Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Using Emotions to Enhance Decision-Making

Dimitrios Antos[1] and **Avi Pfeffer**[2]
[1]Harvard University, [2]Charles River Analytics
Cambridge, MA
[1]antos@fas.harvard.edu, [2]apfeffer@cra.com

Abstract

We present a novel methodology for decision-making by computer agents that leverages a computational concept of emotions. It is believed that emotions help living organisms perform well in complex environments. Can we use them to improve the decision-making performance of computer agents? We explore this possibility by formulating emotions as mathematical operators that serve to update the relative priorities of the agent's goals. The agent uses rudimentary domain knowledge to monitor the expectation that its goals are going to be accomplished in the future, and reacts to changes in this expectation by "experiencing emotions." The end result is a projection of the agent's long-run utility function, which might be too complex to optimize or even represent, to a time-varying valuation function that is being myopically maximized by selecting appropriate actions. Our methodology provides a systematic way to incorporate emotion into a decision-theoretic framework, and also provides a principled, domain-independent methodology for generating heuristics in novel situations. We test our agents in simulation in two domains: restless bandits and a simple foraging environment. Our results indicate that emotion-based agents outperform other reasonable heuristics for such difficult domains, and closely approach computationally expensive near-optimal solutions, whenever these are computable, yet requiring only a fraction of the cost.

1 Introduction

Decision-making in real-world environments is challenging for a number of reasons. First, the real world is complex. An agent should keep track of its changing state, which in most cases is not fully observable—plus, the number of such states in domains of even moderate size is already prohibitively large. Second, the agent should reason about the effects of its actions in every state, both in the short term and in the long run. Such effects could be known, but in real-world systems they almost always have to be learned, either from data or from experience. Unfortunately, past data might not be available or might be sparse, and the agent might not afford to spend time in the environment accumulating experiences. Third, if the environment contains other agents, their strategies and expected behavior need to be considered. However, computing game-theoretic solutions is known to be a hard problem, and in most cases requires making somewhat unreasonable assumptions about the agents' computational and epistemic capacity. Fourth, if the environment is changing stochastically in a complex way, then even excessive learning or computing precise solutions might not provide the agent with robust significant benefits.

Because of these difficulties, researchers often use heuristics to design agents for complex real-world environments. At the expense of optimality, good heuristics may perform well in most cases, evading computational complexity barriers. Living organisms have also been argued [Gigerenzer *et al.*, 2008] to employ a heuristic toolbox that allows them to perform well in tasks which are common in their environment, such as escaping enemies and locating food or potential mates. Yet despite using such heuristics broadly and having limited computational capacity, humans and animals perform admirably well in very complex environments, even without accumulating vast amounts of data, actively (or consciously) representing an exponential number of states, or computing Nash equilibria [Gintis, 2009].

The *emotions* are a psychological mechanism that influences human and animal decision-making in this heuristic way. Emotion has been argued to perform a variety of functions: (*a*) It directs cognitive resources to significant events or aspects of a situation (e.g., fear, upon the animal locating a threat, shifts focus toward escaping or neutralizing it). (*b*) It aids in learning, by storing and easily accessing emotionally intense or similar experiences [Christianson, 1992]. (*c*) It interprets observations in a way that is consistent with the organism's internal state (e.g., while running away from danger, fear will cause many items in the environment to be ignored, retaining perception of only those aiding in defense, evasion or escape). (*d*) It aids in communicating the person's mood and intentions by means of facial expressions, tone of voice or body stance.

