Utilizing Emotions in Strategic Real-Time Artificial Intelligence

Megan Olsen MOLSEN@CS.UMASS.EDU
Kyle Harrington KYLE@KEPHALE.COM
Hava Siegelmann HAVA@CS.UMASS.EDU
University of Massachusetts, Department of Computer Science, 140 Governors Drive, Amherst, MA 01003

Abstract

Strategic real-time systems are of high potential and their applications are growing, although they are mostly prevalent in video games, military training, and military planning. We propose a paradigm to advance current systems by introducing emotions into the simulated agents that make decisions and solve situations cooperatively. By utilizing emotional reactions and communication, we hope to advance these systems so that the decision process better mimics human behavior. Since our system allows sharing of emotions with nearby agents it utilizes both internal and external emotional control.

1. Introduction

Real-Time Artificial Intelligence has been investigated for over a decade (Musliner et al., 1995). A system is considered to be a Real-time AI system if it is able to make decisions within a guaranteed response time and thus meet domain deadlines. These systems face many challenges, including working with partial information, choosing the most crucial action if there are multiple scenarios to react to, and working continuously for an extended period of time without failure. These systems are usually created as expert systems, as they are used for a specific domain. However, they should be able to handle a vast majority of scenarios that occur, not just the specified test scenarios. The results must be returned in a timely fashion (Musliner et al., 1995).

Real-Time Strategy (RTS) is an offshoot of general purpose real-time AI. RTS refers specifically to systems where the primary purpose is to create strategy, usually in a competitive atmosphere. For instance, military training on how to engage the enemy done via simulation is a RTS system. Only training that with a computer strategy aspect is included however, since it is not a RTS system if only the human controls strategy. Currently the military uses simulations heavily for training, and therefore it is crucial for them that these systems advance (Herz & Macedonia, 2002).

Also, many popular video games such as Starcraft and Warcraft incorporate Real-Time Strategy if at least one player is the computer. These games all simulate war among multiple players in which all but at least one player may be computer controlled. Although advances may be made in the AI of these systems, they do not seem to influence the military training development. However, many groups are working to combine the two groups so that meaningful work can be done to advance both fields at once (Herz & Macedonia, 2002; Buro, 2004). Ideally, the creation of war-related video games will be able to influence the military training simulations in years to come (Buro & Furtak, 2003).

Although they may at first seem unrelated, emotions can play a large part in strategy especially when time is limited. Emotions are believed to improve our response time, increase our memory capacity, and provide quick communication (Rolls, 2005). We are able to notice things that we fear quicker than things we enjoy or are indifferent about, showing fear to be crucial to our response time. Remembering an emotion may enable a memory to be more useful for us later, as we can react to the emotion of the experience without needing to remember all of its details. Emotions help us convey our experience to another person; for instance, they will realize danger quicker from noticing our fear than by hearing our explanation. Thus, we propose to include emotion with RTS algorithms to enhance our strategy.

Our system utilizes a current RTS gaming engine as well as its included AIs. We will provide emotions for the game's units (agents), and determine how those

Presented at *North East Student Colloquium on Artificial Intelligence (NESCAI)*, 2008.

emotions affect the game play. We anticipate that emotions will enhance the ability of the agents to react to their environment and influence other agents, thus increasing the performance of the AI. One of our main contributions is the creation of an Emotion Map that enables agents to communicate their emotions with any surrounding agents without direct contact. This Emotion Map saves the emotion of agents and diffuses it for a period of time, enabling neighboring agents to feel the emotion of their peers.

2. Related Work

Many exciting advances in computer science are systems that must function in real time. For example, a model of ship damage control has been created that relies on real time decision making. This model determines the best course of action given the state of the ship and its many control systems. Tested in a simulation environment against actual Navy captains, the model vastly outperformed the humans. This example shows that Real-Time AI can even be valuable in situations where humans are already available to perform the task (Bulitko & Wilkins, 2001).

One type of real time strategy system is the RTS game, which can tackle many different fundamental AI issues. For instance, game AI is closely related to adversarial real-time planning, decision making under uncertainty, opponent modeling, spatial and temporal reasoning, resource management, collaboration, and pathfinding (Buro & Furtak, 2004). One system that is working to improve gaming in all of these aspects is ORTS. This system is an open source game that is utilized in a competition each summer to encourage AI experts to test their skills and create software with a usable combination of solutions. Although we will use a similar system called "Globulation," our enhancements could also be applied to ORTS.

