PATRICK WINSTON: I have extremely bad news.

Halloween falls this year on a Sunday.

But we in 6.034 refuse to suffer the slings and arrows of outrageous fortune.

So we've decided that Halloween is today, as far 6.034 is concerned.

Kenny, could you give me a hand, please?

If you could take that and put it over there.

STUDENT: [INAUDIBLE]?

PATRICK WINSTON: Mm hm.

Just give it to them.

You can take as much of this as you like.

The rest will be given to that herd of stampeding freshman that comes in after.

It's a cornucopia of legal drugs.

Chocolate does produce a kind of mild high, and I recommend it before quizzes and giving lectures.

I have a friend of mine, one of the Nobel laureates in biology, always eats chocolate before he lectures.

Gives him a little edge.

Otherwise, he'll be flat.

So I recommend it.

It will take, I suppose, a little while to digest that neural net stuff.

A little richer than usual in mathematics.

So today, we're going to talk about another effort at mimicking biology.

This is easy stuff.

It's just conceptual.

And you won't see this on the next quiz.

But you will see it on the final.

It's one of those quiz five type problems, where I ask you questions to see if you were here and awake.

So a typical question might be, Professor Winston is a creationist, or something like that.

Not too hard to answer.

In any event, if it's been hard to develop a real understanding of intelligence, occasionally the hope is that by mimicking biology or mimicking evolution, you can circumnavigate all the problems.

And one of those kinds of efforts is ever to imitate evolution.

So we're going to talk today about so-called genetic algorithms, which are naive attempts to mimic naive evolution.

Now, I realize that most MIT students have a basic grasp of sexual reproduction.

But I've found in talking with students that many times, they're a little fuzzy on some of the details.

So let's start off by reflecting a little bit about how that works.

So let's see, we need pink and blue.

And here's our cell, and here is its nucleus, and here are mommy and daddy's chromosomes.

We'll just pretend there's one pair.

Now ordinarily, in ordinary cell division, you get two cells.

Both have a nucleus, and the process of producing them involves the duplication of those chromosomes.

And then one pair ends up in each of the child cells, and that's all there is to that.

That's mitosis.

But then, when we talk about reproduction, it's more complicated because those chromosomes get all twisted up, and they break, and they recombine.

So when we talk about the cells that split off from one of these germ cells, it's no longer appropriate to talk about the pink one and the blue one, because the pink one and the blue one are all mixed up.

So you get two chromosomes here and two here.

But through some miracle of nature, which has always amazed me, these two cells split, in turn, into four cells altogether.

And each of those four cells at the bottom gets one of the chromosomes that was produced by the twisting up of the rope and recombination.

Then, along comes a special occasion.

And now we can think of this as being a blue one and this as being a pink one.

And they come together, and you get a new person, like so.

Note that your mother and father's chromosomes are never, never recombined.

It's your grandparents chromosomes that recombine.

So that's what it's like.

And the main thing to note about this is-- well, a couple things.

If you happen to be female, this part of the process-- this part over here-- took place before you were born.

If you happen to be male, it's going on right now as we speak, which probably explains something.

But in any event, it's going on right now.

But whenever it goes on, there are lots of opportunities for throwing dice.

So God threw all the dice before you were born, if you happen to be a female, in this part of the process over here.

Then, of course, more dice got thrown here when the decision was made about which particular cells got to fuse to form a new individual.

So I want to beat on this idea of lots of choice.

When we talk about genetic algorithms, and we talk about nature, there are lots of choices in there.

And that means there are lots of choices to intervene, to screw around, to make things work out the way you want.

But in any event, there we are.

That's the basic idea, and it all starts with chromosomes.

So we could think of implementing something that imitates that with the ACTG stuff that you learned all about in Seminar 1.

But we're computer scientists.

We don't like Base 4.

We like Base 2.

So I'm going to just suggest that our chromosomes are binary in this system that we're going to build.

So that might be a chromosome.

And it doesn't have to be binary.

