The Utility of Affect in the Selection of Actions and Goals under Real-World Constraints

Paul Schermerhorn and **Matthias Scheutz**
Human-Robot Interaction Laboratory
Cognitive Science Program
Indiana University
Bloomington, IN, USA
{pscherme,mscheutz}@indiana.edu

Abstract

We present a novel affective goal selection mechanism for decision-making in agents with limited computational resources (e.g., such as robots operating under real-time constraints). We argue that when deciding whether to undertake some action, affective states can serve as subjective estimates of the likelihood of that action succeeding. Given that the affective states may reflect, in part, the recent history of successes and failures for a given action type, their roles in action selection can be viewed as analogous to temporal probabilistic decision models such as Markov decision processes. We show how "affect-influenced decision making" can provide low-cost mechanisms to break out of potentially costly sequences of failed actions in the absence of either knowing or being able to compute the actual utility of performing a particular action.

1 Introduction

Making decisions effectively under real-time real-world constraints, as it is often required for autonomous robots, for example, is very hard. The reason for the difficulty lies in the fact that optimal or *rational* decisions are often not feasible, either because the calculations are too time-consuming or because the agent may not always have sufficient reliable knowledge. Specifically, rational choice requires the calculation of expected utilities for each possible action and the selection of the action with the highest expected utility to the agent. Moreover, it requires the agent to have a good-enough knowledge of the current world state and of the world's state transition model (i.e., how likely it is that actions will end of up in the desired state). Humans seem to cope with these complexities not so much using rational utility-theoretic methods, but rather by using affective evaluations of a situation, which can be performed by quick, low-cost heuristics (that can be subsequently verified or corrected by slower, more accurate cognitive processing, if there is enough time).

In this paper, we demonstrate the use of "affective evaluations" to address some difficulties facing strictly "rational" methods of action selection under time, computation, and knowledge constraints. Specifically, the utility of affect in decision making is demonstrated for two classes of problems: deciding whether to repeat a failed attempt to achieve a goal and deciding between actions in the service of (possibly) unrelated goals. In each case, it is shown that an agent's affective states can contribute to beneficial prioritization of actions and goals.

Section 2 applies classical utility theory to decision making, including the use of Markov decision processes to integrate new information. In Section 3, we introduce the method of *affective goal management*, which uses affective states to aid in fast, accurate prioritization of goals. Finally, Section 4 presents some concluding remarks.

2 Probabilistic Decision Strategies

A perfectly rational agent with perfect information can make optimal decisions by selecting the action with the maximum utility. Because the agent knows the costs and benefits of each alternative and the probabilities of each action succeeding, it cannot be wrong about which is the most profitable choice. In reality, however, costs and benefits are only approximately known. More importantly, real-world constraints can make it difficult to estimate accurately the probability of success. These probabilities are, again, approximations, due to incomplete knowledge and chance, but they do allow the agent to make more informed evaluations of the expected utility of its alternatives based on its present knowledge.

2.1 Expected Utility Based Approaches

Given a binary choice point, a choice between undertaking an action or not, it is rational to choose the action when it has positive expected utility u. The probability of the action succeeding is p, the benefit of succeeding is b, and the cost of attempting the action is c. The success component of expected utility is then $p \cdot (b - c)$, and the failure component is $(1 - p) \cdot c$. Hence, the action's expected utility $u = p \cdot (b - c) - (1 - p) \cdot c = pb - pc - c + pc = pb - c$. u can then be used to make the decision: should the action be attempted? If u is positive, yes, otherwise, no.

Take, for example, a robotic agent that (as part of some higher-level task) must move to some location L, but does not know how to get to L from its current position. The robot must decide whether to ask someone for directions. Receiving directions will provide a certain benefit to the agent (via a reduction in the time it takes to reach L, or an increase in the probability that the robot reaches L at all), but asking and listening for the response incurs some cost (e.g., the energy costs of asking and listening, or the time it takes). When the benefit of obtaining directions $b = 100$, the cost of asking for directions is $c = 5$, and the prior probability of the person responding with intelligible directions is $p = 0.5$, the expected utility of a single attempt at asking for directions is $u = 0.5 \cdot 100 - 5 = 45$, so the robot should choose to ask.