Another way to create an RTS game is by controlling characters in games such as Quake. Laird et al. creates bots that can strategize through first person shooter games to beat human players. They create their bots using real-time AI algorithms, giving them the ability to anticipate another player's action, make smart decisions on where to go, and make smart decisions on what actions to take. This type of strategy is different from the type of strategy we will investigate, as it is only a single entity moving in a world against other similar entities (Laird, 2001).

Although there are currently no RTS systems that incorporate emotions that we are aware of, other software systems do exist with them. For instance, the digital life simulation game, the Sims, includes emotions. These emotions control the behavior of in-game agents; an example being that an unhappy agent is less likely to obey the commands of the controlling player. Many other examples of emotions being used in computer systems relate to the field of human-computer interactions (HCI). A great amount of work has been done on improving a computer's ability to detect a user's emotions, and then using that information to change its interaction with the user. Much of this work is in the affective computing field (Picard, 1997; aff, 2007), and tends to relate to voice and facial recognition. A RTS system used for training can benefit from this work, but it is beyond our current scope.

Agents who must communicate indirectly often use collaborating software (not be confused with collaboration software) such as blackboards (Corkill, 2003). These blackboard systems allow agents to post information that other agents can later access and discuss as appropriate. The posted information tends to last for a long period of time, and be accessed when it becomes relevant for that agent. In our system we utilize an "emotion map" that allows a different type of indirect communication. With our map an agent shares its emotions at each time step to the areas immediately surrounding it. Any agents that are nearby will see this information and incorporate it into their own set of emotions. However, the information saved to the map degrades quickly until it disappears, thus preventing the emotions from lingering for more than a few time steps. We thus only allow indirect communication with agents within that locality at the time of the emotion, instead of sharing with any agent at any time. Although this concept seems simple initially, it is not only different from previous communication paradigms but can also lead to powerful and dynamic interactions.

3. Globulation

There are currently a few open source RTS platforms, including ORTS (http://www.cs.ualberta.ca/m̃buro/orts/) and Globulation (http://www.globulation2.org/wiki/Main_Page). Both of these systems run on similar premises, designed as strategy war games where the characters can be controlled by the AI. We chose to work with Globulation, a multi-player game where players compete for resources and territory. A player loses the game if all of their agents have been destroyed. A particularly novel aspect of the game is the lack of control over each individual agent, an approach taken by nearly every RTS. Instead, players can control

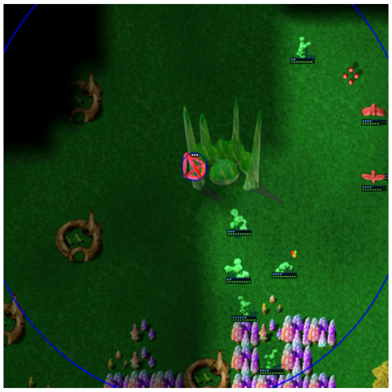


Figure 1. A portion of the Globulation map. The darkly shaded area is area that has not been explored, and therefore cannot be seen. The large item in the middle is a building, whereas the smaller items just south of it are workers. The cluster of items at the bottom are a resource that needs to be gathered.

agents by defining their behavior at each square on the map (e.g., forbidden, harvest, defend, etc.). This allows players to focus on more general strategies as opposed to testing their point and click skills.

Globulation has multiple Artificial Intelligences (AIs) that can be chosen as a player in the beginning of a game. The AI will thus control the actions of its assigned player so that no human intervention is needed. It will not only make overall player choices, but each agent is also given its own set of decision processes. There are many different AIs available for Globulation, each with a different focus, level of detail, and success rate. The AI we will test against is named "Nicowar" and has the highest success rate of the AIs.

To allow our work to concentrate on the emotion aspect, we define emotions as part of each agent's individual decisions such that they can be ported to any of the AIs that already exist for the game. Thus, the decision processes for agents will be a combination of a previously created AI and our emotional paradigm. When designing our emotions we examined the deficiencies of Nicowar. Although it is the most human-competitive AI in the game, its flaws include bottlenecks with pathfinding when dealing with a large numbers of agents, avoiding enemy agents (defensive agents), and finding enemy agents (offensive agents). We will seek to address all of these flaws with our emotional system.

