The Influence of a Domain's Behavioral Laws on On-Line Learning

Thomas Haynes
Department of Mathematics & Computer Sciences
University of Missouri, St. Louis
8001 Natural Bridge Rd
St. Louis, MO 63121-4499
e-mail: haynes@arch.umsl.edu

Sandip Sen
Dept. of Mathematical & Computer Sciences
600 South College Avenue
The University of Tulsa
Tulsa, OK 74104-3189
e-mail: sandip@kolkata.mcs.utulsa.edu

Abstract

In multiagent systems, the potential interactions between agents is combinatorial. Explicitly coding in each behavioral strategy is not an option. The agents can start with a default set of behavioral rules and adapt them on-line to fit in with their experiences. We investigate perhaps the simplest testbed for multiagent systems: the pursuit game. Four predator agents try to capture a prey agent. We show how different assumptions about the domain can drastically alter the need for learning. In one formulation there is no need for learning at all, simple greedy agents can effectively capture the prey (Korf 1992). As we remove layers of abstraction, we find that learning is necessary for the predator agents to effectively capture the prey (Haynes & Sen 1996).

Introduction

The field of multiagent systems (MAS), also traditionally referred to as distributed artificial intelligence (DAI), is concerned with the behavior of computational agents when they must interact. One line of research is into the dynamics of cooperation of the group or team (Haynes & Sen 1997c; Sandholm & Lesser 1995). Another line of research is into the dynamics of competition as either individual agents or groups of agents vie for resources in artificial economies (Mullen & Wellman 1995). An individual agent in a group must balance the pressure of competition versus that of cooperation. It joined the group in order to achieve some expected utility which it believed it could not reach working on its own. Either it is incapable of doing the task, it does not have enough time to do the task, or it brings to the group some skill for which they do not have the capability to perform. However, it may have conflicting goals, prior commitments, hidden agendas, etc.

Multiagent learning combines machine learning with multiagent system research (Sen 1996; Weiß & Sen 1996). The typical goal is for an agent to learn on-line how to interact with a group of agents. On-line learning is utilized because we typically can not specify a priori the dynamics of group behavior. While off-line analysis is helpful, an agent may not have the luxury of disengaging itself from a computational agent society. It is conceivable that an agent may not have access to its "source-code"; it may not be able to go off-line. Also, a task may be mission critical and it may not be possible for an agent to take time out to analyze its actions. In order to minimize further conflict, the agent will have to learn on-line.

While fully deployed industrial multiagent systems are instrumental in understanding how the models of the agent group are effective in the marketplace, such systems are difficult to study as a controlled experiment. If we wish to study balancing the forces of competition and cooperation, we want to remove all other external factors. We want to be able to control the parameters of the experiment such that we can report with certainty our findings. One such testbed domain has been the pursuit game (Benda, Jagannathan, & Dodhiawala 1986), also referred to as the predator-prey game. In its most basic format, four predator agents chase a prey agent on a toroidal grid world. Agents are placed randomly in the world and may only make orthogonal moves. The goal is for the four predator agents to surround and capture the prey agent by occupying its four adjacent orthogonal neighbors.

The pursuit domain had been a standard testbed (Benda, Jagannathan, & Dodhiawala 1986; Gasser *et al.* 1989; Levy & Rosenschein 1992; Singh 1990; Stephens & Merx 1989; 1990) until Korf showed that reactive predator agents employing simple greedy algorithms could effect capture no matter what the starting position of the agents (Korf 1992). This research effectively killed the pursuit game as a testbed for MAS/DAI systems; indeed the game has been labeled as a "toy domain". Haynes and Sen proved that the assumptions made by Korf in setting up the domain were instrumental in allowing the greedy algorithms to effect capture (Haynes & Sen 1996). They showed that with a slight modification of the "physical laws" of the domain, a prey which did not move was able to avoid capture against the greedy algorithms employed by Korf's reactive agents. The real surprise of their research is that while the pursuit game is a "toy domain" in that four children could control the predator agents and coordinate their actions without any explicit communication, no hand-coded algorithm has been able to duplicate such a feat. There are complex interactions, which are readily solved with explicit communication, but which are difficult to solve with only implicit communication. The predator agents either need state, in which case they are no longer reactive, or models of the other agents, which may have to be adapted as the agents interact and discover scenarios not considered by the designer (Haynes & Sen 1997b).