It can be symbolic for fall I care.

But it's just some string of things that determine how the ultimate system behaves.

So it all starts out, then, with some of these chromosomes, some of these simulated chromosomes, simulated, simplified, and naive.

And there's a population of chromosomes.

The population of chromosomes-- it might be subject to a little bit of mutation.

That is to say, a zero becomes a one, or a one becomes a zero.

That happens a lot.

That mutation stuff happens over here when things get twisted up and recombined.

There are copying errors and stuff.

Cosmic rays hit it all the time.

All sorts of reasons why there might be a single point of change.

That produces the mutation effect.

So here, we have a population that starts off over here.

And some of those things are subject to mutation.

And you'll note, here's a whole bunch of choice already.

How many of these mutations do you allow per chromosome, for example?

How many of the chromosomes just slip through without any mutation?

Those are choices you can make.

Once you've made those choices, then we have the crossover phenomenon.

Let's identify one of these guys as the pink one and one of these guys as the blue one.

And so now we have the pink one cruised along as well as the blue one.

The pink one and the blue one cross and produce a new chromosome, just like in nature.

So we take the front part of one, back part of the other, and we fuse them together.

And some may slip by without any of that, like so.

Well, these things are meant to be combined in pairs, like so, but they may not have any crossover in them.

So you have another set of choices.

How many crossovers do you allow per recombination?

You get another set of choices.

So now we've got a population of modified chromosomes through mutation and crossover.

So the next thing to do is we have the genotype to phenotype transition.

That is to say the chromosome determines the individual.

It may be a person.

It may be a cow.

It may be a computer program.

I don't care what.

But that code down there has to be interpreted to be a something.

So it is the genotype, and it has to be interpreted to be something which is the phenotype, the thing that the stuff down there is encoding for.

So here, we have a bunch of individuals.

Now, each of those individuals, because they have varying chromosomal composition, will have a different fitness.

So these fitnesses might be-- well, who knows how they might be scored.

But we're computer scientists.

We might as well use numbers.

So maybe this guy's fitness is 88, and this guy's fitness is 77, and so on.

So now that we've got fitness-- by the way, notice all the choices involved there-- choice of how you interpret the genotype, choice about how the phenotype produces the fitness.

And now we have a choice about how the fitness produces a probability, like 0.8 and 0.1, or something like that-- probability of survival into the next generation.

So now, once we've got those probabilities, we actually have selection.

And those phenotypes out there produce genotypes, a new set of chromosomes, and that completes our loop that goes back in there.

And so that's the new generation.

Sounds simple.

So if you're going to make this work, of course, you have a million choices, as I'm going to emphasize over and over again.

And one of your choices is, for example, how do you compute the probability of survival to the next generation given the fitness?

So we have to go, somehow, from numbers like these to probabilities like those.

So I'm going to talk about several ways of doing it.

None of them are magic.

None of them was specified and stipulated by God as the right way.

But they have increasingly good properties with respect to this kind of processing.

So the simplest thing you can do-- idea number one for computing the-- see, what you do is you get this whole bag of individuals, and you have to decide who's going to survive to the next generation.

So at each step, everything in the bank has a probability of being the one you pick out and put in the next generation.

So at any step, the sum of the probabilities for each of those guys is 1, because that's how high probability works.

The probability of a complete set, added all up, is probability of 1.

So one thing you can do is you can say that the probability that you're going to draw individual i is equal to, or maybe is proportional to, the fitness of that individual.

I haven't completed the expression, so it's not a probability yet, because some piece of it won't add up to 1.

How can I ensure that it will add up to 1?

That's easy.

Right.

All I have to do is divide by the sum of the fitnesses over i.

So there's a probability measure that's produced from the fitnesses.

Yeah.

STUDENT: You need to make sure that the fitnesses aren't negative.

PATRICK WINSTON: Have to make sure the fitnesses are what?

STUDENT: Aren't negative.

PATRICK WINSTON: He says I have to make sure the fitnesses aren't negative.