3.1. Agents

Each player in the game has their own agents that can be controlled by an overarching AI. These agents include warriors, workers, and explorers. Each agent has a numerical amount of health that can decrease if it is injured or increased if it is healed. Each agent is also affected by a need to eat, and has a base desire to do work. All agents are capable of movement in 2-dimensional space within the map boundaries. They will make decisions on what actions to perform based on what they encounter as they move through the map, see Figure 1. Our emotional changes will affect each agent's own decisions, but will not affect the overall AI.

Each agent has its own purpose in the game, and the three types serve very different functions. The workers exist to gather resources, which are needed for the player to build buildings, create more agents, and feed the current agents. The warriors defend the player's buildings and attack the opponent's buildings and agents. The explorers will wander the map to determine the locations of enemies and resources. The warriors need the workers to gather resources for them, whereas the workers need the warriors to defend them. The workers and warriors both need the information the explorers discover to be able to see an oncoming attack, where to go to attack, and the location of additional resources for when current supplies run low.

4. Emotions

4.1. Types of Emotions Modeled

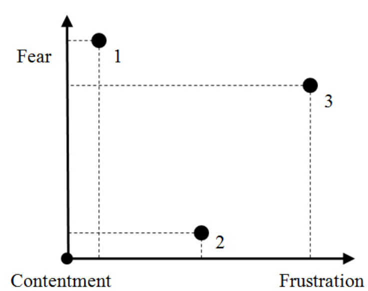


Figure 2. The plane representing the range of an agent's emotions and 4 possible emotion stages: the origin is no fear or frustration, representing contentment; point 1 represents an agent with little frustration but high fear; point 2 is an agent with low fear and medium frustration; and point 3 represents an agent with high frustration and high fear.

We chose to model two different basic emotions which will be the same in each agent type, although each agent type will be affected differently by its emotions. Although we did not choose all of the 6 basic emotions (happiness, fear, frustration, anger, sadness, relief) (Rolls, 2005), we chose two of them coupled with a more advanced emotion. The first basic emotion that we modeled was fear. Fear is increased when an agent is attacked by an enemy agent, an agent is very damaged and close to death, or the player is running low on resources. The first two causes are obvious, and the last cause is because agents will die if they do not have food, which is a resource. The second basic emotion modeled is frustration. Frustration is increased when an agent is unable to perform the task allotted to it or the agent has been on the same task for a significant amount of time (details in the Simulation Details section).

The lack of these two emotions also constitutes an emotion: contentment. For instance, if there is little or no fear the agent feels content as the world seems safe. Also, if the agent has little or no frustration then it is content because everything is working well. Although agents do not make decisions based on the combination of their 2 basic emotions, their emotional state at any time can be represented by a point on a plane with fear as the y-axis and frustration as the x-axis as seen in Figure 2. However, without the emotion map (explained below), emotions would be entirely internal and not shared.

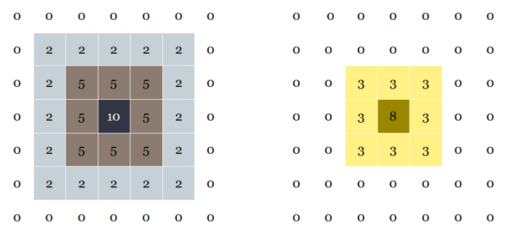
4.2. Effects on Agent Actions

Each emotion affects agents in ways related to five of Eckman's seven characteristics of emotion: Quick onset, automatic appraisal, commonalities in antecedent events, brief duration, and unbidden occurrence (Eckman, 1994). Emotions occur based on events in an agent's neighborhood immediately when that event occurs. The agent does not have time to decide that its surroundings are a problem, but instead there is quick onset due to automatic appraisal of the situation. For all agents of a particular type the same antecedent event types will cause the same amount of the same emotion. Also, emotions are brief unless the same event continues to occur, in which case the emotion will continue to build at a slow rate. Emotions are not consciously caused, as only outside events or the sharing of emotions from another agent can cause them. The two characteristics that we do not relate to do not apply to our situation (presence in other primates, distinctive physiology) (Eckman, 1994). Actions that are taken due to an emotion are decided upon only once the emotion reaches a specified threshold. Once that threshold is reached, the agent acts according to both its current situation and the fact that the particular emotion is strong.

