Velociraptor mongoliensis vs. *Thescelosaurus neglectus* : Winning Strategies and Genetic Algorithms

Team Number 46 for the Mathematical Contest in Modeling

February 10, 1997

1 Restatement of the Problem

During the Cretaceous period (65 to 144 million years ago), eastern Asia and western North America were joined together into one large land mass. Among its inhabitants was an infraorder of predatory dinosaurs called Deinonychosauria, meaning "terrible clawed lizards," so named for the large, sickle shaped claws on their feet. They were relatively small, 1.5 to 4 meters long from head to tail, bipedal with grasping hands, swift, and deadly. There is a distinct possibility that they were intelligent enough to have hunted in packs. Among them, the velociraptor, *Velociraptor mongoliensis*, is of interest to us.

Also at this time, a number of herbivores were evolving, including the family Hypsilophodontidae, so named for the ridges on their teeth. Several *Hypsilophodon* fossils were found together at one important dig, indicating that they may have lived in herds. They were bipedal and agile, possibly living a lifestyle similar that of gazelles or deer, and using their speed to escape predators.

One member of this family was *Thescelosaurus neglectus*, which inhabited what would later become Canada. It was larger and less agile than its relatives, and therefore a likely target for predators.

The subject of this project is to model the possible strategies velociraptors might have used while preying on thescelosaurs, and the related defense strategies used by the prey. Paleontologists will use these results to compare the velociraptor and thescelosaur to modern predators and their prey, such as wolves and deer, and cheetahs and gazelles, in the hope of better understanding their lifestyle. In the first part, the raptor is assumed to be acting alone against a single thescelosaur. The second part considers a pair of raptors and their possible group strategies.

2 Assumptions

The following assumptions are intended to simplify the problem for initial analysis while still retaining its most meaningful aspects.

- The velociraptor is 3 m long and weighs 45 kg. This estimate, given in the original statement of the problem, is contradicted by other sources which indicates that it is not clearly known. However, the raptor's exact size is not central to the mathematical model and a rough estimate of 2-3 m suffices.
- The thescelosaur is approximately the same size as the raptor. Again, this estimate is debatable, but the exact size is not critical.
- If the raptor can get to within arm's reach of the thescelosaur, he can kill it. This assumption is based on [2, page 113]. It describes a fossil site where the skeletons indicate that a velociraptor pounced on another dinosaur, grabbed part of its head, and cut its underside open with the claws on his feet. This suggests that the general hunting strategy was chase, grab, and slash.
- Both dinosaurs can be represented as points. Their shape is partially taken into account by the concept of the predator's arm reach.
- The velociraptor can run at top speed (60 km/h) for fifteen seconds, then must rest. He does not have the option of slowing down once he has reached this speed and must use the 15 seconds all at once.
- The thescelosaur can run at top speed (50 km/h) indefinitely. However, his top speed is significantly slower than the raptor's top speed.
- The velociraptor cannot maintain a speed comparable to its prey for very long and will lose distance if he runs at a comfortable speed. The chase must therefore be rather short.
- Both the velociraptor and the thescelosaur run at top (linear) velocity throughout the chase. This means that the chase is limited to fifteen seconds, because after that the raptor has to stop and rest. Since he can't keep up with the thescelosaur in a prolonged chase, there appears to be no advantage for the raptor in running at a comfortable speed which loses distance.

In fact, in any chase, the velociraptor must eventually kick into high gear to have any hope of catching his prey, and the rest of the fight reduces to the simplified case described here.

- As a consequence of the raptor's limited endurance, if he does not catch the prey during his 15 seconds of high speed running, the thescelosaur is assumed to escape.

- Both dinosaurs have a minimum turning radii which can be used to compute their maximum angular velocities. When they turn, they are agile enough to instantaneously assume this angular velocity. The velociraptor at top speed can run in arc with maximum radius of curvature 1.5 m, which amounts to 11.1 rad/s. For the thescelosaur, the turning radius is 0.5 m and the angular velocity is 27.8 rad/s.

- Both dinosaurs have instant reaction time. That is, if one of them changes direction, the other perceives it immediately. Part of our model involves a simulation, and either dinosaur may take action in as quickly as one time step.

- Initially, the thescelosaur notices the velociraptor when he is between 50 and 15 meters away, depending upon visibility and the alertness of the thescelosaur. Later analysis will show that if the raptor is discovered at more than 40 meters away, the thescelosaur will escape. Therefore, most of the model focuses on what happens if the raptor can sneak in fairly close.

- Once the chase has begun, both dinosaurs know exactly where the other is, how fast he is going, and in what direction.

- Both are also aware of how long the raptor can maintain his top speed, and how much time he has left.

- The chase takes place on a flat, featureless plain. There must be enough vegetation to feed the thescelosaur, so the setting of the hunt must be either a forest, a swamp, or a plain. The forest and swamp are full of obstructions which add too much complexity to the model, so those cases are not considered.

- Both dinosaurs are as intelligent as necessary, within reason. They are assumed not to have tool-making abilities or technology, but any other

strategy is possible. In particular, the velociraptor is capable of stealth and can sneak in fairly close to his prey.

3 Analysis of the Problem

We shall begin by considering the behavior of some simple strategies for a single velociraptor and a thescelosaur.

3.1 The Linear Case

As a simplest case, what happens when we only allow motion in a straight line? Naturally, the velociraptor will run directly towards the thescelosaur, who will be running directly away. Thus the velociraptor can close the initial distance by

$$\text{velociraptor running time} \cdot (\text{ velociraptor speed} - \text{thescelosaur speed})$$

in the time before its muscles give out. Under the assumed values of these variables, this is 40 meters. Remarkably, this means that under good conditions the thescelosaur can perceive the velociraptor far enough away (up to 50 meters) so that it can be assured of escape.

It seems likely that the velociraptor can't always afford to wait for a low-visibility day to eat, and so it follows that the velociraptor *must* be intelligent enough to use stealth or to hunt in groups in order to catch its prey. Even leaving that aside, this result makes good evolutionary sense. For it means that the velociraptor will mostly catch those thescelosaurs in the herd which have diminished perception or diminished running speed, namely the old and the infirm. This prevents the velociraptors from killing off their food supply; it may even help the herd to prosper.