The decision process can be extended to sequences of multiple attempts by calculating the expected utility of an n-attempt sequence. Note that we are concerned here only with sequences of events with 0 or 1 successful attempt, as there is no benefit to making additional attempts once the action succeeds (i.e., the agent continues until it succeeds or reaches n attempts). When the agent succeeds after k cycles, the net utility is $b - kc$—the agent receives the benefit and pays the cost k times. The overall expected utility, then, includes the sum of the net utility that would be realized by a success at each attempt scaled by the probability of succeeding at that attempt. For example, the contribution for the first attempt is $p \cdot (b - c)$ (i.e., with probability p the agent succeeds on that attempt receives the benefit and pays the cost once), the contribution for the second attempt is $(1-p) \cdot p \cdot (b-2c)$ (i.e., the process reaches the second attempt with probability $(1 - p)$, and with probability p succeeds in that attempt, receiving the benefit having paid the cost of attempting twice). Additionally, we must include the expected cost of failing on all n attempts (e.g., for the two attempt case, this is $(1 - p)^2 \cdot 2c$, or the probability of failing twice times the cost of two attempts). In general, the expected utility u for n attempts is:

$$u_n = \sum_{k=1}^{n} (1 - p)^{k-1} \cdot p \cdot (b - kc) - (1 - p)^n \cdot nc \qquad (1)$$

Thus, the probability of succeeding within the n attempts is the sum of the probabilities of succeeding at each attempt, whereas the net reward for succeeding is the benefit less the cost of the attempts, and the agent pays the cost of n attempts without any benefit with a probability that it fails in each attempt.

Returning to the robotic example, if the prior probability of someone responding intelligibly were $p = 0.5$, while b remains 100 and c remains 5, the expected utility of asking up to 20 times is $u_{20} = \sum_{k=1}^{20} 0.5^{k-1} \cdot 0.5 \cdot (100 - 5k) - 0.5^{20} \cdot 100 = 89.9999$. u is positive (because it is very likely that the agent will succeed within the $n = 20$ attempts), so the rational agent will commit to asking up to 20 times.

2.2 Markov Decision Processes

In cases where the agent can retry an action after a failed attempt, calculating the expected utility of n attempts provides a more accurate evaluation of the alternative actions than repeatedly calculating the expected utility of a single attempt, as shown above. The evaluation can be improved even more for a broad class of tasks, in which information regarding previous failures can be used to improve the accuracy of the probabilities used in the calculation. Specifically, the sequence of attempts can be analyzed as a Markov chain

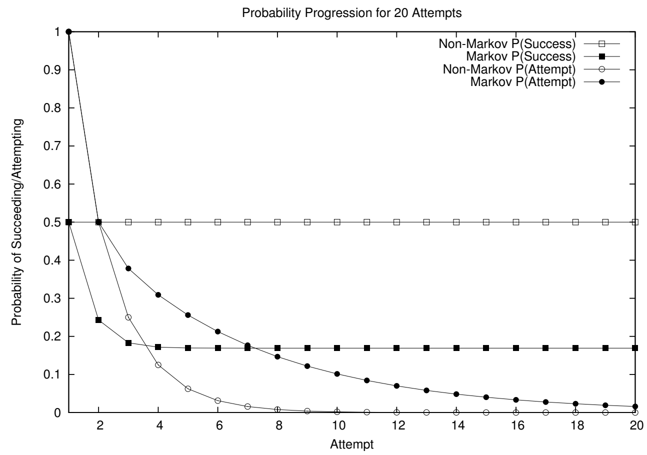

Figure 1: The probabilities of succeeding at attempt k and of reaching attempt k for the robot example.