Yeah, it would be embarrassing if they were.

So we'll just [? strike ?] anything like that as 0.

You've got a lot of choice how you can calculate the fitness.

And maybe you will produce negative numbers, in which case you have to think a little bit more about it.

So now, what about an example?

Well, I'm going to show you an example.

Why don't I show you the example.

What we're going to do is we're going to have a genetic algorithm that looks for an optimal value in a space.

And there's the space.

Now, you'll notice it's a bunch of contour lines, a bunch of hills in that space.

Let me show you how that space was produced.

The fitness is a function of x and y, and it's equal to the sine of some constant times x, quantity squared, times the sine of some constant y, quantity squared, e to the plus x plus y divided by some constant.

So sigma and omega there are just in there so that it kind of makes a nice picture for demonstration.

So there's a space.

And clearly, where you want to be in this space is in the upper right-hand corner.

That's the optimal value.

But we have a genetic algorithm that doesn't know anything.

All it knows how to do is mutate and cross over.

So it's going to start off with a population of 1.

It's a little red dot down in the lower left.

So here's how it's going to evolve.

There's going to be s chromosome consisting of two numbers, an x number and a y number, like, say, 0.3 and 0.7.

Here's another one, which might be 0.6 and 0.2.

So the mutation operator is going to take one of those values and change it a little bit.

So it might say, well, we'll take 3, and we'll make it 0.2.

And the crossover operation is going to exchange the x and y values of pairs.

So if we have a crossover here, then what we're going to get out from this one-- well, we're going to get out a combination of these two.

And it's going to look like this.

Because what we're going to do is we're going to take the x value of 1 and combine it with the y value of the other one.

So this is going to be 0.2-- my mutated value and 0.2-- and this is going to be 0.6 and 0.7.

So that's how my little genetic algorithm heck is going to work.

So having coded this up, we can now see how it flows.

Let's run it 10 generations.

So the population is rapidly expanded to some fixed limit.

I forgot what it is-- 30 or so.

And we can run that 100 generations.

And so this seems to be getting stuck, kind of, right?

So what's the problem?

The problem is local maxima.

This is fundamentally a hill-climbing mechanism.

Note that I have not included any crossover so far.

So if I do have crossover, then if I've got a good x value and a good y value, I can cross them over and get them both in the same situation.

But nevertheless, this thing doesn't seem to be working very well.

STUDENT: Professor, I have a question.

PATRICK WINSTON: Yeah.

STUDENT: That picture is just the contour lines of that function.

PATRICK WINSTON: The contour lines of that function.

So the reason you see a lot of contour lines in the upper right is because it gets much higher because there's that exponential term that increases as you go up to the right.

So I don't know, it looks-- let's put some crossover in and repeat the experience.

We'll run 100 generations.

I don't know.

It just doesn't seem to be going anywhere.

Sometimes, it'll go right to the global maximum.

Sometimes it takes a long time.

It's got a random number generator in there, so I have no control over it.

So it's going to get there.

I couldn't tell whether the crossover was doing any good or not.

Oh, well, here's one.

Let's make this a little bit more complicated.

Suppose that's the space.

Now it's going to be in real trouble, because it'll never get across that moat.

You know, you would think that it would climb up to the x maximum or to the y maximum, but it's not going to do very well.

Even with crossover, it's just not going to do very well, because it's climbing up those local hills.

Anybody got an idea about one simple thing we could do to make it work better?

Yeah, you could increase step size, right?

Let me you see if that will help.

You know, even that doesn't seem to help.

So we have to conclude-- do we conclude that this is a bad idea?

Well, we don't have to conclude it's a bad idea yet, because we may just look at it and ask why five times.

And we might ask, well, maybe we can get a better mechanism in there to translate fitness into probability of survival.

Using this formula is kind of strange, anyway, because suppose temperature is one of your fitness characteristics.

The hotter, the better.