3.2 Thescelosaur Escape Strategies

If, however, a thescelosaur is caught in non-optimal perception conditions, running directly away isn't enough. In this case the velociraptor has the advantage. We consider a velociraptor using the straightforward strategy of always moving directly towards its target. This could easily be improved—for instance by enabling the raptor to aim ahead of a thescelosaur moving at an angle—but for the moment the simple strategy will be more than good enough. So if even this relatively unintelligent velociraptor manages to get close to its prey before being seen (say, within 20 meters), what can the thescelosaur do to escape?

To explore this situation a simple simulator was constructed (see Appendix A for details).

It is fairly evident that the thescelosaur's hope lies in taking advantage of the velociraptor's large turning radius. Because its turning radius is so much smaller than the velociraptor's, it will perhaps be able to maneuver in such a way that the velociraptor is unable to close the distance as swiftly.

An initial attempt examines the thescelosaur strategy of always keeping itself moving in a direction at an angle to the velociraptor's motion (see Figure 1).

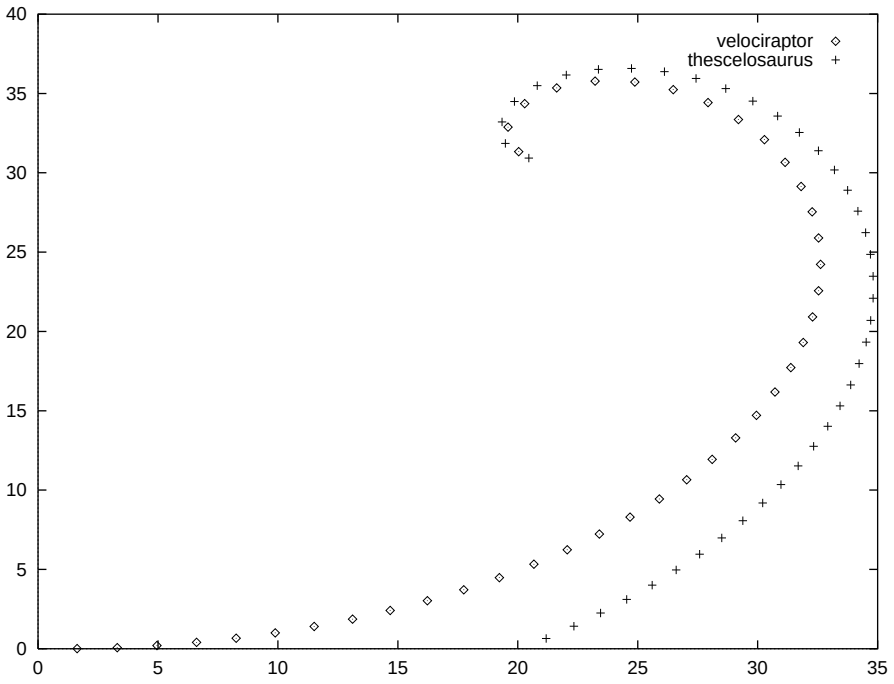

Figure 1: Maintaining angle strategy.

Unfortunately, starting at 20 m the velociraptor has enough time to react to the thescelosaur's turning, and so never has to push the bound on how sharply it can turn. In fact the thescelosaur is caught more quickly using this strategy then when it simply runs directly away.

To take better advantage of the velociraptor's limited turning, it seems that the thescelosaur needs to turn when the velociraptor is *closer*. This will force the velociraptor to make a much sharper turn to move towards its prey. The velociraptor will then overshoot and have to curve back in order to maintain a bead on the thescelosaur.

The most significant gains are made by the thescelosaur when it actually follows a *loopback* strategy, curving sharply around the imminently approaching hunter. The velociraptor will then be forced to turn all the way around in order to follow the thescelosaur, who has gained a significant amount of time (see Figure 2).

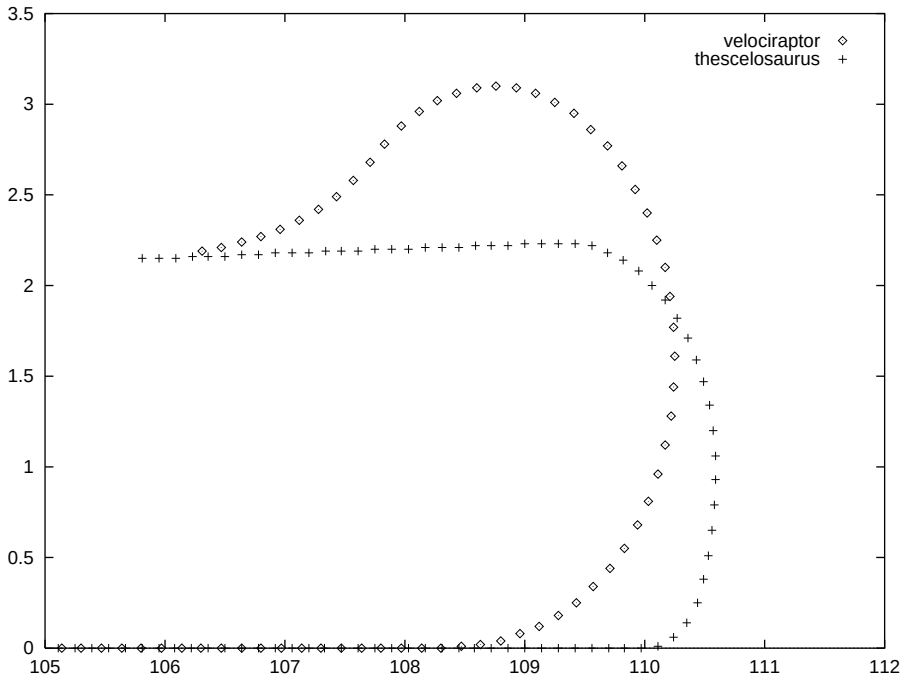

Figure 2: Loopback strategy.

There is a serious flaw with this strategy. Although the thescelosaur gains distance while the velociraptor is executing the upper half of its turn, to do so it has to come as close as a half a meter to the raptor, running perpendicularly in front of the raptor. So it is quite possible that the raptor could make a sudden lunge and bring a quick end to this strategy. Moreover,

since the simulation is representing meters-long creatures by points, it seems even more likely that coming so close would mean death.