Yet, despite the aforementioned benefits of emotion, it has been largely ignored in AI as a concept relevant to decision-making, with very few exceptions (e.g., [Scheutz and Schermerhorn, 2009], [Lisetti and Gmytrasiewicz, 2002], [Minsky, 2007]). Moreover, previous attempts to incorporate emotion in decision-making specify general principles but avoid concrete implementations, or these implementations are highly domain-specific. This paper provides a systematic way to bring emotion into AI decision-making, and demonstrates that measurable benefits may accrue. In particular, we focus on the first of the above functions of emotion, namely, its ability to shift cognitive resources and re-prioritize a person's goals by reacting to significant events in the environment. More specifically, agents in our system have goals with different priority levels. At every point in time, they take actions aiming to accomplish higher-priority goals. Emotions are used to inspire the design of operators that are activated by the agents interpreting their observations with respect to their goals (e.g., fear is elicited when a goal seems to come under threat). These operators' function is to dynamically alter the priorities of goals. For instance, a goal under threat receives higher priority due to the emergence of fear and, as a result, actions are taken to protect it. The net effect of emotions is to "project" the agent's long-run utility function—which might be too complex to directly optimize—down to a valuation function which varies across time, and in which goals are weighted by their current priorities. By myopically maximizing this valuation function in the next time step, the agent essentially ends up performing very well with respect to its actual long-run utility. We test this methodology in two domains: restless bandits, which are provably intractable ([Guha *et al.*, 2010], [Papadimitriou and Tsitsiklis, 1999]), and an artificial foraging environment. We compare our emotion-based methodology against other heuristics and optimal or nearly-optimal solutions, whenever these are computable.

Our paper therefore makes two contributions: First, it formulates emotion-inspired computational operators that adjust the priorities (or weights) of an agent's conflicting goals. In doing so, it provides a principled way of thinking about emotions in a decision-theoretic framework. Second, our work provides a concise methodology for designing heuristics in generic domains. The emotion-based operators we provide are not domain-specific and can be transferred across domains without losing their relevance. Thus, the agent designer may avoid having to design novel heuristics for every new domain.

Related work

Emotion has been widely investigated in the field of psychology, but also in those of economics and computer science. In psychology, cognitive appraisal theories ([Lazarus, 1991], [Oatley and Johnson-Laird, 1987]) have provided the basis for a large number of computational models of emotion (e.g., [Marsella and Gratch, 2009], [Ortony *et al.*, 1988]). Evolutionary psychologists have further demonstrated the connection between emotions and decision-making algorithms, by treating emotions as cognitive operators optimized by natural selection [Cosmides and Tooby, 2008]. Emotions have also been shown to aid interactions between agents and humans and foster relationships [Bickmore and Picard, 2005], as well as influence people's decision-making in a variety of contexts, like negotiation [Van Kleef and De Dreu, 2010], the prisoner's dilemma [De Melo *et al.*, 2009], or educational settings [Conati and Maclaren, 2009].

In relation to decision-making, [Scheutz, 2002] argues in favor of emotions as part of the design of agents and formulates emotions as "clustering concepts," whose function is to group similar situations together (e.g., a flight response might be appropriate in all situations eliciting fear). [Lisetti and Gmytrasiewicz, 2002] also make the case that emotion and rationality are not mutually exclusive and underline the social role of the emotions. In more recent work, [Scheutz and Schermerhorn, 2009] implement a simple emotion-based methodology, in which "positive" and "negative" affect is represented using two variables. These modulate the agent's expectations regarding the predicted reward of its actions (for instance, an agent with a lot of "positive affect" becomes more optimistic). Their model is similar to ours, in that affect is used to modulate weights inside the agent's architecture. However, our model differs in three important ways: First, we formulate actual emotions, such as fear and boredom, instead of undifferentiated positive or negative moods. Second, our emotions are defined in a domain-independent manner, which allows them to be transferrable across domains. And third, in their model emotions act as perceptual distortions, making certain actions seem more or less beneficial than they are—in contrast, in our model they merely update the priorities of goals, without presenting false information about the efficacy of actions.

2 Emotion-based decision-making

Below we describe how our emotion-based agents reason about the world and make decisions. In principle, the agents are using a decision-theoretic framework, selecting actions that maximize a certain valuation function. We discuss how this valuation function is formulated, how it changes across time based on the agents' emotions, and how these emotions are elicited.

Goals and valuation

Any agent is designed to perform certain functions in a particular environment. For instance, a robot exploring the surface of Mars has to locate interesting-looking rocks, collect samples, avoid falling, and return to base. These functions of an agent can be thought of as its *goals* (G). In our system, each goal $g_i \in G$ is associated with three numbers: First, it has a value $v_i \in \mathbb{R}^+$, denoting its relative importance. Second, it has a priority $r_i^t \in \mathbb{N}^+$ in every time step $t \geq 0$. Goals of higher priority are considered more significant in the current time step, whereas goals of lesser priority are less significant. Notice here that the value v_i of a goal does not change over time, but its priority might differ between time steps. Third, each goal has a degree of achievement $d_i^t \in [0, 1]$. This is a measure of the agent's subjective expectation that the goal will be achieved at some point in the future.