Then the ratio of the probability that you'll survive versus the person next to you, that ratio will depend on whether you're measuring the temperature in Celsius or Fahrenheit, right?

Because you've shifted the origin that shifts the ratio that shifts the probability of success.

So it seems kind of strange to just take these things right straight into probabilities.

So a better idea-- idea number two-- is to say, well, shoot, maybe we don't care about what the actual fitnesses are.

All we really care about is the rank order of all the candidates.

So the candidate with the most fitness will have the most probability of getting into the next generation.

The candidate with the second-most fitness will have the second-highest probability, and so on.

But we're not going to use the actual fitnesses themselves to make the determination.

Instead, what we're going to do with this mechanism number two-- this is the rank space method-- is this.

We're going to say that the probability of the highest-ranking individual of getting into the next generation is some constant P sub c, which, of course, you can select.

You have another choice.

Then, if that guy doesn't get selected, the probability of the second-highest-ranking individual getting in the next generation is going to be the probability that that guy didn't get in there.

That's 1 minus P sub c times the same probability constant.

And so you can see how this is going.

P3 will be equal to 1 minus P sub c squared terms P sub c.

P sub n minus 1 will be equal to 1 minus that probability constant to the n minus-- n minus 2 times P sub c.

And then there's only one individual left.

And then if you got through all these guys and haven't got anybody selected, then you've got to select the last guy.

And so the probability you're going to select the last guy is going to be 1 minus P sub c to the n minus 1.

So it's a probability you've missed all those guys in the first n minus 1 choices.

Yeah, it is, honest to God.

See, this is the probability that this last guys is going to get selected.

It's not the probability that it's the last guy getting selected, given that the others haven't been select.

Trust me, it's right.

Are you thinking it ought to be 1?

STUDENT: What?

PATRICK WINSTON: Were you thinking it ought to be 1?

STUDENT: No, I was thinking that I was wondering why you were re-rolling the dice, so to speak.

PATRICK WINSTON: You are re-rolling the dice.

You've got a probability each time, except for the last time, when, of course, you have to take it.

There's nothing left.

There's no other choice.

STUDENT: I have a question.

PATRICK WINSTON: Yeah, [INAUDIBLE].

STUDENT: So when you jump from [INAUDIBLE], that makes sense.

[INAUDIBLE] you're saying [INAUDIBLE].

PATRICK WINSTON: It's the probability [INAUDIBLE] the first two choices.

STUDENT: Yeah, but the second choice had probability one minus P sub c times P sub c, not-- PATRICK WINSTON: Think about it this way.

It's the probability you didn't choose the first two.

So the probability you didn't choose the first one is one minus P sub c.

The probability you didn't choose the next one, as well, because you're choosing that next one with probability P sub c, it's the square of it.

So that might work better.

Let's give it a shot.

Let's go back to our original space choice, and we set and switch to the rank fitness method.

And we'll run out 100 generations.

Whoa!

What happened there?

That was pretty fast.

Maybe I used a big step size.

Yeah, that's a little bit more reasonable.

Oops-- what happened?

It's really getting stuck on a local maximum.

So evidently, I've choosed a constant P sub c such that it just drove it right up the nearest hill.

On the other hand, if I change the step size a little bit, maybe I can get it to spread out.

I sure did.

And now that it's managed to evolve over there to find the maximum value, now I can clamp down on the step size again.

And now it shows no more diversity.

It's just locked on to that global maximum.

So this is not unlike what evolution sometimes does.

Sometimes, species collapse into a state where they don't change for 500 million or 600 million years, like sharks, for example.

Sometimes, they only survive if they've got a lot of diversity built into their way of life so that they can adjust to habitat changes.

Now, when you increase the step size, because you're stuck on a local maximum, it's like heating up a metal.

You make everything kind of vibrate more, make bigger steps.

So this kind of process, where you may start with a big step size and then gradually reduce the step size, is called simulated annealing, because it's like letting a metal cool down.

So you start off with a big temperature-- big step size-- that covers the space.