The goal of this paper is to investigate the ramifications of changing the rules, which govern a simulation of a world, on on-line learning. We utilize the pursuit domain to illustrate a changing world. A key characteristic of the domain is that it models the hunt in the physical world and we all have an intuitive feel for the physical world. We go from a simplistic set of laws, as evidenced by the research of Korf, in which learning is not necessary at all, and scale up the complexity of the simulation until we reach a scenario in which on-line is necessary. Along the way we relate each proposed change in the rules to either a relaxing of the model of "physical" laws or a tightening of the model to follow the "physical" laws.

The Pursuit Problem

The original version of the predator-prey pursuit problem was introduced by Benda, *et al.* (Benda, Jagannathan, & Dodhiawala 1986) and consisted of four blue (predator) agents trying to capture a red (prey) agent by surrounding it from four directions on a grid-world. The movement of the prey agent was random. No two agents were allowed to occupy the same location. Agent movements were limited to either a horizontal or a vertical step per time unit. The goal of this problem was to show the effectiveness of nine organizational structures, with varying degrees of agent cooperation and control, on the efficiency with which the predator agents could capture the prey.

The approach undertaken by Gasser *et al.* postulated that the predators could occupy and maintain a *Lieb configuration* (each predator occupying a different quadrant, where a quadrant is defined by diagonals intersecting at the location of the prey) while homing in on the prey (Gasser *et al.* 1989). This study, as well as the study by Singh on using group intentions for agent coordination (Singh 1990), lacks any experimental results that allow comparison with other work on this problem.

Stephens and Merx performed a series of experiments to demonstrate the relative effectiveness of three different control strategies (Stephens & Merx 1989; 1990). They defined the local control strategy where a predator broadcasts its position to other predators when it occupies a neighboring location to the prey. Other predator agents then concentrate on occupying the other locations neighboring the prey. In the distributed control strategy, the predators broadcast their positions at each step. The predators farther from the prey have priority in choosing their target location from the preys neighboring location. In the centralized-control strategy, a single predator directs the other predators into subregions of the *Lieb configuration*. Stephens and Merx experimented with thirty random initial positions of the predators and prey problem, and discovered that the centralized control mechanism resulted in capture in all configurations. The distributed control mechanism also worked well and was more robust. They also discovered the performance of the local control mechanism was considerably worse. In their research, the predator and prey agents took turns in making their moves. We believe this is not very realistic.

The earlier research into the predator-prey domain involved explicit communication between the predator agents. Korf claimed that such expensive communication was unnecessary, as simple greedy algorithms always lead to capture (Korf 1992). He further claims that the orthogonal game, a discretization of the continuous world which allows only horizontal and vertical movements, is a poor approximation. In a diagonal version of the game (he also considered a hexagonal version), Korf developed several greedy solutions to problems where eight predators are allowed to move orthogonally as well as diagonally. In Korf's solutions, each agent chooses a step that brings it nearest to the prey. As with the research of Stephens and Merx, the prey agent moves and then the predator agents move in increasing order. An agent eliminates from consideration any location already occupied by another agent.

In the paper, Korf considered two distance metrics to evaluate how close a predator agent was to the prey. The *Manhattan* distance (MD) metric is the sum of the differences of the x and y coordinates between two agents. The *max norm* distance metric is the maximum of x and y distance between the two agents. With both algorithms, all ties are randomly broken. Korf selected the max norm metric for the agents to use to chose their steps. The prey was captured in each of a thousand random configurations in these games. But the *max norm* metric does not produce stable captures in the orthogonal game; the predators circle the prey, allowing it to escape.

Korf admits that the MN distance metric, though suitable for the diagonal and the hexagonal game, is difficult to justify for the orthogonal game. To improve the efficiency of capture (steps taken for capture), he adds a term to the evaluation of moves that enforces predators to move away from each other (and hence encircle the prey) before converging on the prey (thus eliminating escape routes). This measure succeeds admirably in the diagonal and hexagonal games but makes the orthogonal game unsolvable. Korf replaces the previously used randomly moving prey with a prey that chooses a move that places it at the maximum distance from the nearest predator. He claims this addition to the prey movements makes the problem considerably more difficult.

Competition versus Cooperation

Central to the study of multiagent systems is the tradeoff that must be made between competition and cooperation. When faced with a set of choices C, a single agent A_i is rational if and only if it selects the one which yields the highest expected utility $U_{i,max}$ (Russell & Norvig 1995). When an agent is a member of a group, it can select an option which does not yield the highest expected utility for it $U_{i,\neg max}$ and still be rational if the selected option yields the highest selected utility for the group $U_{G,max}$. It can put the needs of the many above the needs of the one.