in which previous outcomes influence the current assessment of the probability of success. For example, if the agent is asking for directions, the failure of the previous attempt can be taken as evidence that some other factor, such as ambient noise in the room or poor enunciation by the speaker, is working to reduce the probability of success. Using the conditional probability of succeeding given the result of the previous attempt (i.e., for a first-order Markov chain), we can calculate an updated probability distribution for success on the current attempt as follows:[1]

$$\mathbf{P}(\mathbf{S}_t|\mathbf{a}_{1:t}) = \mathbf{P}(\mathbf{a}_t|\mathbf{S}_t) \cdot \sum_{\mathbf{s}_{t-1}} \mathbf{P}(\mathbf{S}_t|\mathbf{s}_{t-1}) \cdot P(\mathbf{s}_{t-1}|\mathbf{a}_{1:t-1}) \qquad (2)$$

where S_t is the state at iteration t, a_t the new evidence (action outcome) for iteration t, and $a_{1:t}$ the evidence from iterations 1 through t. This probability distribution can be used as part of a partially observable Markov decision process (POMDP) for reasoning about actions and goals [Kaelbling *et al.*, 1998].

Assume for the robotic example that the probability that the speaker says something intelligible given that the previous utterance was intelligible is 0.7, and given that the previous utterance was unintelligible is 0.45. Also, the probability that the speech detector will return a valid parse given an intelligible utterance is 0.75, and given an unintelligible utterance is 0.2. Then the values for p_k begin with the prior probability 0.5 and become progressively worse before converging by the eleventh attempt: 0.5, 0.243129, 0.183152, 0.171684, 0.16958, 0.169197, 0.169127, 0.169114, 0.169112, 0.169111, 0.169112. Again, p_k decreases because for each attempt, the previous attempt failed; otherwise the agent stops trying. Figure 1 compares the probabilities of success for each attempt and the probabilities of reaching each attempt for this particular example.

The probability of success for each iteration p_k then replaces p in the utility calculation:

[1]Note that the basic utility calculation described above can be viewed as a zero-order Markov chain.

$$u_n = \sum_{k=1}^{n} \prod_{j=1}^{k-1} (1-p_j) \cdot p_k \cdot (b-kc) - \prod_{j=1}^{n} (1-p_j) \cdot nc \quad (3)$$

Because p_k changes with each attempt, the probability of $k-1$ failures must be calculated using, not the current p_k, but the values for each prior attempt. However, it is not necessary to recalculate all previous p_k, as the expected utility calculation can be performed recursively. In the robot example, the POMDP calculates a somewhat decreased $u_{20} = 80.5658$. It is unsurprising that the two methods yield similar results, given the high probability of success within 20 attempts and the relatively small cost. If c is increased to 45, the outcome is different: $u_{20} = 7.5$ using the standard utility calculation, whereas $u_{20} = -5.34354$ for the POMDP. The agent's decision will depend in this case on which method is used.

2.3 Repeated Attempts

The POMDP described above is effective at taking into account the information encoded in failed attempts, however, computing the expected utility of a potentially long sequence of attempts may be too expensive for real-time applications. Even if expense were not an issue, neither of the probabilistic strategies says anything about how many attempts should be made, only the expected utility of n attempts (where the single-attempt instance is a special case for which $n = 1$).

It turns out not to be worthwhile to use the expected utility calculations to pinpoint the optimal number of attempts (i.e., by calculating u for $n = 1, 2, 3, \ldots$, and choosing the n with the highest u), because u (as calculated by 1 and 3) is monotonically increasing or decreasing, depending on the values of p, b, and c. This is easiest to see when calculating basic expected utility. The expected utilities of an n iteration sequence and an $n+1$ iteration sequence are identical through the nth iteration. The difference comes after that, where the n-sequence has to subtract the expected utility of paying the cost n times ($(1-p)^n \cdot nc$), and the $n+1$-sequence adds the expected utility of success on the $n+1$st iteration ($(1-p)^n \cdot p(b-(n+1)c)$) and subtracts the expected utility of paying the cost $n+1$ times ($(1-p)^{n+1} \cdot (n+1)c$). Hence, if $-(1-p)^n \cdot nc < (1-p)^n \cdot p(b-(n+1)c) - (1-p)^{n+1} \cdot (n+1)c$, the expected utility will continue to increase as n increases:

$$\begin{aligned}
-(1-p)^n \cdot nc &< (1-p)^n \cdot p(b-(n+1)c) - (1-p)^{(n+1)} \cdot (n+1)c \\
-nc &< p(b-(n+1)c) - (1-p)(n+1)c \\
-nc &< pb - pnc - pc - nc - c + pnc + pc \\
0 &< pb - c
\end{aligned}$$

