein, R. J. (1990). Behavior, reinforcement and utility. *Psychological Science*, *1*, 217–224.

Holroyd, C. B., & Coles, M. G. H. (2002). The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review*, *109*, 679–709.

Howes, A., Lewis, R. L., & Vera, A. (2009). Rational adaptation under task and processing constraints: Implications for testing theories of cognition and action. *Psychological Review*, *116*, 717–751.

Janssen, C. P., & Brumby, D. P. (2010). Strategic adaptation to performance objectives in a dual-task setting. *Cognitive Science*, *34*, 1548–1560.

Janssen, C. P., Brumby, D. P., Dowell, J., Chater, N., & Howes, A. (2011). Identifying optimum performance trade-offs using a cognitively bounded rational analysis model of discretionary task interleaving. *Topics in Cognitive Science*, *3*, 123–139.

Lovett, M. C. (1998). Choice. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 255–296). Mahwah, NJ: Erlbaum.

Montague, P. R., Dayan, P., & Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of Neuroscience*, *16*, 1936–1947.

Morgan, P. L., Patrick, J., Waldron, S. M., King, S. L., & Patrick, T. (2009). Improving memory after interruption: Exploiting soft constraints and manipulating information access cost. *Journal of Experimental Psychology: Applied*, *15*, 291–306.

Napoli, A., & Fum, D. (2010). Rewards and punishments in iterated decision making: An explanation for the frequency of the contingent event effect. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 175–180). Philadelphia, PA: Drexel University.

Nason, S., & Laird, J. E. (2005). SOAR-RL: Integrating reinforcement learning with SOAR. *Cognitive Systems Research*, *6*, 51–59.

Neth, H., Khemlani, S. S., & Gray, W. D. (2008). Feedback design for the control of a dynamic multitasking system: Dissociating outcome feedback from control feedback. *Human Factors*, *50*, 643–651.

Neth, H., Sims, C. R., & Gray, W. D. (2006). Melioration dominates maximization: Stable suboptimal performance despite global feedback. In R. Sun (Ed.), *Proceedings of the 28th Annual Meeting of the Cognitive Science Society* (pp. 627–632). Austin, TX: Cognitive Science Society.

Rieskamp, J., & Otto, P. E. (2006). SSL: A theory of how people learn to select strategies. *Journal of Experimental Psychology: General*, *135*, 207–236.

Schultz, W. (2006). Behavioral theories and the neurophysiology of reward. *Annual Review of Psychology*, *57*, 87–115.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, *275*, 1593–1599.

Shanks, D. R., Tunney, R. J., & McCarthy, J. D. (2002). A re-examination of probability matching and rational choice. *Journal of Behavioral Decision Making*, *15*, 233–250.

Sims, C. R., & Gray, W. D. (2004). Episodic versus semantic memory: An exploration of models of memory decay in the serial attention paradigm. In M. Lovett, C. Schunn, & C. Lebiere (Eds.), *Proceedings of the 6th*

International Conference on Cognitive Modeling (pp. 279–284). Mahwah, NJ: Lawrence Erlbaum Associates.

Singh, S., Lewis, R., & Barto, A. G. (2009). Where do rewards come from? In N. Taatgen & H. Van Rijn (Eds.), *Proceedings of the 31st Annual Meeting of the Cognitive Science Society* (pp. 2601–2606). Austin, TX: Cognitive Science Society.

Soukoreff, R. W., & MacKenzie, I. S. (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in HCI. *International Journal of Human-Computer Studies*, *61*, 751–789.

Sun, R., Merrill, E., & Peterson, T. (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, *25*, 203–244.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Waldron, S. M., Patrick, J., Morgan, P. L., & King, S. (2007). Influencing cognitive strategy by manipulating information access. *The Computer Journal*, *50*, 694–702.

Walsh, M. M., & Anderson, J. R. (2009). The strategic nature of changing your mind. *Cognitive Psychology*, *58*, 416–440.

Wang, D. D., Proctor, R. W., & Pick, D. F. (2007). Acquisition and transfer of attention allocation strategies in a multiple-task work environment. *Human Factors*, *49*, 995–1004.

Watkins, C., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292.

Yechiam, E., & Busemeyer, J. R. (2005). Comparison of basic assumptions embedded in learning models for experience-based decision making. *Psychonomic Bulletin & Review*, *12*, 387–402.