These three figures (value, priority and degree of achievement) combine to give a valuation function $\hat{u}^t = \sum_{g_i \in G} d_i^t \cdot q(v_i, r_i^t)$. Before we explain the function q, notice that the valuation at time t increases with the degree of achievement of the agent's goals. In other words, if the agent is taking actions to maximize its valuation function, it essentially is acting to increase the degree of achievement of its goals. This quantity is, however, weighted by each goal's value and priority, which is what the function $q(\cdot,\cdot)$ is doing. In our experiments we capped $1 \leq r_i^t \leq r_{max}$ and used $q(v_i, r_i^t) = v_i \cdot (r_i^t / r_{max})$. Since goals are weighted by their priorities, by sufficiently increasing the priority of a goal the agent will be geared towards taking actions to accomplish it, rather than lesser-priority goals. Finally, notice that changing $\mathbf{r}^t = (r_1^t, \ldots, r_{|G|}^t)$ provides a way to change an agent's valuation function across time. This change induces adaptive behavior, by essentially switching the agent's focus between its various goals. Notationally, we shall use $\hat{u}(\mathbf{r}^t, \mathbf{d}^t)$ to denote function $\hat{u}$ parametrized by priority and degree of achievement vectors $\mathbf{r}^t$ and $\mathbf{d}^t$, respectively.

Notice that we use the notation $\hat{u}^t$ for the agent's valuation function. The use of the hat ($\hat{}$) operator serves as a reminder that this is not the agent's *actual* utility. This other function (u) might be too complex to even represent, as it needs to incorporate every action taken and every event having occurred across the agent's lifetime, interactions between events, costs and probabilistic dependencies, etc. It might therefore be too complex to even meaningfully write down. In contrast, $\hat{u}^t$ comes merely from the functional specifications (desired behavior) of the agent at every point in time.

Domain Knowledge, Action Selection, Observations

We next turn to how the degrees of achievement d_i^t get assessed and updated. To answer this, we need to enrich our agent with the ability to predict the effect of its actions (A). Hence, we introduce the function $f(\mathbf{d}^t, a)$. This function receives as arguments the degree of achievement vector $\mathbf{d}^t = (d_1^t, \ldots, d_{|G|}^t)$ and a candidate action $a \in A$, and returns the expected degree of achievement vector *after* the action has been taken. To accomplish this, it must incorporate domain knowledge, such as, for instance, the fact that running in the opposite direction of an enemy increases the degree of achievement of goal "avoid enemy," but decreases the degree of goal "preserve charged battery." Given this function f, we have a way of selecting an action in every time step: the agent simply chooses action

$$a_t^* \in \text{argmax}_{a \in A} \hat{u}(\mathbf{r}^t, f(\mathbf{d}^t, a))$$

In other words, the agent selects the action with the highest expected increase in the degree of achievement of its goals, each weighted by its current priority. Notice here that, in maximizing its valuation function, the agent is treating goal priorities as constants, i.e., it does not reason about how these might change in the future because of its actions.

The agent is, however, also able to *observe* things in the environment. Things like the presence of a threat or a reward ought to change its beliefs regarding the degrees of achievement of its goals. To incorporate this functionality, we add the observation function $b(\mathbf{d}^t, e)$, where e is an observation obtained. The function returns—much like f—an updated degree of achievement vector. For instance, if an enemy approaches, the function b will return a degree vector in which the d_i of goal "avoid enemy" is decreased. At every time step an observation e^t is received and

$$\mathbf{d}^{t+1} = b(\mathbf{d}^t, e^t)$$

Notice here that domain-specific knowledge in our system is contained only within the values of goals v_i and within the functions f and b. Furthermore, these are directly derived from the functional specifications of the agent design and are unavoidable in every agent architecture. For instance, a logical or game-theoretic decision-making algorithm would still need to somehow define a goal or utility, and reason about the predicted effects of actions and the interpretation of observations. Also, as explained below, the emotions in our architecture are formulated in an entirely domain-independent manner.