So, as long as $pb > c$, the expected utility is monotonically increasing, and when $pb < c$, the expected utility is monotonically decreasing. When $pb = c$, u is constant at 0 for any value of n—the "break-even" configuration, where the costs are exactly offset by the benefits. In the robot example, this is the case when $c = 50$, and even a slight perturbation up or down leads to increasingly positive or negative u as n increases.[2] Hence, because the decision is determined by the sign of the expected utility, a choice based on the calculation of u_n will be the same for any $n \geq 1$. By itself, the expected utility calculation can recommend only two courses of action: make no attempts, or continue making attempts until success, regardless of how large n gets.

[2]Practically speaking, the values of u converge as the computer precision limits are approached.

The behavior of the Markov process is somewhat complicated by the changing value of p; in the probabilistic version, the break-even configuration represents a strict border between increasing and decreasing utility. There is no corresponding break-even configuration for the POMDP. Like the basic utility calculation, there are large regions of monotonically increasing and decreasing expected utility, but separating them is a narrow region in which expected utility initially rises somewhat before beginning its monotonic decrease. This is due to the initial optimistic value of p_k. After a few failed attempts, p_k is reduced to such a degree that the costs begin to dominate. Hence, POMDPs do not reliably provide results that differ from evaluating the expected utility of a single attempt; in most cases the result will be either zero or unlimited attempts, but there is a small (and difficult to predict) region in which a limited, non-zero value for n would result.

Because neither probabilistic approach distinguishes between attempt sequences of differing length, neither is capable of determining a limit on the number of attempts to make. One approach to ending attempt sequences is an imposed limit that reflects the maximum cost the agent is willing to spend to obtain the benefit. In some cases, for example, strict time constraints exist, providing a practical limit on n. Alternatively, an artificial constraint, such as not continuing after the total costs exceed the benefits ($nc > b$), could provide a practical limit. However, such limits (being artificial) serve as upper bounds on the number of attempts the agent should make; it is likely that in some cases, the agent would maximize utility by stopping earlier.

2.4 Discussion

The probabilistic decision strategies outlined here can, in some circumstances, lead to good choices. However, because of the practical constraints imposed on agents, it may not be feasible to use such approaches. Without knowledge of the probabilities of success associated with each potential alternative, it is not possible to calculate expected utility. Moreover, constraints on computation and time can make extended calculations too costly for the agent. Humans are very good at making fast decisions under these constraints, using affect to help prioritize goals based on so-called "gut feelings" about which one is best (e.g., [Sloman *et al.*, 2005; Clore *et al.*, 2001; Kahneman *et al.*, 1997]. We propose a decision strategy that uses affect to address these challenges for fast, adaptive decision making.

3 Affective Goal Management

In this section, we contrast an affective goal management strategy with the previously-described probabilistic strategies for dealing with repeated attempts and examine affect's role in goal prioritization for two additional related scenarios: choosing between multiple alternatives and evaluating the utility of actions that may satisfy multiple goals.

3.1 Affective Decision Making

The *affective goal manager* (AGM) uses a simple history for successes and failures of atomic or basic actions, which effectively provides a subjective assessments of the success of actions based on recent experience–we construe these simple evaluations as primitive "affective states". These affective states are updated based on the performance of system components, giving the same advantage as the Markov process: as additional information is acquired, estimates of the likelihood of failure and success are adjusted. In fact, we claim that the AGM implements an approximation of a POMDP, albeit one that does not require knowledge beforehand of any conditional probabilities.

The AGM employs a fast, online decision-making process, avoiding potentially costly expected utility calculations for long sequences of iterations. The decision mechanism does not take the expected value of future attempts into account, instead choosing for the current iteration whether to attempt the action. This, combined with the use of affective states as constantly updated subjective estimates of success, allows the AGM to limit the number of attempts the agent will make based on its recent experience, an approach preferable to imposing artificial limits.