And then you slowly reduce the step size, so you actually crawl up to the local maxima that are available.

So that seemed to work pretty well.

Let's see if we can get it to work on the harder problem of the moat.

So it's not doing very well.

Better increase the step size.

No, it's still kind of stuck.

Even though it's got the capacity to cross over, it's so stuck on that lower right-hand corner, it can't get up that vertical branch to get to a point where a crossover will produce a value up there in the upper right-hand corner.

So we're still not home yet.

So what's the trouble?

The trouble is that the fitness mechanism is just driving things up to the local maximum.

It's just terribly unfortunate.

What to do?

Well, here's something you could do.

You can say, well, if the problem is we've lost the diversity in our population, then we can measure the diversity-- not only the fitness of the set of individuals we're selecting from, but we can measure how different they are on the individuals we've already selected for the next population.

In other words, we can get a diverse population as well as a fit population if, when we make our selection, we consider not only their fitness but how different they are from the individuals that have already been selected.

So that's going to be mechanism number three.

So now we have a space, and we can measure fitness along one axis-- ordinary fitness-- and this is rank space fitness, so that's going to be P sub c.

There will always be some individual with the highest fitness.

And over here-- that might not be P sub c, actually.

But there'll be some individual with a maximal fitness, and at any given step in the selection of the next population, there'll be some individual that's maximally diverse from all of the individuals that have been selected for the next

generation so far.

So what kind of individual would you like to pick for the next generation?

Well, the one with the highest fitness rank and the one with the highest diversity rank.

So what you'd really like is you'd like to have somebody right there.

And if you can't have somebody right there, if there's nobody right there with a maximum fitness, a maximum diversity at the same time, then maybe you can draw in iso-goodness lines, like so, which are just how far you are from that ideal.

So let's summarize.

You've got to pick some individuals for the next population.

When we pick the first individual, all we've got to go on is how fit the individual is, because there's nobody else in that next generation.

After the first individual is selected, then we can look at our set of candidates, and we can say which candidate would be more different from the set of things we've already selected than all the others.

That would get the highest diversity rank and so on down the candidate list.

So let's see how that might work.

So we're going to use a combination of fitness rank and diversity rank.

And we'll just use the simple one so far.

We'll use a small step size, and we'll let this run 100 generations to see what happens.

Bingo.

It crawls right up there, because it's trying to keep itself spread out.

It uses that diversity measurement to do that.

And at the same time, it's seeking high fitness, so that's why it's crawling up to the upper right-hand corner.

But in the end, that diversity piece of it is keeping the things spread out.

So suppose you're a shark or something.

You don't care about diversity anymore, And we could just turn that off.

Is that thing still running?

Go back to fitness rank-- bingo.

So there you are-- you're stuck for 600 million years.

So let's see if this will handle the moat problem.

See, our step size is still small.

We'll just let this run.

So the diversity of P sub is keeping it spread out, pretty soon, bingo, it's right in there.

It's across that big moat, because it's got the crossover mechanism that combines the best of the x's and the best of the y's.

So that seems to work pretty well.

OK, so see, these are some of the things that you can think about when you're thinking-- oh, and of course, we're a shark, we're going to forget about diversity.

We'll change the selection method from fitness and diversity rank to just diversity.

It collapses down on to the highest hill.

Yeah, [INAUDIBLE], what?

STUDENT: How does step size translate into mutations?

PATRICK WINSTON: Oh, just the-- question is, how does step size translate into mutation?

Instead of allowing myself to take steps as big as 1/10, I might allow myself to take steps as big as 3/10, according to some distribution.

So what to say about all this?

It's very seductive, because it's nature, right?

The trouble is, it's naive nature.

And as evolutionary theories go, this is horrible.

This is naive.

So we'd like to use real evolutionary theory, except we don't have real evolutionary theory.

Evolution is still a mystery.

Some things are pretty obvious.

You can breed fast race horses.

That works just like so.