Emotion-inspired operators

We are now ready to define emotion-inspired operators, whose function is to update the priority vector $\mathbf{r}^t$. Emotions are of two types: *Goal-specific* emotions are elicited in connection to, and update the priority of a particular goal. For example, fear is elicited when a particular goal is threatened. *Goal-independent* emotions are elicited without a particular goal in mind, and may change the entire priority vector. For instance, the emotion of boredom is elicited when nothing out of the ordinary happens for a while, but is not associated with any particular goal.

Each emotion operator $m \in M$ is associated with two functions: on the one hand, the *activation* function λ_m contains the elicitation condition for the emotion and returns TRUE if that condition is met; on the other hand, the *consequence* function κ_m describes how the priority vector is to be changed when the emotion is present. Below we present the functional form for some emotions:

- Hope and fear are reactions to changes in the expectation of the future accomplishment of a goal, positive or negative (hence, hope and fear are goal-specific emotions). We define λ_m for fear in this case to be TRUE iff $d_i^{t-1} - d_i^t \geq \theta_1$, where θ_1 is an externally set threshold. A similar condition and threshold can be defined for hope. In essence, fear and hope are elicited when there is a significant change in the expectation that a particular goal g_i will be accomplished. For instance, if an enemy shows up, function b will cause a decrease in the d_i for goal "avoid enemy" and this will elicit fear (if larger than θ_1). We also define κ_m to increase the r_i^t of the threatened (hoped-for) goal by a constant c. This increase, if sufficient, will direct the agent's efforts into protective actions, such as running away.
- Boredom, a goal-independent emotion, is elicited when the $\hat{u}$ experienced by a number of rounds has not changed significantly. In particular, λ_m is TRUE iff the standard deviation of payoffs $\{\hat{u}^{t-\tau}, \ldots, \hat{u}^{t-1}\}$ does not exceed a certain threshold θ_2 (we set τ, the length of history considered for the elicitation of boredom, to 10). When activated, the emotion of boredom perturbs the priority vector $\mathbf{r}^t$ at random. The net effect of this is an avoidance of "local maxima," i.e., actions that lead to historically good payoffs, but which might prevent the agent from exploring even better alternatives.

- Anger is an emotion that gets elicited when blameworthiness can be attributed to another party for some negative-impact event [Ortony *et al.*, 1988]. As such, its elicitation λ_m must consider whether an observed change in the environment has been caused by some other agents, and whether they had an alternative course of action or they specifically intended to cause it, as opposed to it being a side-effect of their plans (to establish blameworthiness). Anger will result in the raising of priorities among goals geared toward harming the supposed perpetrators or negating the effect of their actions.
- Sadness is an emotion that is elicited from repeated or significant failure. In humans it elicits withdrawal from activity and rigorous thinking, aimed at re-planning one's course of action. Consistent with this function of sadness, our λ_m was set to be TRUE when a series of payoffs $\{\hat{u}^{t-\tau}, \ldots, \hat{u}^{t-1}\}$ all lie below a certain fraction δ of the historically average or expected reward. The result of sadness is to suppress the priority of higher-priority goals, and increase that of low-priority goals, essentially "switching focus" to a potentially more promising set of actions.

Notice that the above emotion operators are defined in a general-purpose and not domain-specific manner. Between domains, the definition of the emotions does not change, although their elicitation thresholds could be adjusted accordingly. (As we show in the next section, however, performance is very robust with respect to the threshold values chosen.) In our simulations, we implemented the emotion operators of hope, fear, as well as the emotion of boredom, to keep the model simple. We also used the same threshold for hope and fear in each case.

3 Experimental evidence

To evaluate our methodology we selected two domains that are characterized by high uncertainty and complex stochastic change. Those two features are very common in real-world environments, for which ample data or full observability might not be available. Furthermore, uncertainty and stochasticity make the computation of optimal strategies difficult and also might impede learning, thus accentuating the need for heuristic approaches. Finally, the reason we chose two domains (instead of one) was to illustrate that our approach is not fine-tuned to the specifics of a particular environment, but might have potentially broader applicability to a variety of domains.