A good way to imagine this process is as follows. The thescelosaur will run directly away from the velociraptor as long as possible. Then, just as the raptor is about to leap to the kill, the thescelosaur swerves sharply around its hunter. At this point there is a chance that the velociraptor will be able to lunge and wound the thescelosaur enough to stop its flight. On the other hand there is a certain probability that the raptor will miss because of the extreme speed of the chase. Moreover, it should be remarked that *Thescelosaurus* developed protective bony studs in its back, perhaps for just this kind of encounter ([2, page 141]). Nonetheless, the thescelosaur has significant chance of death using this strategy.

On the other hand, this loopback strategy was probably the thescelosaur's only chance for life. It is clear that the thescelosaur needs to use its more agile turning to evade its predator. However, if it turns while the velociraptor is still at a significant distance, the velociraptor—even one using the "dumb" directional strategy outlined above—is more than able to compensate. Also, if it tries to turn and still continue forward—for instance, in a forward zigzag motion—our simulation indicates that not nearly enough time is gained, and the velociraptor will almost certainly be able to close before its muscles fatigue. A loopback seems to be absolutely necessary.

It should be noted that this maneuver can be iterated to prolong the chase until the velociraptor gives up; however, with each swerve the thescelosaur runs a large chance of cutting it just too close. The iterated pattern can take on any number of interesting shapes (see Figures 3, 4, 5), depending on the thescelosaur's precise strategy for how close it allows the velociraptor and how sharply it turns; the basic idea is the same for all.

We feel that it is unlikely that these kinds of maneuvers serve as anything more than a last-ditch effort. It is conjectured that *Thescelosaurus* was a herd animal, and as such survival would be guaranteed by the fact that the old and infirm would be killed first, not quite so much by the ability of any individual to evade the determined hunter.

3.3 Velociraptor Hunting Strategies

So far we have concentrated on the optimal strategy for the thescelosaur, assuming that the velociraptor is using the straightforward strategy of always running directly toward its prey. The question next arises: how much is it

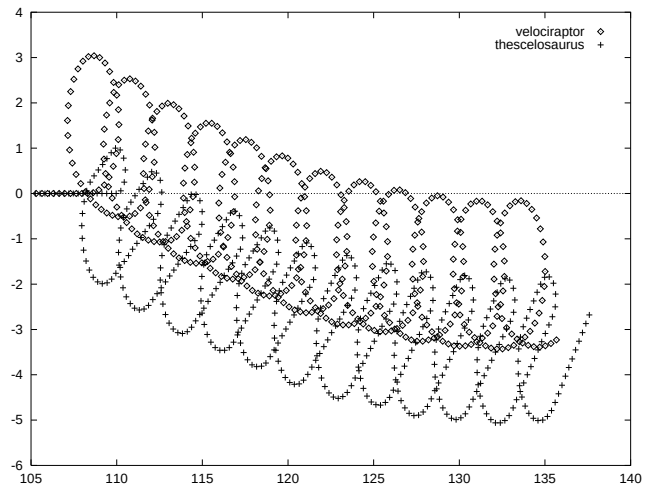

Figure 3: Zigzag loopback. That the thescelosaur seems to run toward the velociraptor at points is an illusion caused by the scale and lack of time reference.

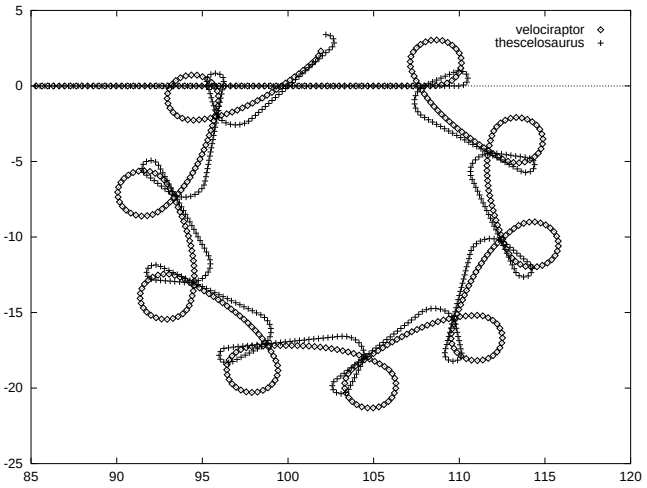

Figure 4: Dance of death.

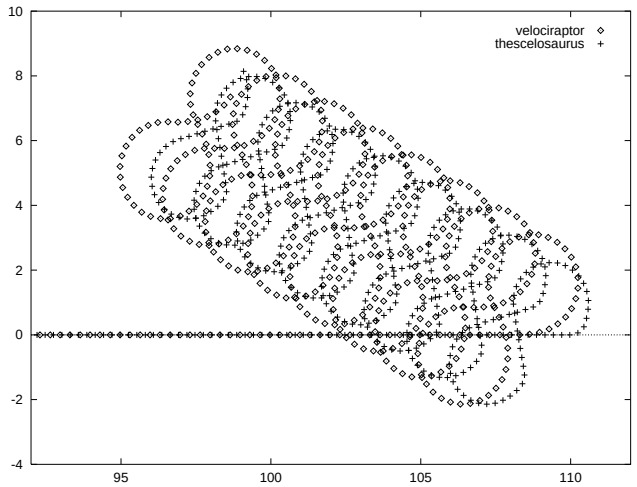

Figure 5: Backwards figure-8 loopbacks.

possible to improve on this hunting strategy for a single hunter?

Our conjecture is: not much. It has been seen that under good conditions for the thescelosaur's perception, a single velociraptor will be unable to approach closely enough to catch the other dinosaur; on the other hand, when the velociraptor is stealthy enough to approach more closely, the thescelosaur is unlikely to escape. (It can only escape even the "dumb" strategy through a series of swerves, each of which the thescelosaur will be lucky to survive.) In other words, whether the single velociraptor catches its prey seems to depend almost entirely on how closely it can approach before beginning its short sprint.

We certainly imagine that the intelligent hunter would follow a slightly better strategy, one that would anticipate its prey's location when the prey is running at an angle. That is, instead of always aiming directly at its prey, it would aim slightly ahead. It seems, though, that this strategy would at best provide a marginally better chance for the velociraptor to take down the thescelosaur when it swerves—a chance which is already quite high.