The agent's overall affective state is represented by two state variables, one which records positive affect (A_P), and the other of which records negative affect (A_N) [Sloman *et al.*, 2005]. A_P and A_N are reals in the interval $[0, 1]$ that are influenced by the performance of the agent's various subsystems (e.g., speech recognition). When a subsystem records a success, it increases the level of positive affect, and when it fails, it increases the level of negative affect. Specifically, success increases A_P by $\Delta A_P = (1 - A_P) \cdot trig$ (failure updates A_N analogously). This update function ensures that A_P remains in the interval $[0, 1]$. Both affective states are also subject to regular decay, bringing their activations in the absence of triggering events back to their rest values (i.e., 0): $\Delta A_P = A_P \cdot dec$ [Scheutz, 2001].

All else being equal (i.e., with both affect states starting at rest and no affect triggers from other sources), the value of $trig$ determines how many failed attempts the agent will make of an action before giving up. With greater $trig$, the value of A_N rises faster, leading the agent to reduce its subjective assessment of the expected benefit (i.e., to become "pessimistic" that the benefit will be realized).

The AGM makes online choices based on the expected utility of a single attempt, similar to the basic expected utility calculation above. However, as noted, accurate probabilities may not be available to the agent. For this reason, the affect states A_P and A_N are used to generate an "affective estimate" of the likelihood of success: $a = (1 + A_P^2 - A_N^2)$.[3] This value is then used in the calculation of the expected utility of an action: $u = a \cdot b - c$.

The effect of positive and negative affect is to modify the benefit the agent expects to receive from attempting the action. When both A_P and A_N are neutral (i.e., $A_P = A_N =$ 0), the decision is based solely on a comparison of the benefit and the cost. However, given a history of actions, the agent may view the benefit more optimistically (if $A_P > A_N$) or pessimistically (if $A_P < A_N$), potentially leading it to make decisions that differ from the purely rational algorithm.

[3] A_P and A_N are squared to amplify the difference between the two, which amplifies the effect of the dominant state on the agent's decision process. Also, if p is available, then $a = (p + A_P^2 - A_N^2)$.

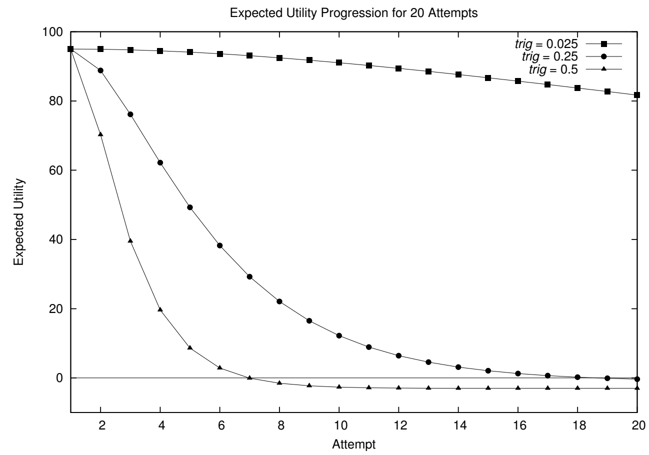

Figure 2: The expected utilities calculated at each attempt by the AGM for various values of $trig$.

3.2 Repeated Attempts

Figure 2 depicts for the robot example the effect of various triggers: one that is too optimistic, continuing into the foreseeable future; one that is too pessimistic, stopping fairly early; and one that is more reasonable, stopping at about the point where the costs will outweigh the benefits. This suggests that the value of $trig$ could be defined as a function of b and c to improve the likelihood that A_N will rise quickly enough to end the series of attempts before costs exceed benefits. The agent could employ negative reinforcement learning to determine the value of $trig$ for individual actions.