3.1 Restless bandits

Restless bandits (RB) are an extension of stochastic multi-armed bandits (MAB). In a typical setting, the bandit consists of a number (k) of arms, each of whom delivers a reward when selected. In the simple MAB case, the reward r_i of each arm i is drawn from a probability distribution that depends on the arm's state (s_i). In each time step, the agent may choose only one arm and obtain the reward from it; after this, the arm that was selected transitions stochastically to another state according to a transition matrix $\mathbf{T}_i$. The restless bandit case extends the above framework by allowing all arms to undergo a transition in each round, even those not selected. In particular, each arm i transitions to a different state in each round according to matrix $\mathbf{T}_i$ (if selected) or matrix $\tilde{\mathbf{T}}_i$ (if not selected). The goal in both cases is to maximize average reward. But whereas MABs admit an optimal solution, termed the "Gittins index" [Gittins, 1989], restless bandits have been shown to be a hard problem. According to [Papadimitriou and Tsitsiklis, 1999] even with deterministic transitions the problem is intractable in the general case. Recently, work has been done to be able to compute solutions for subclasses of the problem (e.g., [Slivkins and Upfal, 2008]), most of which follow the "Whittle index" (for a good review and an approximation algorithm see [Guha *et al.*, 2010]). Such solutions, however, suffer from assuming that too much is known: payoff distributions and transitions matrices, as well as the initial state of each arm, are usually considered known, and the problem is cast as being able to reap high rewards despite the state of all arms changing over time. However, this does not directly apply in situations where neither the stochasticity in payoffs nor in transitions is known, as in out setting.

In our simulations there are $k = 5$ arms, each of which can be in one of three states: "good," "medium" and "bad." In each state s of arm i, payoffs are given by a Gaussian with mean μ_i^s and standard deviation σ_i^s. Naturally, good states have higher means than medium ones, which have higher means than bad ones. An agent is allowed to make choices for a number R of steps (the value of which we varied among 30, 100, 500 and 1500) and the average payoff (as well as its variance) were recorded. For every agent tested, the experiment was repeated 100 times.

Our emotion-based agent was given five goals, one for each arm, of the form g_i = "obtain high reward from arm i." All goals had the same value $v_i = v$. The agent did *not* track the states of the various arms across time. It merely assumed that the state of every arm remained unchanged since the last time it was selected. When the agent selected an arm, it compared the payoff received with historical payoffs from the same arm, and made a crude estimate whether its state was good, medium or bad. Given its assessment of the arms' states, action "select arm i" was expected (in function f) to increase the degree of achievement of goal g_i if arm i was believed to be in the good state, decrease it if i was believed to be in the bad state, and leave it unchanged otherwise. This is basic domain knowledge, merely stating that selecting good-state arms is better than selecting bad-state ones. After a reward was obtained, the degree of achievement of the corresponding goal would be adjusted accordingly (function b), increasing upon high rewards and decreasing upon low rewards ($d_i^{t+1} \leftarrow$ payoff received at t / max payoff). The agent employed the three aforementioned emotion operators of hope, fear and boredom to update the priorities of its goals, which were all initialized to 5 and were allowed to range between 1 and $r_{max} = 10$. We tried different values for the emotion thresholds ($\theta_1 \in \{0, 0.1, 0.3\}$ and $\theta_2 \in \{0, 1\}$). Hence a total of six variants of the emotion-based agent were tested, in order to examine the robustness of our emotion-based methodology with respect to the choice of thresholds. The approach's performance was taken to be the average among the six variants examined in order to prevent any observed performance benefits being obtained by a fine-tuning of the threshold values.