This analysis certainly makes sense considering what is conjectured about the hunting behavior of *Velociraptor*. Firstly, it was hunting a herd animal, so there would be the opportunity for solo hunters to pick off the weak and the stragglers. However, it certainly would have needed the ability to kill

stronger members of the herd when it was absolutely necessary. To this end, *Velociraptor* seems to have organized in hunting packs, and would have more likely hunted in pairs. This naturally leads to the consideration of optimal hunting strategy for a velociraptor hunting pair.

Before advancing to that particular question, we would like to discuss an entirely different approach to solving this problem. So far all we have really done is examine what the possible hunting and evasion strategies are, and use a simulator to see which are most effective. However, there is no guarantee that our consideration of various strategies has been exhaustive.

Biology seems full of examples of animal instincts which lead to behavior which is very effective, but which no human would ever consider. In particular, we are not hunters or prey, and so our perspective on this situation is totally foreign. We need some approach to the problem which will provide possibilities which we would never think of on our own. A natural way to explore the various possibilities is to use the same process as life itself.

3.4 Using a Genetic Algorithm

While it is fairly easy to model any given strategy for the predator and the prey, coming up with those strategies is not. The revised model is not concerned with any single strategy, but rather with the natural forces that would have caused those strategies to evolve in living dinosaurs.

Each creature in the physical simulation has a "brain" which tells it what direction to go next. So far, only a handful of hard-wired brains have been tried. Consider a more flexible brain which can take information about both dinosaurs and output one of the actions "turn left," "turn right," or "go straight." This process is repeated for each time step of the physical simulation. Since both animals are assumed to be going at top speed, there is no need to include "slow down" or "speed up." Since angular acceleration is instantaneous, there is also no need specify how far to turn. Smooth turns are made by intermixing a bunch of "turn" and "straight" instructions, and sharp turns are made by using more "turns." Keep in mind that the "brain" here an abstract combination of the physical brain, reflexes, instincts, and so on of a real dinosaur.

In nature, the dinosaur's instincts, reasoning, and reflexes are all combined in the decision making process. These factors are encoded in their genes and are passed on from one generation to the next. A dinosaur with a "good" brain is most likely to reproduce. A "dumb" predator on the other hand is likely to die of starvation, and a "dumb" prey will probably get caught and eaten. Under these circumstances, natural selection takes over, and as the best members of a species reproduce and recombine their genes, successively better dinosaurs evolve.

It is possible to simulate evolution with a computer program. Each brain is represented as a bunch of bits (0s and 1s) which take several bits of input, and produces one of the three strategies "left," "right," or "straight." The physical simulator described in the first part of this section determines which brains are better than others and may be used to sort the population in order of fitness. The best brains reproduce and the worst brains die and get replaced. Starting with a population of randomly created brains, the population will eventually home in on the optimum strategies. The details of this *genetic algorithm* are in **THE APPENDIX**.

This simple model has the ability to create any possible strategy that either dinosaur might use, even ones that are counterintuitive to humans and would not otherwise be discovered.

The great advantage to genetic algorithms is their speed. There are 2^{1024} possible brains in the implementation used in this project. Searching all of them exhaustively for the best ones for each species would take on the order of 10^{300} years, which is many orders of magnitude larger than the age of the universe. On the other hand, we estimate the running time of our genetic algorithm combined with the physical simulation to take on the order of a few years to run on a Sparc 5 computer. That's still too long to get results for this paper, but much better than the alternative.

The algorithm can be sped up by decreasing its precision. Currently, each time tick in the physical simulation is 5 ms, and could be made larger by a factor of ten. That sacrifice of accuracy brings the running time down to a matter of months on a Sparc 5, although the chance of getting meaningful results is difficult to predict. It can also be computed in parallel, with each processor of a super computer performing the work for a handful of dinosaurs. In this case, the algorithm could compute thousands of generations in a matter of hours. Therefore, despite the fact that this simulation cannot be run in time to give results for this paper, it is still a reasonable computation.

The disk included with this paper contains the code for the physical simulation, the genetic algorithm, a sample problem for the genetic algorithm which finishes running in just a few hours, and the combined evolution engine and dinosaur simulation.

3.5 Hunting Strategies for a Velociraptor Pair

We would like to note that the genetic algorithm model can easily be extended to more complicated hunting situations. In particular, one can begin with velociraptor and thescelosaur genomes that have already evolved powerful one-on-one strategies, and then put them in a two-on-one situation. This will force the genomes to adapt. By starting with already highly evolved genomes, one can save a significant amount of computer time.

We recommend the genetic algorithm approach because of its robustness. It will discover effective strategies that might be overlooked. However, we will also consider here some particular strategies for a hunting pair that we have devised.

When exploring the dynamic between a single velociraptor and a thescelosaur, we concentrated on thescelosaur strategies for evasion when the velociraptor had managed a close approach; for, with such a close approach, it seemed the velociraptor was almost assured of catching its prey. Moreover, when the velociraptor was farther away, the thescelosaur was assured of escape.

When the velociraptors hunt in pairs, the situation changes. Since the likelihood of a kill given a close approach is already high for one predator, we shall concentrate on strategies for the velociraptor pair when they are detected at a greater distance; in fact, near the upper limit of thescelosaur perception (50 meters).

3.5.1 Naive Strategy

The simple velociraptor strategy we have already seen is quite effective if the velociraptor begins close to its prey. In this section, we consider the effectiveness of this strategy from a greater distance with two hunters.

The raptors in Figure 6 begin at significant distance from their prey on opposite sides of it. They then proceed directly towards it. The thescelosaur takes its simple evasion strategy of running directly away from the closest raptor.