An emotion's effect on each agent is homogeneous throughout that agent type and differs between agent types. The effects of emotion on our agents are based on the idea of fight versus flight. If a worker experiences enough fear, it will move in a direction toward less fear until its fear falls below the required amount for it to withdraw. If possible, it will continue to move in a direction that will decrease its fear. This effect minimizes the problem with the AI that causes agents to not escape their enemies. A warrior, however, will advance toward the cause of its fear (up to a specific threshold) in hopes of vanquishing the source. This reaction causes a warrior to move toward nearby enemy agents and attack, which is beneficial in both offense and defense. This effect will improve on the AI's problem of not moving toward enemy agents often enough. However, if a warrior's fear level crosses a higher threshold it will retreat, improving on it's ability to survive. The definition of "higher" was tested to determine an appropriate level, and is discussed in the Simulation Details section.

The agent reactions are similar for frustration. If a worker is feeling frustrated it will look elsewhere for work, which will usually involve looking for resources to gather. If the worker is already in a location with resources but still feels frustration, it is likely due to a large number of workers gathered who are causing a bottleneck for retrieving resources. If a warrior has frustration it will explore to look for enemies or will wander around acting as a lookout, as frustration is likely a result of no danger in its current location. Frustration directly combats the AI's problem of failed pathfinding. The AI already has a built-in check for "boredom" that verifies that a worker is not idle for a long period of time. Frustration, however, will solve the problem of workers being crowded together at a single entrance to an area with resources, or otherwise needing to move to continue succeeding toward their goals. Thus frustration can create a more efficient resource gathering mechanism for workers, and a higher likelihood of encountering enemies for warriors.

The baseline for each agent is to have no fear and no frustration (i.e. be content). Over time, any emotion felt will decrease until it reaches this baseline or a new experience replenishes the emotion. For example, if a worker is running from an enemy agent and is not chased, its fear level will continue to decrease with each time step until it no longer has fear. However, if the enemy chases it such that they continue to be the same distance apart the agent will maintain the same amount of fear. If the enemy is moving closer to the agent, the fear will increase.



(a) Immediate emotion diffusion

(b) Decay of emotion after 1 time step

Figure 3. Approximate diffusion concept. The map in 3(a) depicts the values in the squares under and surrounding a agent that just experienced an emotion of strength 10. The map in 3(b) depicts the values in those same squares after a single time step, assuming that no agent is present to modify the emotions of its surroundings with emotions having a decay value of 2.

4.3. Emotion Map

For emotions to be most effective there must be a mechanism for agents to infer each other's emotions. For humans, emotions are exceptionally useful as a way to communicate. An agent's emotion is therefore influenced by the emotions of other agents under the same player via an Emotion Map and gradient. Agent emotions cannot be interpreted or felt by an opponent's agents. At each time step, an agent's emotion will be saved to the map. Fear and frustration are kept separately on the emotion map. The emotion map affects an agent's emotions and is updated by an agent's emotions at every time step. This frequency is because emotions are vital to an agent's decision making, so it is therefore necessary that both the map and the emotions felt by an agent from the map are as accurate as possible. The agent's emotion will be added to the emotion on that square, and will be diffused out to the adjacent sets of squares within a specified cityblock distance. A distance of 2 is shown in Figure 3(a), assuming an emotion with value 10 was just experienced. Given a *max_radius* that defines the furthest distance an emotion can travel and a *value* of the emotion being felt, the amount of emotion that will be saved on the map at a location that is *distance* away from the original point of the emotion is

$$Map(distance) = \frac{value - \frac{value}{max\_radius}}{distance}. \qquad (1)$$

The emotion on the map will decrease over time in the same way that the individual agent's internal emotions will decrease over time. At each time step, the current emotion will decrease as shown in Figure 3(b), and then any new emotions will be added.

Each agent can see a gradient of the map, and is therefore affected by this gradient with each decision made. The agent's own emotions are affected by the map such that a small percentage of each of its emotions is derived from the emotions on the map from the end of the previous time step. The map therefore allows agents to communicate indirectly, since the emotions held on the map are due to another agent's recent emotions. Therefore, if an agent recently encountered an enemy in a particular location, all close by agents will be aware due to the emotions on the map. Also, any other agents that come to the area within a short time span will be aware as well. This use of the emotion map can be equated to people hearing each other yell in fear or anticipation of a fight, or grunting from boredom. It is also related to the fact that another individual that arrives slightly after the other agents may never know that something recently occurred there. An example of the emotion map being used can be seen in Figure 4.