The agent can still be competitive within the context of such cooperation. If the group has as a global goal T_G and the agent has a local goal T_i, then it will select actions that allow it to reach its local goal in the context of the group reaching the global goal. If it has a choice between multiple actions which both lead to the global goal, it will select the one which brings it closer to its local goal. If it has a choice between multiple actions, none of which lead to its local goal, it may select an action that leads another agent A_j away from its own local goal T_j.

Given a set $C' \subset C$ in which all the expected utility of all choices $U_{C',j}$ are such that $U_{G,max} - \delta \leq U_{C'} \leq U_{G,max}$ and δ is some constant denoting the latitude of an agent to accomplish a task, then the agent may select the choice $C_j \in C'$ such that it maximizes its expected utility given the need to cooperate with the other group members, $U'_{i,max}$.

Consider a blue–collar worker in an assembly line. In order for the company to stay solvent, the worker must process W widgets per hour. If a worker does not average W widgets per hour, then that worker is fired. The worker gets paid H dollars per hour, regardless of how many widgets produced in that hour. There is also a failure rate of ϵ, widgets are inspected during the next shift, and the worker's average only includes widgets that do not fail during inspection. Even though the company can make more profit if more than W widgets are produced per hour, and thus help ensure the company will not go bankrupt, which will cause the worker to lose the job, the worker has no incentive to produce more than the minimal widgets required. A worker may actually process $1.25 * W$ widgets per hour at the start of the day to make sure the quota of $W + \epsilon$ will be met, but during the latter part of the day may only produce $0.75 * W$ widgets per hour.

Finally, consider that such a job is stress free and the workers are able to chat to their friends as they are processing widgets. Also, assume that the worker gets paid by the number of widgets processed, but still has to process at least W widgets per hour. If a worker can average $2.0 * W$ widgets per hour, then he or she will be promoted to a higher paying job. However, in the new job, there is much more stress and less opportunity to chat to one's friends. The rational worker, who values both money and friends, will manage to average somewhere close to $2.0 * W - \delta$ widgets per hour, where δ is a comfort zone to ensure both the failure rate ϵ for making the most allowable average number of widgets and that too much is not done if the actual failure rate is $\epsilon' << \epsilon$. Indeed, such a worker would meticulously count the widgets and take one's time on the last widget to be processed.

Motivation for Learning

The classic tradeoff between cooperation and competition certainly applies to the greedy predator agents advocated by Korf. The goal of the individual predator is to get as close to the prey as possible. The goal of the group is to capture the prey. There are scenarios where these two goals conflict, see for example Figure 1. At this point the agents need to coordinate their moves. Korf claims that this coordination is not necessary. If we examine his basic algorithm, we see that such coordination is still necessary.

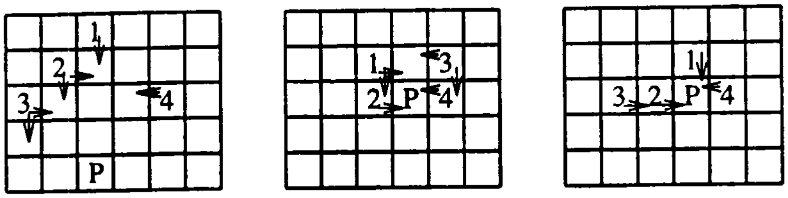
In the laws governing the simulation in Korf's research, the prey agent moves first and then the predators move in order. Consider Figure 2, in which the prey agent from the initial placements given in Figure 1 has already moved. In each case, the potential conflict vanishes. The predator agents are still employing a greedy heuristic, but they are coordinating their moves. Consider predator **2**'s options in Figure 1(b), it can either stay still or move *South*, further away from the prey. With either choice **2** makes, a capture is imminent. If it stays still, as it does in Figure 2(b), it will take predator **3** three moves to get to the capture position (assuming **3** does not go *North*). However, if **2** were to instead move *South*, it would take two moves for a capture to be effected.

Since the potential for conflict has been removed, the agents are engaging in communication, see Theorem 1. If we view the movement of all agents as a round in the simulation, then during a round the prey informs the predators where it is going to move before they decide where they will move. Likewise, each predator agent $\mathbf{p}_i$ informs all other predator agents $\mathbf{p}_{j,j>i}$ where it plans to move before they move.