The trouble is, we don't have any real good idea about how speciation takes place and how a lot of evolution works, because all these chromosomes are connected to their phenotype consequences in very complicated ways that nobody fully understands.

So there's a great deal of magic in that genotype to phenotype transition that nobody really understands very well.

So when people write these programs that are in the style of so-called genetic algorithm, they're taking a photograph of high school biology, and they're spending a long time building programs based on that naive idea.

But that naive idea has lots of places for intervention, because look at all the things you can screw around with in that process of going from one generation to the next.

By the way, what does mutation do?

It's basically hill climbing, right?

It's producing a little spread out, and you're using the fitness thing to climb the hill.

So you get a lot of choices about how you handle that.

Then you get a lot of choices about much crossover you're doing.

What does crossover do?

It kind of combines strong features of multiple individuals into one individual, maybe.

So you've got all kinds of choices there.

And then your genotype to phenotype translation-- how do you interpret something like those zeroes and ones as an if-then rule, for example, as something that's in the hands of the designer?

Then you've got all the rest of those things, all which are left up to the designer.

So in the end, you really have to ask-- when you see an impressive demonstration, you have to say, where does the credit lie?

And I mean that pun intentionally, because usually the people who are claiming the credit are lying about where it's coming from.

But nevertheless, let me give you a couple of examples of where this has found actual, bona fide practical application.

So when you look for practical application, you might say, well, in what kind of problem does a good front piece combine with a good back piece to produce a good thing overall?

And the answer is, when you're making a plan.

So you might have a problem in planning that requires you to take a series of steps.

And you might have two plans, each of which is a series of steps.

And you might combine these to produce something new that's the front half of one and the back half of another.

So that's practical application number one.

And that requires you to interpret your chromosome as an indicator of the steps in the plan.

Another example is drawn from a [? UROP ?] project a student did for me some years ago.

He was a freshman.

He came to me and said, I want to do a [? UROP ?] project.

And I said, have you taken 6.034?

And he said no.

And I said, go away.

And he said, I don't want to go away.

I want to do a [? UROP ?] project.

So I said, have you read my book?

He said no.

I said, well, go away, then.

And he said, I don't want to go away.

I want to do a UROP project.

So I said, I don't have any [? UROP ?] projects.

He said, that's OK.

I've got my own.

He's a finance-type guy, so he was interested in whether he could build a rule-based expert system that could predict the winners at horse races.

So his rule-based expert system consisted of rules like this.

If x and y, then some conclusion.

If l and m, then some kind of conclusion.

And from these, he would produce rules like if x prime-- that's a slightly mutated version of the x antecedent-- and m, then some conclusion.

So it's mutation and crossover.

And he was able to produce a system that seemed to work about as well as the handicappers in the newspaper.

So he started losing money at a less fast rate.

He is now doing something in the stock market, they say.

Doesn't talk so much about it, though.

But an interesting application.

He came up with rules like, if the sum of the jockey's weight on the post position is low, that's good.

Well, that makes sense in the end, because the jockey's weight is always between 100 and 110 pounds, and the post position is always between 1 and 10 or something, so they were commensurate values.

And a low one is good, in fact.

Not bad.

But neither of those-- I mean, this is real stuff.

My company uses this sort of stuff to do some planning work.

But neither of those is as impressive as the demonstration I'm about to show you that involves the evolution of creatures.

And these creatures consist of block-like objects, like so.

And they combine like this, and so on.

And so how can you make a feature like that from a [? 0-1 ?] chromosome?

Well, some of the bits in the chromosome are interpreted as the number of objects.

Others are interpreted as the sizes of the objects.

Others are interpreted as the structure of how the objects are articulated.

And still others are interpreted as fixing the control algorithm by which the creature operates.

So you see how that roughly goes?

Would you like to see a film of that in action?

Yes.

OK.

[INAUDIBLE] always likes to see the films.

STUDENT: How would you measure diversity in that graph?

PATRICK WINSTON: The question is, how do I measure the diversity of the graph?