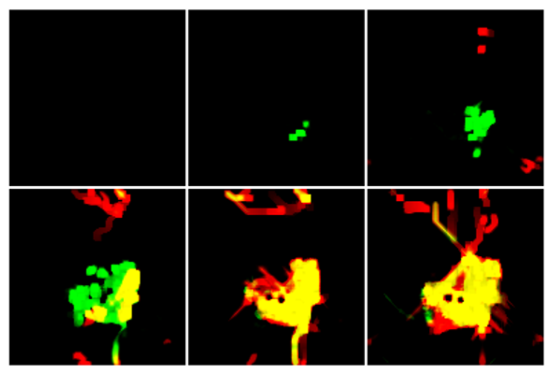


Figure 4. A series of images demonstrating an emotive agent's emotion map changing over time. Images are taken every 4,000 time steps. Frustration is shown in the middle shade of gray, fear is shown in the darker shade, and the lightest shade is the overlap of the two emotions. Images are organized chronologically from left-to-right and top-to-bottom. Initially frustration is experienced by workers in the home base of the agent. Eventually the opponent attacks the agent and the agent's home base is filled with both fear and frustration.

5. Simulation Details

5.1. Globulation Set-Up

Simulations were run with version 0.9.1 of Globulation 2. Evaluations were performed on the map Muka, which is a one player versus one player map. Each player has all necessary resources contained within a region that is connected to the opponent via two land bridges (at the top and bottom). The map wraps from right to left, creating a land bridge from the left side to the right side of each player's region. Both players also have an additional smaller peninsula containing resources. The map is essentially symmetric although it was created by hand due to limitations of the map editor.

5.2. Emotion Controls

All emotions exist on a scale of 0 to 100. Both emotions on the emotion map are initialized to 0 at every location. Emotions then decay linearly at a rate of 1 every time step. The constant diffusion radius for both fear and frustration is 5. An example emotion map changing over time can be seen in Figure 4. Both fear and frustration were discounted by a factor of 0.1, meaning that 10% of the previous emotion level is added to the current emotion level. Fear is affected by two factors: medical condition and surrounding enemy agents. Fear is increased by 1 every time step that the agent is damaged, and increased by 10 for every surrounding enemy agent. Frustration is increased in a particular agent by one tenth of the amount of time spent continuously performing the same task. This increase of frustration allows agents stuck in a location to free themselves by moving away from the frustration gradient.

Thresholds were required for specifying emotion controlled behaviors. Worker and warrior agents surpassing their frustration threshold will begin to move in the opposite direction from the frustration gradient. For both workers and warriors, the frustration threshold is 85. Worker agents with fear greater than 75 attempt to evade the source of the fear by moving in the opposite direction of the fear gradient whereas warrior agents with fear greater than 55 are drawn toward the source of fear, following the fear gradient. Once a warrior's fear increases above 90 it will retreat as well. These values were determined via tests on AI Nicowar with emotion.

5.3. Agent Decisions

Each agent has two sets of controls that are the same across all AIs: the built-in decision controls, and the emotion-based decision controls. The built-in decision controls check for life threatening situations and are executed first. Without the emotion-based decision controls, agents would choose an action randomly if there was no life threatening situation (high need for medical care or food). However, AIs that utilize emotions use the emotion-based decision controls if no other decision has been made by that agent. The built-in and emotion-based controls are mutually exclusive for each time tick, i.e. a decision is made by either one or the other for each agent.

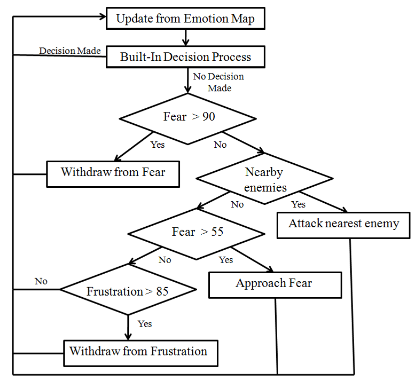


Figure 5. The decision tree for a warrior at each time step.