While the activation of each affective state is subject to decay, the rate of decay is slow enough that they can serve as *affective memory*, carrying the subjective estimates of the likelihood of success and failure ahead for a period after the events that modified the states. Returning again to the robot example, after a series of failures leading to the agent deciding not to attempt to ask directions again, the activation of A_N begins to decay. If, after some period of time, the agent is again faced with the choice of whether to ask for directions, any remaining activation of A_N will reduce the likelihood that it will choose to do so. In this way, the agent "remembers" that it has failed recently, and pessimistically "believes" that its chances of failing again are relatively high (e.g., because it has likely not left the noisy room it was in). Figure 3 shows the expected utility of asking for directions calculated by an agent 25 cycles after a series of failed attempts (e.g., Figure 2). For sufficiently high values of $trig$, the calculated utilities drop off faster than before, and the agent decides sooner that it is not worth continuing to ask. When $trig$ is very low, however, the expected utility actually *rises* as the number of failures (and, hence, the value of A_N) increases. This is due to an asym-

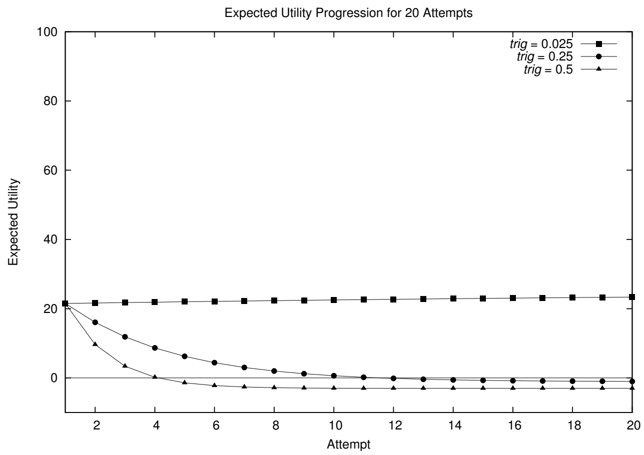

Figure 3: The expected utilities calculated at each attempt by the AGM for various values of $trig$, *after* an extended series of failures and 25 cycles of decay.

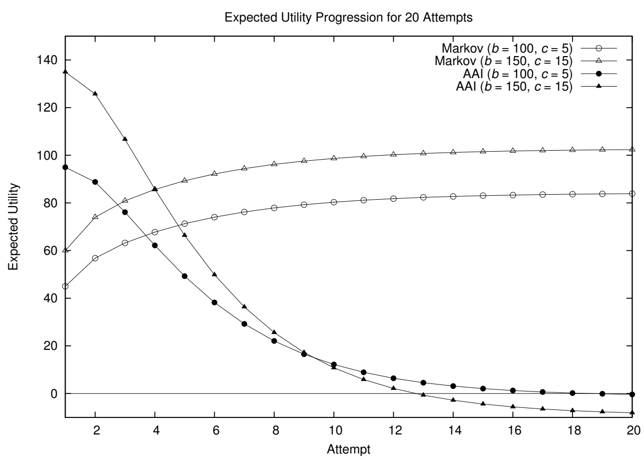

Figure 4: The expected utilities calculated for attempt by the AGM and the POMDP for two alternatives.

metry in the application of the trigger and the decay. Because A_N must be constrained by its upper bound (1), the magnitude of the trigger increment ($(1 - A_N) \cdot trig$) decreases and the magnitude of the decay decrement ($A_N \cdot dec$) increases as A_N increases; with sufficiently high A_N, even though $trig$ is 5 times dec in Figure 3, the decay dominates. The net effect is a reduction in negative affect, hence an increase in the estimate of success and a rise in expected utility. This highlights the importance of selecting an appropriate value for $trig$.

3.3 Two Alternatives

To this point, the examples and discussion have been limited mostly to choices between attempting some action or doing nothing. However, these methods can also be used to choose between alternative actions: the agent simply calculates the expected utility of each alternative and selects the one with the highest u. Figure 4 compares the behavior of the POMDP and the AGM on a selection between two alternatives. The first alternative is the one used above, in which $b = 100$ and $c = 5$. In the second alternative, $b = 150$ and $c = 15$. For the POMDP, the probabilities for both alternatives are also the same as above (i.e., the prior probability is 0.5, etc.), and the AGM has $trig = 0.25$.