We compared this to the following agents: (i) a random agent, which selected arms with uniform probability; (ii) a reactive agent, which selected an arm until a significant decrease (greater than 30%) in payoffs was observed, then switched to a different arm at random; (iii) a learning agent, which made assessments (based on past data) about the mean and variance of each state, but did not track the change in states across time; (iv) an "all-seeing" agent, who would (somehow) know the payoff means of all the states, as well as the current state of each arm; (v) a "half-seeing" agent, who would know the payoffs means of all the states but would not know their exact current state; and (vi) a collection of indexing agents. These indexing agents are guaranteed according to [Guha *et al.*, 2010] to contain the optimal policy, and work as described below.

For every indexing agent, an integer value t_i (the index) is associated with each arm i. The agent then acts as follows: Initially, it selects a few arms at random to assess the mean of good and bad states across the five arms. Then, it chooses its next action by looking at the reward obtained by its last choice. If the arm selected in the last round is evaluated as being in the good state, the agent selects it again; otherwise, it chooses at random from the subset of arms i that have not been selected for at least t_i time steps, where t_i is the index for that arm. In other words, the index of an arm denotes how long an agent must wait, after the arm is detected in the bad state, until it starts considering it again.

We simulated 5^{15} such index policies ($t_i \in [1, 15], \forall i$).[1] Results from our simulation are presented in Table 1. (All differences are significant at the 1% level according to pairwise non-parametric Wilcoxon tests.) Naturally, policies (iv)–(vi) are not candidates for direct comparison, because they "cheat" by having access to knowledge the remaining agents do not; nonetheless, they are good indicators for the performance of our agents.

Agent	$R = 30$	100	500	1500
Average Emotion	3.64	3.26	3.07	3.17
i. Random	1.91	3.03	1.94	2.37
ii. Reactive	1.29	2.98	2.83	2.45
iii. Learner	3.15	2.93	2.32	2.22
iv. All-seeing	8.75	8.73	8.83	8.72
v. Half-seeing	3.99	3.70	3.62	2.92
vi-i. Best-index	5.64	4.46	4.13	4.16
vi-ii. Avg-index	2.44	2.47	2.46	2.43

Table 1: Average payoff of agents in restless bandit domain

As can be seen in the table, the emotion-based agent consistently outperforms the random, learning, and reactive heuristics, as well as the average index policy, and comes close to the optimal index policy (which, is, however, not computable without a priori knowledge of the parameters of the bandit arms). As a matter of fact, only about 50 of the 5^{15} index policies outperformed the emotion-based agent for every choice of R.

Moreover, the performance of the emotion-based agent seems robust with respect to the time R allotted for learning and experimentation. Next, we tested the robustness of our agents with respect to the choice of thresholds θ_1 and θ_2. Across all choices of R, the variance in the payoff obtained among the six variants of the emotion-based agents (with different choices of θ_1 and θ_2) never exceeded 0.3. Table 2 shows the performance of six variants for $R = 1500$. As can be seen, all but one variant (6) still outperform the other heuristics (last column of Table 1).

Variant	1	2	3	4	5	6
Avg. payoff	3.51	3.79	3.01	3.76	3.23	2.36

Table 2: Average payoff of emotion variants for $R = 1500$

3.2 Foraging

We next evaluated our methodology in a foraging domain consisting of a 10×10 square field. Random units of food are placed on squares of the board, such that their total number does not exceed 15. A single agent starts in location $(0, 0)$ and is allowed, in every time step, to either (a) walk in one of the four directions, spending 5 HP, (b) run, covering twice as much space, but spending 20 HP, (c) pick up food, if it stands on it, or (d) eat food, if it carries some. An agent may carry up to one unit of food at each time, and picking up a unit of food causes one more to randomly appear somewhere, such that the total remains 15. The field is also populated by a number of enemies (8) and a number of friends (8), which are indistinguishable to the agent. However, standing on the same square as an enemy in the end of a round causes the agent to suffer damage of -100, while a friend provides 100 units of benefit. Both friends and enemies will move in every round toward the agent, if it finds itself next to them. Eating food also gives the agent 200 HP. There are two seasons (winter and summer) in this environment, and the season changes stochastically with probability 0.02 after every round. The location of friends, enemies, and food is not uniform; they are all clustered in parts of the field based on the current season (e.g., in the winter there are more enemies in the top-left corner). Agents are assumed to perceive the area that is one step away from their current location, counting diagonal steps (i.e., the 3×3 square surrounding them). However, agents do not observe the season, do not know the transition probability for its change, and have no knowledge of the movement patterns of friends or enemies.