Each agent uses a specific decision tree when determining what action to perform each time step. The decision tree for the emotion-based controls on warriors can be seen in Figure 5. These decisions are primarily based on the thresholds discussed previously in the paper. All updates from the emotion map occur at the beginning of each time tick. Thus, the fear (F) of an agent at time t in location λ if it is surrounded by ϕ enemies is shown in Equation 2 if $Map(Fear, \lambda)$ refers to the amount of fear on the Emotion Map in location λ and ω is a binary number that is 1 if the agent is damaged and 0 otherwise.

$$F(t) = 0.8 \cdot F(t-1) + 10\phi + \omega + 0.1 \cdot Map(F, \lambda) \quad (2)$$

An agent's frustration (A) at time t can be similarly set as seen in Equation 3 if χ is a binary number that is 1 if actionTickTimer represents the time the agent has been doing the same action, ($actionTickTimer > 50$), and ($actionTickTimer/10$).

$$A(t) = 0.8 \cdot A(t-1) + \chi + 0.1 \cdot Map(A, \lambda) \quad (3)$$

6. Results

We will test multiple AIs against themselves both with and without emotions to determine if the AI was improved with the emotions we implemented. We will also test AIs that have only fear or only frustration. The emotive AI will always be player 1, whereas the non-emotive AI will always be player 2. For all statistics of results we therefore compare player 1 to player

2. This ensures that our results are not biased due to starting location on the map, as the non-emotive versus non-emotive results are also calculated this way.

To analyze the success of a game we can examine the hit points (HP) per agent for each player, with agents referring to regular agents and buildings. HP represent the health of an agent, which decreases as an agent is injured and increases when it is healed; if an agent reaches 0 HP it dies. A high HP per agent ratio can signify that either the player has a high number of agents in various stages of health, or that all agents have high HP. HP also rises as the level of an agent increases, so high HP can signify more powerful warriors as well. Since all of these scenarios can represent a successful game, they also imply good performance. We can therefore use the ratio of HP to agents to determine whether the emotions improved the AI. Since each of these games is two AIs playing against each other, we can take the difference of their ratios at each time step and then average them. This average represents how much better the HP/agent ratio for the first AI is over the second AI for the duration of the game. We can also use the number of buildings and agents as a measurement of success. A higher number of buildings and agents on average implies that enough resources were gathered and that defense was relatively strong.

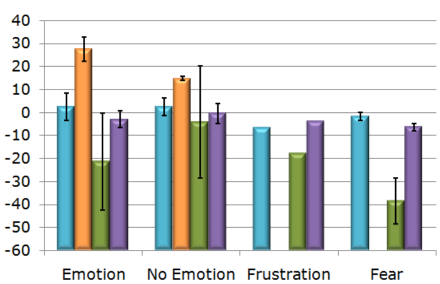
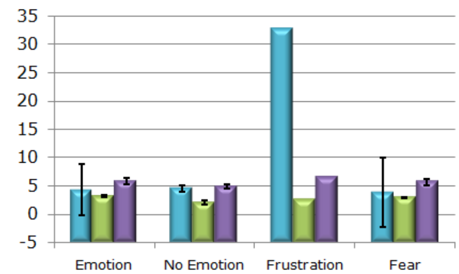
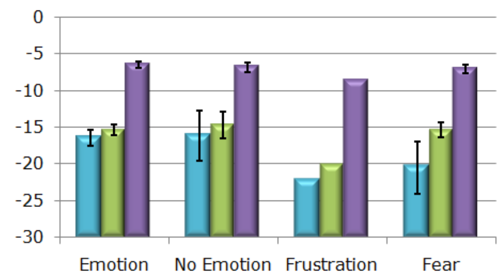


Figure 6. Results from Nicowar versus Nicowar. Each label on the X-axis represents the version of Nicowar (player 1) played against a non-emotive Nicowar (player 2). From left to right the bars represent: difference of average hp/unit (blue), difference of average hp/unit when player 1 one (orange), average number of agents (green), average number of buildings (purple).

Initially we test Nicowar vs. Nicowar, as seen in Figure 6. Player 1 wins 3/8 of the games when neither AI has emotion, and wins 1/4 of the games when player 1 has emotion and player 2 does not. At this point it is difficult to say if there is a significant change in wins. However we do see significant improvement of HP/agent when the emotion AI wins the game over when the non-emotion AI wins. It seems likely that the emotions give the AI an advantage overall. These results also show us that the AI performs worse when only fear or only frustration are used in the game, as opposed to the combination, but only in the sense that the emotion agent never wins a game. The difference of the overall HP/agent for each situation is not statistically significant.



(a) Warrush vs Warrush



(b) Numbi vs Numbi

Figure 7. The difference between the average number of hit points per agent on player 1's team versus the same average for player 2. Each label on the X-axis represents the version of the AI played against a non-emotive version of the same AI. From left to right the bars represent: difference of average hp/unit (blue), average number of agents (green), average number of buildings (purple).