In a one raptor scenario this case would clearly end with a safe getaway for the thescelosaur. It is outside the 40m safe range in a straight chase; however, the thescelosaur quickly discovers that it cannot run straight away from either raptor (without entering the arms of the other), and in compromising on a new path to keep away from both raptors, it loses significant distance. The

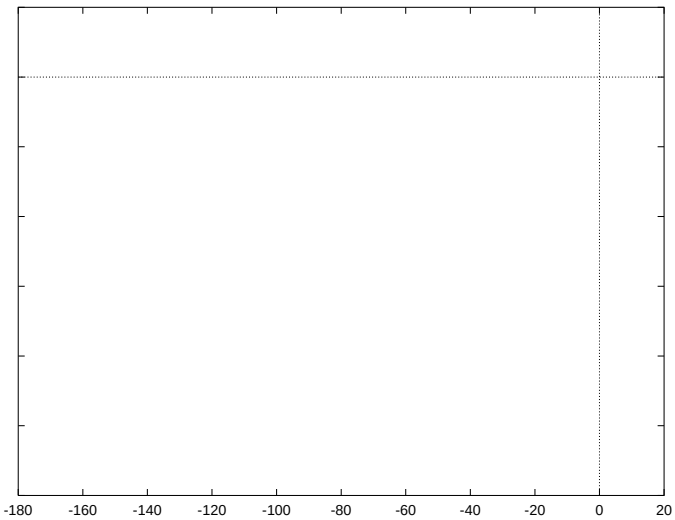

Figure 6: Naive hunting pair strategy.

raptors cut the corners off the thescelosaur's lead.

As simple as this case is, it nonetheless illustrates kernels of several important strategies: the raptors reduce the thescelosaur's range of options by beginning the chase from diametric positions; they reduce the thescelosaur's lead by herding it towards the more distant raptor; and they flank the thescelosaur, keeping it from effectively using its turning capabilities.

3.5.2 Initial Placement

As in the one raptor case, the initial positions of the raptors when they "appear" to the startled thescelosaur are vital factors in determining the outcome of the chase. The sample two raptor case in Figure 6 illustrates an excellent geometry: the raptors come from opposite directions. This dramatically reduces the thescelosaur's alley of escape.

Another important part of the raptors' initial placement would likely involve timing their overt attacks to coincide. An ill-timed attack might drastically reduce the effectiveness of the late raptor.

3.5.3 Herding strategies

Once the chase in Figure 6 begins moving, the thescelosaur immediately runs from the closest raptor; by "frightening" the thescelosaur away from it, the first raptor is thus bringing the second (and more distant) raptor much closer to the prey. While raptor 2 starts 10m further from the thescelosaur, it ends the chase neck and neck with its colleague.

This sort of herding strategy can compensate for timing problem in the raptors attacks, but it also seems possible that one "obvious" raptor could chase an unsuspecting thescelosaur perilously close to a well-hidden raptor before the prey would take note of the second hunter.

3.5.4 Flanking Strategies

Next we shall consider strategies which two velociraptors can use when they are detected at more or less the same time. Although the raptors in the sample case in Figure 6 did fence the thescelosaur in, it would seem that a more coordinated and careful attack could be executed by a pair of intelligent, consenting raptors.

If the two raptors aimed ahead of the position of the thescelosaurus, they could cut down their approach time. This alone would extend the sure kill

zone against this simple thescelosaur strategy to well beyond the worst case detection radius. However, with a richer defensive strategy, the raptors will be forced to curtail the effectiveness of thescelosaur loopbacks.

We feel that the flanking strategy can also provide a solution to this difficulty if the raptors maintain a few metres margin between them as they approach striking distance from the prey. If the thescelosaur attempts a turn, the raptor on the side it turns to can attempt to catch it as it turns while the raptor on the opposite side can turn to follow. Even if the first raptor does not catch the thescelosaur, the second raptor will have extra room to turn and will not be likely to overshoot the thescelosaur's turn. As can be seen in Figure 2 that gives the thescelosaur its lead. Thus, it seems likely that one or the other of the raptors will be succesful under these conditions.

3.5.5 Further Strategies

Other velociraptor strategies certainly seem possible, e.g., spiraling in towards the prey; however, even the simplest of these strategies seems devastatingly effective. If no effective thescelosaur escape strategy can be found even by powerful simulation techniques like the genetic algorithm, we would suggest that perhaps

4 Strengths and Weaknesses

4.1 Strengths of the model

- Flexibility — the greatest strength of our model is certainly its overall flexibility. The separation of perception from thinking allows us to easily change the focus or capabilities of either the hunter or the prey; the genetic algorithm allows us to construct a wealth of possible strategies; and the configuring the simulator's tickTime allows us to control the runtime and granularity of all the action.

- Independence — this is independence from human thought patterns. The genome system "lives" in a universe where the only carrots and sticks come from the success or failure of its reptilian "body". As such, its strategies will evolve to maximize success in that world. These strategies will not be filtered by the authors' human perspective.

- Robust Extensibility (parts of the model code) — new types of raptors and thescelosaurs, new perceptive capabilities and emphases, and new conditions on the success of the hunt could all be easily added to the model. These areas would soon need to be extended as assumptions like total prey and predator awareness (perception) were discarded. Thus, the model is quite ready for more realistic circumstances.

4.2 Weaknesses of the model

- No accommodation for the physical dimensions of prey and predator — this weakness of the model is perhaps the most fundamental in light of the project: to model the hunting strategies of **velociraptors**. While we have simulated the common constraints of the hunter (speed, endurance, and turning radius), only with the specific body form of the raptor and thescelosaur in mind can we accurately distinguish the hunting capabilities of a raptor with these characteristics from a generic hunter (i.e., a cheetah).

- Memory — even during the heat of the chase, it is likely that an effective predator would use the course of the chase so far to determine its next move. This weakness could perhaps be alleviated by maintaining

"sticky" protein concentrations in the genetic algorithm. These left-over proteins from previous cycles would represent memory of previous perceptions and decisions.

- Weak extensibility (parts of the model code) — the greatest code weakness in the model is that the code is unprepared for expansion to more than two raptors or more than one thescelosaur. This is not explicitly requested in the problem statement, but one can extrapolate that studying pack hunting and herd defenses might be useful to the clients of our model.

A The Physical Simulation

In order to explore the behavior of raptors and thescelosaurs, we constructed the following computer simulation. (For programs, please see enclosed disk.)

A.1 The Model

Each creature is represented as a point in space. Associated with the point is a vector representing the creature's velocity. (Recall that we assume that the creatures will be running full speed during the entirety of the chase; although in the two-velociraptor case, it is possible that one will appear to the thescelosaur later than the other.)

Time is represented discretely; that is, we only update the positions and velocities, and the creatures only make decisions, after a certain number of milliseconds. We call these units *ticks*. The coarseness of the ticks can be changed to adjust for limits of computation time, or to model finite reaction times for the creatures. We typically used between 1 and 5 ms.