Theorem 1 *The deterministic ordering of agent movement is a form of explicit communication.*

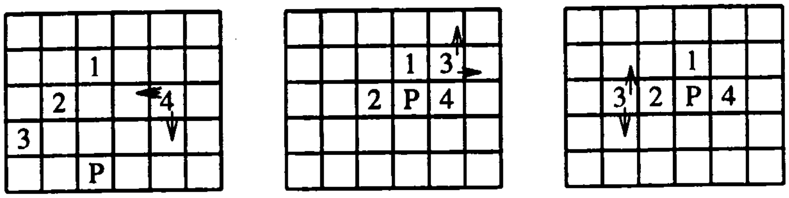
Also, Theorem 2 states that any prey which moves slower than the predators can be caught. If the prey moves at a speed 90% that of the predators, we can say that every tenth round, the prey does not move[1]. If the predators can manage to not loose ground to the prey when it moves, which the greedy algorithm is geared to do, then they are guaranteed to be able to move closer

[1] In the typical implementation, the availability of the prey to move is determined randomly each round. On the average, it moves at 90% of the predator's speed.



(a) (b) (c)

Figure 1: Possible scenarios in which the individual goal conflicts with the global goal. (a) Predator 2's best moves conflict with either that of 1 or 3. (b) Either predators 1 and 3 have a potential conflict with each other or 2 and 4 already occupy desirable positions. (c) Predator 3's best move is already occupied by 2.



(a) (b) (c)

Figure 2: From the initial positions in Figure 1, the prey P has taken its move, which was to stay still. (a) Predator 1 has moved and removed the potential conflict with 2. Predator 2 then moves, removing the potential conflict with 3. (b) Predators 1 and 2 have moved and it is 3's turn to move. Note that even though 4 has not moved, it still blocks 3 from moving *South*. At this point, 3 must move away from the prey P (unless it has the option to stay still). (c) Predators 1 and 2 have moved and it is 3's turn to move. This time, 4's placement does not conflict with that of 3's desires. Note that even though the move *South* is better, i.e., there is an open capture position, with either the MD or MN algorithms 3 is just as likely to move *North*.

when it is motionless. The algorithm employed by the prey can delay the inevitable, but because it is not called when the prey does not move, the predators will eventually catch it irregardless of the prey's algorithm.

Theorem 2 *Regardless of the behavioral algorithm employed by the prey, if the prey agent effectively moves slower than the predator agents and it selects and takes its move before the predators, then it can be caught in a finite number of steps.*

Korf claims that a prey agent which consistently moves away from the nearest predator (MAFNP) is harder to catch than one that moves randomly. If we consider a prey which moves randomly from the set {*North, South, East, West, Here*}, then it is effectively moving at 72% of the speed of the predators. While the prey utilizing MAFNP can avoid locality to delay capture, it will still suffer from not be able to reason and move when it is forced to stay still.

The final factor which contributes to the number of steps needed to capture the prey is the initial placement of the predators with respect to the prey. In Theorem 3, the minimum number of steps is a factor of the effective speed of the prey. If the farthest predator is n locations distant and the prey does not move with a probability of p, then it will take a minimum number of steps $\frac{n}{1-p}$ for the farthest predator to get into a capture position. This number of steps is a minimum because of the potential conflict of two or more predators for a capture position.

Theorem 3 *If the order of agent movement is fixed and the prey moves slower than the predators, then the number of steps to capture depends strictly on the initial placement of the agents and the effective speed of the prey.*

If the predators are placed in different quadrants, then the likelihood of conflict decreases. If however, two or more predators are placed in the same quadrant, then all but one will have to move away from the prey in order to capture it. For example, in Figure 2(b), predator 3 is in the same quadrant as 4 and 4 beat 3 to a capture position. To get to the final capture position, 3 must go around 4.

With the MD algorithm, a capture may not take place in a situation like Figure 2(b). Predator 3 will move away from the prey, but since it does not have any memory, the very next round it will move back to its previous position. By using the MN algorithm, 4 can eventually move *South*, as each of *North, Here,* and *South* are equally desirable at a distance of 1 and a random tie breaking will allow it to go *South*.