Our emotion-based agent had two goals: g_1 ="avoid danger" and g_2 ="replenish HP." The presence of food and the acts of picking it up and eating it were set to increase the degree of achievement of g_2, while the presence of another creature (friend or enemy) was initially expected to decrease the degree of g_1. However, after an interaction, and depending on its outcome (experiencing damage or benefit), the degree of achievement of g_1 would go down or up. Hope would

[1] This brute force approach was chosen due to the fact that the optimal index policy cannot be computed without having knowledge of the means of all states and the transition probabilities between them, which our agents did not know.

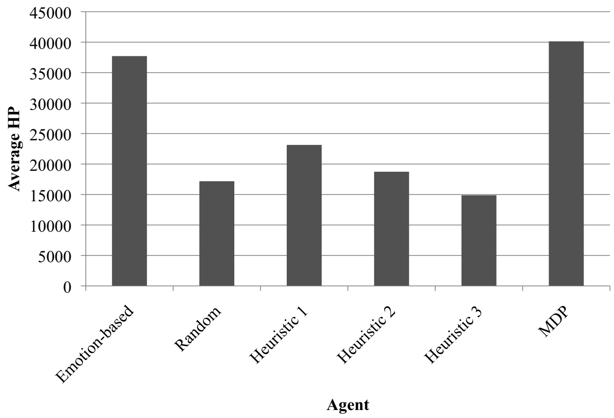

Figure 1: Average 500-round HP in foraging environment

ensue after a positive interaction and decrease the priority of g_1, while fear, caused upon suffering damage, would increase it. The net effect of hope and fear was, therefore, a tendency to avoid creatures right after being damaged by one (since now g_1 had a high priority), and not avoid them after being benefited (since now g_1 had a low priority). This emotion-based agent was compared against a random policy, and three scripted heuristic policies of the form: "walk about at random, pick up and eat food when you see it, and move in the opposite direction of other creatures." The three heuristic policies differed in whether the agent would walk or run away from enemies, and whether it would consider interacting with a creature to see if it is a friend or not. The agents were allowed 500 moves on the field and then their HP was recorded; for all agents the experiment was repeated 100 times.

The optimal policy in this problem would be the solution to a very complex POMDP with over 8 million states. To get a better sense of the performance of our system, however, we abstracted features of the problem and reduced the number of states by assuming certain independencies. Thus, we only accounted for the location of nearby creatures (up, down, left, right, center, none), the location of food, whether food is being carried and whether the very last interaction was with a friend or enemy, for an MDP with a total number of 144 states. (The abstracted problem is an MDP since unobserved variables, like the location of unseen creatures, are assumed not to matter.) In Figure 1 we present the performance of the various agents. (Differences are significant at the 1% level.)

As can be seen the emotion-based agent outperforms the most obvious heuristics and comes very close to the MDP which was formulated on the reduced (abstracted) problem. Note that, although the emotion-based agent underperforms the MDP, the latter requires more computational resources, and also requires some knowledge about things that our emotion-based agent does not know. For example, the MDP needs to know the transition probabilities between states, which in turn requires making assumptions of the behavior of friends and enemies in the environment.