Each tick the creature makes one decision: move ahead full speed, turn left full speed, or turn right full speed. This significantly simplifies the model. If a creature wants to make a more gentle turn, it has to turn sharply, move ahead a few ticks, turn sharply, move ahead a few ticks, and so on. If the tick time is not too coarse (say, less than 10 ms), this approximates a more gentle turn very accurately.

The initial positions of the creatures are changed based on what level of thescelosaur perception we are considering; the raptor(s) will be between 15 and 50 meters away, and the second raptor might not appear until a certain time into the simulation.

The simulation ends when either 15 seconds have elapsed (a victory for the thescelosaur) or the thescelosaur-point is within a certain reach distance of a velociraptor-point (a victory for the velociraptors). We typically used a reach distance of between half a meter and one meter. This is actually quite small; we believe that if the thescelosaur gets that close, it can only be said to have a certain probability of survival. For the moment though we will always give the thescelosaur the benefit of the doubt.

A.2 Interfacing with the Genetic Algorithm

The remaining consideration is how the creatures make their decisions. The simplest way is simply to hardwire a strategy. However, you are limited by the strategies you can devise and your ability to program them. Instead, we recommend interfacing the physical simulation with a genetic algorithm.

Each creature is considered to have a "brain" which is a genome. Each tick the creature does three things: *perceive*, *think*, and *act*. Perceiving is interpreting data about the world. The important data is the relative position and velocity of the other creatures, the amount of time each raptor has before muscle fatigue, and, in the two-raptor scenario, whether the thescelosaur has detected the second raptor yet. Each of these data is discretized for simplicity; so for example instead of knowing the exact distance to its prey, the velociraptor will only be able to tell what range of distance it is, out of say

- ¡1 m
- 1–2 m
- 2–3 m
- 3–5 m
- 5–10 m
- 10–15 m
- 15–40 m
- ¿40 m

This discretization could be made coarser or finer, depending on computer constraints. It could also be used to represent what is believed to be the dinosaur's inherent ability to perceive. It should be noted that the discretization will affect the evolution of strategies, because it limits what changes in the environment can be considered important. Under ideal conditions the discretization would be made quite fine, and what changes in the environment are considered important would be left up to evolution to determine.

The discretization allows us to represent each datum as an bit string. For instance the eight distance ranges allow distance to be represented using

three bits. The bit strings for each perception data are then concatenated, into one long string of bits, which can be considered as an integer. This integer encodes all relevant data about the creature's world. This integer is now passed to the brain.

After perception, the creature proceeds to think. The genome which is the creature's brain will take the integer representing perception, and using this input will change protein concentrations (see Appendix B for details). In particular there will be a "turn left" protein, a "straight" protein, and a "turn right" protein. Which of these proteins has the highest concentration will be the decision of the brain.

Finally, the creature acts on this decision. In the simulation, this means that its location and velocity are updated based on which action it performed. It is assumed to have followed that action for one whole tick.

A.3 Evolution

So how do we get effective strategies? We use a process of evolution. For technical details, please see Appendix B.

Begin with a population of random genomes for each creature (we used 500 for each). For each genome, play a certain number of "games" (we used ten) with the other creature (or creatures) chosen randomly. That is, take the genomes, put them in the simulation as the brains of the creatures, as described above, and see what happens. Each simulation has a winner and a loser: the thescelosaur wins if it escapes; the raptor wins if it eats. Take a certain perception of the genomes which win the most often (we used thirty percent) and kill the rest. Cross breed the living genomes, introduce appropriate mutations, and compete again. Each of these iterations is one generation. After many generations (say, one thousand) the creatures have learned how to win. The velociraptor has learned how to catch its prey, and the thescelosaur has learned how to escape. Those that did not learn how to win, died.

It is remarkable that the genomes have no idea what they are or what they are doing. All they have is an input bit string, and at the end they know whether they win or lose. It makes no difference to them whether they are the brains of velociraptors chasing thescelosaurs, or the brains of velociraptors playing chess. It is very much a game.

B The Genetic Algorithm

B.1 Overview

A mathematical game is a contest between two opponents, each of which may select among several actions. The payoff to each player under every possible combination of actions is known. The problem is to discover strategies under which a player can maximize his payoff. Most such games can be analyzed with techniques from linear algebra.

Unfortunately, if the strategies and payoffs are not completely understood, the linear algebra techniques are useless. A better idea might be to model the circumstances which shape the game rather than the strategies themselves, and hope that good strategies will emerge. In this case, a method of analysis which can invent strategies is needed, and genetic algorithms do just that.

The idea is simple: We present the members of a species with a problem and ask them to solve it. The ones that can solve it the best are kept and the others are fed to the velociraptors. The very best of the best are allowed to breed with one another and produce new creatures with hybrid strategies. An element of randomness is present to prevent stagnation. The best strategies for the problem eventually appear. If there is one single best strategy, it will dominate the population. If instead there are multiple optima, they will dominate together in proportion to their utility. For a good treatment of evolution and game theory, see *The Selfish Gene*, by Richard Dawkins.

Methods of determining strategies are called *genomes* and are divided into components called *genes* which favor selection of one action over another. These are the players' brains, so to speak. Deciding how good a genome is requires putting it through a simulation or some other computation and assigning it a rating. The best genomes can then be determined by any sorting algorithm. The breeding process amounts to taking genes from two different genomes and patching them together into a new genome, which is then added to the population.

In a computer model of evolution, genes are represented as strings of bits (0 or 1) which must consider several bits of input, and decide which strategy to use.

If the genetic algorithm is to work well, the problem must satisfy a number of conditions:

- The problem must be one of searching for the optimum solutions to a

question. Finding good hunting strategies falls into this category.

- It should be fairly simple to pose.
- Similar strategies should yield similar payoffs. The breeding process assumes that crossing two good creatures stands a good chance of yielding another.

The creatures' genomes must also satisfy some constraints for evolution to be computable.