The AGM lines in Figure 4 represent the expected utilities computed at each iteration, which would then be used to decide upon an action. The Markov lines represent the expected utility computed before the first iteration by the POMDP for $n = 1$ to 20. Regardless of how many iterations the POMDP considers, the second alternative will be the one chosen. Also, the Markov process has no way of determining value of n it should use to compute u. Of course, one could relax the constraints placed on the POMDP method described here, and potentially improve the decision-making process. For example, rather than requiring the process to choose one alternative at the start and stick with it, an optimal decision making process would examine every possible combination of the two alternatives to come up with the sequence of attempts that yields the greatest expected utility overall. Evaluating each possible sequence, however, requires exponential effort, which is often practically impossible.

Another alternative that approximates the optimal strategy, but at a substantially lower cost, is to allow the decision maker to decide after each turn. The AGM also will initially select the second alternative. However, by the tenth iteration the AGM will change to the first alternative (if it is still available). By that point, the second alternative would only have gone on for another three iterations before yielding a negative expected utility, whereas the first alternative has nine iterations remaining. This is a substantial advantage of not committing early: the online, single-attempt decision model makes informed, non-arbitrary assessments of when it is best to stop trying an action, allowing it to change course midstream when a better alternative exists. The AGM's low-cost probability approximation is an efficient mechanism for evaluating expected utility even in the absence of conditional probabilities.

4 Conclusion

Making beneficial decisions in the face of limited resources and constraints on time and knowledge can be very challenging. Classical decision-theoretic methods of action and goal selection can make good decisions, however, there may be cases in which the computational overhead of calculating the expected utility of attempting to satisfy a goal is too high. Moreover, although it was shown that the basic expected utility calculation (always) and the POMDP (normally) both quickly settle on a single choice (suggesting that it is unnecessary to make the calculation for long sequences), they will not perform well in the absence of accurate probability estimates. The affective goal manager introduced here is a low-cost mechanism for decision-making that does not require information about the prior or conditional probabilities of success. The agent's affective states reflect its recent history of success and failure, and can, therefore, influence subjective estimates of the likely outcomes of alternative actions.

When deciding whether to attempt a single action, the AGM is able to take its history of failures into account, allowing it to terminate detrimental series of attempts. Similarly, when choosing between two alternatives, the online AGM can recognize after a sequence of failed attempts of the first alternative that it would be beneficial to change to the second alternative. Given its low cost, flexibility, and capability, the AGM has been shown to be an effective decision mechanism under time, computation, and knowledge constraints. A robotic architecture for human-robot interaction that implements the AGM is under development in our lab, and has been used successfully in experiments with human subjects [Scheutz *et al.*, 2006].

References

[Blaney, 1986] P. H. Blaney. Affect and memory: A review. *Psychological Bulletin*, 99(2):229–246, 1986.

[Bless *et al.*, 1996] H. Bless, N. Schwarz, and R. Wieland. Mood and the impact of category membership and individuating information. *European Journal of Social Psychology*, 26:935–959, 1996.

[Clore *et al.*, 2001] G.L. Clore, K. Gasper, and H. Conway. Affect as information. In J.P. Forgas, editor, *Handbook of affect and social cognition*, pages 121–144. Erlbaum, Mahwah, NJ, 2001.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

[Kahneman *et al.*, 1997] D. Kahneman, P.P. Wakker, and R. Sarin. Back to Bentham? Explorations of experienced utility. *Quarterly Journal of Economics*, 112:375–405, 1997.

[Scheutz *et al.*, 2006] Matthias Scheutz, Paul Schermerhorn, James Kramer, and Christopher Middendorff. The utility of affect expression in natural language interactions in joint human-robot tasks. In *Proceedings of the 1st ACM International Conference on Human-Robot Interaction*, pages 226–233, 2006.

[Scheutz, 2001] Matthias Scheutz. The evolution of simple affective states in multi-agent environments. In Dolores Cañamero, editor, *Proceedings of AAAI Fall Symposium*, pages 123–128, Falmouth, MA, 2001. AAAI Press.

[Sloman *et al.*, 2005] Aaron Sloman, R. Chrisley, and Matthias Scheutz. The architectural basis of affective states and processes. In J.M. Fellous and M.A. Arbib, editors, *Who needs emotions? The Brain Meets the Machine*. Oxford University Press, New York, 2005.