4 Discussion & Conclusion

We have presented a methodology for emotion-based decision-making in complex environments with high uncertainty and stochasticity, and demonstrated that benefits may accrue with very little computational cost. Furthermore, emotions in our methodology were formalized as domain-independent operators, making them applicable to virtually every situation and providing a way for the design of heuristics in novel environments. However, two issues need to be clarified. First, our use of the word "emotion" must be understood as detached from the biological and psychological complexities of emotions in living organisms. Although cognitive appraisal theories have taken an information-processing approach to analyzing the emotions, the biological substrate and the psychological influences of emotions are far too complex to account for. Moreover, we are merely inspired by certain functions of affect and wish to replicate them in computer agents to the extent that they confer a performance advantage or reduce computational cost. In that spirit, our goal is not to replicate human emotional reactions with high fidelity, but to adapt basic notions of emotion-based decision-making to the practical needs of AI agents. Second, it must be noted that our tests examined very complex domains, for which optimal solutions are either not tractable or very hard to compute. We feel this illustrates the value in further investigating the benefits from emotion-based decision-making. However, because of the absence of more sophisticated algorithms, we had to compare our agent against fairly simple heuristics. This was necessary as, to the best of our knowledge, no better methods exist that can be used without extensive periods of learning. Despite such heuristics being justified as competitors, however, our simulations say little about domains for which optimal solutions are currently within our reach. We plan to explore how this methodology lends itself to such domains and how it compares with currently optimal solutions. One of the avenues we are exploring is comparing it with "anytime" algorithms, and thus assessing the computational cost (in time and resources) for these algorithms to catch up with an emotion-based decision-maker which operates "out of the box." Quantifying in this way the limits of our current algorithms, we will have a better sense of when an emotion-based methodology is appropriate or not.

References

[Bickmore and Picard, 2005] Timothy W. Bickmore and Rosalind W. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Trans. Comput.-Hum. Interact.*, 12:293–327, June 2005.

[Christianson, 1992] Sven Ake Christianson. *The Handbook of Emotion and Memory: Research and Theory*. Psychology Press, 1992.

[Conati and Maclaren, 2009] Cristina Conati and Heather Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19:267–303, August 2009.

[Cosmides and Tooby, 2008] L. Cosmides and J. Tooby. *Handbook of Emotions*, chapter Evolutionary Psychology and the Emotions. NY: Guilford, 3rd edition, 2008.

[De Melo *et al.*, 2009] Celso M. De Melo, Liang Zheng, and Jonathan Gratch. Expression of moral emotions in cooperating agents. In *IVA*, pages 301–307, 2009.

[Gigerenzer *et al.*, 2008] Gerd Gigerenzer, Henry Brighton, and Max Planck. Homo heuristicus: Why biased minds make better inferences, 2008.

[Gintis, 2009] Herbert Gintis. *Game Theory Evolving*. Princeton University Press, 2nd edition, 2009.

[Gittins, 1989] John C. Gittins. Multi-armed bandit allocation indices. *Wiley-Intersc. Series in Systems and Optimization*, 1989.

[Guha *et al.*, 2010] Sudipto Guha, Kamesh Munagala, and Peng Shi. Approximation algorithms for restless bandit problems. *J. ACM*, 58:3:1–3:50, 2010.

[Lazarus, 1991] R. S. Lazarus. *Emotion and Adaptation*. Oxford University Press, 1991.

[Lisetti and Gmytrasiewicz, 2002] C. L. Lisetti and P. Gmytrasiewicz. Can a rational agent afford to be affectless? a formal approach. *Applied Artificial Intelligence*, (16):577–609, 2002.

[Marsella and Gratch, 2009] Stacy C. Marsella and Jonathan Gratch. Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70 – 90, 2009. Modeling the Cognitive Antecedents and Consequences of Emotion.

[Minsky, 2007] Marvin Minsky. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon and Schuster, 2007.

[Oatley and Johnson-Laird, 1987] K. Oatley and P. N. Johnson-Laird. Towards a cognitive theory of emotions. In *Cognition and Emotion*, volume 1, pages 29–50, 1987.

[Ortony *et al.*, 1988] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. New York: Cambridge University Press, 1988.

[Papadimitriou and Tsitsiklis, 1999] C.H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queuing network control. *Math. Oper. Res.*, 24(2):293–305, 1999.

[Scheutz and Schermerhorn, 2009] Matthias Scheutz and Paul Schermerhorn. *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*, chapter Affective Goal and Task Selection for Social Robots, pages 74–87. IGI Global, 2009.

[Scheutz, 2002] Matthias Scheutz. Agents with or without emotions? In *15th Intl. Florida AI Research Society*, pages 89–93, 2002.

[Slivkins and Upfal, 2008] R.S. Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits, 2008.

[Van Kleef and De Dreu, 2010] G. A. Van Kleef and C.K.D. De Dreu. Longer-term consequences of anger expression in negotiation: Retaliation or spillover? *Journal of Experimental Social Psychology*, 46(5):753 – 760, 2010.