Removing the Communication

If the agents move in a deterministic order, simple greedy predator behavioral strategies are effective in capturing the prey. With the MD algorithm, deadlock situations can occur[2]. With the MN algorithm, the predators are not guaranteed to stay in a capture position (Korf 1992). Since we have shown, with an ordering on movements, the greedy agents are communicating, we would like to see if greedy agents are effective if we do not order the movements. An argument can be made that the ordering of the moves is not natural. While lions may prefer that the antelope move first, natural selection will favor any antelope which can move at the same time as the lions[3]. It has been shown by Haynes and Sen that using this new rule, there are many deadlock situations for the predator agents when pitted against simple prey algorithms of moving in a straight line (Linear) or even not moving at all (Still) (Haynes & Sen 1997a)!

The MN algorithm, as described by Korf, does not allow the predators to move to the cell occupied by the prey. (In his research, the prey moves first, followed by the predators in order. Thus conflicts are resolved between predators and prey by serialization.) Figure 3 illustrates a problem with this restriction. The cells to the *North* and *South* of predator 4 are as equally distant from the prey **P** as the cell currently occupied by predator 4. Since all ties are non-deterministically broken, with each movement of the agents, there is a 66% probability that predator 4 will allow the prey **P** to escape.

Assuming a Linear prey moving *East*, Figure 3 also illustrates the failure of the MN metric algorithms to capture a Linear prey. It is possible that a predator can manage to block the prey, but it is not very likely that it can keep the prey blocked long enough for a capture to take place. It is also possible that once captured, the prey may escape the MN metric algorithms. The MD metric algorithms do not suffer from this inability to make stable captures. They do however have a drawback which both the Linear and Still prey algorithms expose. Haynes and Sen found that MD metric algorithms stop a Linear prey from advancing. Their original hypothesis was that the Linear prey moved in such a manner so as to always keep the predators "behind" it. Thus, the inability to capture it was due to not stopping its forward motion. They started keeping track of blocks, i.e., a situation in which a predator blocks the motion of the prey, and discovered that the MD metric algorithms were very good at blocking the Linear prey.

The MD strategy is more successful than the MN in capturing a Linear prey (22% vs 0%) (Haynes & Sen 1997a). Despite the fact that it can often block the forward motion of the prey, its success is still very low. The MD metric algorithms are very susceptible to deadlock situations, such as in Figure 4. If, as in Figure 4(a), a predator manages to block a Linear prey, it is typical for the other predators to be strung out behind the prey. The basic nature of the algorithm ensures that positions orthogonal to the prey are closer than positions off the axis. Thus, as shown in Figures 4(b) and (c), the remaining predators manage to close in on the prey, with the exception being any agents who are blocked from further advancement by other agents. The greedy nature of this algorithm ensures that in situations similar to Figure 4(c), neither will predator **2** yield to predator **3** nor will predator **3** go around predator **2**. While the MN metric algorithms can perform either of these two actions, predator agents employing it are not able to keep the Linear prey from advancing. It is also evident that once the Linear prey has been blocked by a MD metric algorithm, the prey algorithm degenerates into the Still algorithm. This explains the surprising lack of captures for a prey which does not move.

For the majority of moves in the predator–prey domain, either the max norm or MD metric algorithms suffice in at least keeping the predator agents the same distance away from the prey. As discussed earlier, the prey effectively moves 10% slower than the predators, the grid world is toroidal and the prey must occasionally move towards some predators to move away from others. Therefore the predators will eventually catch up with it. Contention for desirable cells begins when the predators either get close to the prey or are bunched up on one of the orthogonal axes. What the predators need to learn is table manners. Under certain conditions, i.e., when two or more predator agents vie for a cell, the greedy nature of the above algorithms must be overridden. We could simply order the movements of the predators, allowing predator **1** to always go first (in essence regressing to the implementation used by Korf!). But it might not always be the fastest way to capture the prey. No ordering is likely to be more economical than others under all circumstances.

Also, if we consider real world predator–prey situations, the artificial ordering cannot always be adhered to. Consider for example a combat engagement between fighter aircraft and a bomber. If there are only two fighters, the ordering rule suggests that fighter **1** always moves before fighter **2**. If they are in the situation depicted in Figure 5(a), then fighter **1** cannot fire on the bomber **B**, because doing so will hit fighter **2**. Clearly, fighter **2** should first move *North* or *South*, allowing both it and the other fighter to have clear fire lanes. But under the proposed ordering of the movements, it cannot move in such a manner. So, the default rules is that fighter **1** moves before **2**, with an exception if

[2] Which indicates that on-line learning could be beneficial in this setup.