- The genome must be representable as a string of bits which takes several bits of input and outputs some strategy.
- Genomes must be crossable. That is, if you take parts of two genomes and patch them together into a third, the result must be something other than nonsense. For example, if you cross Word Perfect with MathCad, the result is not a mathematical word processor or even a runnable computer program, just garbage. Hence, CPU-executable machine code is a bad genome format. On the other had, treating the bits as digits of an integer in binary notation is very good, since the crossing of two such integers results in another perfectly valid integer.

B.2 The Protein Population Model

In the computer simulation, genomes are represented as strings of bits. It is possible to model the interpretation of these bits after the way cells use their DNA, and the result is a genome which satisfies the conditions of the genetic algorithm.

In a living cell, the genome is stored in long molecules called DNA. Life processes take place because its genes encode other molecules called proteins which in turn execute other genes or cause the cell to take various actions, such as taking in water or moving. Genes are executed only when there are proteins floating around in the cell which read their particular section of DNA, resulting in a feed-back relationship between proteins and genes. If more proteins which read a particular gene are floating around in a cell, it's more likely that that gene will be read; consequently, the population of the protein it encodes will increase. Proteins also affect each other's function, and can disable one another, resulting in feed-back between various types of proteins. Male mammals, for example, contain all the genetic code for female

reproductive organs, but they are never executed thanks to these feed-back mechanisms. Other feed-back loops ensure that a cell contains the proper amount of water, and so on.

The same feed-back process can be used in a computer implementation of a genome. The genome consists of a number of pairs of genes. Each pair is associated with a probability index. To determine what action the genome wants to take, go through each pair of genes, and compute a random number. If that number is less than or equal to that pair's probability index, the genes are executed. This process constitutes one *thinking cycle.*

Each gene encodes one of two kinds of actions, `p` or `s`. `P` genes change probabilities and correspond to proteins in a cell which read DNA. For example, the `p+13` gene increments the probability of gene pair thirteen. If the first gene is read, the second is more likely to be read at a later time. Similarly, there are `p-` genes which decrement the probabilities of a gene pair.

`S` genes contribute to the decision of what strategy the genome should employ. For example, `s+1` indicates a vote for action number 1, and `s-7` indicates a vote against action number 7.

Input to the genome consists of a string of bits. During the initial thinking cycle, all probabilities are set to zero except for those of the pairs bound to input. If bit n is set to zero, the probability index of pair $2n$ is maximized, and that of pair $2n + 1$ is minimized. The reverse is true if bit n is set to one. This corresponds to detection molecules in cells which activate certain proteins when they encounter the substance they are looking for.

After several thinking cycles, the votes for and against the strategies are examined, and the creature performs the one that got the most votes.

Incidentally, the reason why genes are grouped into pairs is to facilitate feed-back mechanisms and to emulate real genes which tend to have side effects. For instance, in humans, the gene for sickle cell anemia also provides resistance to malaria.

Note also that there is a terminology problem: In humans, the "gene" for blue eyes actually consists of several "genes" which encode different proteins, each of which consist of dozens of "genes" for amino acids and protein folding instructions. In this discussion, the term "gene" will usually refer to any single `p` or `s` code, although the pairs of genes, and even the pairs which interact with one another, could also be referred to as "genes."

B.3 Technical Information about the Dinosaur Genetic Algorithm

The dinosaur simulation requires a parallel genetic algorithm, since the raptor strategies and thescelosaur strategies both have an influence on one another. For example, if the raptor's strategy is to run around in circles, the thescelosaur can escape by simply running away in a straight line. However once one raptor discovers the concept of running towards its prey at top speed, its genome soon dominates the predator population, and the simple run-away strategy is not effective and the thescelosaurs will have to come up with something else. Improvements in one species translate into improvements in the other.

To implement this double evolution, each genome of either species is assigned a rating. In the single velociraptor simulation, this involves pairing each raptor up with randomly selected thescelosaurs, and each thescelosaur with several velociraptors. The chase begins and each dinosaur uses a genome thinking for 20 cycle s to decide where to move next. The predator scores one point each time it catches its prey, and loses one point each time it gets away. The prey scores the opposite of the predator. On average, each dinosaur participates in the same number of tests.

Once all members of both species have been rated, they are partially sorted through an algorithm similar to quick-sort. The average rank of the population is estimated from a small sample, and those with above average ratings are moved to the top of the list, while those with below average ratings are moved to the bottom of the list. The two halves of the list are then organized recursively. The process stops early and does not sort creatures that will be bred no matter what order they're in, nor does it sort creatures which will be replaced.

The ranking is not completely accurate. Sometime good genomes are discarded and bad genomes are kept. This is to keep the population from converging too quickly and possibly getting trapped at a local maximum.

In each generation, the lower 40 percent of the population is replaced by the offspring of the upper 30 percent. Two genomes produce an offspring by crossover. Both are split at a randomly determined point and the top half of one is combined with the bottom half of the other. The new genome also goes through a mutation loop where some of its bits, approximately 1 percent, are toggled at random. Again this is to maintain variety and prevent the population from homing in on a local optimum. These percentages are

arbitrary and could probably use some tweaking. The mutation rate is very high compared to nature, but works well enough with small populations.

Each gene is represented by a string of bits. In the specific implementation used here, each genome contains 64 pairs of genes, each 8 bits long, so there are a total of 256 different actions.

Since there are 64 pairs of `s+` and `s-` genes, the genome can select from up to 64 different strategies. This model doesn't use more than a handful. Rather than ignore genes such as `s+45`, we take the strategy number modulo the total number of strategies. If a genome is supposed to decide among 7 actions, `s+45` is interpreted exactly like `s+3` because $45 \equiv 3 \pmod 7$.

B.4 A Simple Strategy Selection Example

Imagine for a moment that the strategies for the velociraptor are as follows: sleep, walk, hop, use photon torpedoes, polka...; and the strategies for the thescelosaur are: shake hands, sing, erect magneto-quantum deflector shield, shout.... Clearly, the raptor's best bet is number 3 and the thescelosaur's is number 2 (numbering the options from zero, as computers usually do). To get things going, all creatures are always given sixteen 1s for their input bits. Starting with a random population of both species and playing by the rule "The raptor wins if his 3rd tally is greater than or equal to the thescelosaur's second tally," yields some interesting results in a few minutes.