I did it the same way I measured the fitness.

That is to say, I calculated the distance-- the actual metric distance-- of all the candidates for the next generation from all of the candidates that had already been selected.

I summed that up.

And from that sum, I could rank them according to how different they were from the individuals that were already in the next generation.

It's like giving a rank, and then from the rank, I use that kind of calculation to determine a fitness, ie, a probability of survival, and then I just combine the two kinds of probabilities.

STUDENT: So you always kept-- every time that you started something, [? you cached ?] those.

And you kept everything that you've ever [INAUDIBLE].

PATRICK WINSTON: I'm always using the individuals that have already been selected at every step, so every step is a little different because it's working with a new set of individuals that have already been selected for the next generation.

OK?

So let's see how this works.

So this is showing the evolution of some swimming creatures.

And they're evolved according to how well they can swim, how fast they can go.

Some of them have quite exotic mechanisms, and some of them quite natural.

That looked like a sperm cell floating away there.

Once you have these things evolving, then of course, you can get groups of them to evolve together.

So you saw already some that were evolving to swim.

These are evolving to move around on the land.

It's interesting-- this was done by Karl Sims, who at the time was at a then-thriving company, Thinking Machines, a fresh spinoff from MIT.

So he was using a vastly parallel computer, super powerful for its day, thousands of processors, to do this.

And it was a demonstration of what you could do with lots of computing.

In the early stages of the experimentation, though, its notion of physics wasn't quite complete, so some of the creatures evolved to move by hitting themselves in the chest and not knowing about the conservation of momentum.

I thought that was just great.

So here they are, out doing some further-- So you look at these, and you say, wow, there must be something to this.

This is interesting.

These are complicated.

I think this is one of the ones that was trained, initially, to swim and then to do land locomotion.

So eventually, Karl got around to thinking about how to make these things evolve so that they would compete for food.

That's the fastest, I think, by the way, of the land locomotors.

So that was training them to-- evolving them to jump.

This is evolving them to follow a little red dot.

Some of them have stumbled upon quite exotic methods, as you can see.

Seem to be flailing around, but somehow manage to-- sort of like watching people take a quiz.

Making progress on it.

But now we're on to the food competition.

So some of them go for the food, and some of them go to excluding their opponent from the food, not actually caring too much about whether they get it.

That's sort of what children do.

There's a kind of hockey player-- now, here's two hockey players.

Watch this.

They're kind of-- one succeeds-- it reminds me a little bit of hockey, rugby, something like that.

Sometimes, they just get kind of confused, go right after their opponent, forgetting about the food.

Gives up.

I think these are the overall winners in this elimination contest.

I can't quite get there.

OK, so you look at that, and you say, wow, that's cool.

Genetic algorithms must be the way to go.

I remember the first time I saw this film.

It was over in Kresge.

I was walking out of the auditorium with Toma Poggio And we looked at each other, and we said the same thing simultaneously.

We didn't say that genetic algorithms were the way to go.

What we said was, wow, that space is rich in solutions.

What we were amazed by was not that simple-minded genetic algorithms produced solutions but that the space was so rich with solutions that almost any mechanism that was looking around in that space would find them.

But there's yet another way of thinking about it, and that is you could say, wow, look at how smart Karl Sims is, because Karl Sims is the one who had his hands on all the levers, all those choices.

And I kept emphasizing, all those choices that enabled him to trick this thing, in some sense, into stumbling across the solutions in a space that was guaranteed to be rich with solutions.

So you have to ask-- so first of all, diversity is good.

We noticed when we put diversity into the genetic algorithm calculations, we were much better at finding solutions.

But the next gold star idea that I'd really like to have you go away with is the idea that you have to ask where the credit lies.

Does it lie with the ingenuity of the programmer or with the value of the algorithm itself?

In this case, impressive as it is, the credit lies in the richness of the space and in the intelligence of the programmer, not necessarily in the idea of genetic algorithms.