[3] This argument must be adapted to the discrete model we are applying for the game. In a continuous world, moving first denotes a slower reflex animal and is not a factor of the relative speeds of the animals.

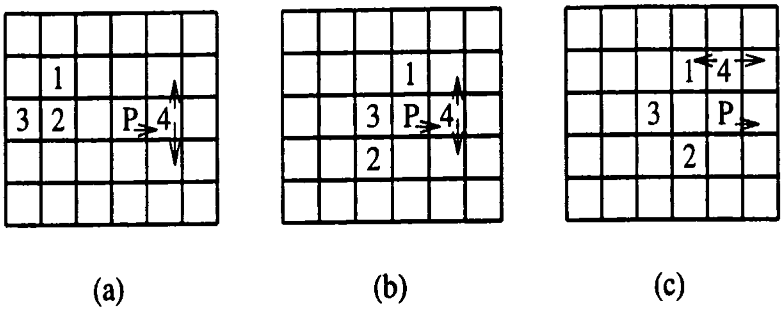


Figure 3: A possible sequence of movements in which a MN metric based predator tries to block the prey **P**. (a) Predator **4** manages to block **P**. Note that **4** is just as likely to stay still as move *North* or *South*. (b) Predators **1** and **3** have moved into a capture position, and predator **2** is about to do so. Note that **4** is just as likely to stay still as move *North* or *South*. (c) Predator **4** opts to move to the *North*, allowing the prey **P** to escape. Note that **4** is just as likely to stay still as move *East* or *West*.

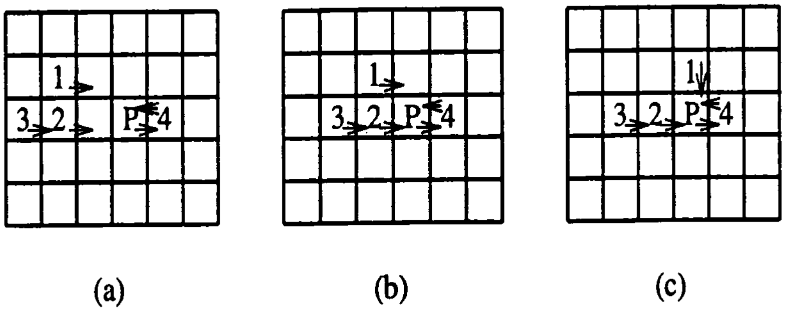


Figure 4: A possible scenario in which a MD metric based predator tries to block the prey **P**. (a) Predator **4** manages to block **P**. Predators **1**, **2**, and **3** move in for the capture. (b) Predator **2** has moved into a capture position. (c) Predator **1** has moved into a capture position. Predator **2** will not yield to predator **3**. They are in deadlock, and the prey **P** will never be captured.

fighter **2** is in front of **1**. The rule can be modified such that the agent in front gets to move first. However, if we add more fighters, then the situation in Figure 5(b) does not get handled very well. How do fighters **2** and **4** decide who shall go first? What if they both move to the same cell *North* of fighter **2**? These are the very problems we have been discussing with the MD metric algorithm.

Theorem 4 follows from our analysis of prey agent employing just the Linear and Still behavioral algorithms. It has been shown that if the prey utilizes more complex behaviors, e.g. maximize its distance from all predators, then Theorem 4 still holds (Haynes & Sen 1997a). In order to reduce contention for cells and to resolve deadlock situations, the predators must learn to cooperate. Either the designer of the behavioral algorithms has to enumerate each potential adverse situation or the agents can adapt through on–line learning.

Theorem 4 *Even if the prey agent effectively moves slower than the predator agents, if it selects and takes its move at the same time as the predators, then there exist many initial placements of the agents in which the predators cannot catch the prey if they are reactive*

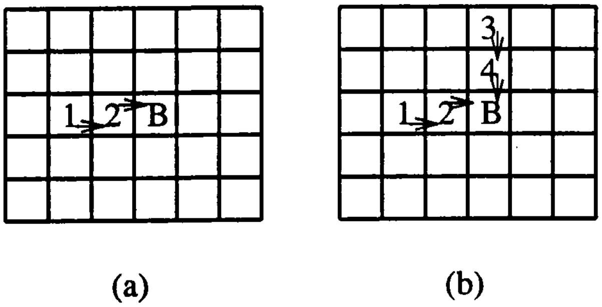


Figure 5: Conflicts in firing lanes for fighter planes strafing a bomber **B**. (a) fighter **1** is blocked from firing by **2**, and (b) Not only is fighter **1** blocked, but so is **3** by **4**.