We can also test other AIs versus themselves, as seen in Figure 7. For Warrush, there is no significant difference between using both emotions or no emotion. However, there is an extremely significant difference when using only frustration. In this case, the agents of Warrush retain a very high amount of health throughout the game, most likely due to agents surviving longer and thus gaining levels and increasing their maximum amount of health. There could also be an improved ability for finding the healers.

Numbi, however, does not see any change with the addition of emotion (Figure 7(b)). The chosen emotions and how they were modeled probably do not affect Numbi as it already performs those specific tasks well. This verifies that emotions, as with all decision processes, should be created with details designed for the specific situation.

It is also interesting to note that in all situations, using only frustration results in completely no variation in game outcome. As can be seen in all figures, there is no standard error between runs. This may be due to frustration somehow removing the uncertainty in agent action, such that no random choice is ever made. This may imply that makes decisions based on fear results in the later need to rely on the original basic decision making processes.

7. Conclusions

The addition of emotions has shown an overall improvement in the performance of the AIs. As expected, some AIs improved more than others and responded more favorably to certain scenarios. For instance, the AI Warrush improved significantly more with only frustration as opposed to utilizing both. These results suggest that the two emotions may conflict in certain scenarios. One such scenario is enemies in the base area, as the workers fear may stop the warriors from attacking or the warrior frustration of constant attacks may influence worker movement patterns. Depending on the scenarios likely to occur, different emotional influences should be utilized for different AIs.

The Nicowar AI, however, improved substantially more with both emotions, and was the only emotive AI to win multiple games. As both emotions were designed specifically to counteract problems in the Nicowar AI, this may imply that emotions designed for a specific system will improve that system's functionality the most although it may also improve similar systems. For real-time AI, these results suggest that emotions should be explored for designing new decision processes.

We also conclude that an emotion map is an efficient way to allow agents to communicate emotion with neighbors. It does not require direct communication but is more reminiscent of cellular communication. Since the interactions and social constructs that arise from emotion sharing are one of the key aspects of emotions, such an ability is necessary. We are currently continuing this work to increase the success of the emotions in controlling the agents by adding other emotions and modifying the emotional reactions of agents. It seems reasonable to utilize these fairly simple constructs to improve an agent's decisions.

The spatio-temporal evocation of emotion can be seen visually in Figure 4 at a few critical points in time. From these images it is possible to see that emotional evocation occurs during relevant times during the game play. Given that the expression of emotion occurs in the proper situations and that results show performance improvements for the emotive case, it is reasonable to conclude that an emotional framework will benefit the AIs in Globulation. Since the tested paradigm shows improvement over no emotion, more complex emotional systems should be examined. The relative success of our system demonstrates that adding emotions to a real-time system with artificial intelligence is feasible and warrants further study.

References

(2007). *Affective computing and intelligent interaction*, vol. 4738/2007. Springer.

Bulitko, V., & Wilkins, D. (2001). Real-time decision making for shipboard damage control. *AAAI*.

Buro, M. (2004). Call for ai research in rts games. *AAAI, AI in Games Workshop*.

Buro, M., & Furtak, T. (2003). Rts games as test-bed for real-time research. *Workshop on Game AI, JCIS*.

Buro, M., & Furtak, T. (2004). Rts games and real-time ai research. *Proceedings of the Behavior Representation in Modeling and Simulation Conference (BRIMS)*.

Corkill, D. (2003). Collaborating software: Blackboard and multi-agent systems & the future. *Proceedings of the International Lisp Conference*.

Eckman, P. (1994). *All emotions are basic*. Oxford University Press.

Herz, J., & Macedonia, M. (2002). *Computer games and the military: Two views* (Technical Report). Center for Technology and National Security Policy, National Defense University.

Laird, J. (2001). Using a computer game to develop advanced ai. *IEEE Computer*.

Musliner, D. J., Hendler, J. A., Agrawala, A. K., Durfee, E. H., Strosnider, J. K., & Paul, C. (1995). The challenges of real-time ai. *Computer*, *28*, 58–66.

Picard, R. (1997). *Affective computing*. MIT Press.

Rolls, E. (2005). *What are emotions, why do we have emotions, and what is their computational basis in the brain?* Oxford University Press.