Clearly, the best strategy in both cases is to have a lot of `s+` genes for the appropriate tally. After the first ten generations or so, not much has happened: the important tallies are close to twenty or thirty. This is because the best arrangements consist of a couple of `s+` commands in the genes that are turned on by the input. After a few dozen generations however, things start to get more intricate. Take this prey gene for example:

```
  1:  15: 1b=p+27. 92=s+2.
  5:  15: d8=s-8.  a2=s+2.
 11:  15: 82=s+2.  9=p+9.
 12:  14: c0=s-0.  92=s+2.  (Turned on by 56)
 17:  15: 38=p+56. 8a=s+10.
 15:  15: ba=s+10. 82=s+2.
 25:  15: 72=p-50. a2=s+2.
 26:   0: 82=s+2.  70=p-48.
 28:   0: a2=s+2.  71=p-49.
```

```
 31:  15: 92=s+2.  64=p-36.
 56:  15: bc=s+12.  c=p+12. (Turned on by 17)
```

The order of the columns is pair number, probability index (0 to 15), and the two genes. The genes which don't affect bucket number 2 or these selected genes have been omitted. Pairs 1, 5, 11, 15, 17, 25, and 31 are turned on by the input, but pair 12 is turned on indirectly by pairs 17 and 56. A couple of `s+2` genes are also present but are apparently being ignored. They may have been part of a gene combination in an ancestor's genome that was scrambled up a bit by crossover. This genome produced a tally of around 130. After 150 generations, the tally count of both species goes up above 200.

This predator gene, from the 1000th generation, produces a tally of around 420.

```
  1:  14: 1c=p+28. 39=p+57. (Turned on by input)
  2:  15: a3=s+3.  d7=s-7. (Turned on by 21)
  3:  15: a3=s+3.  3b=p+59. (Turned on by input)
  5:  15: c6=s-6.  93=s+3. (Turned on by input)
  7:  14: a3=s+3.  3f=p+63. (Turned on by input)
  8:  15: a3=s+3.  a3=s+3. (Turned on by 17, 34, and 63)
  9:  15: 2f=p+47. 83=s+3. (Turned on by input)
 11:  15: b3=s+3.  f8=s-8. (Turned on by input)
 12:  12: 1d=p+29. 9f=s+15. (Turned on by 57)
 13:  15: 93=s+3.  83=s+3. (Turned on by input)
 15:  15: f8=s-8.  b3=s+3. (Turned on by input)
 17:  15:  8=p+8.  83=s+3. (Turned on by input)
 18:   2: 5d=p-29. 88=s+8. (Turned on by 34)
 19:  15: a3=s+3.  8d=s+13. (Turned on by input)
 21:  15: 78=p-56.  2=p+2. (Turned on by input)
 23:  15: 93=s+3.  d9=s-9. (Turned on by input)
 25:  15: 3b=p+59. 93=s+3. (Turned on by input)
 29:  15: b3=s+3.  83=s+3. (Turned on by input, 12, off by 18)
 31:  15: b3=s+3.  93=s+3. (Turned on by input)
 34:   3: 12=p+18.  8=p+8. (Turned on by 56)
 47:  15: a3=s+3.  83=s+3. (Turned on by 9)
 49:   0: 95=s+5.  c3=s-3. (Never turned on)
 52:   1: 89=s+9.  38=p+56.
 56:   0: 22=p+34. 28=p+40. (Turned on by 52, off by 21)
 57:  15:  c=p+12. a3=s+3. (Turned on by 1)
```

```
59:  15: ec=s-12. 93=s+3. (Turned on by 3 and 25)
62:   0: d9=s-9.  93=s+3. (Never turned on)
63:  15: b3=s+3.   8=p+8. (Turned on by 7)
```

(Note: Probabilities of 0 do not indicate that a gene is never executed, because the random number test selects an integer in between 0 and 15 inclusive and executes the gene pair if the random number is less than or equal to the index.) This genome is full of complex behavior, with genes turning other genes on indirectly, feed-back relationships, and so on. For example, gene pair 21 does the double duty of turning on number 2, which has an `s+3` code, and of suppressing number 56, which tries to turn off the valuable number 29 through the chain 56, 34, 18, 29.

This particular example was done to test the computer program for correctness. It took approximately five hours to run 1000 generations starting from a completely random population on a Sparc 5 computer.

It also uses a different comparison method than the velociraptor simulation: the vote counts in a specific tally are considered, which encourages complex interaction among the genes to evolve quickly, as in the examples. For raptors and thescelosaurs, the exact number of votes in each bucket doesn't matter; only which one has the most votes is significant.

B.5 Analysis of Genetic Algorithms

The evolutionary process is remarkably stable. In some early tests of the computer program, good results still appeared despite the fact that there were numerous bugs in the code, some of them quite serious.

There is no question that the algorithm made progress in the above examples. Consider a genome generated at random. About half the genes are `s` type, with half of those `+` and half `-`. Therefore, the tallies for that genome should be close to zero, as confirmed by numerical experiment. A population of genomes which consistently produce tallies of over 400, as in the thousandth generation of the example, is most definitely not a random occurrence and indicates that the algorithm is indeed doing work.

The disadvantage to this approach is the time. This particular implementation is well optimized, performing $\Theta(n)$ simulations[1] per genome, and better than $O(n \log n)$ rating comparisons per generation. Unfortunately, it took 5 hours for the simple test experiment described above. A test for the

[1] Knuth's Θ and O notations for asymptotic orders of magnitude.

velociraptor simulation using a population of 50 and playing an average of two simulations per creature took several minutes for one generation, most of which was spent performing 4 million think cycles. A properly formed raptor and thescelosaur algorithm with 500 individuals playing 20 simulated hunts against each other would take on the order of 6 hours per generation, which means it would take nearly a year to really make progress (1000 generations).

On the other hand, the advantage to this approach is the time. The obvious alternative of comparing every possible raptor genome to every possible thescelosaur genome would take on the order of 2^{1024} simulations. 2^{1024} is a three hundred digit number, and human civilization will have no doubt gone the way of the dinosaurs many times over by the time that computation is finished.

References

[1] Alexander, R. McNeill. *Dynamics of Dinosaurs and Other Extinct Giants.* Columbia University Press: New York, 1989.

[2] Dixon, Dougal et al. *The Macmillan Illustrated Encyclopedia of Dinosaurs and Prehistoric Animals.* Macmillian: New York, 1988.