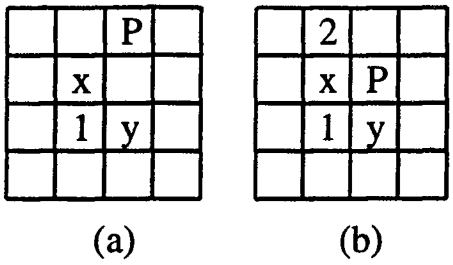


Figure 6: In both (a) and (b), the cells marked x and y are equal distant via the MD metric from the prey **P** for predator **1**. (a) x is chosen because the sum of the possible moves from it to prey **P** is less than the y's sum of moves, and in (b) y is chosen because while the look ahead is equal, there is a potential for conflict with predator **2** at x.

and employ strictly greedy algorithms.

Adding Behavioral Rules

Haynes and Sen have explored adding additional domain knowledge to the simple greedy agents (Haynes & Sen 1997b). In order to facilitate efficient capture, i.e., provide the agents with the best set of default rules, they enhanced the basic MD algorithm. If we consider human children as the agents playing a predator–prey game, we would see more sophisticated reasoning than simple greedy behavioral rules. When faced with two or more equally attractive actions, a human will spend extra computational effort to break the tie. Let us introduce some human agents: Alex, Bob, Cathy, and Debbie. Bob and Debbie have had a fight, and Bob wants to make up with her. He asks Alex what should he do. Alex replies that in similar situations he takes Cathy out to dinner. Bob decides that either Burger King or Denny's will do the trick (He is a college student, and hence broke most of the time). In trying to decide which of his two choices is better, he predicts how Debbie will react to both restaurants. Denny's is a step up from Burger King, and she will probably appreciate the more congenial atmosphere.

In the predator–prey domain, such a situation is shown in Figure 6(a). Predator **1** has a dilemma: both of the cells denoted by x and y are 2 cells away from the prey **P**, using the MD metric. The sum of the distances between all the possible moves from x and prey **P** is 8 and the sum from y to the prey **P** is 10. Therefore using this algorithm, which we call the look ahead tie–breaker, predator **1** should chose x over y.

A second refinement comes from what happens if the look ahead tie–breaker yields equal distances for x and y? Such a scenario is shown in Figure 6(b). Then predator **1** should determine which of the two cells is less likely to be in contention with another agent. Predators do not mind contending for cells with the prey, but they do not want to waste a move contending with another predator. By the least conflict tie–breaking algorithm, predator **1** should pick y over x (y has 0 contentions, while x has 1).

Suppose that Bob and Debbie have had another fight, but this time Alex and Cathy also have fought. Furthermore, a new restaurant, the Kettle, has opened up in town. Since the Kettle is on par with Denny's, Bob is again faced with a need to break a tie. As he knows that Alex and Cathy have fought, he believes that Alex will be taking her out to make up with her. Bob does not want to end up at the same restaurant, as he and Debbie will have to join the other couple, which is hardly conducive to a romantic atmosphere. He decides to model Alex's behavior. Like Bob, Alex is a student and has never eaten at the Kettle. Since Cathy is placated by being taken to Denny's and Alex does not like changing his routine, then Alex will most likely take her there. Thus Bob decides to take Debbie to the Kettle. Notice that if Bob had not accounted for a change in the environment, then his case would have caused a conflict with his goal.

A final enhancement is to restructure the order of the actions such that staying still is always the last action considered. This has the benefit of when the predators are in a capture position, they attack the prey and when all forward moves are blocked, the predators move away from the prey. By attacking, the predators are likely to follow the prey as it moves. By moving away from the prey, predators can avoid certain deadlock situations.

With these three enhancements, Haynes and Sen reported a significant improvement in the capture of the Still prey over the MD algorithm (46 versus 3 in 100 test cases). However, there is still vast room for improvement. If we instead use as a base algorithm the MN distance metric, predator agents enhanced with these rules will always capture and the prey agent employing the Still algorithm. Figure 7 illustrates some scenarios in which the enhancements facilitate capture. Notice that if the prey agent can move, the enhanced MN algorithm is still not guaranteed to capture the prey. Consider Figure 7(a), while the prey agents are repulsing each other to move into the capture positions, the prey could conceivably escape. There will be turns when 3 of the allowable moves will be open. These predator agents are not even guaranteed to keep the Linear prey blocked.

Conclusion

We have shown in the simple predator–prey domain that even greedy agents must communicate in order to capture the prey. If we remove communication, then predators need to learn to interact. We can improve the greedy algorithms employed by the predators, but without explicit communication, history, and/or full models of the other agents, situations will arise in which the agents will need to learn. Unless we can provide the full model of the other agents, which might be feasible in this domain, but certainly not in more "realistic"

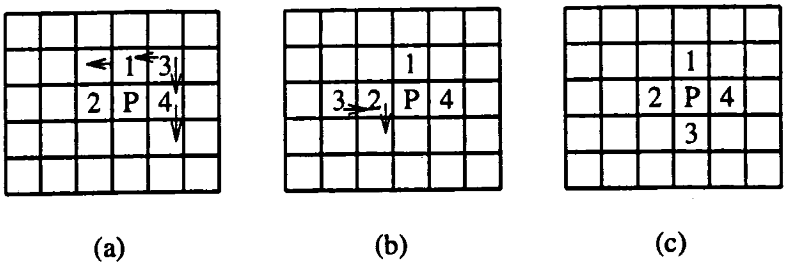


Figure 7: Possible scenarios for which the enhancements facilitate capture: (a) Predators **1** and **4** are "repulsed" from **3**'s location. Eventually non-determinism ensures that cycle of moving away from **3** and then back is broken and a capture ensues. (b) Predator **2** will move *South* has it has the least conflict. Predator **3** will move *East* as it is the closest to the prey **P**. In the next time step, **2** will move into the final capture position. (c) No predator will move to a diagonal cell as these cells are under contention. Hence the MN algorithm with the enhancements will *hold* a capture.

domains, the agents must adapt to each other by engaging in on-line learning. We can also make the laws governing the simulation more realistic. However, we find that previous models which did not employ learning now fail, and once again we go back to the need of on-line learning.

References

Benda, M.; Jagannathan, V.; and Dodhiawala, R. 1986. On optimal cooperation of knowledge sources - an empirical investigation. Technical Report BCS-G2010-28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, Washington.

Gasser, L.; Rouquette, N.; Hill, R. W.; and Lieb, J. 1989. Representing and using organizational knowledge in DAI systems. In Gasser, L., and Huhns, M. N., eds., *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*. Pitman. 55–78.

Haynes, T., and Sen, S. 1996. Evolving behavioral strategies in predators and prey. In Weiß, G., and Sen, S., eds., *Adaptation and Learning in Multi-Agent Systems*, Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag. 113–126.

Haynes, T., and Sen, S. 1997a. The evolution of multiagent coordination strategies. *Adaptive Behavior*. (submitted for review).

Haynes, T., and Sen, S. 1997b. Learning cases to resolve conflicts and improve group behavior. *International Journal of Human-Computer Studies (IJHCS)*. (accepted for publication).

Haynes, T. D., and Sen, S. 1997c. Co-adaptation in a team. *International Journal of Computational Intelligence and Organizations (IJCIO)*. (accepted for publication).

Korf, R. E. 1992. A simple solution to pursuit games. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, 183–194.

Levy, R., and Rosenschein, J. S. 1992. A game theoretic approach to the pursuit problem. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, 195–213.

Mullen, T., and Wellman, M. P. 1995. A simple computational market for network information services. In Lesser, V., ed., *Proceedings of the First International Conference on Multi-Agent Systems*, 283–289. San Francisco, CA: MIT Press.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Sandholm, T. W., and Lesser, V. R. 1995. Coalition formation amoung bounded rational agents. In Mellish, C. S., ed., *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 662–669.

Sen, S. 1996. Adaptation, coevolution and learning in multiagent systems. Technical Report SS-96-01, AAAI Press, Stanford, CA.

Singh, M. P. 1990. The effect of agent control strategy on the performance of a DAI pursuit problem. In *Working Papers of the 10th International Workshop on Distributed Artificial Intelligence.*

Stephens, L. M., and Merx, M. B. 1989. Agent organization as an effector of DAI system performance. In *Working Papers of the 9th International Workshop on Distributed Artificial Intelligence.*

Stephens, L. M., and Merx, M. B. 1990. The effect of agent control strategy on the performance of a DAI pursuit problem. In *Proceedings of the 1990 Distributed AI Workshop.*

Weiß, G., and Sen, S., eds. 1996. *Adaptation and Learning in Multi-Agent Systems.* Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag.