UNCOMMON DESCENT
Serving the Intelligent Design Community

- Home
- About
- Resources
- Search

« Hunter Baker's THE END OF...
Cambrian Explosion Caught on Film... »

26 August 2009

Uncommon Descent Contest Question 10: Provide the Code for Dawkins’ WEASEL Program

O'Leary

Special invitation for Richard Dawkins – but any civil person is entitled to enter.

There’s been some discussion here and elsewhere whether the the recent IEEE article by Dembski and Marks correctly characterizes Richard Dawkins’ famous METHINKS IT IS LIKE A WEASEL program.

Does the program ratchet correct letters or does it let them vary? One is a partitioned or stair-step search, the other a more realistic evolutionary search. From *The Blind Watchmaker*, where Dawkins describes the program, its performance suggests that it could be either of these options (though he doesn’t say).

On the other hand, from a (video-run of the program , go to 6:15), it seems to be the latter.

It’s easy enough to settle this question: Make the code for the program public. Perhaps Richard Dawkins himself or his friends at RichardDawkins.net can finally provide this code (apparently a program written in BASIC).

The prize is a copy of either Stephen Meyer’s new Signature in the Cell or Richard Dawkins’ soon-to-be-out The Greatest Show on Earth.

Should the winner choose the latter, I will ask Dawkins’s publicist to mail the copy. Given that at his site, he calls himself “the most formidable intellect in public discourse,” I would assume that if he signs the copy, it will be worth millions.

But wait. Let’s see that code first.

This entry was posted Wednesday, August 26th, 2009 at 8:46 am and is filed under Darwinism. You can skip to the end and leave a response. Pinging is currently not allowed.

277 Responses

1

kibitzer

08/26/2009

9:10 am

Scientists, when they write computer programs and make scientific claims based on their performance, are supposed to make the code available. Thomas Schneider has done this with EV. Christof Adami has done this with AVIDA. It is simply unconscionable that over 20 years after the program has been out and used to argue for Darwinism, Dawkins still has not made this code publicly available.

2

Atom

08/26/2009

9:22 am

kibitzer,

I’ve heard it reported that there was a time when the code was in fact available, but I can’t verify the veracity of this claim. We’ll see if it turns up.

3

KRiS_Censored

08/26/2009

9:32 am

A single Google search has turned up 3 different listings of source code for the “Weasel” program written in 3 different languages, 1 of which includes both a locking and a non-locking mechanism built into it for easy comparison. All three show up on the very first page of results. Are you really having difficulty finding any source code for the Weasel algorithm? Or are you trying to pretend that the algorithm that he describes is completely irrelevant and it’s his original code written by him and him alone that’s really important?

4

kibitzer

08/26/2009

9:36 am

Atom,

I’m not sure I’m being too hard on Dawkins. Granted, The Blind Watchmaker and his WEASEL program came out in the 80s before easy dissemination of the program code on the Internet would have been available. But the program has been much discussed on the Internet in the last decade. So where is the code?

5

kibitzer

08/26/2009

9:40 am

KRiS_Censored,

Three different languages? Were those all written by Dawkins? What was Dawkins’ original code that he cited in The Blind Watchmaker. You make it sound as though the original code is unimportant. It’s the original code on which his inflated claims are based. Let’s see it.

6

specs

08/26/2009

9:44 am

So, you want Dawkins to provide you with a 23 year old computer program that has probably been reproduced hundreds, if not thousands of times, in a variety of programming languages, including by one regular contributor to this site (Atom). Have I got that right?

7

Indium

08/26/2009

9:58 am

I have written a few programs in the nineties. Some even for university stuff. I have no copy left. I guess I lost a bit everytime I changed to a new OS or Computer.

Anyway, the algorithm as described in the BW is enough to reproduce and many people have done so. This includes Atom from this very site (”Proximity Reward” algorithm from the tool on the EIL site).

8

Indium

08/26/2009

10:00 am

Oh, and btw, the BW is not presenting original scientific research in this case. Weasel is a pet algorithm developed for a popular audience almost 25 years ago – it´s scientific relevance is more or less zero.

9

kibitzer

08/26/2009

10:39 am

Indium,

“It’s scientific relevance is more or less zero.” Since Dawkins used it to support evolution, are you saying that evolution is unscientific?

10

DiEb

08/26/2009

10:52 am

Dawkins describes his algorithm in the following way:

> It again begins by choosing a random sequence of 28 letters, just as before:
>
> WDLTMNLT DTJBKWIRZREZLMQCO P
>
> It now ‘breeds from’ this random phrase. It duplicates it repeatedly, but with a certain chance of random error – ‘mutation’ – in the copying. The computer examines the mutant nonsense phrases, the ‘progeny’ of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL. [...] the procedure is repeated.

This is enough to replicate his program:

> 1. chose random string
> 2. copy string n times with mutation. **NOTE**: at this step you don’t know which letters are correct in place, so no letter is safe from being mutated!
> 3. chose best fitting string. **NOTE**: best fitting seems to be generally understood to be the string with the most correct letters, the fitting is expressed by a number between 1 and 28
> 4. Stop, if the number of correct letters is 28, otherwise
> 5. goto 2

The parameters which you can chose is the number of copies n, and the probability that a letter in a string is mutated, p. You may even chose another procedure of mutation – but keep in mind the **NOTE** of step 2.

It’s really basic to realize steps 1 – 5 in the programming language of your choice…

11

KRiS_Censored

08/26/2009

10:53 am

kibitzer

> You make it sound as though the original code is unimportant. It’s the original code on which his inflated claims are based.

Apparently you are in fact “trying to pretend that the algorithm that he describes is completely irrelevant and it’s his original code written by him and him alone that’s really important.”

As a matter of fact it’s the original *methodology* (or in computer speak “algorithm”) upon which his claims are based. The code itself was simply a way of demonstrating that methodology and is completely irrelevant.

Let me ask you this, if he re-wrote the program today and it was exactly the same in every possible way, except that one variable was named “y” instead of “x”, would that program be unusable? The *methodology* employed by the program is identical in either case, yet the *code* used to do it is different. How about if he changed a “for” loop into a “while” loop? Again, the final program would be identical in it’s methodology, but the code would be different.

It’s the methodology that’s of real import. The specific code used to implement it is irrelevant.

12

DiEb

08/26/2009

10:58 am

a number between **0** and 28, of course…

13

Indium

08/26/2009

10:59 am

kibitzer: No. Evolution does not need Weasel as a support. It was a simple model to educate a broad audience about the power of cumulative selection 25 years ago. I don´t really understand your logic there.

BTW, above I wanted to write “…the algorithm as described in the BW is easy enough to reproduce …”.

14

kibitzer

08/26/2009

11:01 am

Indium,

“… is enough to reproduce …” I assume you mean “… is EASY enough to reproduce …” But what do you mean by “reproduce”? Do you mean, (1) Come up with the exact program that Dawkins wrote? Obviously not. (2) Come up with a functionally equivalent program? But that’s the question, isn’t it, whether the program ratchets letters or doesn’t? Without the original code it’s not clear whether a given piece of code is functionally equivalent? Or do you mean (3) Come up with any old program that’s close enough? In that case, I assume you’re okay with Dembski characterizing Dawkins’ algorithm as ratcheting correct letters. No, you say? Dembski was wrong in that characterization? Then provide the original code. Repeat after me: WE WANT TO SEE THE ORIGINAL CODE, WE WANT TO SEE THE ORIGINAL CODE, WE WANT TO SEE THE ORIGINAL CODE …

15

lars

08/26/2009

11:02 am

@Kris_censored:

> Or are you trying to pretend that the algorithm that he describes is completely irrelevant and it’s his original code written by him and him alone that’s really important?

The operative question here is, what algorithm did Dawkins use? That question has never been definitively answered, yet arguments for evolution have been made based on the program's performance. The program's relevance as a demo of what evolution-like processes can do depends on what algorithm was used.
Having a public copy of the source code would allow everyone to verify what algorithm was used.

If he's lost the source code, he should say so. If he has it, he should publish it.

I don't think the Weasel program is now important as direct evidence for/against evolution, since more recent simulation programs have gone into much more detail and the algorithms have been published. However, the Weasel program is a familiar example to many people, and thus a useful reference point (which is why Dembski and Marks talk about it). Significant questions about the accuracy and relevance of Weasel-based arguments for evolution could be answered if the algorithm(s?) were published.

But possibly Dawkins and his allies don't like the answers.

16

kibitzer

08/26/2009

11:08 am

KRiS_Censored,

Ah, it's the METHOLODOLOGY that's important. And how do you figure out the methodology without the original code. Are you clairvoyant?

17

DeLurker

08/26/2009

11:40 am

lars#15

> The operative question here is, what algorithm did Dawkins use? That question has never been definitively answered,

Actually, it has. For example, a step by step implementation of the algorithm described in *The Blind Watchmaker* is available here:

http://www.softwarematters.org/more-weasel.html

A report of Dawkins stating that his program does not fix correct letters in place is here:

http://austringer.net/wp/index.....r-sweater/

A video of the algorithm running, showing that letters are not fixed, is here:

http://www.youtube.com/watch?v=5sUQIpFajsg

All of this information, and more, has been posted on UD repeatedly every time the Weasel algorithm comes up.

> yet arguments for evolution have been made based on the program's performance.

You've got it backwards, I'm afraid. The Weasel algorithm was used to demonstrate the power of mutation and selection over random search, in a popular science book published 23 years ago. It was used to *describe* one component of evolutionary theory, not to *support* evolutionary theory.

> The program's relevance as a demo of what evolution-like processes can do depends on what algorithm was used.

Indeed. If letters were fixed in place once correct, that would not be at all relevant to evolutionary theory.

18

DeLurker

08/26/2009

11:42 am

kibitzer#16

> Ah, it's the METHOLODOLOGY that's important. And how do you figure out the methodology without the original code.

By following the very clear description in *The Blind Watchmaker*, of course. Many people have done so successfully.

19

kibitzer

08/26/2009

11:53 am

DeLurker,

Of course, as programs go, Dawkins' WEASEL is trivial and it's easy enough to reconstruct something that's close to it. But given the controversy surrounding it, let's see the original program. Why is that so difficult?

20

DeLurker

08/26/2009

12:04 pm

kibitzer#19

> Of course, as programs go, Dawkins' WEASEL is trivial and it's easy enough to reconstruct something that's close to it. But given the controversy surrounding it, let's see the original program. Why is that so difficult?

The program was originally written in Apple BASIC over twenty years ago. Neither the language interpreter nor the computer itself have been available for some time. I suspect that, even if Dawkins has a copy, there are no disk readers still available for that format.

The algorithm is very clearly described in *The Blind Watchmaker*. The only people I know of who have misunderstood it are Royal Truman, Dembski, and Marks. The only "controversy" I have ever seen over this simple, well documented algorithm is here on UD.

Please read the full description of the Weasel algorithm in TBW and explain how anyone could come to the conclusion that letters are fixed once correct.

21

DNA_Jock

08/26/2009

12:11 pm

It only evidence to support the idea that the BW weasel has a latching mechanism, is that it *appears* to latch.
Many people have subsequently shown that this **behavior** is readily achieved by algorithms that do have any latching **mechanism**.

Now if anyone wants to argue that Dembski and Marks are accurate in their characterization of the BW weasel, they need to explain the observed difference in **behavior** of the BW weasel and the partitioned search that D&M describe.

22

DNA_Jock

08/26/2009

12:14 pm

Arrgh , English, that should be “The only evidence” and “algorithms that do **NOT** have any latching mechanism.
Sorry

23

kibitzer

08/26/2009

12:17 pm

DeLurker,

You wrote: “Please read the full description of the Weasel algorithm in TBW and explain how anyone could come to the conclusion that letters are fixed once correct.”

Glad you asked. Look at the output of the program in The Blind Watchmaker, and you find that once a letter corresponds to the target sequence, it doesn’t change. And sorry, the book didn’t refer to the subsequent 1987 video in which letters do change after they match the target.

So the question remains, did Dawkins write the program so that it ratchets correct letters or not. I’ll grant you, probably not. But Dawkins’ throughout The Blind Watchmaker prefers “cumulative selection” to “natural selection.” Cumulative suggests ratcheting. So Dembski’s interpretation is hardly as far-fetched as you make out.

24

DeLurker

08/26/2009

12:23 pm

kibitzer#21

> You wrote: “Please read the full description of the Weasel algorithm in TBW and explain how anyone could come to the conclusion that letters are fixed once correct.”
>
> Glad you asked. Look at the output of the program in The Blind Watchmaker, and you find that once a letter corresponds to the target sequence, it doesn’t change.

You are not taking into consideration the full description of the algorithm. Looking at the most fit daughter strings every 10 generations is obviously very unlikely to show reversions.

Please show how the full, clear, detailed explanation of the Weasel algorithm can possibly be interpreted to even suggest that letters are fixed once correct.

25

R0b

08/26/2009

1:24 pm

kibitzer:

> But Dawkins' throughout The Blind Watchmaker prefers "cumulative selection" to "natural selection." Cumulative suggests ratcheting. So Dembski's interpretation is hardly as far-fetched as you make out.

"Cumulative" and "natural" are orthogonal. In nature, selection is both cumulative and natural. In WEASEL, it's cumulative but not natural.

As Dave Wisker mentioned in another thread, Dawkins explicitly stated what he means by cumulative selection, and it has nothing to do with latching. From TBW page 45:

> The essential difference between single-step selection and cumulative selection is this. In single-step selection the entities selected or sorted, pebbles or whateverthay are, are sorted once and for all. In cumulative slelection, on the other hand, they 'reproduce'; or in some other way the results of one sieving process are fed into a subsequent sieving, which is fed into into…and so on. The entities are subjected to selection or sorting over many 'generations' in succession. The end-product of one genration of selection is the starting point for the next generation of selection, and so on for many generations.

And Dembski's interpretation, which he has maintained throughout the years, mutates every unlatched letter and involves no selection. This contradicts both the description of the algorithm and the output reported in TBW, so I would call it not only far-fetched, but clearly wrong.

26

Indium

08/26/2009

1:32 pm

kibitzer, may I ask, what is your evidence FOR a latching weasel? The output tables of resulting strings?

27

tsmith

08/26/2009

1:32 pm

I always find it interesting when a darwinist uses INTELLIGENT DESIGN to try to prove some component of evolution….but here's my stab at it…

```
printf ("METHINKS IT IS LIKE A WEASEL ");
```

28

tsmith

08/26/2009

1:34 pm

do I win a prize????

29

R0b

08/26/2009

1:53 pm

Denyse:

> From The Blind Watchmaker, where Dawkins describes the program, its performance suggests that it could be either of these options (though he doesn't say).

Then why does Dembski continue to say that it's the former?

Dawkins also doesn't say whether WEASEL includes code that steals my credit card numbers. Until he provides the code, it's a toss-up. Should I cancel my credit cards?

30

kibitzer

08/26/2009

1:58 pm

Ladies and Gentlemen,

We're all beating our gums. Please, let's see the original code. Why is that so much to ask? To paraphrase Ben Stein, Does anyone have it? Anyone?

31

R0b

08/26/2009

2:29 pm

kibitzer, none of us have it. Have you asked Dawkins?

32

Indium

08/26/2009

2:29 pm

kibitzer, I guess you may have missed my question. If you don´t mind: What is your evidence FOR a latching weasel? The output tables of resulting strings?

33

fmarotta

08/26/2009

2:31 pm

Why does the original code matter?

The fact is algorithms which use latching and non-latching have been coded by others and are easily found on the web. Do these programs prove anything of substance relative to evolution or intelligent design? I tend to think not.

34

DNA_Jock

08/26/2009

2:34 pm

I have tried to reconcile the **performance** seen in TBW with the no-selection-partitioned-search described by D&M.
D&M explicitly state that the mutation rate for correct letters is 0%, and they state the number of progeny = 1; they do not, however, *explicitly* state that the mutation rate (for unlatched letters) has to be 100% (I know, I am being generous in my reading of "so we take 26 new letters"). So I tried to adjust the mutation rate such that a partitioned search would perform in a manner consistent with the performance seen in TWB.
Err, it cannot be done. Forget about latching, you cannot have a partitioned search that changes only one element in its first iteration, and reaches a solution within 43…

35

DeLurker

08/26/2009

2:34 pm

kibitzer#30

> We're all beating our gums. Please, let's see the original code.

As I noted above in #20, it's entirely possible that the code hasn't survived changes in technology of the past 23 years.

Dawkins' very lucid description of the algorithm, however, has survived. Please explain how that description can be logically construed to involve explicit fixing of letters in place once they are correct.

36

Spiny Norman

08/26/2009

2:50 pm

I can't speak for Dawkins of course, so I am in the position of offering external criticism. Like many here (presumably), I have been a programmer for several decades. The trivial nature of the Weasel "demonstration" is what irks me. It seems clear to me that the algorithm misses the mark in what it apparently seeks to show. I'd therefore like to propose an alternative, which might be closer to the mark in the sense of demonstrating natural selection from an evolutionary perspective.

My first suggestion is that the number of mutations should be set to a value that is consistent with what we actually know about the frequency of mutations (i.e. the likelihood that any one letter would be mutated in any generation).

My second suggestion revolves around the idea of using a dictionary (electronic of course). As the letters are 'mutated' in the text, sections of the text should be compared to the dictionary to see if they are a functional word. For example, if the string RIKE gets mutated to BIKE or to HIKE then that is now a functional bit of text and will be highly conserved. The possibility that it will be converted to LIKE (part of Dawkins' preferred outcome) is now greatly reduced.

These modifications would seem to go some of the way towards addressing criticisms of Dawkins' algorithm, however the downside from his p.o.v. is (in my estimation) that the algorithm will now be extremely unlikely to produce "METHINKS IT IS LIKE A WEASEL" and so of course his purpose in introducing it in the first place, which is apparently to hide from the general public the very real probability problems that evolutionary theory must overcome, would be frustrated … and why would he want to demonstrate the very point he is trying to hide in the first place?

37

steveO

08/26/2009

2:59 pm

Sorry, a bit OT.

Here's Dawkins near the end of the "weasel" chapter – referring to the other simulation. Oh the excitement! the drama!

> When I wrote the program, I never thought that it would evolve anything more than a variety of tree-like shapes… Nothing in my biologist's intuition, nothing in my 20 years' experience of programming computers, and nothing in my wildest dreams, prepared me for what actually emerged on the screen. I can't remember exactly when in the sequence it first began to dawn on me that an evolved resemblance to something like an insect was possible. With a wild surmise, I began to breed, generation after generation, from whichever child looked most like an insect. My incredulity grew in parallel with the evolving resemblance… I still
> cannot conceal from you my feeling of exultation as I first watched these exquisite creatures emerging before my eyes. I distinctly heard
> the triumphal opening chords of Also spiach Zaiathustia (the '2001
> theme') in my mind. I couldn't eat, and that night 'my' insects' swarmed behind my eyelids as I tried to sleep.

Of course, in the excitement of the action, it would be easy to miss the man behind the curtains operating the buttons and levers!

38

HouseStreetRoom

08/26/2009

3:14 pm

tsmith @27,

I know, right?

I think I may be able to improve upon your code some:

```
int main()
{
printf (”METHINKS IT IS LIKE A WEASEL “);
}
```

Look, I couldn't resist.

39

kairosfocus

08/26/2009

3:39 pm

Okay:

Mrs O'Leary, I can point to the EIL weasel programs and I do believe the software is available. But alas that is by Atom, and it is associated with Dembski not Dawkins.

I can also point out that in 1986, the o/p of Weasel was showcased as cumulatively marching to the target, and with samples that in 200 of 300 places, and in EVERY case where a reversion was a priori possible, we never saw ONE reversion. Such a significant sample suggests that the o/p effectively ratchets its way to the target, ansd that the successful letters are therefore latched.

The real issue is HOW, and the answers are that there are two ways: explicitly, and implicitly.

The latter working by virtue of there being a match of pop per generation, mutation rate per letter ansd fulter to select even the slightest advance to the target. For under relevant cases, no-change members will be by overwhelming odds present, and so if 1 change members are frequent, no-change and one step increments will dominate in such a way as to preserve currently correct letters.

Nor is this theory, there are actual runs that do that. (HT: Atom and EIL.)

Now, in 1987, the video runs most definiteoly do not have the latching action that is credible for 1986. This is suggestive of detuning of parameters and possibly a different filter.

That should not even be controversial, but since the ratcheting-latching issue highlights the targetted search that lies at the heart of Weasel, and this in turn turns it into a "misleading" –Dawkins' word! — example, this has become a focus for red herrings and strawmen, soaked in ad hominems and ignited.

Sad. But sadly revealing.

GEM of TKI

40

BillB

08/26/2009

3:46 pm

O'Leary:

How will you tell if the code is genuine, given that it will be just pasted into this discussion?

Should I just write out the algorithm Dawkins describes in Apple Basic and submit it? It would produce the results he published.

41

DeLurker

08/26/2009

3:56 pm

kairosfocus#39

> I can also point out that in 1986, the o/p of Weasel was showcased as cumulatively marching to the target, and with samples that in 200 of 300 places, and in EVERY case where a reversion was a priori possible, we never saw ONE reversion.

That is because the few strings shown in *The Blind Watchmaker* are the most fit daughter strings from every tenth generation.

You're doing your math incorrectly. Dawkins' Weasel algorithm, unlike that used by Dembski and Marks in their recent IEEE paper, uses a relatively low mutation rate across all letters in the string and population size much greater than 1. Taking a typical population size of 200, the number of strings generated in 43 generations (the time required to converge to the target string in TBW), there were 8600 strings generated. Of those, only 4 were shown in the book.

In his second run, the target was found in 64 generations, for a total of 12,800 strings, of which only 6 were shown.

You are considering only approximately 0.047 percent of the strings in both cases. The likelihood of seeing a reversion is, of course, very low.

Since the output shown in TBW clearly cannot be used to claim that correct letters are locked in place once found, please explain based on Dawkins' description of the algorithm how you can possibly come to the conclusion that they are.

42

kibitzer

08/26/2009

4:40 pm

BillB,

I'm sure we would all trust if Richard D. himself assured us that the code in question was his original. Also, like the forged Air National Guard documents surrounding George Bush in "Rathergate," it will probably be easy enough to tell if the code was written recently.

43

Anthony09

08/26/2009

5:01 pm

After the "Accidental Origin of Life" contest question, I think the newxt few questions would be:

1. Why are their still apes, if we evolved from apes and evolution is true?

2. Why don't dogs ever give birth to cats if evolution is true?

But this latest one is almost as good.

44

Anthony09

08/26/2009

5:02 pm

Whoops, haste makes waste. I meant:

"I thought the next few questions would be:"

45

Oatmeal Stout

08/26/2009

5:03 pm

Programs implement algorithms, and algorithms are not cooked up after the fact to describe what programs happen to do.

Those of you familiar with commercial software development might think in terms of “That’s not a bug, it’s a feature.”

Those of you familiar with Peewee Herman might think in terms of “I meant to do that.”

Something I forgot to mention is the convenience of turning Dawkin’s algorithm into the algorithm Dembski and Marks analyze. This makes them look so bad. It is much easier to come up with the active information of the D&M algorithm than the Dawkins algorithm.

If we ID proponents want to gain credibility, we are going to have to exhibit exemplary scholarship, not look for excuses for what we do. Demanding program code when the algorithm is the issue is really shabby excuse-making.

46

kibitzer

08/26/2009

5:08 pm

Anthony09,

In place of your list, let’s try

1. Why are there still Darwinists given the pathetic state of their theory?

2. Do Darwinists reproduce in the ordinary way or by parasitizing the wider public and by siphoning their tax dollars?

47

BillB

08/26/2009

5:10 pm

> it will probably be easy enough to tell if the code was written recently.

Not really, so long as the person writing it knows how to write in apple basic (which I don’t) and doesn’t use variable names that relate to current events or trends then there is no way of telling. Typically variable names in a programme like this would be obvious things like ‘Population’ and this would be consistent with code written then, or yesterday.

48

DeLurker

08/26/2009

5:17 pm

kibitzer#45

Let’s try this instead (from my #20):
“Please read the full description of the Weasel algorithm in TBW and explain how anyone could come to the conclusion that letters are fixed once correct.”

You have yet to address this core issue.

49

jerry

08/26/2009

5:39 pm

The amazing thing is that all agree that the WEASEL program is meaningless for anything and the concept of cumulative selection is meaningless in the scheme of evolution other than the trivial. But here we are debating this nonsense like the fate of the world depended on it. For one reason only. Whether Bill Dembski interpreted the original code correctly. There is this intense obsession by the anti ID people to impugn Bill Dembski in any way they can but in the process they end up revealing how intellectually barren they are.

The anti ID people essentially admit they have nothing of substance to say. So have at it with these feeble complaints. Every comment just makes it easier for the pro ID people.

50

O'Leary

08/26/2009

5:55 pm

BillB at 40:

No use asking me.

Everything I know about computers, I learned from WordPerfect 5.0 20 years ago, and have since updated to WordPerfect 12.

But lots of cats here ARE computer whizzes, and will sure know if Dawkins is funnin' them.

You may wish to consider pasting Dawkins's original code onto a Web page, in case you have problems with length restrictions or other problems in our combox.

Ours is a good system but was not intended to accommodate long strings of code, so I can't answer for what will happen if you try to do it that way.

51

Mapou

08/26/2009

6:35 pm

As a programmer, I can tell you that Dawkins' weasel algorithm is useless as an argument for evolution. It is just a trick to impress fools because it assumes an end goal. Intelligent design is written all over it.

Darwinian evolution, on the other end, has no end goal. It is dead on arrival.

52

Joseph

08/26/2009

7:13 pm

DeLurker:

> "Please read the full description of the Weasel algorithm in TBW and explain how anyone could come to the conclusion that letters are fixed once correct."

Using the description of *cumulative selection* along with the illustration using the weasel program (both in TBW), the only inference is one of a ratcheting process.

Otherwise he should have chosen a better term like “lost, found, lost, found again, then lost, and found again selection”

Or “back and forth selection”.

Or the “whatever bloody survives selection”.

Or better yet “badda-bing, badda-boom selection”.

But that wouldn’t have misled the populace into thinking the blind watchmaker is something that “he” isn’t.

53

Joseph

08/26/2009

7:17 pm

Also the program is evidence for ID as ID can be reduced to nothing more than a targeted search- along with the resources required to reach that target.

54

wrf3

08/26/2009

7:18 pm

I guess I have to be counted among the group that wonders why having the original code is so important. Both latching and non-latching algorithms can be coded, the various parameters (such as mutation rate) set, and the algorithms analyzed.

As a historical note, there is code that I wrote 20+ years ago that I no longer have, but wish I did. Neither the listings, nor the punch tape, survived.

IMO, this is much ado about nothing. Is there something I didn’t understand?

55

yakky d

08/26/2009

7:37 pm

Joseph,

> Also the program is evidence for ID as ID can be reduced to nothing more than a targeted search- along with the resources required to reach that target.

This makes no sense. Is Guitar Hero also evidence for ID?

56

R0b

08/26/2009

8:18 pm

kairosfocus:

> Now, in 1987, the video runs most definiteoly do not have the latching action that is credible for 1986. This is suggestive of detuning of parameters and possibly a different filter.

Keeping in mind that the video shows all candidates, not just the winners, do you see any correct letters being lost from one generation to the next? How can you tell?

57

DeLurker

08/26/2009

9:02 pm

Joseph#50

> Using the description of cumulative selection along with the illustration using the weasel program (both in TBW), the only inference is one of a ratcheting process.

Like kibitzer and kairosfocus, you are not addressing the full description of the Weasel algorithm as clearly and cogently explained by Dawkins in *The Blind Watchmaker*. Picking one term (”cumulative selection”) out of a two page explanation is unconvincing in the extreme.

Please refer to http://www.softwarematters.org/more-weasel.html for an example of how to go through the details of the Weasel algorithm step by step to create an implementation. There is no way that I can see to come to the conclusion that the mutation operator has any knowledge of the target string. If you can logically defend that proposition, please do so.

58

random coder

08/26/2009

9:29 pm

Any semi-competent programmer can implement Dawkin’s weasel algorithm.

Here’s my version in Common Lisp – the language God used to create the universe – I’m pretty sure it behaves the same as Dawkins’ original.

I used SBCL to run it, but any Common Lisp implementation should do.

Note that in this version, you can make the target string be anything you want, just change it in the first defparameter. You can also change the mutation rate & population size to see how they affect the evolution of the string.

Note that the copy-mutate function has no knowledge of the target string.

```
;;;;;;;;;;;;;;;;
(defparameter target-string “methinks it is like a weasel”)
(defparameter population-size 10000)
(defparameter p-mutation 0.001)

(defun count-matches (x y)
(length (loop for n from 0 to (1- (length x))
when (eql (aref x n) (aref y n))
collect n)))
```

```
(defun copy-mutate (x mutation)
(let ((cp (make-string (length x))))
(loop for n from 0 to (1- (length x)) do
(setf (aref cp n) (if ( (setf new-score
(count-matches (setf child (copy-mutate x mutation)) target))
score)
(format *standard-output* ” Best match so far (this generation) == ~D : |~D|~%” new-score child)
(setf x child score new-score)))
(cons x score)))

(defun random-string (len)
(let ((str (make-string len)))
(loop for n from 1 to (1- len) do
(setf (aref str n) (code-char (+ 32 (random 95)))))
str))

(defun weasel (&key (target target-string)
(pop-size population-size)
(mutation p-mutation))
(format *standard-output* “Population size: ~D~%P(Mutation): ~D~%Target: |~D|~%” pop-size mutation target)
(let (gen matches (gen-count 0) (leader (random-string (length target))))
(format *standard-output* “Initial (random) string: |~D|~%” leader)
(setf matches (count-matches target leader))
(loop while (< matches (length target)) do
(setf gen (best-match leader pop-size mutation target))
(setf leader (car gen) matches (cdr gen))
(format *standard-output* "Best generation ~D match == ~D : |~D|~%" (incf gen-count) matches leader))))
```

59

random coder

08/26/2009

9:50 pm

This comment system doesn’t seem to know how to handle greater-than / less-than characters, so let’s try encoding them. You can also email me, and I’ll be happy to provide this as a text file:

```
;;;;;;;;;;;;;;;

(defparameter target-string “methinks it is like a weasel”)
(defparameter population-size 10000)
(defparameter p-mutation 0.001)

(defun count-matches (x y)
(length (loop for n from 0 to (1- (length x))
when (eql (aref x n) (aref y n))
collect n)))

(defun copy-mutate (x mutation)
(let ((cp (make-string (length x))))
(loop for n from 0 to (1- (length x)) do
(setf (aref cp n) (if (< (random 1.0) mutation)
(code-char (+ 32 (random 95)))
(aref x n))))
cp))

(defun best-match (x pop-size mutation target)
(let (child new-score (score 0))
(loop for n from 1 to pop-size do
(when (> (setf new-score
(count-matches
(setf child (copy-mutate x mutation)) target))
score)
(format *standard-output*
” Best match so far (this generation) == ~D : |~D|~%”
new-score child)
(setf x child score new-score)))
(cons x score)))

(defun random-string (len)
(let ((str (make-string len)))
(loop for n from 1 to (1- len) do
(setf (aref str n) (code-char (+ 32 (random 95)))))
str))

(defun weasel (&key (target target-string)
(pop-size population-size)
(mutation p-mutation))
(format *standard-output*
“Population size: ~D~%P(Mutation): ~D~%Target: |~D|~%”
pop-size mutation target)
(let (gen matches (gen-count 0) (leader (random-string (length target))))
(format *standard-output* “Initial (random) string: |~D|~%” leader)
(setf matches (count-matches target leader))
(loop while (< matches (length target)) do
(setf gen (best-match leader pop-size mutation target))
(setf leader (car gen) matches (cdr gen))
(format *standard-output*
“Best generation ~D match == ~D : |~D|~%”
(incf gen-count) matches leader))))
```

60

mike1962

08/26/2009

10:07 pm

OP, why is anyone hung up on Dawkin’s weasle programs? It’s ONLY purpose was to demonstrate stepwise build up of information. It’s was not meant as complete analogy of evolution. As Dawkin’s said in Blind Watchmaker:

“Although the [WEASEL] model is useful for explaining the distinction between single-step selection and cumulative selection, it is misleading in important ways. One of these is that, in each generation of selective ‘breeding’, the mutant ‘progeny’ phrases were judged according to the criterion of resemblance to a distant ideal target, the phrase METHINKS IT IS LIKE A WEASEL. Life isn’t like that. Evolution has no long-term goal. There is no long-distance target, no final perfection to serve as a criterion for selection, although human vanity cherishes the absurd notion that our species is the final goal of evolution. In real life, the criterion for selection is always short-term, either simple survival or, more generally, reproductive success.”

61

Indium

08/27/2009

1:05 am

mike, I know why us “Darwinists” are so hung up with it: Because Dembski for some reason continues to misrepresent it. This fact is, btw, independend of the latching issue at hand. Dembski describes a partitioned search: No population, no selection, 100% mutation rate of incorrect letters. Dawkins describes a GA: He selects strings from a population using a fitness criterion. Incorrect letters have a biologically at least more or less realistic mutation rate <10%.
Also, it is fun to see kf and others come up with rethoric herring oil soaked smokescreens like "implicit latching"!

62

djmullen

08/27/2009

1:44 am

denyse, kairosfocus and reader from riesel, are you aware that you’re using the word “ratchet” instead of “latching”?

Darwinian evolution does “ratchet” information into the DNA. It’s been described that way by scientists for decades.

It does this ratcheting through the simple technique of making many copies of successful DNA strings and letting natural selection get rid of any unsuccessful mutations to those strings – such as restoring a former incorrect letter.

This is EXACTLY what happens in Dawkins’ program or any other program that successfully mimics evolution. That is why Dawkins did not have to put any kind of latching into his program – the latching / ratcheting is inherent in Darwinian evolution and his program merely simulates one part of it.

Spiny Norman: Dawkins wrote “Weasel” as a pedagogical tool to demonstrate how the cumulative selection that is used by Darwinian evolution is almost infinitely faster than the type of “all-at-once” selection that creationists and IDists typically use. (You know, where they calculate that it would take 20^100 tries to find a 100 amino acid length protein by chance or 4^150 tries to find a 150 base pair long stretch of DNA by chance.) Because it was a teaching tool, he selected a specific target for it to find rather than confuse the issue by cobbling together some sort of moving target.

However, as I’ve written on this blog, if you re-write the program to look into an external file for the “target”, you can change that target whenever you wish and the program will continue to find the new strings just as quickly as it finds the fixed “Methinks it is like a weasel”.

63

feebish

08/27/2009

1:54 am

I don’t understand why so many here are arguing against letting Dr Dawkins release his program. Or I should say “programS.” Inspired by the many posts about this Weasel idea, I stopped at the library on my way home from work and picked up a copy of The Blind Watchmaker. It turns out that the Weasel program was written not only in Basic, but also in Pascal (see page 49).
These would be fairly small programs, so it would not surprise me if Dr Dawkins printed them out when he first made them, as I used to do when programming in Fortan during that same time period.

Finally, many thanks to kibitzer for advocating so strongly for the position of openness in these matters. You are not afraid to mix it up with people who don’t share your views. I dare say you could not be advocating this position better if you were Dr Dembski himself.

Now, let’s see BOTH these original programs, to get to the bottom of these issues of demi-ratcheting, cumulative selection, and so forth.

64

BillB

08/27/2009

2:31 am

Joseph:

> Using the description of cumulative selection along with the illustration using the weasel program (both in TBW), the only inference is one of a ratcheting process.
>
> Otherwise he should have chosen a better term like “lost, found, lost, found again, then lost, and found again selection”

Or he could have use cumulative, as in cumulative height, cumulative wealth…

If he wanted to describe a letter locking mechanism he could have just said that once a letter is found the search for it is over.

65

BillB

08/27/2009

3:02 am

O’Leary:

> But lots of cats here ARE computer whizzes, and will sure know if Dawkins is funnin’ them.

Ok, so a follow up question.

Lets say Dawkins or anyone else posts some code, in the correct language, that produces the correct behaviour.

If the code does not contain a latching mechanism, includes a population of more than one, and a mutation rate not fixed at 0 or 100, that applies to all letters … Will the judges here conclude that it is the correct code, or will they cite these differences between it and Dembski and Mark’s algorithm as evidence that it is a fake?

I have a strong feeling that some here would.

66

kairosfocus

08/27/2009

4:03 am

Onlookers:

First, the bottomline: after many months of exchanges, it is clear that –absent a near-miracle — no credible Weasel code c 1986 [and yes, per CRD's statements in BW, the basic version ran across a lunchtime and the Pascal one did so in 11 seconds] will be forthcoming.

So, what we have to deal with is a minor (but instructive) bit of real-world scientific analysis, as I have documented here, months ago now. That is, we have to look at empirical evidence that is not complete — here, published showcased samples of runs of generational champions c 1986, and descriptions — and come up with a reasonable best explanation, using replication of results as a good cross-check, e.g through Atom's adjustable Weasel, here.

Also, it seems that the exchanges aptly model the patterns of contention and conflict that often happen at the cusp of scientific revolutions, as the old order reacts with cognitive dissonance — and, often, rage and associated idea hit-man rhetorical tactics designed to restore the old order and defeat the new by "any means necessary" — to that which threatens to overturn their comfortable world.

Let us note:

1 –> The printoffs of sampled, showcased generational champions c 1986 show altogether 300+ letters, 200+ of which are of letters that were initially correct in the seed or which have subsequently gone correct.

2 –> IN EVERY INSTANCE WHERE A LETTER GOES CORRECT IN ANY ONE GENERATION, IT REMAINS SO IN ALL FURTHER SAMPLES UNTIL THE PROGRAM HITS THE FULL TARGET.

3 –> On law of large no's [the correct form of the layman's crude "law of averages"], that strongly supports the inference that the samples do not revert because the generational champions preserve correct letters very strongly.

4 –> Examination of Dawkins' remarks on the program (and Joseph and I are by now tired of the dozens of times we have had to point out what should be obvious facts of basic reading of direct statements in rather plain English] support that observation:

> [Weasel] . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. *The computer examines **the mutant nonsense phrases**, the 'progeny' of the original phrase, and **chooses** the one which, **however slightly, most resembles the target*** phrase, METHINKS IT IS LIKE A WEASEL . . . . What matters is the difference between the time taken by **cumulative selection**, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of **single-step selection**

5 –> Weasel is targetted search that rewards mere proximity of non-functional phrases through a process of random variation of a seed to create a population and artificial selection based on mere proximity to a set target.

6 –> Weasel's gain in performance over random search is therefore directly — though inadvertently — acknowledged to be due to ACTIVE INFORMATION. (Thus, the relevance of Dembski's and Marks' analysis.)

7 –> And, the failure to use realistic degree of functionality as a threshold of preservation — note the dismissal of "single step selection" begs the question of first getting TO shores of function in large config spaces before climbing to peaks of performance.

8 –> In short, the root issue is being distracted from and misrepresented then dismissed. No wonder Weasel is "misleading."

9 –> However, we still have to explain the behaviour of the run of generational champions as evidenced by teh showcased samples c 1986.

10 –> CRD speaks of selecting slightest increments to target, which by virtue of the digital, alphanumeric character system, is one letter.

11 –> Similarly, he speaks of "cumulative selection."

12 –> Cumulative, as a word, means: "Increasing or enlarging by successive addition." The increment in question plainly being that of proximity to target, per the selection of the greatest relative advance to target for generational champions.

13 –> Thus, Weasel ratchets towards the target. Ratcheting, being:" To cause to increase or decrease by increments." (Implicit in that is the dogging action of a pawl, whereby progress is one-way.)

14 –> Thus also, a ratcheted advance latches progress so far. Latching being: "To close or lock with or as if with a latch." (In this case, correct letters of the generational champions in the showcased runs c 1986, credibly appear to latch.)

15 –> Such ratcheting-latching cumulative progress to target of course first and foremost is NOT a de novo creation of complex functional information. For, the information is preloaded through the presence of a target and a cumulative progressive mechanism that step by step advances to an already present target.

16 –> This latching action can be fairly simply achieved EXPLICITLY by using a masking filter:

a] say, let letter correct = 1, letter incorrect = 0. Then with perfect match we have 28 as a metric, and with perfect mismatch, we have 0.

b] So, we compare members of the mutated population of a given generation with the target, assigning to each one a metric string, with 1's and 0's to match correct or incorrect letters.

c] closest to target becomes new champion, and the 1's act as mask bits that block further mutation of correct letters. (After all, they are perfectly functional and would be preserved by "natural selection.")

d] Each new generation will at least preserve what has been achieved and the march of generations will cumulatively ratchet their way to the target.

17 –> The result can also be achieved — for "good runs" — IMPLICITLY. In this case:

e] when population size and per letter mutation rate are set so that with very high probability there will be no change members of the population, and single changes will prevail otherwise, e.g. 4% rate, 50 – 100 member population seems a plausible set of parameters — there will be a strong tendency to preserve progress to darte and/or to reward single step advances that do not corrupt already correct letters.

f] with a suitable filter, the resulting trend will be to choose generational champions that are either the same or single step advances.

g]the showcased 1986 runs hit target in 40+ and 60+ generations, supporting that about half the time, no-change was the generational champion.

h] similarly, single step advances predominated the rest.

i] but with other parameter values, various effects uch as substitutions of correct and incorrect letters, or reversions will occur.

j] As already noted and linked, these patterns have been demonstrated through actual runs of Atom's adjustable Weasel, from EIL.

18 –> Under other circumstances, weasel will show quasi-latched behaviour with relatively rare reversions on correct letters [similar to how anglers who use baitcasters will know that a ratchet and pawl may begin to slip if the dog is worn].

19 –> under yet other circumstances, far from latched behaviour will occur.

20 –> It has recetnly been suggested that the 1987 run is simply showing all members of population, and is in a condition where champions will show fairly frequent reversion. this can be accounted for by a case of a Weasel that is detuned sufficiently for that to happen.

21 –> But, the 1986 runs of champions — here, taking into reckoning CRD's claim that he never wrote an explicitly latched weasel — would very easily be explained as "good" runs of a tuned set of parameters, so that we see latching action.

22 –> Now also there has been much digital ink spilled over how Marks and Dembski in their recvent paper have distorted the weasel relative to CRD's program.

23 –> however, on p. 1055, they simply describe, exemplify and analyse a partitioned search. Whether such partitioning of runs of generational champions into the active search subset of the string and the locked in subset is achieved explicitly or implicitly makes no material difference tot he analysis of how many generations on average it takes to hit target or the associated probabilities associated with the injection of active information.

24 –> And, no, M & D do not provide an algorithm for a weasel, just an analysis of the sort of run of champions showcased in BW,

1986, that shows latched behaviour based on targeted search; to deduce the effect of active information by comparison with the yardstick, random walk search.

25 –> As CRD remarked in BW and as was cited above, targetted search's injection of active information makes a HUGE difference, one that M & D provide a quantitative analysis for and thence a metric of active information injected.

26 –> And the relevance of active information to Intelligent Design is that it gives us a mechanism for explaining the action and impact of intelligence.

27 –> Intelligent search works batter than random search [or a search algorithm picked at random from a large pool of diverse algorithms] because the active information coming from knowledge and creative imagination sharply restricts the scope of search to a zone much more likely to be close to success than a blind process.

28 –> Going back tot he flooded fitness function landscape model, we can also se that an intelligent search that is not successful can cotribute to success. (For, if the bottom of the sea of non-function slopes up towards the beach of function, partial success and trends of partial success can guide further intelligent search by sending warmer-colder signals and trends, i.e increments in active information through a beacon.)

29 –> In fact, Weasel is an inadvertent example of this! (Non-functional nonsense phrases progress to target through gradual warmer-colder signals emitted by an algorithm that has a distance to target/success metric.)

++++++++++++

Now, onlookers, let us observe and analyse how the Darwinist critics of the latching analysis respond to the above.

GEM of TKI

67

Laughable

08/27/2009

4:12 am

I don't really understand why it is so important to get Dawkin's original code – even if Dawkins never wrote any code, and just imagined up the algorithm, the fact that it has been reproduced by many different people and operates exactly according to his description would indicate that the idea behind it and the principles involved in it are correct.

It's like not believing that aeroplanes exist until you see the original blueprints for the Wright Brother's aeroplane. It has been done, its been reproduced, it works.

Atom's code at http://www.evoinfo.org/Resourc.....elGUI.html
for the "proximity reward search" is a faithful reproduction of Dawkins code – there isn't any latching (run a few trials and occasionally you'll see a correct letter being dropped).

What else do you need? Dawkins described an algorithm (without latching), and Atom and many others have programmed an algorithm that behaves exactly how he said it would. Even if the code didn't exist when Dawkins wrote his book, it does now, so the points he was making in regards to the search are valid.

(note that the "partitioned search" is an incorrect model of Dawkin's search).

68

kairosfocus

08/27/2009

4:40 am

PS: Atom's source code is available, through the linked GUI page, here.

69

Indium

08/27/2009

6:41 am

kf, what you write boils down to “Just from the output we can´t decide if latching was used, but we will assume it was anyway. Without any proof or any mentioning of latching in the explanation if Weasel in the BW we will assume Dawkins lies and the video from 1987 is a different and more or less faked algorithm. We have to admit that the algorithm is good enough to preserve good letters, as a smokescreen we call that implicit latching.”.
The question is not if Weasel latches implicitly, we know that it does (for certain parameters) and that´s exactly the demonstration of cumulative selection.
The question is wether Weasel latches explicitly or, put in another way, if it treats correct and incorrect letters differently during the mutaion routine. It doesn´t. It doesn´t need to. Dawkins said it doesn´t. A video that shows the algoritthm running with different parameters shows it doesn´t. Mutations are supposed to be independend of resulting fitness, so there would be no reason to let it latch. When it is decided which “gene” mutates, the algorithm does not know which letters are correct. The orakel, as D+M call it, does not return the exact correct letter indices, it just returns a total fitness value (as it should).

From Atoms software suite, which algorithm do you think is representative of Dawkins´ Weasel and which one corresponds to the algorithm used by D+M in their paper?

I further note that you still do not adress the fundamental differences:
DM don´t use a population, they don´t use selection. Weasel uses both. It has to, otherwise it would not work. A fundamental difference if I have ever seen one!

70

kairosfocus

08/27/2009

7:09 am

Indium:

Clearly you have a problem with evidence-based reasoning.

Here is the showcased 1986 o/p|:

```
___________

>> 1 WDL*MNLT*DTJBKWIRZREZLMQCO*P
2? WDLTMNLT*DTJBSWIRZREZLMQCO*P
10 MDLDMNLS*ITJISWHRZREZ*MECS*P
20 MELDINLS*IT*ISWPRKE*Z*WECSEL
30 METHINGS*IT*ISWLIKE*B*WECSEL
40 METHINKS*IT*IS*LIKE*I*WEASEL
43 METHINKS*IT*IS*LIKE*A*WEASEL

1 Y*YVMQKZPFJXWVHGLAWFVCHQXYPY
10 Y*YVMQKSPFTXWSHLIKEFV*HQYSPY
20 YETHINKSPITXISHLIKEFA*WQYSEY
30 METHINKS*IT*ISSLIKE*A*WEFSEY
40 METHINKS*IT*ISBLIKE*A*WEASES
50 METHINKS*IT*ISJLIKE*A*WEASEO
60 METHINKS*IT*IS*LIKE*A*WEASEP
64 METHINKS*IT*IS*LIKE*A*WEASEL >>

_______________
```

1] Kindly find a case of reversion in the 200+ possible cases in the runs of champions above, where such could happen; or else discuss their absence cogently in light of the law of large numbers.

2] Kindly compare CRD's remarks on how the program cumulatively progresses to target (as already excerpted), demonstratin how reading the remarks as meaninfg what they say is not correct.

Otherwise, you are simply "bravely" whistling by the graveyard in the dark; while the duppies leaning on the fence are looking on and shaking their heads.

GEM of TKI

71

Indium

08/27/2009

7:21 am

kf, the absence of reversing correct letters is easily explained by what you call implicit latching*, eg the power of random mutation + selection. This is exactly what was supposed to be demonstrated by Weasel and is verified by what you say in your point number 17. Non reversing letters are no proof of anything. Since this is the only argument *for* latching you have you really have no argument at all.

Since you still evade some important issues it is probably a good idea to remind you of them:

From Atoms software suite, which algorithm do you think is representative of Dawkins´ Weasel and which one corresponds to the algorithm used by D+M in their paper?

Main differences between the Partitioned Search and Weasel:
D+M don´t use a population, they don´t use selection. Weasel uses both. It has to, otherwise it would not work. A fundamental difference if I have ever seen one!

* I hesitate to use your "implicit" smokescreen, because incorrect letters are not protected from mutations when generating the next population, so there is no latching at all, just a successfull evolutionary search.

72

kairosfocus

08/27/2009

7:45 am

PS: Here is Simon Greenleaf, a founding father of the modern theory of evidence, on the error of the skeptic, which I have descriptively tagged, selective hyperskepticism:

>> [26] . . . It should be observed that the subject of inquiry [i.e. evidence relating to the credibility of the New Testament accounts, but also generally applicable to matters of fact on the balance of evidence] is a matter of fact, and not of abstract mathematical proof. The latter alone is susceptible of that high degree of proof, usually termed demonstration, which excludes the possibility of error . . . *In the ordinary affairs of life we do not require nor expect demonstrative evidence, because it is inconsistent with the nature of matters of fact, and to insist on its production would be unreasonable and absurd* . . . **The error of the skeptic consists in pretending or supposing that there is a difference in the nature of things to be proved; and in demanding demonstrative evidence concerning things which are not susceptible of any other than moral evidence alone, and of which the utmost that can be said is, that there is no reasonable doubt about their truth** . . . .

[27] . . . . In proceeding to weigh the evidence of any proposition of fact, the previous question to be determined is, when may it be said to be proved? The answer to this question is furnished by another rule of municipal law, which may be thus stated:

A proposition of fact is proved, when its truth is established by competent and satisfactory evidence.

By competent evidence, is meant such as the nature of the thing to be proved requires; and by satisfactory evidence, is meant that amount of proof, which ordinarily satisfies an unprejudiced mind, beyond any reasonable doubt. . . . . If, therefore, the subject is a problem in mathematics, its truth is to be shown by the certainty of demonstrative evidence. But if it is a question of fact in human affairs, nothing more than moral evidence can be required, for this is the best evidence which, from the nature of the case, is attainable. Now as the facts, stated in Scripture History, are not of the former kind, but are cognizable by the senses, they may be said to be proved when they are established by that kind and degree of evidence which, as we have just observed, would, in the affairs of human life, satisfy the mind and conscience of a common man. [Testimony, Sections 26, 27, emphases added.] >>

———————————

Selective hyperskepticism reveals itself — and simultaneously reduces itself to absurdity — by the fact of inconsistency and unreasonableness in demand for evidence on matters of fact.

In this case, no-one would even be pretending that the o/p of Weasel 1986 as showcased did not show a latched, ratcheting pattern and/or that both explicit and implicit latching has been demonstrated [with code accessible for inspection!], apart from other extraneous considerations that are irrelevant to the weight/balance of evidence sufficient to decide an ordinary, unprejudiced mind.

Or, the man in the Clapham Bus Stop if you will.

73

tsmith

08/27/2009

7:56 am

> 38
> HouseStreetRoom

LOL

74

DNA_Jock

08/27/2009

8:00 am

kf, you are correct in that Weasel is a GA that exhibits latching **behaviour**, which can be achieved without any latching **mechanism**, for the reasons you outline in your point 17 and acknowledge when you say “samples do not revert because the generational champions preserve correct letters very strongly.” (#3) Isn’t it cool?
You are also correct when you say (#23) ” on p. 1055, they[D&M] simply describe, exemplify and analyse a partitioned search. ”
The **ONLY** issue in this particular round of Weasel Wars is that D&M mis-characterize Weasel, as described in TBW, and that this mischaracterization has been previously pointed out to them.
It is obvious from the **behavior** of Weasel in TBW that it is NOT a partitioned search as described by D&M. A high school student can see this. Why then did the paper cite TBW as the source of the partitioned search?

75

kairosfocus

08/27/2009

8:00 am

Indium:

1] You now concede that IMPLICIT latching works. Implicit latching is real.

2] You then attribute it — in the face of the explicitly tagetted search based on reward of mere proximity of “nonsense phrases” to the target — to “the power of random mutation + **selection**”

3] Onlookers, observe, **not** “NATURAL selection.” A telling omission indeed. In short, Indium cannot bring her-/him-self to explicitly acknowledge that the selection in Weasel is **artificial** and intelligent based on active information, and is not parallel to what is said of natural selection. (CRD concedes this in some “fine print.”)

4] Indium, had you simply bothered to read the always linked app 7, s/he would have seen that on the balance of evidence and in particular a reported claim by CRD c 2000, the best explanation is that the evident latching is IMPLICIT.

5] You then proceed to discuss irrelevancies. D & M use an illustration of what partitioning is, and provide an analysis that will apply to any march of generational champions that exhibits latching. That is, partitioning into the on-target and active search zones, however achieved, will have the same relevant mathematical characteristics. This I have already pointed out.

6] In short, the analysis in the paper is independent of the algorithm, so long as ratcheting action shows up. Partitioning is a cognate of ratcheting and associated latching in a targetted search.

GEM of TKI

76

DeLurker

08/27/2009

8:01 am

kairosfocus#66

> So, what we have to deal with is a minor (but instructive) bit of real-world scientific analysis, as I have documented here, months ago now. That is, we have to look at empirical evidence that is not complete — here, published showcased samples of runs of generational champions c 1986, and descriptions — and come up with a reasonable best explanation, using replication of results as a good cross-check, e.g through Atom’s adjustable Weasel, here.

No, that’s not what we have to do. What we have to do is read Dawkins’ very clear explanation of the Weasel algorithm as written in *The Blind Watchmaker*. Here it is:

> So much for single-step selection of random variation. What about cumulative selection; how much more effective should this be? Very very much more effective, perhaps more so than we at first realize, although it is almost obvious when we reflect further. We again use our computer monkey, but with a crucial difference in its program. It again begins by choosing a random sequence of 28 letters, just as before:
>
> WDLMNLT DTJBKWIRZREZLMQCO P
> It now ‘breeds from’ this random phrase. It duplicates it repeatedly, but with a certain chance of random error – ‘mutation’ – in the copying. The computer examines the mutant nonsense phrases, the ‘progeny’ of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL. In this instance the winning phrase of the next ‘generation’ happened to be:
>
> WDLTMNLT DTJBSWIRZREZLMQCO P
> Not an obvious improvement! But the procedure is repeated, again mutant ‘progeny’ are ‘bred from’ the phrase, and a new ‘winner’ is chosen. This goes on, generation after generation. After 10 generations, the phrase chosen for ‘breeding’ was:
>
> MDLDMNLS ITpSWHRZREZ MECS P
> After 20 generations it was:
>
> MELDINLS IT ISWPRKE Z WECSEL
> By now, the eye of faith fancies that it can see a resemblance to the target phrase. By 30 generations there can be no doubt:
>
> METHINGS IT ISWLIKE B WECSEL
> Generation 40 takes us to within one letter of the target:
>
> METHINKS IT IS LIKE I WEASEL
> And the target was finally reached in generation 43.

There is no rational reading of that description that could possibly suggest that letters are fixed in place once correct. All daughter strings are created by mutation of the parent and the mutation mechanism has no knowledge of the fitness of the resulting string.

If you disagree, show how Dawkins' own words can be construed otherwise.

77

kairosfocus

08/27/2009

8:06 am

PS: The search used for the printoffs I have given in earlier threads is the obvious one: the proximity reward search algorithm. And this is the one that is probably closest in concept to — certainly thee 1987 algor, and with parameters co-tuned to give latching, probably the 1986 ones.

PPS: An explicitly latched algor can also use a population of mutants in each generation and pick the closest to target, but that will make but little difference other than somewhat speeding up run to target. (Cases where a gen size of 1 is used are simplifications for convenience; but such should now be avoided as they will become targets for idea hit-man rhetoric.)

78

DeLurker

08/27/2009

8:06 am

kairosfocus#69
You have apparently missed my explanation in #41 of how you are doing your math incorrectly. As I point out in that post, you are considering only approximately 0.047 percent of all the strings generated by Dawkins' Weasel algorithm. It is not surprising that you don't see a letter reversion.

You need to read the entire description of the algorithm instead of referring to just a few sample lines of output.

79

Joseph

08/27/2009

8:08 am

DeLurker,

I am addressing the full description made by Dawkins and illustrated by Dawkins in "The Blind Watchmaker".

Again if he didn't mean "cumulative" he should not have used that word.

And his illustration of the process using WEASEL there is only ONE inference to be made- it is a ratcheting process.

OTOH you have not provided anything from the book that would change nor challenge that inference.

80

Joseph

08/27/2009

8:10 am

Also the program is evidence for ID as ID can be reduced to nothing more than a targeted search- along with the resources required to reach that target.

yakky d:

> This makes no sense.

It makes perfect sense if you understand ID.

So what do you know of ID and how did you come to that?

81

Joseph

08/27/2009

8:11 am

DeLurker,

The weasel program is only concerned with the output.

Nothing else is relevant.

82

kairosfocus

08/27/2009

8:12 am

DL:

Simply read the excerpt above and the runs as shown.

The runs as shown together with the remarks on cumulative selection make latching and ratcheting a very reasonable and straightforward understanding of CRD's words.

Focus on the 1986 printoffs and the meaning of CUMULATIVE in its context.

the general rule of hermeneutics is to take the plain meaning in context as the meaning, save where it makes no sense.

And, plainly the obvious meaning — cumulative, ratcheting, latching action as Weasel moves to target — makes a lot of sense.

GEM of TKI.

83

DeLurker

08/27/2009

8:12 am

Joseph#70

> I am addressing the full description made by Dawkins and illustrated by Dawkins in "The Blind Watchmaker".

No, you are continuing to focus solely on the word “cumulative” and are applying your own, very restricted, interpretation of what that word means.

When climbing a hill, one cumulatively increases one’s altitude, even though an individual step may decrease it temporarily. With a genetic algorithm, one cumulatively increases fitness, even though one generation may be less fit than the previous generation due to the vagaries of the random number generator.

I posted the full description of the Weasel algorithm in #75. Please show how that could possibly be interpreted, in context, as supporting the idea that letters are protected from mutation once correct.

84

Indium

08/27/2009

8:13 am

kf, I will happily admit that the algorithm is good at preserving correct letters most of the time. Call that implicit latching if you must, but this is just a distraction on your part.

Since it is an artificial algorithm this is artificial slection. Or natural selection if you call CPU, harddisk and RAM the natural environment of algorithms! 😉

Since you admit, that the latching is implicit the answer to the original question of the thread “Does the program let correct letters vary?” is YES! Most of the time these variations do not show up because of the selection routine, but they are still mutated.

Now, this is sorted then.

The next question then is if D+M misrepresent Dawkins. Since they use explicit latching instead of implicit, since they use no population and no selection the answer is also “YES!”.
I suppose that the active info concept can be applied to the real Weasel, and Random Mutation / Selection are indeed discussed in the paper! But discussing Weasel using a different algorithm where the only similarity is that it also preserves correct letters most of the time is a blatant misrepresentation.

I ask again: Which algorithm from Atoms suite corresponds to Weasel and which one corresponds tpo the Partitioned Search discussed by D+M?

85

DeLurker

08/27/2009

8:17 am

kairosfocus#81

> The runs as shown together with the remarks on cumulative selection make latching and ratcheting a very reasonable and straightforward understanding of CRD’s words.

Here are Dawkins’ actual words, again:

> It now ‘breeds from’ this random phrase. It duplicates it repeatedly, but with a certain chance of random error – ‘mutation’ – in the copying.

Note that there is no suggestion of mutating only incorrect letters. It is a very simple statement.
Dawkins continues:

> The computer examines the mutant nonsense phrases, the ‘progeny’ of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL.

The selection mechanism is the only step of the algorithm that refers to the target phrase. The mutation mechanism does not. There is no explicit preservation of correct letters.
It is very clear from a straightforward reading of Dawkins' own words that Dembski and Marks have misrepresented the Weasel algorithm in their paper.

86

Joseph

08/27/2009

8:17 am

BillB:

> Or he could have use cumulative, as in cumulative height, cumulative wealth…

Without qualifying the word there is no reason to think that is how he used it.

IOW if he is using a word in a way it is not generally accepted then it is up to him to qualify it.

He didn't so there is no reason to think he did.

> If he wanted to describe a letter locking mechanism he could have just said that once a letter is found the search for it is over.

How he describes the difference between a random search and cumulative selection, especially given the weasel illustration, that is teh only inference to make.

IOW there wasn't any need to come right out and say it.

And taken in context what others say about **mutations**:

> A mutation is a permanent change in the DNA sequence of a gene.

There just isn't any other inference to be had.

87

Joseph

08/27/2009

8:22 am

DeLurker,

You don't get to tell me what I am doing.

Also I am using the word "cumulative" how it is defined- the standard and accepted definition.

If Dawkins is using it any differently than the standard and accepted definition it is up to him to explain that.

As I said if Dawkins wanted people to know the reality behind evolution he would not have used the word "cumulative" and he wouldn't have used a program in which the output is always equal to or greater than the input.

IOW there isn't anything in TBW that supports what you are saying.

Nothing- not one word.

If there were you would have posted it by now.

88

kairosfocus

08/27/2009

8:22 am

PS: DL, I am focussing on the *showcased* runs c. 1986 [as you cirted but seem to have missed the sifgnificancfe of in 41], which is what is to be explained as factual data. That data contains 200+ letters whose values could vary from correct a priori, but do not. (If you look up the demonstrated runs starting here, you will see cases where such reversions with detuned parameters are possible, and how long they tend to persist. If reversions were at all common in the relevant runs, reversions should have been easily observable in a sample of 200+ characters: taking Np = 0.1, and N = 200, p = 5 * 10^-4.)

89

Joseph

08/27/2009

8:23 am

DeLurker,

There doesn't have to be any explicit code for latching.

Latching/ ratcheting occurs as a matter of course given a small enough mutation rate and a large enough population size.

What part of that don't you understand?

90

DeLurker

08/27/2009

8:24 am

Joseph#85

> Without qualifying the word there is no reason to think that is how he used it.
>
> IOW if he is using a word in a way it is not generally accepted then it is up to him to qualify it.

The usage described by BillB is perfectly standard. It is your restricted definition that is not.
Hundreds of programmers around the world have implemented the Weasel algorithm in a wide variety of languages, based solely on the material in *The Blind Watchmaker*. Based on that, it appears that yours is the idiosyncratic interpretation, unsupported by the actual text.

91

DeLurker

08/27/2009

8:27 am

Joseph#86

> IOW there isn't anything in TBW that supports what you are saying.

> Nothing- not one word.
>
> If there were you would have posted it by now.

I have, as have others, several times. Again, in Dawkins' own words:

> It now 'breeds from' this random phrase. It duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying.

The algorithm clearly specifies mutation without respect to the target string. That is the obvious reading of the text and the most biologically realistic mechanism.

92

Joseph

08/27/2009

8:27 am

indium:

> D+M don´t use a population, they don´t use selection.

You have said that before but never substantiated that claim.

IOW that appears to be all in your head.

Ya see if they referenced TBW, which they did, then it is a safe bet that they used the same stuff- population and selection- that Dawkins used.

93

kairosfocus

08/27/2009

8:28 am

Indium:

Red herring.

the facts in evidence to be explained are the showcased runs of champions c 1986.

These on good empirical reasoning, show latching, and we have provided two mechanisms that can account for it.

FYI, the champions emerge not when we have mutated a seed 50 or so times, but when we have filtered for closest to target. Thus mut rate, pop size AND filter affect the production ocf champions.

After the ingredients are mixed, we see that under certain circumstances, already correct letters will be strongly preserved — and that under others, they are most emphatically not.

So, implicit latching is not a nullity.

GEM of TKI

94

Joseph

08/27/2009

8:28 am

DELurker,

I know the strings mutate.

That was never the question.

Latching/ ratcheting occurs as a matter of course given a small enough mutation rate and a large enough population size.

What part of that don't you understand?

95

DeLurker

08/27/2009

8:29 am

kairosfocus#87

> DL, I am focussing on the showcased runs c. 1986 [as you cirted but seem to have missed the sifgnificancfe of in 41], which is what is to be explained as factual data.

That is obviously what you are focusing on, to the exclusion of the clear description of the algorithm that contradicts your claims. Look at the description, not the small amount of sample output.

96

DeLurker

08/27/2009

8:32 am

Joseph#88

> There doesn't have to be any explicit code for latching.
>
> Latching/ ratcheting occurs as a matter of course given a small enough mutation rate and a large enough population size.

Fantastic! We're in agreement!
Since there is no explicit code for latching in Dawkins' Weasel algorithm, the IEEE paper by Dembski and Marks misrepresents it.

97

Joseph

08/27/2009

8:32 am

The usage by BillB may be standard but is also EXPLAINED.

Dawkins did not explain cumulative as a process that could also lose what it had.

Ya see saying that would noyt have helped his case.

But I understand your position needs to redefine things on the fly.

And I understand that commonly accepted definitions do not mean anything to you.

But then again that is why your position is nonsense.

98

Joseph

08/27/2009

8:34 am

DELurker:

> Since there is no explicit code for latching in Dawkins’ Weasel algorithm, the IEEE paper by Dembski and Marks misrepresents it.

Except that Marks/Dembski never stated the program enlists explicit code for latching.

However reading TBW and looking at the weasel illustration, a ratcheting process is easily inferred.

99

kairosfocus

08/27/2009

8:36 am

Joseph,

the only case of EIL providing an algorithm for Weasel is Atom’s adjustable weasel. (In the paper they are simply analysing the dynamics of ratcheted Weasel type programs; applicable to any latched version, implicit or explicit. But some strawman factories are running 24/7 at UD these days.)

When they provide such a case, we can observe that EIL provides not only source code but text boxes for you to put in your preferred pop size and mut rate!

Anybody prepared to bet that they did not do a LOT of simulation before sticking with what they know months ago was a controversial paragraph?

If someone is, I have some prime, hot — hot, hot, hot! — real estate on Chances Peak Montserrat to sell for a great price . . .

GEM of TKI

100

DeLurker

08/27/2009

8:39 am

Joseph#91

> Indium:
>
> > D+M don´t use a population, they don’t use selection.

> You have said that before but never substantiated that claim.
>
> IOW that appears to be all in your head.
>
> Ya see if they referenced TBW, which they did, then it is a safe bet that they used the same stuff- population and selection- that Dawkins used.

Read the Dembski and Marks paper, section E, Partitioned Search:

> Two of the letters {E,S} are in the correct position. They are shown in a bold font. In partitioned search, our search for these letters is finished. For the incorrect letters, we select 26 new letters and obtain
>
> OOT*DENGISEDESEHT*ERA*NETSIL. (21)
> Five new letters are found, bringing the cumulatie tally of discovered characters to {T,S,E,*,E,S,L} All seven characters are ratcheted into place. The 19 new letters are chosen, and the process is repeated until the entire target phrase is found.

No population, no selection. Indium's description is correct.

101

Joseph

08/27/2009

8:41 am

I read the paper.

Just because they do not explicitly say there is a population of size X and Y selection, doesn't mean they didn't use them both.

Do you have the code they used?

If you don't then you just don't know.

102

DeLurker

08/27/2009

8:42 am

Joseph#97

> However reading TBW and looking at the weasel illustration, a ratcheting process is easily inferred.

You keep claiming this, but you have yet to support your claim with reference to Dawkins' description of the algorithm in *The Blind Watchmaker*. The text is clear — mutation is independent of fitness. There is no ratcheting.

103

kairosfocus

08/27/2009

8:43 am

DL:

You are simply asserting that the world is as you wish it to be, not as it is.

Denial is not a river in Egypt.

Onlookers,

Observe CRD again, as already excerpted — and recall the showcased runs show evident latching and ratcheting to go with these remarks:

____

>> It . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. *The computer examines the mutant nonsense phrases, the 'progeny' of the original phrase, and chooses the one which, however slightly, most resembles the target phrase*, METHINKS IT IS LIKE A WEASEL . . . . What matters is the difference between the time taken by **cumulative** selection, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of single-step selection: about a million million million million million years. This is more than a million million million times as long as the universe has so far existed . . . . >>

____

Compare the plain vanilla defn of "cumulative: "Increasing or enlarging by successive addition." And you can refer above to the step by step discussion at 66. (One of the effects of objection waves, however irrelevant or distractive, is to bury substantial comments so that they are easily overlooked.)

QED.

Good day

GEM of TKI

104

Joseph

08/27/2009

8:44 am

KF,

Had Dawkins printed outputs that showed character reversals he would have screwed up many readers.

IOW people would have noticed the contradiction.

105

DeLurker

08/27/2009

8:44 am

Joseph#100

> I read the paper.
>
> Just because they do not explicitly say there is a population of size X and Y selection, doesn't mean they didn't use them both.
>
> Do you have the code they used?
>
> If you don't then you just don't know.

They describe a single string, not a population. They mutate every incorrect letter, rather than creating multiple daughter strings and selecting the most fit. No population, no selection. How can you read it otherwise?

106

kairosfocus

08/27/2009

8:45 am

PS: DL: "cumulative: "Increasing or enlarging by **successive addition.**"

107

Joseph

08/27/2009

8:48 am

DeLurker,

1- In TBW the weasel program never shows a reversal of charcters. Not one reversal is shown.

2- People reading the book would look up the word "cumulative" because "cumulative selection" is new (was new at the time).

3- Putting the two together- the definition of "cumulative", Dawkins description of slight improvements and the fact the illustration doesn't show any reversals, the inference is clear. Cumulative selection is a ratcheting process.

If it wasn't he should have been explicit in saying exactly how it works.

However had he been honest his point would have been lost.

108

DeLurker

08/27/2009

8:50 am

kairosfocus#102

> it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. The computer examines the mutant nonsense phrases, the 'progeny' of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL . . . . What matters is the difference between the time taken by cumulative selection, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of single-step selection

The ellipses you inserted actually span quite a bit of text. The discussion of cumulative selection is not part of the description of the algorithm, it is part of the description of the results.
The simple genetic algorithm described by Dawkins results in cumulative selection, without the mutation mechanism having any knowledge of the fitness function. That's the entire point of the example.

109

Joseph

08/27/2009

8:50 am

DeLurker,

Do you have their code or not?

I say since they referenced TBW they used the same stuff Dawkins used.

A partioned search would be a selection process- keep what matches and mutate the rest.

That is selection.

110

DeLurker

08/27/2009

8:56 am

Joseph#106

> 1- In TBW the weasel program never shows a reversal of charcters. Not one reversal is shown.

This has been addressed repeatedly by several people during the interminable discussions on this topic. See my #41 in this thread if you missed those explanations.

> 2- People reading the book would look up the word "cumulative" because "cumulative selection" is new (was new at the time).

People with average or better vocabularies would understand that "cumulative" does not mean "monotonically increasing." If Dawkins meant to say that, he has more than sufficient writing skill to do so.

> 3- Putting the two together- the definition of "cumulative", Dawkins description of slight improvements and the fact the illustration doesn't show any reversals, the inference is clear. Cumulative selection is a ratcheting process.

Putting two incorrect assumptions together and ignoring the clear description of the algorithm might lead to any number of incorrect conclusions.

> If it wasn't he should have been explicit in saying exactly how it works.

He was, but no writer can protect against deliberate misreading.

> However had he been honest his point would have been lost.

Unfounded accusations of dishonesty are not part of civil discourse.

111

DeLurker

08/27/2009

8:59 am

Joseph#108

> Do you have their code or not?

Why would their code implement a different algorithm than they describe in the paper? If you've read the paper, please show exactly where they describe a population and selection.

> I say since they referenced TBW they used the same stuff Dawkins used.

That is exactly the point under discussion. They describe a completely different algorithm, thereby misrepresenting Dawkins.

112

Indium

08/27/2009

9:27 am

Joseph,

so, you believe that D+M use a population and selection?
The wording in the paper is very clear:

> Two of the letters {E,S} are in the correct position. They are shown in a bold font. In partitioned search, our search for these letters is finished. **For the incorrect letters, we select 26 new letters and obtain**
>
> **OOT*DENGISEDESEHT*ERA*NETSIL. (21)**
> Five new letters are found, bringing the cumulatie tally of discovered characters to {T,S,E,*,E,S,L} All seven characters are ratcheted into place. The 19 new letters are chosen, and the process is repeated until the entire target phrase is found.

The article could not be clearer: NO population, no selection.

So, you now claim D+M use a differnet algorithm than the one they describe in the paper. Great idea, I love it!

113

Indium

08/27/2009

9:36 am

The math they use easily destroys your argument Joseph. Look at the formula for finding a letter after Q queries: This formula is only correct when there is no population but only single queries where every wrong letter is mutated.

114

DNA_Jock

08/27/2009

9:38 am

Joseph, thank you for making our point, viz. D&M’s Section E is misleading:
“I say since they referenced TBW they used the same stuff Dawkins used.”

And yet they did not.

“Do you have their code?”

No, I have equation (22) on p1055 which is mathematically correct ONLY IF the mutation rate is 100% and the population per generation is 1.
(You will notice that the formula does not include a term for either parameter…bit of a give-away)
Equation 22 can be easily adjusted to account for mutation rates other than 100%; but if you introduce multiple offspring in one generation, then the math is completely different. I believe D&M are aware of this fact.
And even with variations in the mutation rate, equation 22 cannot be made consistent with the observed behavior of Weasel in TBW.
Case closed.

115

DiEb

08/27/2009

10:00 am

I took the string
SCITAMROFN*IYRANOITULOVE*SAM
and calculated a next generation using Dawkins’s algorithms with populations of 10,50 and 100 – and mutation rates of .04, .05 and .1.
The tenth string in the list is the second generation given in the paper of Mark and Dembski. The differences with the first generation are in bold face:

1. SCITAMROFN*IYRANOI**E**ULOVE*SAM
2. SCITAMROFN*IYRANOITULO**G**E*SAM
3. **E**CITAMR**I***N*IY**Z**ANOITULOVE*SAM
4. SCITAMROFN*IYRANOITUL*****VE*SAM
5. SCITAMROFN*IYRANOITULOVE*S**E**M
6. SCITAM**O**O**L**N**O**IYRA**M**OITULOVE*S**E**M
7. SCITA**N**ROFN*IY**Y**ANOITULOVE*SAM
8. SCIT**I**MROFN***J**YRANOITULOVE*SAM
9. SCITAMROFN*I**C**R**H**NOIT**S**LO**W**E*SA**V**
10. **OOT*DENGISEDESEHT*ERA*N**E**T**S**IL**

Can anyone spot a difference in the design of the strings? Anyone? KF? Anyone?

116

seanbutnotheard

08/27/2009

10:11 am

Forgive me if I’m oversimplifying, but doesn’t the line (in the LISP version):

(defparameter target-string “methinks it is like a weasel”)

demonstrate that the algorithm is not a good model of natural selection? It seems more analogous to front-loading to me.

I think a more realistic test would be to see if a similar neo-Darwinian algorithm, with **no** target string, could produce a program that does anything useful — replicate itself, for example. How long would that take? What are the chances that it would complete the task within 10^150 CPU cycles?

To put it another way, I wonder if it makes any sense to entertain the possibility that computer viruses, with their self-replicating and self-modifying nature, evolved solely and spontaneously by random chance and necessity?

117

Indium

08/27/2009

10:24 am

seanbutnotheard,

you are right, natural evolution has no fixed target.

Weasel can be and has been critized because of this. But mostly by people who don´t understand its educational purpose.

Your example of a better genetic algorithm is a bit strange. Evolution works with mutating replicators, so the ability to reproduce is already there. Where did that come from? That´s a completely different question! -> Origin Of Life! Weasel does not address this of course.

118

DeLurker

08/27/2009

10:26 am

seanbutnotheard#114

> I think a more realistic test would be to see if a similar neo-Darwinian algorithm, with no target string, could produce a program that does anything useful

That wouldn’t be biologically realistic. While biological evolution has no ultimate goal, it does have immediate goals. Dawkins makes this clear in the same chapter in which he describes the Weasel algorithm:

> Evolution has no long-term goal. There is no long-distance target, no final perfection to serve as a criterion for selection, although human vanity cherishes the absurd notion that our species is the final goal of evolution. In real life, the criterion for selection is always short-term, either simple survival or, more generally, reproductive success.

That’s quite different from having no target at all.

119

R0b

08/27/2009

10:58 am

kairosfocus, a couple of items that you might want to stop ignoring at some point:

- There is no need to argue about what Dawkins means by cumulative selection, since he tells us explicitly. Webster does not trump Dawkins on the question of what Dawkins means.

- There is no evidence that the program in the 1987 video is any less “implicitly latched” than the 1986 program.

- There is no required “matching” of parameters in order to have “implicit latching”. The lower the mutation rate, the better the latching, regardless of the population. The higher the population, the better the latching, regardless of the mutation rate.

- I'm still waiting for you to show us how M&D's math applies to "implicit latching". What is Q?

120

seanbutnotheard

08/27/2009

11:52 am

@ Indium & DeLurker,

Thanks for the clarification, the purpose of the software makes a little more sense to me now; Its scope is limited.

I'll admit to not having read Dawkins as thoroughly as I should on the subject… which I will do before interjecting again.

121

R0b

08/27/2009

12:31 pm

kairosfocus:

> When they provide such a case, we can observe that EIL provides not only source code but text boxes for you to put in your preferred pop size and mut rate!

Yes, Atom covered all the bases in his code. But turn to the Weasel Ware page. Here we find a mathematical description of what is explicitly claimed to be Dr. Dawkins' search, but the math contradicts both the description and the output in TBW.

122

Tomato Addict

08/27/2009

3:13 pm

What's the big deal? This algorithm is so simple I can create a version of it in Excel, and I did: http://dreadtomatoaddiction.bl.....easel.html

It works too. No latching, no active information, just a crude random search.

123

kairosfocus

08/27/2009

3:50 pm

Rob (and others):

Re 119: Rob, kindly, read Eqn 22 p 1055, in its immediate context and tell me just what part of . . .

> Assuming uniformity, the probability of successfully identi-fying a speci?ed letter with sample replacement at least once in Q queries is 1 ? (1 ? 1/N)^Q, and the probability of identifying
> all L characters in **Q queries** is
>
> q = (1 ? (1 ? 1/N)^Q)^L (22)
>
> For the alternate search using purely random queries of the entire phrase, a sequence of L letters is chosen. The result is either a success and matches the target phrase, or does not. If there is no match, a completely new sequence of letters is chosen.

. . . it is that you do not understand, as a mathematically sophisticated person.

Again, once the run of generational champions takes on the cumulative progress, ratcheting-latching pattern [and cf the showcased runs of 1986 on that], it makes but little difference whether it is produced explicitly or implicitly.

And, it has been SHOWN that it can do so implicitly, and the mechanism involved has been explained several times, hinging on the very high odds of a no-change case being present once we have a sufficiently large po per generation, and with a sufficiently low odds of mutation that single letter changes are the dominant form for advances. And, the matching of parameters is relative to having these effects turn up.(Too much of the above critical commentary reminds me of Galileo’s opponents who refused to look through his telescope for themselves but were all too copious with criticisms.)

For the rest, please read no 66 supra and make reference to the onward linked, esp App 7 the always linked, which gives the background context for why there is a controversy since Dec last, and why the reality of implicit latching is important.

G’day again.

GEM of TKI

124

DNA_Jock

08/27/2009

4:26 pm

kf, the point is that eqn 22 does not (and cannot) take generational champions into account. The generation size is one. Given this fact alone, the algorithm described in section E is unrelated to the Weasel in TBW.
Latching, whether mechanistic (”explicit”, IYW) or behavioral (”implicit”, IYW), don’t enter into it.
D&M are wrong.

125

R0b

08/27/2009

4:50 pm

kairosfocus, so you’re saying that Q is the total number of queries, and not the number of generations. Thank you.

For WEASEL with a population of 200 and mutation rate of 5%, empirically the median number of generations is 45. Let’s see if the math agrees.

q = (1-(1-(1/N))^Q)^L
= (1-(1-(1/27))^(45*200))^28
= .9999999999something

But empirically it should be .5. Do you still hold that D&M’s math applies to algorithms that latch implicitly?

126

Tomato Addict

08/27/2009

4:55 pm

KF writes> And, it has been SHOWN that it can do so implicitly, and the mechanism involved has been explained several times, hinging on the very high odds of a no-change case being present once we have a sufficiently large po per generation, and with a sufficiently low odds of mutation that single letter changes are the dominant form for advances. …

This only makes sense under a scenario where the target string is known, and a very restricted view of the population of potential changes.
If we have the target string "METHINKS" and start with the random string "QWERTYUI", then a single mutation to "KWERTYUI" is no gain. It is also no loss. Retaining this mutation, suppose a second mutation happens to give us "KWERTYUH", also no gain and no loss.

Now suppose there is an alternate target "KENSMITH" which is equally viable. The second string "KWERTYUH" is effectively a double mutation towards "KENSMITH". If our search is restricted to "METHINKS" this gets us nowhere, but with no predetermined target will now continue uphill towards "KENSMITH" instead. Under a different randomization the search might have reached "METHINKS" instead, or perhaps it could branch and both targets would be found. Speciation, anyone?

The search never "sees" the target, only the hill. Forget the target entirely, the search is the thing.

127

wrf3

08/27/2009

5:51 pm

Through Google, I found Elsberry's Javascript implementation of the Weasel program. When I ran it, it sure looked like it was ratcheting letters in place. However, I looked at the Javascript source code — and there was no provision for ratcheting. So, like random coder, I threw together some Lisp code to play with the algorithm.

If the mutation rate and population size are just so, then the algorithm **appears** to latch. Increase the mutation rate and no latching behavior is observed. In fact, with higher mutation rates, the target string would likely not be found.

Would it be too provocative to say that the mutation and population parameters have to be tuned by the programmer in order to get this to work?

128

kairosfocus

08/28/2009

12:25 am

A few Footnotes:

1] Weasel 1987:

The proposal by Rob [that the BBC Horizon video is showing all pop members] would explain the winking effect on observed reversions, but at the price — as he implies — of removing it as evidence that implicit ratcheting/latching is presumably "not" happening.

Hitherto, the 1987 video was usually cited by objectors tot he concept of latching as "proof" of non-latching behaviour. (Can anyone identify tehrun of generational champions from 1987 and show whether or not this run showed latching or quasi-latching [i.e. rare reversions of correct letters . . . probabilistic barriers are not generally 100% effective . . .]?)

Beyond this, it still remains that [a] implicit latching, [b] quasi-latching and of course [c] far from latching behaviour are demonstrated on adjusting mut rate, pop per gen and of course filter effects.

2] When is a query a query?

Notoriously, the Clinton era white House said that "it depends on wha the meaning of is, is."

We face just above [with reference to Rob etc], a situation where it depends on what a "query" is, in the context of the Dembski-Marks analysis.

In short, **the whole generation champion selection process can be *legitimately* seen as one query.**

For, if we have implicit latching happening — and remember latching-ratcheting relates to the run of champions . . . and that is D & M's EXPLICIT context — in such a case, the Dembski-Marks analysis will apply.

Thus, Q can legitimately be understood in context — StephenB keeps drawing our attention to this factor . . . — as a metric of number of generational champions in sequence. In short,t eh query ain't over till the factor that is "ratcheting" has been identified. In the case of a procedure that enfolds population generation and proximity to target selection to get to that stage, the proper interpretation of Q is as number of generations of query.

For — repeat — it is the generational champions that would be ratcheting to target.

And, since a pop generation and champion selection procedure can be used with EXPLICIT latching, the same result applies for this case too.

Next, partitioning is a term that describes a search procedure based on effective divide and conquer: once a partial success has been achieved, by whatever mechanism it is preserved and t5he rest of the search is confined tot he remaining part of the string. That can be done explicitly (cf mask filter concept in 66) or implicitly (see interaction of pop per gen, mut rate per letter in the seed for a generation, proximity to target metric, and filtering also in 66).

In short, the associated objection turns on misreading what is going on in the analysis and in particular, misunderstanding of what constitutes a "query" in the contextually relevant sense.

Ah. "Context . . ."

3] When is a target a target?

A glance at CRD's description of Weasel should make it clear that he EXPLICITLY includes a proximity-to -target filter which promotes "nonsense phrases" — i.e. plainly non-functional ones — on a metric of proximity to a target.

Let us remind ourselves [with special reference to TA], from BW:

> It [Weasel] . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. *The computer **examines** the **mutant nonsense phrases**, the 'progeny' of the original phrase, and **chooses** the one which, however slightly, **most resembles** the **target** phrase*, METHINKS IT IS LIKE A WEASEL . . . . What matters is **the difference** between the **time** taken by **cumulative selection**, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of **single-step selection** . . .

This immediately implies that the target is preloaded in the algorithm, and is not created by it. (A GA version of Weasel that does not have a loaded target will not in general converge on the specific phrase "Methinks …" either.)

That is, the analysis in 66 above and in the appendix it links, is correct.

Also, t6he injection of targetting and of proximity reward as "fitness" rather than reasonable threshold of independent function — dismisswd as "single step selection" — inadvertently demonstrates that Weasel exhibits active information in action.

GEM of TKI

129

Indium

08/28/2009

1:12 am

So, now, kf, you are reduced to claiming that an algorithm that does a query by constructing a population by copying a string with random mutations and then selecting the best fitting member is the same as an algorithm which gueries by just randomly selecting a new letter for every wrong one.

This is so obviously wrong that it doesn´t even need a refutation. It is plain denial of the obvious.

130

DiEb

08/28/2009

1:50 am

@wtf3:

If you take *number of correct letters* as the fitness function, the probability of observing a change from a better to a worse state between two generations is dependent on the rate of mutation and the size of the population. As an example, for a mutation rate of 4% one gets:
population of 10: 95.7 %
population of 50: 0.0000026 %
(pic)
i.e., in a billion runs of Dawkins's weasel program with a population size of 50, one expects 26 incidents of such obvious non-latching behaviour.

131

kairosfocus

08/28/2009

5:43 am

Indium:

Please.

You force me to be direct: **You need to stop misreading to object, and start reading to understand**; for which reading *in context in a conceptually and dynamically connected way* is vital.

(Contrary to the Dawkinsian propaganda, we are not ignorant, stupid, insane or wicked. Now, too, Marks is a PhD Electrical engineer, while Dembski is a PhD mathematician. As well, the paper — a peer reviewed publication of an IEEE professional society — presumably passed the scrutiny of a panel of at least three comparably qualified and experienced electrical engineers. As one who helped design an electrical engineering degree program, I can assure you that such men will have a solid background in mathematics and logical reasoning, as well as in modelling and analysis of mathematical models, and of course relevant computer science. In short, the attitude that ever so clearly lurks behind your objections does not pass the smell test.)

So, it is no surprise that a closer look is revealing on the gaps in the objection you so facilely made. For just one instance, let us now observe the lead in to Eqn 22:

> . . . the probability of **identifying all L characters** in Q queries is . . . .

1 –> That is, the natural point of completion of a query is the point where the issue of distance of approach to target is definitively addressed as at that time.

2 –> This, plainly, is not the point where mutants in any given generation have simply been generated by the seed with the per letter mutation rate applied [that just gives us a cluster of n mutants].

3 –> Instead, it is the point where you have applied the selection filter in the generation and have chosen a champion, identifying its Hamming distance to the target as the least. Probability metric q (the subject of Eqn 22) is then assessing the likelihood that one is then

on target, i.e. ALL L LETTERS are then correct after Q queries.

4 –> In short, just as I have already observed, **the proper understanding of the point where a query is completed by the relevant algorithms is the point where you have a definite generation champion at a definite nearest distance to target so far.**

5 –> If the algorithm uses a generation population size 1, then it would be the immediate subject of the distance to target metric. But such a generation size is a simplification; CRD explicitly speaks of multiple member generations.

6 –> And, it is when the champion on proximity to target has been chosen that we have something for which, across a sequence of generations, we may observe whether it ratchets (thus latches) or has occasional slips, or has no behaviour that looks anything like a ratchet.

7 –> Recall, too, the further context of all this is the very explicit context in which: *"char-acters are **ratcheted** into place . . . "*

8 –> In either an explicit or an implicit latching case, that can only happen after the generation champion has been chosen by the distance to target filter and a contest.

____

Therefore, I am not being "reduced" to anything. I am simply reading for sense in context; as opposed to reading to make — too often, specious and even sophomoric — objections.

GEM of TKI

132

kairosfocus

08/28/2009

5:49 am

PD: DIEB, re : *If you take number of correct letters as the fitness function*

While I am aware of the 'standard" terminology, I must point out how it is loaded and misleading here. For, the Weasel does not assess fitness in any sense of functionality. It only assess distance to pre-loaded target [i.e. Weasel is not creating de novo information out of noise!], rewarding "nonsense" — thus precisely non-functional and unfit — phrases for proximity. this is at the heart of the question being begged by the program, as I pointed out in 66 above, and of course in my always linked appendix 7.

133

kairosfocus

08/28/2009

6:05 am

WRF3, 127:

Re: >> If the mutation rate and population size are just so, then the algorithm appears to latch. Increase the mutation rate and no latching behavior is observed. In fact, with higher mutation rates, the target string would likely not be found. >>

In short, under certain conditions, the algorithm will implicitly latch [a certain proportion of the time, quasi-latching otherwise one presumes . . . probabilistic barriers do not impose solid barriers], but under others, it will not.

If we, a la CRD 1986, are looking for "**cumulative** selection" then we will showcase the examples that "best" illustrate steady ratcheting progress to target, and voila, the Weasel 86 published results:

____

>> 1 WDL*MNLT*DTJBKWIRZREZLMQCO*P
2? WDLTMNLT*DTJBSWIRZREZLMQCO*P

```
10 MDLDMNLS*ITJISWHRZREZ*MECS*P
20 MELDINLS*IT*ISWPRKE*Z*WECSEL
30 METHINGS*IT*ISWLIKE*B*WECSEL
40 METHINKS*IT*IS*LIKE*I*WEASEL
43 METHINKS*IT*IS*LIKE*A*WEASEL

1 Y*YVMQKZPFJXWVHGLAWFVCHQXYPY
10 Y*YVMQKSPFTXWSHLIKEFV*HQYSPY
20 YETHINKSPITXISHLIKEFA*WQYSEY
30 METHINKS*IT*ISSLIKE*A*WEFSEY
40 METHINKS*IT*ISBLIKE*A*WEASES
50 METHINKS*IT*ISJLIKE*A*WEASEO
60 METHINKS*IT*IS*LIKE*A*WEASEP
64 METHINKS*IT*IS*LIKE*A*WEASEL >>
```

We may observe in these decimated samples that of 300 + letters we observe 200 of letters that have gone correct and not one instance of such a letter ever reverting to incorrect status. On la of large numbers, it makes it morally certain that he programs — proudly announced as cumulatively selective — indeed latched success to date and ratcheted their way to target in these cases.

this can be achieved explicitly or implicitly, by suitably matching pop size, mut rate and filter, and of course showcasing good results [which as you suggest may not be particularly hard to find].

A mountain has been made out of a molehill since December last, and now that a peer-reviewed paper hasd been published with an analysis based on ratcheting, it is being exgended into a mountain range.

But in fact, ever since 1986, it should have been obvious to conscientious commenters, that Weasel begged the main questions at stake, was fundamentally dis-analogous to the3 claimed mechanisms of chance variation and blind natural selection, and in fact showed intelligently designed artificial selection at work.

Unfortunately, such substitution of artificial for natural selection and inference of dubious results traces all the way back to Origin of Species.

Sad. Ever so sad.

GEM of TKI

134

Indium

08/28/2009

6:26 am

kf,
far from being irrelevant, how the query is done is exactly the algorithm we talk about!

For D+M, a query is being done by randomly choosing new letters for incorrect ones. The probability to find the target after Q queries is given by Eq 22 in their paper.

Dawkins looks for a champion of a pool of sequences he builds by copying a parent string with random mutations (which are independent of the letter being correct or not). The chance to have the complete target string reproduced differs greatly from Eq. Nr. 22 from the D+M paper and obviously depends on population size and mutation rates, which are not used in the partinioned search D+M apply.

Different kind of query, different algorithm. And, it bears to be mentioned again: No population and no selction are used in the D+M paper. As I said, denying of the obvious. Quite funny, please go on with it. It makes your evasion and failure to admit a simple error all the more visible for everybody.

Oh, and to complete my catalogue of open questions/remarks: In Atoms GUI, which algorithm corresponds to the partitioned search from the D+M paper and which search algorithm corresponds to Weasel? As a new bonus: Of these two, which one seems to use more active info?

135

kairosfocus

08/28/2009

6:57 am

Indium:

You are confusing a description of the EFFECT of ratcheting with the mechanisms that generate it.

As has been shown, ratcheting can be achieved explicitly OR implicitly.

ONCE RATCHETING EXISTS, THE MARKS AND DEMBSKI ANALYSIS APPLIES.

Your basic error is that you insist on seeing an explicitly latched mechanism in the Dembski and Marks analysis, whilst in fact they are starting from the OBSERVED ratcheting and are working out its mathematical implications for such searches. (Their example is one of what ratcheting looks like, presumably after several generations. Note that one of Dawkins' examples, the 43 gen run, starts with three correct letters that never revert. By generation 10, something like six letters are correct and thereafter they never revert, steady cumulative progress being made onward until by gen 43 all letters are correct. This example is of course in BW, which is referenced by M & D in teh context.)

You are reading explicit latching INTO the analysis, not drawing it out of it.

Again, kindly stop reading to object and start reading to understand, bearing in mind that you are dealing with reasonably experienced and qualified persons.

That would save you from making gross blunders in criticisms, at least.

GEM of TKI

136

DNA_Jock

08/28/2009

7:08 am

Indium, you beat me to it. But kf keeps failing to get the point, so to repeat:
D&M describe a partitioned search with a generation size of one. Any algorithm that selected a generational champion would have a completely different equation to describe its behavior.
kf does however bring up an interesting point when he popints out that this is a peer reviewed paper that has

> "presumably passed the scrutiny of a panel of at least three comparably qualified and experienced electrical engineers".

kf notes that

> " As one who helped design an electrical engineering degree program, I can assure you that such men will have a solid background in mathematics and logical reasoning, as well as in modelling and analysis of mathematical models, and of course relevant computer science."

Ignoring the rather silly argument from authority here, I will note that "as a PhD who has published and has reviewed papers" the peer-review process is not perfect.
As a reviewer, you usually would not bother to check the appropriateness of the citations. The reviewer assumes that the citation is accurate and appropriate. So an electrical engineer reviewing this paper is going to assume that ref[12] describes a partitioned search with a generation size of one. The reviewer assumes that the authors are honest.

137

DNA_Jock

08/28/2009

7:23 am

> kf:
> ONCE RATCHETING EXISTS, THE MARKS AND DEMBSKI ANALYSIS APPLIES.

This is just plain wrong.
Eqn 22 does not describe the mechanism of Weasel, and it does not describe the behavior of Weasel. And the reason is that eqn22 requires a generation size of one; no selection of a generational champion is permitted.
It is impossible to achieve the Weasel run of 1986 via partitioned search.

138

kairosfocus

08/28/2009

7:36 am

DNA Jock:

I see you can repeat the idea hitman talking points quite well. (And please, turn off the strawman and red herring factories: *it should be clear that I am asking for a fair reading on the grounds that people of a certain background have a reasonable level of basic competence, not appealing to blind submission to the authority of peer review.* [See what troubled waters reading to object lands you in?])

Now, please observe what has been pointed out, most recently in 131, and explain how ratcheting is observable apart from after a single representative of each generation has been selected, and for sufficient generations that we may reasonably see non-reversion.

And, tell me how the generation of a cluster of mutants from a seed and selection based on closest approach does not generate a SINGLE champion as representative per generation; as we may see listed in the printoffs from Weasel 1986.

The D & M analysis occurs in the context of such observable ratcheting, and addresses the question in the first instance of probability of hitting home by Query number Q. And, the focus of that is the identification of the member of the latest generation which has least Hamming distance to target.

Att hat point, once we have a viable mechanism to selecty a champion per generaiton, we have a basis for an analysis of ratcehting behaviour, which is what Dembski and marks undertake. They have said precisely nothing about how we get to the closest-approach string, only that it is in the context that such strings, in succession ratchet towards the target.

It is others who have asked how can that be, and our answer, with demonstration — facilitated by Atom's adjustable weasel produced and supported by the same EIL — is that Weasel algorithms can be latched explicitly or implicitly, and with algorithms that latch implicitly, that depends on matching of generation pop size, mutation rate per letter and proximity selection filter.

And, we can show marches of implicitly latched champions, which would be the result of queries in the relevant sense, as discussed in 131.

once we see the ratcheting, cumulative march to target in action, the M & D analysis applies.

And so, onlookers, we again see the sort of problem Galileo faced: many of his critics refused to look through his telescope and examine the actual evidence on the merits as a basis for informed discussion. their idea hitman talking points were quirte persuasive to many who were schooled in their view of things, but lacked grounding in the evident facts.

And here, the latest red herring and strawman models intended to dismiss the M & D analysis hinge on a distorted reading of what a query is in teh M & D analysis.

of courese, all of this is w=ever so far form the real issue:

1 –> M & D have produced a significant analysis that introduces Active Information solidfly to the world of peers.

2 –> In so doing, they are accounting for the impact of itnelligence on the dcreation of fucntionaly specific complex informaiton

3 –> this extends the existing analysis of ID.

4 –> Last and least, Weasel c 1986 aptly illustrates the impact of such active information. indeed, without realising it, Dawkins says as much in BW:

> It [Weasel] . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. The computer examines the mutant nonsense phrases, the 'progeny' of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL . . . . **What matters is the difference between the time taken by cumulative selection, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of single-step selection** . . .

5 –> that difference of course comes out of active information.

Onlookers, I think enough has been said for now to help spot the gaps in the objectors' latest set of idea hitman talking points.

G'day.

GEM of TKI

139

DeLurker

08/28/2009

7:36 am

kairosfocus#135

> You are confusing a description of the EFFECT of ratcheting with the mechanisms that generate it.
>
> As has been shown, ratcheting can be achieved explicitly OR implicitly.
>
> ONCE RATCHETING EXISTS, THE MARKS AND DEMBSKI ANALYSIS APPLIES.

The behavior of ratcheting only exists for certain values of population size and mutation rate, and even then only probabilistically. As Indium points out in #134:

> Dawkins looks for a champion of a pool of sequences he builds by copying a parent string with random mutations (which are independent of the letter being correct or not). The chance to have the complete target string reproduced differs greatly from Eq. Nr. 22 from the D+M paper and obviously depends on population size and mutation rates, which are not used in the partinioned search D+M apply.

The Weasel algorithm documented by Dawkins and the the partitioned search algorithm documented by Dembski and Marks are completely different and not amenable to the same type of mathematical analysis.
Further, your statement

> Note that one of Dawkins' examples, the 43 gen run, starts with three correct letters that never revert.

is incorrect. You have no way of knowing if those letters revert because only the best daughter strings from every tenth generation are shown. With Dawkins' clearly documented algorithm, it is mathematically possible for a reversion to occur. It is impossible for a reversion to occur in Dembski and Marks partitioned search.
This brings us back to the core issue, namely that Dembski and Marks did mischaracterize Dawkins' algorithm in their IEEE paper.

140

Indium

08/28/2009

7:52 am

kf, you are so fixated on the implicit or explicit ratcheting or latching, it´s amazing.

Please acknowledge that everybody has more or less understood this latching behaviour. From the output strings in the BW we can´t decide which internal behaviour the algorithm has. So, for the moment let´s pretend the latching issue is solved.
But, what we **can** see very easily just from looking at EQ. 22 is that the analysis from Dembski and Marks is not correctly describing Weasel. The formula, plain and simple, is an incorrect description of Weasel. I don´t even need to do any calculation to prove it, I just have to note that the behaviour of Weasel strongly depends on mutation rates and population sizes, which is of course not true or EQ 22 or the partitioned search D+M in general.

So, D+M use a completely different algorithm, the math is plainly incorrect for Dawkins Weasel. It is correct only for the partitioned search they describe in the text.

You can check this easily once you finally let me know which of Atoms algorithms is Weasel and which one the partitioned search. But since you have ignored this questions about 10 times now I guess you will do it an eleventh time, too, right?

141

kairosfocus

08/28/2009

8:04 am

DL:

Passing by.

For the sake of onlookers, i will add some pointers.

You have summarised implicit latching as though you disagree with me.

As to he case of the 43 gen program, I think the sample was nearly 90 letters in the sample that go correct and in the samples never revet. I suspect that a 90 or so sample will be representative, of the program as a whole, especially as CRD used it to illustrate cumulative selection, which is of course a near synonym for ratcheting.

In short, you are indulging here in selective hyperskepticism.

Indium:

Not at all, you are reading INTO the remarks on p 1055, an algorithm that is not there. the analysis is of ratcheting behaviour as observed, not of how it may arise — apart from the associated effect of partitioning [which is produced by latching whether implicit or explicit].

This can happen explicitly and implicitly, i.e. the analysis of ratcheting is not algorithm-specific, but observation responsive. Implications of observed ratcheting, not inference to best explanatory mechanism.

AND, you need to realise that for the past 6 or so months there have been wave after wave of objections tot he reality and significance of implicit latching, including some aspects of your remarks.

Onlookers, hope this helps.

G’day again

GEM of TKI

142

Indium

08/28/2009

8:11 am

No, sorry, it doesn´t help at all because you just ignore every argument and question.

143

kairosfocus

08/28/2009

8:18 am

Indium:

I am not ignoring questions, I have answered them on the merits.

Just, my answers do not fit your preconceptions.

Why not pause and review the thread above back to say 131, and then go way back up to 66 and come back froward? (Not to mention my always linked, appendix 7.)

While you are at it, reread pp. 1055 – 6 in the paper.

Then come back on the issues.

Okay, I *really* have to go now.

GEM of TKI

144

kairosfocus

08/28/2009

8:20 am

PS: Onlookers, do the same. is it fair to say that I have **ignored** “every argument and question”?

145

Indium

08/28/2009

8:28 am

Ok, let´s try again.

Which of Atoms algorithms is the partitioned search and which one Weasel?

146

Tomato Addict

08/28/2009

8:38 am

KF:

I was trying to point out that the search will work equally well for any equivalent gradient function. For this example the target is known and fixed, but that is not a requirement: The search will work equally well even if a specific target is unknown (granted we might not know immediately when the search is completed).
My comments perhaps stray too far from the topic at hand, so I won’t pursue this line of discussion.

However …

It seems that the definition of latching and ratcheting being used is that of any successful search. By this standard any algorithm capable of climbing a gradient is latching and ratcheting, and that seems to included all possible search algorithms. What would it take to have a search algorithm that does not, implicitly or explicitly, ratchet and latch? I would suggest that an algorithm with non-zero probability of back-sliding (moving down the gradient instead of up) is not doing either.

I confess some confusion about the concept of “active information”. It appears to me like any use of information from the local or global search environment is being considered to as “active”. This would then include all search algorithms other than a “blind” random walk, and I have a hard time considering that to be any sort of a search. Is there such a thing as a search algorithm that does not use active information?

147

Joseph

08/28/2009

8:40 am

DeLurker,

I know only a sample of outputs were shown in the book.

In that sample we never observe a character reversal.

That is a fact.

Take that with the definition of “cumulative” and his description of cumulative selection, there is no reason to think that character reversals are possible.

Yes intlligent agencies can take a step back in order to gain ground going forward.

But the weasel program is not imitating intelligent agencies.

IOW your deception is duly noted.

And if deception is all you have to make your case then you have already lost.

148

Indium

08/28/2009

8:43 am

In the meantime you have ignored my question in two additional posts of yours, btw, kf.

I will add a bonus question: Look at the number of queries Atom tracks for both algorithms. Do they match your definition = finding one champion is one query?
This is rather irrelevant for the question wether D+M misrepresent Dawkins but it shows very nicely that you just make up your definitions as you go along.

149

DeLurker

08/28/2009

8:52 am

kairosfocus#141

> As to he case of the 43 gen program, I think the sample was nearly 90 letters in the sample that go correct and in the samples never revet. I suspect that a 90 or so sample will be representative, of the program as a whole

and
Joseph#147

> I know only a sample of outputs were shown in the book.
>
> In that sample we never observe a character reversal.

You are both ignoring my #41 and #78 in which I show that you are considering only approximately 0.047 percent of the strings generated by a typical run of the Weasel algorithm. Looking only at the most fit string every 10 generations is unlikely in the extreme to show reversions.
Look at the algorithm description. It is very clear.

150

Joseph

08/28/2009

8:52 am

The Dembski/ Marks paper pertaining to a partitioned search uses a selection process similar to Dawkins in “WEASEL”.

They also use a population of 1.

They also appear to be using a very high mutation rate- 5 correct letters appearing in one whack.

They do not describe the starting sequence.

IOW yes I see issues with their algorithm if it was supposed to exactly mimic Dawkins’ “weasel”.

That said I don’t believe that was their intent. Dawkins was trying to illustrate a different idea than Dembski/ Marks.

151

DNA_Jock

08/28/2009

8:55 am

OK, let's imagine, just for the sake of illustration, that ref[12] described a Weasel program with an **explicit** latching **mechanism**.
D&M would still wrong to equate ref[12] with a partitioned search.
Eqn 22 describes a partitioned search. It cannot be used to describe a process that includes the selection of a generational champion.
This has been pointed out to you a number of times, and yet you continue to assert.

> ONCE RATCHETING EXISTS, THE MARKS AND DEMBSKI ANALYSIS APPLIES.

Stripping away the rhetoric, your argument appears to be "I cannot see the difference between eqn 22 (partitioned search) and a Latching-Weasel, therefore there is none."
This sounds like an argument from personal stupidity, but I'll give you the benefit of the doubt here and blame Morton's Demon.

152

Indium

08/28/2009

9:00 am

Joseph,

admitting an error (even if it is not your own) is not usual here on this board, I applaud your honesty in #150.

Cheers!

153

R0b

08/28/2009

9:02 am

kairosfocus:

> The proposal by Rob [that the BBC Horizon video is showing all pop members] would explain the winking effect on observed reversions

The winking effect has been explained several times by various people in the WEASEL discussions. Are you just now taking note of it?

> but at the price — as he implies — of removing it as evidence that implicit ratcheting/latching is presumably "not" happening.

No, it doesn't remove it as evidence that implicit latching is not happening, because it never was such evidence.

> Hitherto, the 1987 video was usually cited by objectors tot he concept of latching as "proof" of non-latching behaviour.

And we're exactly right, because we have always used the term *latching* to refer to what you call *explicit latching*, and the video does prove that the letters are not explicitly latched.

> the proper interpretation of Q is as number of generations of query

Earlier it was "tell me just what part of . . . **Q queries** . . . it is that you do not understand." Now you're saying that a query isn't actually a single inquiry of the oracle, but rather a whole generation. Of course, that's what you meant all along, right?

But you seem to have not put much thought into this. How do you explain the fact that M&D's math doesn't take into account the population size and mutation rate?

Let's do the same math as #125 with Q=generations:

q = (1-(1-(1/N))^Q)^L
= (1-(1-(1/27))^45)^28
= .003

but it should be .5.

154

DNA_Jock

08/28/2009

9:09 am

Good idea, Indium, to point people to the EIL site, where they can see for themselves that partitioned search and Weasel are quite different beasts.

My personal favorite – take parameters that produce kf's "quasi-latching", say gen size 100 and mutation rate of 10%, but then increase the length of the target string…At 57 characters, the partitioned search needs a median of 117 queries, but Weasel has gone past 36,000 (x100/gen = 3.6 MM offspring) without success.
It is left as an exercise for the reader to find Weasel parameters that produce "apparent latching" but query runs over a thousand generations.
Now, how do I adjust eqn22 to model these searches? That's a puzzler.

155

Joseph

08/28/2009

9:15 am

DeLurker:

> Nonetheless, the Weasel algorithm as described can, in fact, take a step back.

Not as described by Dawkins in TBW.

> You continue to ignore the fact that Dembski and Marks mischaracterized the Weasel algorithm in their IEEE paper

That is not a fact. And as a matter of fact the only inference one can take from TBW is that cumulative selection, as described by Dawkins and illustrated by the weasel program is a ratcheting process.

156

Indium

08/28/2009

9:15 am

DNA_Jock: Be careful! You have just proven that evolution is wrong! 😉

157

Joseph

08/28/2009

9:17 am

DeLurker,

We cannot consider what we cannot observe.

All we have to go by is what is in the book.

That is why your comment in 41 is irrelevant.

158

Joseph

08/28/2009

9:19 am

BTW by saying non-telic processes can mimic agency, that is very deceptive.

Cumulative in a non-telic scenrio can only refer to "increasing by succesive additions".

159

Indium

08/28/2009

9:21 am

Joseph,
yes, evolution tends to keep positive mutations in the population and so does Weasel.

From the output in the 1987 video you can see that this is not true for every individual member of the population though. This is in line with the explanation of Weasel in TBW: The mutation rate is independend of the resulting fitness. This is, btw, the most important difference between Weasel and the partitioned search from the D+M article.

160

Indium

08/28/2009

11:03 am

Oh, and btw, as far as I can see a few posts from DeLurker have been removed from this forum. This usually means that DeLurker has been banned, Joseph. So, you are probably arguing with an empty chair. This is easy (for most people, some even then mess it up!), but not much fun I suppose.

161

kairosfocus

08/29/2009

5:48 am

Mrs O’Leary (and onlookers):

It is clear that no authentic code by CRD c. 1986 in Basic or Pascal will be forthcoming. So, we will have to work with the showcased print-offs and with the descriptions he has given.

I: The impact of Active Information

CRD’s remarks in BW, 1986, bring out a remarkable key fact that we might easily miss in the clouds of digital ink above on all sorts of side issues and side-tracks.

Namely, *in 1986, Mr Dawkins inadvertently testified to the power of intelligent design over blind chance and necessity, and in so doing also testified to the impact of active information provided by intelligence in making a search for functionally specific complex information become feasible*:

____

>> It [Weasel] . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – ‘mutation’ – in the copying. *The computer [indirectly, the programmer!]* ***examines*** *the mutant* **nonsense phrases***, the ‘progeny’ of the original phrase, and* **chooses the one which, however slightly, most resembles the** ***target*** **phrase**, METHINKS IT IS LIKE A WEASEL . . . . *What matters is* ***the difference*** *between the time taken by* ***cumulative selection****, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of* ***single-step selection*** [i.e. "more than a million million million times as long as the universe has so far existed"] . . . >>

____

a –> Weasel c. 1986 (and maybe 1987 too) is cumulatively progressive, targetted search that on the showcased sequences of generational champions, ratchets forward to the target [latching the already achieved successful letters, even if they are in "nonsense phrases"], through a filtering mechanism that picks the closest approach to target in each generation as the seed for the next.

b –> It is therefore unsurprising that when CRD makes qualifying statements, he admits that Weasel *“is* ***misleading*** *in important ways. One of these is that, in each generation of selective ‘breeding’,* ***the mutant ‘progeny’ phrases were judged according to the criterion of resemblance to a distant ideal target****.”*

c –> So, Weasel is by its author’s direct admission, misleading because it is foresighted, targetted search, search that rewards increments in proximity rather than of function [note his "nonsense phrases remark].

d –> Likewise, he admits that if instead Weasel had to produce functionality before any hill climbing could begin, it would fail, exhausting the available search resources.

e –> And so, we see what accounts for the relatively remarkable capacity of cumulative, targetted selection on proximity: information abovut warmer/colder has been given, allowing for proximity-climbing.

f –> i.e., the ***active information*** *injected by the intelligent designer of the program* has had a dramatic impact on the capability of Weasel.

g –> It is equally worth noting that CRD acknowledges: “In real life, the criterion for selection is always short-term, either simple survival or, more generally, reproductive success.” That is, present functionality is critical to the working of natural selection, and so the dismissal of “single step selection” rather begs the main question, and helps Weasel achieve an unwarranted degree of plausibility and influence of making it seem that chance variation and natural selection can credibly create novel complex functionally specific information.

h –> So, Weasel needs to be put out to pasture as another exploded icon of evolution. (But ID-ers can then hire it back out of retirement as evidence of the power of active information based intelligent design.)

____

A bit of an irony, nuh?

[. . .]

162

kairosfocus

08/29/2009

5:49 am

Now, too, there are a few mopping up points that need to be addressed:

II: The Dembski-Marks "Algorithm":

Above, much has been made of an alleged algorithm developed by M & D, and how it so far differs from the Weasel 1986, to the point where issues of academic misrepresentation have been hinted at in purple passages.

However, especially given the fact that EIL hosts Atom's adjustable weasel that makes its source code available — hint, hint, Mr Dawkins — and covers the various possible weasels admirably, this is a case of making a rhetorical mountain out of a mole-hill. A glance at the section of p 1055 of the IEEE paper will help us see why:

____

>> Partitioned search [12] is a "divide and conquer" procedure best introduced by example. Consider the L =28 character phrase

METHINKS ? IT ? IS ? LIKE ? A ? WEASEL. (19)

Suppose that the result of our ?rst query of L =28 charac-ters is

SCITAMROFN ? IYRANOITULOVE ? SAM. (20)

Two of the letters {E, S} are in the correct position. They are shown in a bold font. In partitioned search, our search for these letters is finished. For the incorrect letters, we select 26 new letters and obtain

OOT ? DENGISEDESEHT ? ERA?NETSIL. (21)

Five new letters are found, bringing the cumulative tally of discovered characters to {T, S,E, ?,E, S,L}. All seven char-acters are ratcheted into place. The 19 new letters are chosen,
and the process is repeated until the entire target phrase is found. >>

____

1 –> First, the example is explicitly given to illustrate what partitioned search is like, and starts with a set of letters in 20, then immediately proceeds to vary all incorrect letters, giving us 21 with several more correct letters [itself rather unlikely and a clue that this is didactic, not an algorithm in progress], then says "the process is repeated until the entire target phrase is found."

2 –> Let us, just for argument, accept that this is meant to be a realistic algorithm, with partitioning and 100% variation of letters remaining.

3 –> Now, in sampling with replacement, the idea is that once a letter may vary, it can take up any of the 27 available states. Of these, one is correct, and 26 are not. So, immediately, to plausibly gain a five-letter increment in proximity to target, a very large pool of mutant progeny would have had to be created so that far-tail highly improbable outcomes of multiple correct letters would show up among the generation of mutants.

4 –> And, a proximity to target filter would then be required to select the closest to target, which –voila! — has five more correct letters.

5 –> So, immediately, if an alleged algorithm is being shown, it is credibly based on a very large population per generation and a proximity to target filter. (And, as the demonstration at 237 in the March 26 Wm A Dembski thread in this blog on simulation wars shows, such a case would race home to target through picking multiply correct mutants. For, there, we may see how a 999 pop, 8% all-letters may vary case runs home in 22 gens.)

6 –> Of course, such an imaginary algorithm would have to be explicitly latched. For as the case F from 239 in the thread — a 999 pop, 25% case — shows, with those high rates, reversions will otherwise make it hard to close the deal.

7 –> "GOTCHA!" Nope: STRAWMAN!

8 –> For, already, we have seen that the *projected* D & M algorithm shows that a ratcheting action will close to target, and that it implies targeted, proximity filtering based search that uses generations of sufficient size to see far tail population effects such as multiple go-correct mutations. And, that the projection is based on an illustrative pedagogical example rather than a sample run. So,

there is plainly a caricature being set up to be knocked over here.

9 –> On a more reasonable, less rhetorically loaded discussion, M & D used a didactic illustration of what partitioning means, and how a partitioned search ratchets, then analysed what that implies about active information and its impact on success.

10 –> Now, we have long since *seen* from 234 in the Sim wars thread that such partitioning can be achieved not only EXPLICITLY, but also IMPLICITLY. Similarly, EIL just happens to sponsor Atom's adjustable weasel, which provides explicitly latched search and a second algorithm that under certain cases — as in 234 Sim wars thread, etc — will implicitly latch as pop per generation, mut rate per letter and filter interact. [Observe how those objectors who so stridently claim to see an algorithm convenient to their debating points in the IEEE paper glide by the provision of the adjustable weasel as quickly as they can, even repeating silly questions on which algorithm was used to do the implicitly latched demonstrations even after it had not only been obvious but explicitly answered.]

11 –> So, once we respect context, we can easily see that the illustration at p 1055 in the IEEE paper is didactic rather than a description of a serious algorithm.

12 –> But also, we see that the illustration even taken as a caricature is actually going to have to imply a very large population per generation and aggressive proximity filtering that can capture cases where multiple letters advance.

13 –> this is suggestive o0n a second point of debate as already addressed: what a "Query" means in the context of p. 1055. the answer is again plain, i.e my earlier analysis in what is now 131 is correct:

> . . . let us now observe the lead in to Eqn 22 [on p. 1055]:
>
> > . . . the probability of identifying all L characters in Q queries is . . . .
>
> 1 –> That is, **the natural point of completion of a query is the point where the issue of distance of approach to target is definitively addressed as at that time.**
>
> 2 –> This, plainly, is not the point where mutants in any given generation have simply been generated by the seed with the per letter mutation rate applied [that just gives us a cluster of n mutants].
>
> 3 –> Instead, **it is the point where you have applied the selection filter in the generation and have chosen a champion**, identifying its Hamming distance to the target as the least. Probability metric q (the subject of Eqn 22) is then assessing the likelihood that one is then on target, i.e. ALL L LETTERS are then correct after Q queries.
>
> 4 –> In short, just as I have already observed, **the proper understanding of the point where a query is completed by the relevant algorithms is the point where you have a definite generation champion at a definite nearest distance to target so far.**

14 –> So, there are no real — as opposed to strawman –barriers to looking at the M & D analysis as an analysis of partitioned, ratcheted, cumulatively progressive search for a target, however achieved: explicitly or implicitly.

15 –> And that is what M & D present, and in so presenting they highlight the impact of active information coming from intelligent designers — including of course Mr Clinton Richard Dawkins.

[. . .]

163

kairosfocus

08/29/2009

5:50 am

III: Minor issues

Several debate points have been raised that require a footnote or two:

1] Latching must only be explicit:

We have of course demonstrated that it can be implicit, especially when one is looking at showcased "good" runs.

That is, if there is a big enough population and a small enough mutation rate that we have a high probability fo no-change members in each generation, and single step changes are otherwise significant, then with an appropriate filter, we can fairly easily observe cases where once a letter goes correct, it will remain so all the way to the target. (The first such demo used 4%, 50 on Atom’s adjustable weasel.)

In short, latching, an aspect of ratcheted, cumulative progress to target, can be achieved explicitly and implicitly.

2] What about quasi-latching?

Since we are looking at showcased runs, it is the cases that show no reversions that become what we need to explain.

The fact that — since, probabilistic barriers are inherently porous — other runs on the same parameters may show occasional reversions therefore makes no relevant difference. (I used the term quasi-latched to predict such cases, which have ALSO been observed.)

3] Which Weasel shows implicit latching per Atom’s lists of algorithms?

Already answered but recirculated as though unanswered: PROXIMITY REWARD SEARCH.

4] Parsing EQN 22, p. 1055:

In introducing the expression, M & D note: “Assuming uniformity, the probability of successfully identi-fying a specified letter with sample replacement at least once in Q queries is 1 – (1 – 1/N)^Q.”

So, in this analysis, they are seeing letters in parallel columns of search, each in effect independent of the others. N is number of letters in the alphabet, and Q the number of queries completed. The probability of being right are 1 – that of being wrong after Q tries.

Eqn 22 extends this to L letters, L the length of the phrase.

But, there is an evident subtlety there. This can be seen in that each of the L letters is free to come home at any point up to *and including* the Qth query. (So, it seems we are working on the odds of hitting the target AT the Qth try, not the cumulative odds up to that point.)

This is fine for the context of comparative purposes as is shown in EQN 23 — on the similar odds to get to the target on simple random search — and following but it makes for a subtlety if one is expecting the cumulative odds to that point.

5] Partitioned search and weasel are different beats!

Not so fast, pardnuh!

Up to 2000, the information the public has had in hand was the showcased runs of 1986, and the description of cumulative targetted search. on this premise, it is reasonable — thus legitimate — to construct a weasel that uses the obvious mechanism, explicit latching.

Subsequent to that time, we have had statements from Mr Dawkins and co that Weasel did not explicitly latch at any stage — but mo credible c. 1986 code. (And BTW, commenter Apollos on this site has shown that explicitly latched weasels can be programed to show reversions. ONLY CREDIBLE CODE WILL BE DEMONSTRATIVE, IN THIS LIGHT.)

We have accepted the testimony on the presumption of charity, and have demonstrated that latching is feasible on an IMPLICIT basis. (Note, this makes no difference to the already discussed point that weasel demonstrates the power of active information, not chance variation and natural selection based on complex function.)

An explicitly latched weasel is a legitimate interpretation of the information given. And that is part of the point of the EIL demonstration: many weasels are possible and they give various patterns of behaviour, sometimes convergent, sometimes divergent. And in particular, implicit latching is a point of convergence between explicitly latched partitioned search and proximity reward search.

6] What about the BBC Horizon 1987 program?

This has often been held to demonstrate that Weasel as originally coded did not latch. Of course, it now is apparent that the winking effect is possibly due to the video highlighting the members of generations, not the generational champions.

A result that — as was acknowledged above by the commenter who raised it — is entirely compatible with: IMPLICIT LATCHING.

Beyond that, the gap between 1986 and 1987 together with the demonstrated impact of shifting parameters on a proximity reward algorithm without EXPLICIT latching — we can get latching, quasi-latching and far from latched behaviour — indicates that the 1987 run does not ground a claim that Weasel c 1986 as showcased did not latch and ratchet to the target in a cumulative process.

7] Somebody was banned!

This occasionally happens at UD, generally for abusive commentary; i.e for good — though regrettable — reason.

(For comparison, there was a recent case where Mr Barrett Brown — a columnist at HuffPo — made some slanderous remarks there. A UD thread was put up to discuss the mater, and it turned out that at HuffPo: (i) comments were very restricted as to length [250 words in response to nearly 3,000], (ii) critical comments were subject to suppression on non-transparent grounds, with no defensible reason, (iii) slander and abuse were tolerated and even encouraged there. On balance UD is the better forum by far.)

____

So, much of the above shows how Darwinist talking points can make the weaker appear the stronger case, to the unwary onlooker.

So, let us take note.

GEM of TKI

164

DNA_Jock

08/29/2009

7:42 am

Yowser.

With reference to kf's first post, where he points out (in his language) that Weasel uses tons of 'active information', i.e. it is better than a random search. He is making CRDs point for him. Thank you.

In the second post, kf tries to deduce a lot from the fact that D&M's exemplar goes from 2 hits to 7 in a single generation.
He claims this is "rather unlikely" and concludes that both of the following must be true
1) it is clearly a didactic example, and not a real sample
2) it shows that their must be a "very large pool of mutant progeny would have had to be created so that far-tail highly improbable outcomes of multiple correct letters would show up among the generation of mutants"

Obviously, these two cannot simultaneously be valid conclusions, but, not to worry, the premise is wrong: the chances of getting an iteration this good are one in 419, well within the bounds of reasonable 'data selection', given the time and computing power available….

But the third post is the winner. Here kf argues that equation 22 describes the probability that the Qth query is correct, rather than the cumulative probability.

> But, there is an evident subtlety there. This can be seen in that each of the L letters is free to come home at any point up to and including the Qth query. (So, it seems we are working on the odds of hitting the target AT the Qth try, not the cumulative odds up to that point.)
> This is fine for the context of comparative purposes as is shown in EQN 23 — on the similar odds to get to the target on simple random search — and following but it makes for a subtlety if one is expecting the cumulative odds to that point

Wow. Wow. Wow. This is so obviously wrong I don't know how to explain it. Here goes: since (1-1/N) is less than one, (1-1/N)^Q tends towards zero as Q gets very large. Thus equation 22 tends towards 1 as Q gets very large (for any fixed value of L). It's a cumulative probability distribution. As D&M make really clear in the text.
To repeat : **eqn 22 shows the *cumulative* probability that a partioned search will hit its target. It cannot be adjusted to account for a generation size greater than one.** Go to EIL and compare "Proximity Reward Search" (Weasel) with "Partitioned Search". Due to the way they cite TBW in their paper, D&M are claiming that these two beasts are the same, when they clearly know better.

2,740 words, all heat and no light. I guess I was wrong about Morton's Demon.

165

kairosfocus

08/29/2009

8:18 am

Onlookers:

Simply compare what I wrote and DJ’s critiques.

See where the balance on the merits is for yourselves, and who is dealing with specifics vs who is making simple dismissals based on snippets out of context and caricatured.

GEM of TKI

166

DNA_Jock

08/29/2009

8:37 am

Sorry about the bad link. Please go to the EIL site and compare the behavior of “Partitioned Search” and “Proximity Reward Search”.

I lowered the mutation rate to 3% to ensure I still get kf’s so-called quasi-latching behavior with longer Search Phrases, and I had a lot of fun watching the Proximity Search get worse and worse (relative to the Partitioned) as I increased the length of the Search Phrase.
Don’t trust either me or kf, *onlookers*, go and try it for yourself.
D&M’s paper maintains that these searches are the same…

167

DiEb

08/29/2009

8:44 am

I propose the following experiment:

> Take 100 programmer.
> Show them the two pages of Dawkins book.
> Ask them to realize the algorithm in the language of their choice in an hour of time.

Then, we can see which percentage understands the algorithm to latch explicitly, which fitness function is used in general, etc…

168

BillB

08/29/2009

9:16 am

KF, I think we can now sum up your claims as follows:

1-> An explicit, required, latching mechanism **is the same** as non-explicit, non-required, not-always-latching behaviour

2-> A mutation rate that has to be **between** zero and one hundred percent **is the same** as a mutation rate that has to be **either** zero or one hundred percent

3-> A population of one, where no selection can occur **is the same** as a population of many from which one is selected

Now:

> many weasels are possible and they give various patterns of behaviour,

Nonsense, many types of search algorithm are possible, they can use many different strategies and mechanisms, and produce different results. But they are all **different** types of search algorithm.

WEASEL is an algorithm defined by Dawkins in The Blind Watchmaker. Many people have written software based on it, and others have made software inspired by it, but which employs different strategies. There is only one WEASEL though and it is described quite clearly by Dawkins.

Onlookers will have observed that you have expended a huge amount of text over several threads trying to rhetorically and semantically get around the basic fact that Dawkins algorithm is different that Dembski and Marks. One could be forgiven for thinking that your inability to admit error is pathological.

Contrary to your allegations, those of us who keep arguing this point are doing so because it is a matter of fact, not because we are trying to confuse and poison the debate, or because we don’t understand the issues, don’t read your posts properly or are trying to avoid specifics.

The irony of all this is it makes no difference to the content of D and M’s paper if they simply removed the reference to The Blind Watchmaker, and replaced it with a correct reference to an actual peer-reviewed paper or an academic volume describing a partitioned random search, rather than referencing a crude pedagogical example in a popular science book.

All this work just to avoid admitting that a Weasel is not the same as a Fox!

169

Joseph

08/29/2009

9:26 am

Indium,

The debate isn’t about “evolution”.

It is about directed vs undirected processes.

And even the most beneficial mutation has a better chance of becoming lost before it becomes fixed- especially in sexually reproducing populations.

But anyway I have the book on order from the local library.

I will have it next week and post again at that time.

170

R0b

08/29/2009

10:39 am

kairosfocus:

> So, once we respect context, we can easily see that the illustration at p 1055 in the IEEE paper is didactic rather than a description of a serious algorithm.

That’s a nice theory, but it’s belied by the fact that both Dembski and Marks have stated that this algorithm is, in fact, *Dawkins’* WEASEL algorithm. Dembski has been saying it for years, even after correction. The EIL website still says it.

171

mindbleach

08/29/2009

1:51 pm

In the Blind Watchmaker video… you do realize that you can see the "correct" letters briefly change, right? In the linked video, around 6:24, you can see the K in METHINKS turn into an N, then an X, then an S, each for a fraction of a second. All of the letters flicker between correct and incorrect. Say what you will about the applicability or accuracy of Dawkin's program, but it obviously doesn't lock the letters once they're right.

172

Oatmeal Stout

08/29/2009

3:09 pm

kairosfocus,

With friends like you, who needs enemies? Some of us ID proponents with genuine competence in engineering would like for you to stop with your obfuscation, just as we'd like for Dembski to admit once in a while that he made a mistake. You seem to think there's huge value in the cultural war in presenting Dembski as an inerrant genius. I'd remind you that even the original Isaac Newton spent at least as much of his time on alchemy as he did math and science.

In scholarly circles, unlike political circles, owning up to errors is a key part of gaining credence.

The truly sad aspect of this run-around is that Dembski and Marks actually do analyze something like Dawkins' procedure in their article. More on this in my next comment. But in the meantime I ask, why are you wasting so much energy on muddying the waters when there is more valuable work to address?

173

Oatmeal Stout

08/29/2009

4:01 pm

Dembski and Marks actually do analyze something fairly close to the Weasel procedure in section III-F.2, "Optimization by Mutation." There are several differences from what Dawkins describes:

1. The target is specified over a binary alphabet.

2. The number of offspring is limited to 2.

3. The mutation rate is so low that in most generations both offspring are perfect copies of the parent.

The difference in alphabet is trivial. The target is arbitrary, as in the Weasel problem, and may be set to a string of 1's to reveal that the optimization problem is the heavily studied ONEMAX.

Solving ONEMAX is easy, yet it is clear in Figure 2 that D&M's simulation runs typically required a great many trials (the horizontal axis should be labeled "trials" or "generations," but the runs are very long in either case). The reason is that the mutation rate is 5 in 100,000 bits, when the length of bit strings is only 100. In other words, few offspring are mutants. From an engineering perspective, appropriate to an engineering journal, it is absurd to perform a stochastic search in which most trials generate no movement in the search space. The simulation does nothing but to give evidence that the simplifying assumptions in D&M's derivation of (28) do not introduce substantive errors, provided that the mutation rate is extremely low.

It is worth noting in the caption of Figure 2 that D&M do acknowledge that the decreases in fitness they assume, in the appendix, do not occur actually do occur rarely — even with an incredibly low mutation rate. It should be obvious, though no one has mentioned it here, that the probability of a decrease in fitness increases with fitness of the parent. When the parent matches the target, the condition for a decrease in fitness is that both offspring be mutants.

I believe that D&M would have done better not to restrict the number of offspring to 2, and to have obtained a low probability of decrease in fitness not by setting the mutation rate extremely low, but through a combination of a moderately low mutation rate and a somewhat higher number of offspring.

174

Indium

08/29/2009

5:18 pm

kf, as expected you have once again not answered my simple questions. But your arguments are becoming more and more entertaining, if this it at all possible! And I am still trying to read all your stuff, so please keep it up.

A small history of your arguments:
- an algorithm that doesn´t protect correct letters in a search is the same as one that does (the famous implicit latching!)

- an algorithm that does a query by replacing every wrong letter with a new random one is the same as one that builds a population (n>1) by copying a parent string with mutations and selecting the best one

- you have redefined "query" to mean the determination of the next parent string. Everybody can look at Atoms GUI to see that you are wrong of course.

- Somehow the paragraph/pictures/formulas in the D+M paper are only a pedagogical something that not criticizes Weasel directly but is somehow relevant anyway (this argument I don´t really get, I guess I am not alone here…). This obviously contradicts what Demsbi has said on this very website.

Dr Dembski started a thread here where he explicitly said that his article criticizes Weasel and therefore evolution. I hope every onlooker now understands that he did nothing of the sort. He discusses a partitioned search which is completely different from the search performed by Weasel. Neither the explanation by Demsbki nor Eq 22 can be mapped onto Weasel and your inability to recognize this or maybe to admit this rather obvious problem is not reflecting well on you.
I think your friend Joseph has now understood the difference, I think he has enough patience to explain it to you!

So, I have to ask again, just for fun: Which algorithm in Atoms software suite is Weasel and which one the partitioned search? Are they the same?

-
@Oatmeal Stout
Yes while the D+M can be criticized for not representing Weasel correctly that does not mean that the active info concept can´t be applied to Weasel. I am sure that it can be done in a similar way as it is done in the "Random Mutations" paragraph. Therere, however, D+M at least implicitly acknowledge where evolution gets its feedback (or active information) from: From the environment of course. But this is another topic.

-
@ Joseph
It´s not about evolution? Yes, I agree.

_
I have a brilliant idea: Why doesn´t Dr. Dembski present *his* Weasel algorithm here or just points us to the relevant code in Atoms suite? Maybe he can even get the prize then?

175

Tomato Addict

08/29/2009

11:06 pm

> 2] What about quasi-latching?
>
> Since we are looking at showcased runs, it is the cases that show no reversions that become what we need to explain.
>
> The fact that — since, probabilistic barriers are inherently porous — other runs on the same parameters may show occasional reversions therefore makes no relevant difference. (I used the term quasi-latched to predict such cases, which have ALSO been observed.)

So let me see if I have this straight. Latching (quasi- if you like) is still considered to occur even with occasional reversions. Hence you do not accept my suggestion that a non-zero probability of reversion is non-latching.

How about this then: Latching occurs when the probability of moving "up" the gradient is greater than the probability of moving down. This would again seem to include any search other than a blind random walk.

—and—

Joseph wrote> It is about directed vs undirected processes.

Is there any search other than a blind random walk that is undirected?
That doesn't seem like a useful distinction.

176

Tomato Addict

08/29/2009

11:24 pm

Some administrative questions related to this contest.

I have received no acknowledgment of my entry into the contest. Who may I contact about this?

When is the closing date for entries, and are multiple entries allowed?

How will the winner be determined?

It IS a contest, right?

177

Tomato Addict

08/29/2009

11:29 pm

There was an HTML failure in post #175: I was quoting *kairosfocus* at the beginning.

178

O'Leary

08/30/2009

12:44 am

Tomato Addict at 176:

1. There is no acknowledgement apart from publishing in the combox here. We assume that if you do, you have implicitly entered. If you win, you must send me a mailing address to claim the prize, but it need not be your home and will not be harvested anyway.

2. You can enter as many comments as you like, but you can only be one winner in a given contest. I have been known to declare two winners, in which case both get prizes.

3. I usually judge the entries and my task journal reminds me when. With many entries, there could be delay. When the subject is technical, as in this case, I will seek help, possibly introducing more delay.

4. We are all volunteers here. But yes, it IS a contest. There is a literal stack of books and DVDs right here in my office that will leave individually, when awarded.

179

yakky d

08/30/2009

1:12 am

If anyone's interested, Richard Dawkins himself has apparently commented on a recent thread PZ put up related to this contest. RD does comment on PZ's blog from time to time, so I think it's most likely really him.

Anyway, RD says he no longer has the original program but that it did not "latch". Not that latching vs. nonlatching is that important in the grand scheme of things, but it's helpful to get confirmation of this detail from the source.

180

sandlinjohnb

08/30/2009

2:00 am

First, the program was not part of scientific research – it was a demonstration of function. Second, many thousands have replicated the method and results without coding in explicit latching. Third, the discussion about a demo program as if it were research is absurd. Fourth, the original code is presumed lost. The point was never about the program, it was about the algorithm and how it demonstrates the effect of selection pressure on random changes.

181

Oatmeal Stout

08/30/2009

2:06 am

[My entry, made polite.] Let's step through this "Weasel" controversy.

1. Dawkins stated an algorithm.

2. Dawkins claimed to have implemented the algorithm in his Weasel program.

3. Dawkins provided a sample of the outputs from one run of the program.

4. Dembski, observing that something possible in execution of the algorithm was not evidenced in the sample, inferred that the program did not implement the stated algorithm.

5. Dembski went beyond this inference that the program did not correctly implement the stated algorithm to an inference of the algorithm the program actually did implement.

6. Dembski and Marks attributed the inferred algorithm to Dawkins, giving no indication whatsoever that it was not the algorithm Dawkins stated, and went on to analyze it.

Step 4 is logically invalid. Steps 5 and 6 are poor scholarship.

Regarding step 4, the program is randomized, and the fact that something that might have happened appears not to have happened is not sufficient evidence for concluding that the program is incorrect. At best, one may question the probability that the program correctly implements the stated algorithm.

Regarding steps 5 and 6, when you doubt that a program correctly implements an algorithm, you do not jump to abductive inference of the algorithm that the program does correctly implement, and then make your guess the topic of discussion. Proper conduct for a scholar would be to a) work with the author of the algorithm and program to determine if the program is correct, and b) develop a correct program if the original program is incorrect. It is never appropriate to say, “Well, let’s just change the algorithm to match the program.” Programs implement algorithms, and algorithms are not cooked up after the fact to describe what programs happen to do.

Whether or not Dawkins has provided the Weasel program is utterly irrelevant. Dembski and Marks analyze an algorithm, not a program. The correctness of a mathematical analysis of the algorithm is independent of the correctness of any implementation of the algorithm.

The appropriate challenge here is to provide the email in which Dembski and Marks ask Dawkins to confirm that the algorithm they attribute to him is in fact his algorithm.

182

sandlinjohnb

08/30/2009

2:52 am

I should also add I wrote a short BASIC program coded for the Algorithm Dr. Dawkins explains in the Blind Watchmaker. It isn’t Dr. Dawkins’ original code, but it works the same as the sample shown in the 1987 video. I wrote it for the MS-DOS 3.3 version of BASIC, not Apple][, but the differences should be minor.

183

nephmon

08/30/2009

3:51 am

I gave up reading the detailed arguments a while back, but I think I’m right in saying that that kf (verbose chap, isn’t he?) accepts at this point that the original TBW program didn’t contain any explicit latching but gives the appearance of it because only every 10th generation is shown.

I’ve observed this in my own (non-latching) implementation too, with n(offspring)=100 and p(mutation)=0.01. What’s interesting is that even if I print out the survivor for *every* generation, I still don’t see any reversion of previously correct characters to incorrect ones. Obviously the mechanism of selecting the fittest offspring from a suitably large population with a small mutation rate is a very powerful one. Which is what RD was trying to illustrate?

Since it’s been shown over and over (and over) again that the algorithm he described produces exactly the results he printed in the book without any explicit latching, can someone explain to me why seeing the original source code is remotely important?

184

kairosfocus

08/30/2009

4:41 am

Onlookers:

Shaking my head . . . sigh.

It is sadly evident from the above that we have much of straining at gnats while swallowing camels, as well as barking up wrong trees. Etc, etc.

But first, let’s go back a moment to where I left off, on minor issues.

I: More on Q

One of the peculiarities of Weasel algors, is that they halt when they hit home; of course halting being a key property of algorithms. As a result, if they hit home before a generation number G, they do not get to G. Thus, if Q means number of mutants to date, then if size of generation is S, Q = G*S.

Immediately, if a run completes [all L letters correct] at gen G, it cannot have been complete before G.

So, the paper’s p.1055 discussion would on this reading of Q would be of a ratcheted run that shows a march of champions with latching up to G, when it halts: completion AT G. (In an explicitly latched Weasel this would be automatic, in a version that on being tuned and giving a good run latches — as observed [our first swallowed camel . . .] — this implies that when metric falls to distance to target = 0, there is a latching action imposed.)

And there is also the second camel: debates over the meaning of Q do not affect the OBSERVED FACT of latching (regardless of other runs that may not do so — remember, we are accounting not for typical or overall behaviour but for runs that are showcased c. 1986 that were showcased because of their cumulative progress to target).

And that is camel no 3: implicit latching is an observed phenomenon, one that answers to CRD’s enthusiastic description and showcased printoffs. (The description and printoffs that the objectors above are ever so eager to direct our attention away from.)

2: Camel no 4: distractions over code and algorithms

The primary fact is that Weasel is a confessed, targetted search which makes cumulative progress to target, even through generations championed by “nonsense phrases.” Something which by CRD’s confession is “misleading” due to the long term targetting and associated artificial selection.

Indeed, CRD also highlights that the targetting and artificial selection make a big difference to time to target: tha tis we have a case of active informaiton in action.

This, we can see form BW, and it is apparently necessary to give it again, as last cited at 161:

———————————

>> It [Weasel] . . . begins by **choosing** a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – ‘mutation’ – in the copying. *The computer [indirectly, the programmer!]* ***examines*** *the mutant* ***nonsense phrases****, the ‘progeny’ of the original phrase, and* ***chooses*** *the one which, however slightly, most resembles* ***the target phrase***, METHINKS IT IS LIKE A WEASEL . . . . **What matters is the difference between the time taken by cumulative selection, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of single-step selection** [i.e. "more than a million million million times as long as the universe has so far existed" -- this is where CRD admits that active information has played a key role in the speedup] . . . >>

———————————

As such, Weasel c 1986 as presented is “fair game” for an analysis as-is, on the implications of the active information manifested by such cumulative, evidently ratcheted and latched search on mere proximity not relevant complex functionality.

And, since the analysis of ratcheted progress to target does not depend on whether the latching is or is not implicit or explicit — these are mechanisms to get to the observations of evident latched, cumulative, ratcheting progress to target c. 1986 — then, how that latching is achieved is irrelevant to the point that active information is a key reason for the performance above unassisted random search, and to quantify the injected active information.

[. . .]

185

kairosfocus

08/30/2009

4:42 am

3: Selected bloopers (too many to address one by one . . . :

a: BB@ 168: *An explicit, required, latching mechanism is the same as non-explicit, non-required, not-always-latching behaviour*

–> Not at all: Strawman. (This, buy one who has already stepped outside the pale of civil discourse.)

–> For the purposes of the M & D analysis, how latching is achieved for showcased runs is irrelevant. (remember, the issue is to account for the SHOWCASED runs, which will not necessarily be typical. [THAT IS PART OF WHY CRD'S CODE C. 1986 WOULD BE HELPFUL.])

–> Once latching evidently exists, the analysis applies.

–> And, let us recall: CRD in 1986 inadvertently admitted that the speedup was due to the targetting on proximity. That is, active information.

b: *A mutation rate that has to be between zero and one hundred percent is the same as a mutation rate that has to be either zero or one hundred percent*

–> Strawman, again. A caricature loaded with ad hominems is being set up.

–> How latching is achieved is — for the purposes of the actual analysis — irrelevant to that it is achieved.

–> This is now willful obtuseness, as well. [One who is involved in a civil discussion has a duty of care to seek to understand an interlocutor, not to twist words taken out of context to suit one's self. But then, sadly, BillB has long since demonstrated want of civility and a habit of twisting words self-servingly to accuse falsely.]

c: *A population of one, where no selection can occur is the same as a population of many from which one is selected*

–> Again, it is quite evident from p. 1055 as already analysed, that M & D gave a simplistic pedagogical example of what partitioning looks like. (Of course they did not reckon with the sort of word twisting rhetoric we are seeing in this thread and doubtless elsewhere. I am sure the IEEE engineers looking on are beginning to see what is wrong in the state of Darwinland.)

–> A simple scrollup to 162 will suffice to show that a mutation of five letters going correct at once would only be plausible for a very large population indeed, with an aggressive selection filter. So, the notion of a population of one is read into the example, not drawn out of it and its context.

–> And, the other half of this objection, that Q is number of mutants to date, then runs into the point that as discussed above, we would then have Q = G* S.

–> the rest of BillB's analysis collapses due to strawman premises.

d: *WEASEL is an algorithm defined by Dawkins in The Blind Watchmaker. Many people have written software based on it, and others have made software inspired by it, but which employs different strategies. There is only one WEASEL though and it is described quite clearly by Dawkins.*

–> from the above, it is clear that there are many legitimate ALGORITHMS for Weasel that will fit with Dawkins' description c 1986.

–> In short, this is mere caricature.

e: Rob @ 170:*That's a nice theory, but it's belied by the fact that both Dembski and Marks have stated that this algorithm is, in fact, Dawkins' WEASEL algorithm. Dembski has been saying it for years, even after correction. The EIL website still says it.*

This is what The EIL page linked by Rob says: *First, let's look at **partitioned search** used by Dr. Dawkins. Assuming uniformity, the probability of successfully identifying a specified letter with sample replacement at least once in Q queries is . . . [leading up to the same summarised math as appears in p. 1055 of the IEEE paper]*

–> Of course, the "belying" is based on forcing the observation of partitioning — i.e evident cumulative, ratcheted search that advances to a target on proximity as showcased c 1986 — into a particular algorithm that implements it; and algorithm that does not appear in the IEEE paper, nor for that matter in the EIL page as linked.

–> AND, we note that it is demonstrated that implicit latching is possible, which will produce the same run of champions effect as seen in the 1986 showcased run excerpts. [**And if the 1986 run excerpts were ATYPICAL BEHAVIOUR (which is what some above suggest, while claiming that he program did not explicitly latch), then, that raises questions on the integrity of the Weasel program as presented at that time**; questions that should be answered by opening to public inspection credible code. Recall, Joseph and I have been trying to account for the observed behaviour c 1986, on the claim that CRD did not explicitly latch his program c 1986. If implicit latching is possible but atypical, that itself raises questions about what was going on in the showcased 1986 runs.]

–> In short, more word twisting and strawmanising. With some troubling possible implicaitons that call for credible code c 1986.

f: mb, 171: *In the Blind Watchmaker video… you do realize that you can see the "correct" letters briefly change, right? . . . Say what you will about the applicability or accuracy of Dawkin's program, but it obviously doesn't lock the letters once they're right.*

–> this of course raises the issue of the apparent gap between the showcased runs c 1986 and the video c 1987.

–> the first serious option is that the 1987 video is a detuned run of Weasel that shows unlatched behaviour due to the detuning from the matched pop size, mutation rate and filter in 1986.

–> the second [suggested by an objector to the idea of implicit latching], which does account for the winking effect, is that we are looking not at generation champions, but at the raw members of the population. (This runs into the problem that if MB's observations are accurate, then mutation per letter rate is rather high — known to lead to one form of detuning and non-latched behaviour.)

g: OS, 176:*You seem to think there's huge value in the cultural war in presenting Dembski as an inerrant genius*

–> Where have I ever said or implied such? [I think I am on record that we are all finite, fallible, and indeed fallen.]

–> I happen to think that in this case, the M 7 D analysis — as opposed tot he caricatures presented above — is reasonable, and have given my reasons.

–> And, pardon my suspicions when I see talking points of the now all too familiar form *"I am a supporter of X, but I think the supporters of X are idiots or worse . . . "*

–> BTW, Weasel c 1986 is admittedly targetted search that rewards mere proximity.

h: Indium, 174: *an algorithm that doesn´t protect correct letters in a search is the same as one that does (the famous implicit latching!)*

–> Notice how this strawman distortion [cf bloopers a - c supra] has now become a repeated mantra, to be taken as gospel truth on the power of sheer brassy repetition in the teeth of the facts. [I shudder to think of what is going on in Darwinland echo chambers on this . . .]

–> Indium, in case you don't recognise the tactic, this one is called the big lie, adn I need not list its well-known exponents — who BTW, projected it unto their intended victims, instead of telling the truth that they were the ones using it [in short, turnabout tactics].

–> Please, don't be taken in by it.

–> And as for the "lost in the laugh" remark above, that too is a well-known agit-prop tactic. Stop laughing and start reading more carefully to UNDERSTAND before you criticise, please.

[. . .]

186

kairosfocus

08/30/2009

4:45 am

i: *you have redefined "query" to mean the determination of the next parent string. Everybody can look at Atoms GUI to see that you are wrong of course.*

–> The problem at root is in the absurdities thrown out by trying to read the didactic example as an algorithm.

–> Where I do have what I think on further thought overnight is an error, is that I have taken Q to be number of generations at the first, in that context. Q is — on second thought — number of generations multiplied by size of generations.

–> And, that makes no difference to the evident pattern of ratcheting from generation to generation in the line of champions [which is what the printoffs c 1986 show; and what it therefore the empirical foundation of all discussion] that is a key part of the analysis, or tot he comparison of effect of active information based vs random search, as Q is in any case consistent across the two.

j: *Somehow the paragraph/pictures/formulas in the D+M paper are only a pedagogical something that not criticizes Weasel directly but is somehow relevant anyway*

–> Utter misrepresentation, laced with ad hominems. (Advice: If you do not understand, ask, don't assert. Please.)

–> A reading of 161 ff, for instance [not to mention all the way back to the always linked app 7] will easily and clearly show that I have said that the didactic example presented of what partitioning of a search looks like, is not a realistic representation of an algorithm, but a simple illustration of a behaviour of observed outputs: *in ratcheted searches, once letters go correct they are preserved correct, and more and more letters go correct and are preserved until the target phrase is complete.* [And, M & D say just about as much in pretty close words to what I have just said.]

–> once that is seen, we can see the relevance and accuracy of the basic analysis of what such a ratcheted search looks like probabilistically, on the mere fact of ratcheting. (Which raises no commitments on what the ratcheting comes from, whether explicit or implicit.)

–> And, I have said precisely nothing about the onward analysis in the IEEE paper, as this has not come up, i.e you are putting words in my mouth above, words that simply do not belong there.

k: TA, 175: *How about this then: Latching occurs when the probability of moving "up" the gradient is greater than the probability of moving down. This would again seem to include any search other than a blind random walk.*

–> Remember, we are starting form observing an evident o/p pattern, per showcased examples of "cumulative selection" in action.

–> in those examples, for 200 cases were letters go correct, and can revert, the excerpts never show a reversion. And since listing every 10th generation's champion is unlikely to correlate with the search process, then we can infer that the description and the showcased runs coincide: there is ratcheted progress to target.

–> After that, we look a the "occasional slips" case; one that is also observed on producing a program capable with certain parameters being matched, of latching. this makes sense,a s probabilistic barriers are not absolute. [All the oxygen molecules in the room where you sit can conceivably rush to one end, leaving you choking; no physical barrier absolutely forbids that. But, that is rather unlikely, and unobserved.]

–> And a third case is possible, where there is no evident ratcheting.

l: YD, 179: *RD says he no longer has the original program but that it did not "latch". Not that latching vs. nonlatching is that important in the grand scheme of things, but it's helpful to get confirmation of this detail from the source.*

–> And how did CRD explain the showcased runs and gushing remarks on the wonderful power of cumulative selection c 1986?

–> Other than, that he is claiming that he did not EXPLICITLY latch the program, which would be the same thing he is reported to have said c 2000. [In short, the issue of implicit latching is still very much on the table, and recall, such behaviour on "good runs" is DEMONSTRATED. If CRD's actual o/p's on the showcased runs did not latch, implicitly or explicitly, then to present them as if they did while gushing on the power of cumulative selection will require a bit of explaining on how the results and remarks were not presented in a misleading manner.]

–> In any case we have it that no credible code will be forthcoming. **Contest over, unless someone can dig up a credible copy from somewhere that has a reasonable chain of custody.**

m: OS, 180:*Dembski, observing that something possible in execution of the algorithm was not evidenced in the sample, inferred that the program did not implement the stated algorithm*

–> Bold denial of stated facts on the record.

–> the claim by Dawkins p 48 ff of BW, was that Weasel exhibited "cumulative [and targetted] selection," based on proximity to target, which conferred a major advantage over "single step selection." Where, cumulative NORMALLY means: *Increasing or enlarging by successive addition.* [1st meaning AmHD.]

–> in support of this, he produced listings c 1986 in BW and NewScientist, that showed over 200 cases of letters going correct and then open to reversion, without a single reversion in evidence; on two runs, one of 40+ and one of 60+ generations. [That OS thinks there was only one published run shows that he has not investigated carefully before commenting adversely.]

–> On such — multiplying the two lines of evidence together — it is evident beyond reasonable dispute that Weasel c 1986 generational champions ratcheted to target with associated latching of successful to date letters on "good" runs.

–> on “forensically” reconstructing the algorithm to do that, two main approaches are possible: explicit latching and implicit where the pop per gen, mut per letter rate and filter interact to at least some of the time give runs that ratchet. Both have been demonstrated and are legitimate readings on the evidence of 1986.

–> on subsequent statements (and possibly the 1987 video] the latter is the — on balance of evidence — best explanation for the observed published runs and descriptions c 1986.

–> the rest of OS’s case foes downhill from there, repeating a now familiar line of talking points.

+++++++++++++

After taking time to go through the above, I am still shaking my head.

GEM of TKI

187

kairosfocus

08/30/2009

5:52 am

PS: OS, If you rake time to look at my characteristic thought on ID [e.g through the always linked] in terms of functionally specific, complex information and its roots in thermodynamical and informational thinking (which trace back through Thaxton et al, not Dembski), as well as my related look at the Caputo case, you will see that my thought is significantly independent from that of Mr Dembski. I happen to think that Mr Dembski — though finite, fallible and fallen as we all are — has got some things right, things that are too often caricatured and wrenched by objectors to improperly dismiss them through strawman fallacies. And the habit of such strawmannising by denizens of Darwinland is I believe abundantly evident above.

188

kairosfocus

08/30/2009

6:08 am

PPS: I also happen to think his partnership with Dr Marks has enriched his work. And, that on a topic known for months to be controversial, there would have been significant cross-checking before publication. Now, compare that with the sort of srtrawmannising above, and it should be evident why I draw the conclusions I do on who is more likely to be correct in this case. then, multiply by the obvious didactic example context of the alleged algorithm that they are being castigated for. And mix in the fact that explicitly AND implicitly latched weasel programs have been demonstrated on actual runs — programs sponsored on the web by the same EIL. After such factors are in evidence, what makes the best overall explanation? On what grounds?

189

Blue Lotus

08/30/2009

6:08 am

Kariosfocus

> –> on “forensically” reconstructing the algorithm to do that, two main approaches are possible: explicit latching and implicit where the pop per gen, mut per letter rate and filter interact to at least some of the time give runs that ratchet. Both have been demonstrated and are legitimate readings on the evidence of 1986.
>
> –> on subsequent statements (and possibly the 1987 video] the latter is the — on balance of evidence — best explanation

> for the observed published runs and descriptions c 1986.

Richard Dawkins has responded and has said

> –> on "forensically" reconstructing the algorithm to do that, two main approaches are possible: explicit latching and implicit where the pop per gen, mut per letter rate and filter interact to at least some of the time give runs that ratchet. Both have been demonstrated and are legitimate readings on the evidence of 1986.
>
> –> on subsequent statements (and possibly the 1987 video] the latter is the — on balance of evidence — best explanation for the observed published runs and descriptions c 1986.

http://scienceblogs.com/pharyn.....nt-1887052

And

> http://www.youtube.com/watch?v=5sUQIpFajsg
> About 6:10 seconds in, watch the 'Darwin' algorithm home in on 'METHINKS IT IS LIKE A WEASEL'. You will see (for instance at 6:16) that the program does not 'latch', because the W of weasel mutates and then comes back. It keeps winking on and off from W. Clearly no latching. However, as PZ and others have said, it really doesn't make a lot of difference whether the program 'latches' or not. These people are so unbelievably stupid.
>
> I claim my trip to the Bahamas.
>
> Richard

http://scienceblogs.com/pharyn.....nt-1886689

Note that Richard says "the" program not "This version of Weasel (which is different to the version on TBW)". Therefore according to Richard Dawkins there is only 1 version of Dawkins' Weasel and it behaves as per the video (no latching).

Will you now withdraw your opinion or continue to say that Richard Dawkins does not know how Richard Dawkins Weasel operates?

190

Blue Lotus

08/30/2009

6:13 am

The first Dawkins quote should have been

> Alas, I no longer have the original program. It seemed too trivial to be worth keeping. Obviously any half way decent programmer could knock it up in a trice.
>
> Fortunately, however, there is a film of the program in operation (see post #53) above, and you can clearly see from the film that there was no 'latching'.

http://scienceblogs.com/pharyn.....nt-1887052

191

Blue Lotus

08/30/2009

6:26 am

KariosFocus,
I ask you one more time.

If I have two programs.

Once always outputs data similar to

```
Generation 1: XXXYYYXXXZZZ
Generation 2: XXXYYXXXYZZY
```

The other always outputs data similar to

```
Generation 1: XXXYYYXXXZZZ
Generation 2: REJSJTHVHASG
```

Would any reasonable person claim that they were in fact the same program producing similar outputs?

No.

Yet given two programs where the outputs are

```
Generation 1 Y*YVMQKZPFJXWVHGLAWFVCHQXYPY
Generation 10 Y*YVMQKSPFTXWSHLIKEFV*HQYSPY
and
Generation 1: XYBPFPBMCLPGFONJYWFKXPFOVMKDX
Generation 2: XPJSQJMGPRYPONAIRJSGXZRWQJQBX
```

you continue to claim that they are the same program!

Please note the latter set of results were generated via WeaselWare here
http://www.evoinfo.org/WeaselGUI.html
So they are valid results. The previous generation 1 and 10 are obviously from TBW.

Please explain Kariosfocus how these two outputs are in fact being generated by the same program?

192

Indium

08/30/2009

6:49 am

kf,
anorther few thousand words ignoring my question, as expected. Too keep your evasion on display I will just repeat it: Which one of Atoms sehr algorithms is Weasel, which one the partitioned search? Are they the same?

I will just briefly comment on a few of your points, there is too much noise in there to spend more time on it. Whenever you are ready to make a clear argument without all this obfuscation I will again be more exhaustive.

> implicit latching is an observed phenomenon, one that answers to CRD's enthusiastic description and showcased printoffs.

Of course it is: Nobody denies that evolution or Weasel are highly efficient searches. We are denying that that

193

kairosfocus

08/30/2009

7:02 am

Indium:

You first.

I have repeatedly pointed out to you that most of your relevant queries were long since answered, starting with the always linked.

So, your mantra on “ignored” is rising to the level of reite3rated false allegations, i.e slander.

Kindly stop it. Point out where my answers are INADEQUATE on the merits — if you can — then you may go on from there.

GEM of TKI

194

Indium

08/30/2009

7:13 am

kf,
anorther few thousand words ignoring my question, as expected. Too keep your evasion on display I will just repeat it: Which one of Atoms sehr algorithms is Weasel, which one the partitioned search? Are they the same?

I will just briefly comment on a few of your points, there is too much noise in there to spend more time on it. Whenever you are ready to make a clear argument without all this obfuscation I will again be more exhaustive.

> implicit latching is an observed phenomenon, one that answers to CRD’s enthusiastic description and showcased printoffs.

Of course it is: Nobody denies that evolution or Weasel are highly efficient searches. We are denying that that Weasel protects correct letters from mutating.

> The primary fact is that Weasel is a confessed, targetted search

Confessed targeted search? What are you talking about? It´s written with big letters in the book, there is nothing to confess at all! And this is not the question at hand anyway. The question is: Doe D+M misrepresent it?

> As such, Weasel c 1986 as presented is “fair game” for an analysis as-is, on the implications of the active information manifested by such cumulative, evidently ratcheted and latched search on mere proximity not relevant complex functionality.

This alone is enough to amke any further argument with you rather unnecessary. Weasel is completely misrepresented in the paper by D+M. It´s not fair game to just plug in a different algorithm that happens to also keep correct letters (which Weasel sometimes even doesn´t do) and say your discussing Weasel. Also, please notice that D+M explicitly cite TBW and D announced on this blog that his paper criticizes Weasel and therefore even evolution. They give very specific results for the active info, which are plainly wrong for Weasel. Eq 22 is in no way consistent with Weasel (independend from latching). They are discussing a completely different algorithm, it is not fair game at all. That you consider this to be fair game speaks volume about a few things…

> how that latching is achieved is irrelevant to the point that active information is a key reason for the performance above unassisted random search, and to quantify the injected active information

Quantifying active info for different searches is the main point of the paper by D+M. And yet the results for the partitioned search and Weasel are completely different which you would notice if you would just fire up Atoms software and check it. Or you can verify that Eq 22 is simply wrong for Weasel (latching or not latching, doesn´t matter) and therefore also the calculation is wrong. That does not imply that the active info concept can´t be applied to Weasel, I am sure it can be done! But it´s done wrong in the paper because Weasel simply is no partitioned search. Ask Joseph.
kf, honestly, these things are now so clear to see for everyone that you are just damaging your reputation.

The discussion and formulas in the paper are not correct for Weasel because quite simply Weasel is no partitioned search. Which is again a good point to repeat my question: Which of the Search Algorithms in Atoms suite is Weasel and which one the partitioned search? Maybe asking it two times in one post helps you to focus.
Oh, and the bonus question: Do they seem to have the same amount of active info?

195

Indium

08/30/2009

7:17 am

Oh, sorry for the doulbe post.

196

Blue Lotus

08/30/2009

7:17 am

Kariosfocus

> So, your mantra on “ignored” is rising to the level of reite3rated false allegations, i.e slander.

One could say the same about you.

You have ignored Dawkins own comment on latching and Weasel.

You have ignored comments that you are only seeing 0.047% of the total data and making assumptions based on that tiny fragment and applying the to the whole data set, as per comments #41, #78 and #149.

You have ignored Indium’s simple question – “Which one of Atoms algorithms is Weasel, which one the partitioned search? Are they the same?”

You have ignored my question #190, asked several times elsewhere and answered with a flurry of strawmen but no comment on the substance of the question itself.

> It is sadly evident from the above that we have much of straining at gnats while swallowing camels, as well as barking up wrong trees. Etc, etc.

Perhaps you would be better served by addressing the issues on their merits rather then trying to accuse everyone who disagrees with you of slander.

197

Indium

08/30/2009

7:19 am

kf, you can I point you to where you ignore my question? You ignore it in all your thousands of words. Why not just answer it? Which algorithm from Atoms suite is Weasel, which one the partitioned search? Are they the same? Same active info? Same number of queries?

198

Indium

08/30/2009

7:20 am

Sorry again. First sentence should begin with “how”, not “you”.

199

DNA_Jock

08/30/2009

7:59 am

To Oatmeal’s excellent history of the whole affair at 181 , I would only add the following- the search shown in TBW **cannot** be a partitioned search, for the reasons Blue Lotus points out in 190 – the difference between generation 1 and generation 2. Something that D&M should be aware of.

Kf has carefully avoided my post (164). He keeps saying “once latching occurs the D&M analysis applies”. **This has been demonstrated to be false.**

Unfortunately (for him) he has now been backed into the position that Q is the total number of queries.

> kf@185:
> Where I do have what I think on further thought overnight is an error, is that I have taken Q to be number of generations at the first, in that context. Q is — on second thought — number of generations multiplied by size of generations.

I insist that anyone who is having trouble following the math here, just go the
http://www.evoinfo.org/Resourc.....elGUI.html
and run a search for ‘Methinks it is like a weasel’. Then compare the Query Count for the Partitioned Search (about 98) with the Query Count for the “Proximity Reward Search” (which kf states (@163) is the Weasel algorithm)

These were my runs (using G=50, MR=4%):

> Target: METHINKS IT IS LIKE A WEASEL
> Search Type: English Phrase
> Partitioned / Deterministic / Prox Reward
> 1) 68 25 5850
> 2) 85 26 4400
> 3) 94 27 6800
> 4) 156 27 6100
> 5) 88 27 7800
> 6) 78 23 11950
> 7) 96 26 4800
> 😎 90 27 9800
> 9) 100 27 4000
> 10) 93 26 5600
> 11) 62 27 8700
> 12) 127 27 4500
> 13) 85 27 10150
> 14) 115 27 5200
> 15) 53 27 11850
> 16) 90 27 6350
> 17) 98 26 13950
> 18) 83 26 5650
> 19) 160 25 7700
> 20) 121 27 5700

So partitioned takes between 60 and 160 queries, but Weasel(50,4%) takes (using kf’s definitions throughout) between 4,000 and 13,000.
These are “equivalent”?
But don’t rely on anyone’s assertions – **try it for yourself!**
Enjoy!

200

kairosfocus

08/30/2009

8:04 am

BL:

Your links above and quotes are, unfortunately, not helping us resolve the evident differences between the showcased runs of 1986, and the videotape of 1987.

In short, the cites from Mr Dawkins raise — or miss — more questions than they answer. [And the ad hominems are unfortunately of a piece with his longstanding contempt for those who differ with him: *ignorant, stupid, insane or wicked.* Mr Dawkins: FYI, I have what seem to me to be some serious questions, and I am not simply setting out to smear and distort, I would like to have them resolved in a positive fashion. In that light I do not find off- the- cuff prejudice and dismissal civil; but instead to sound far too much like the incivility and regrettably ill-informed commentary that I and others, e.g, start here, find in your The God Delusion.]

Some of the following has been raised several times but it will plainly bear repetition:

1 –> the showcased runs of 1986, with gushing commentary on the wonderful power of CUMULATIVE selection, demonstrate that there is good reason to infer that the showcased runs did ratchet and latch their way to the target:

We may conveniently begin by inspecting the published o/p patterns circa 1986, thusly [being derived from Dawkins, R, The Blind Watchmaker , pp 48 ff, and New Scientist, 34, Sept. 25, 1986; p. 34 HT: Dembski, Truman]:

```
1 WDL*MNLT*DTJBKWIRZREZLMQCO*P
2? WDLTMNLT*DTJBSWIRZREZLMQCO*P
10 MDLDMNLS*ITJISWHRZREZ*MECS*P
20 MELDINLS*IT*ISWPRKE*Z*WECSEL
30 METHINGS*IT*ISWLIKE*B*WECSEL
40 METHINKS*IT*IS*LIKE*I*WEASEL
43 METHINKS*IT*IS*LIKE*A*WEASEL
```

```
1 Y*YVMQKZPFJXWVHGLAWFVCHQXYPY
10 Y*YVMQKSPFTXWSHLIKEFV*HQYSPY
20 YETHINKSPITXISHLIKEFA*WQYSEY
30 METHINKS*IT*ISSLIKE*A*WEFSEY
40 METHINKS*IT*ISBLIKE*A*WEASES
50 METHINKS*IT*ISJLIKE*A*WEASEO
60 METHINKS*IT*IS*LIKE*A*WEASEP
64 METHINKS*IT*IS*LIKE*A*WEASEL
```

[Cf my comments in the always linked app 7]

2 –> This pattern from 1986 — 200+ candidates that could revert in the samples if no latching is happening, but do not ever do so [consider on the import of the law of large numbers on this; i.e the simplest explanation that fits the facts is that the sample reflects the run of generational champions here . . .] — cannot simply be dismissed by making an offhand comment on events of 1987. *For, it is known separately that adjustment of parameters and/or different runs will produce cases that latch and cases that do not.*

3 –> In the case of the 1987 run, if the intent by Mr Dawkins is to say that he is showing the run of generational champions [a la 1986] — and that seems hard to avoid — then he is not explaining an apparent difference in showcased runs c 1986 and 1987, but distracting from it. (And Mr Dawkins is known to make significant use of misdirection and mischaracterisation in his public discussions, as the linked rebuttal to his The God Delusion documents.)

4 –> If instead, CRD is showing the run of individual members of the population in the video, that these do not latch letters is a built in characteristic of the implicit latching case: it is the run of champions — the evidence that we are seeking to explain — that would show latching in the case where: [1] to a sufficiently high likelihood no-change members appear in the generation’s pop, [2] single step advances prevail otherwise. On this case, we will see latching of the run of champions as the current state will be locked in by [1] and advances if they happen will appear at [2] preserving the advances to date and occasionally adding to them. (And in some cases one run will latch and another will not on similar parameters. So, we are back at: wha tis being showcased in 1986 and in 1987, and how represenattive of the runs and population is that.)

5 –> Not to mention, how does the behaviour ofthe population fit with the gushing claims from BW that accompany the above runs:

>> It [Weasel] . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – ‘mutation’ – in the copying. The computer [indirectly, the programmer!] examines the mutant nonsense phrases, the

'progeny' of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL . . . . What matters is the difference between the time taken by cumulative selection, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of single-step selection [i.e. "more than a million million million times as long as the universe has so far existed" -- this is where CRD admits that active information has played a key role in the speedup] . . . >>

____

6 –> If double changes etc appear with sufficient frequency, we will see another demonstrated characteristic: substitutions where one letter advances, and another retreats — which then tend to run across decadal generation boundaries. [this would be in either a quasi-latched or an unlatched case.]

7 –> In the showcased 1986 runs, it took 40+ and 60+ generations to hit target, so that no-advance won the generation contests about half the time, i.e. we have separate reason to see that we may well be in the implicit latching regime described.

8 –> And, if the showcased runs were unrepresetnative of he performance of Weasel c 1986, then CRD has some expalining to do on his gushign comments about cumulative selection.

9 –> Lastly, he seems to be suggestig that he weawsels out there more or less capture the patterns of behaviour. In this case, then explict latching and implicit latchigtn are legitimate interpretations, and ont he evidence that he presented he4 has showcased implicitly latched runs c 1986, and either internal members of po per gen in 18987 or a case taht is distinctly dfifferent formt he 1986 runs.

10 –> Or else, if the showcased 1986 runs did not ratchet to target, then his published excerpts were misleading – and that in a context that he admitted was in the broader sense "misleading."

GEM of TKI

PS: BL, 190: You are missing the point, and are reading INTO the remarks in the paper something that is not drawn out of it. This happens fairly frequently, so we do have to watch out for it.

Try reading on a different "scenario," and see if this helps you see where I am coming from: *"On p. 1055 M & D presented a didactic example of what partitoning of a search means, not a real world output of a program or illustration of an algorithm's typical output in one step."*

This, I have pointed out step by step from 161 above on, as a more credible explanation. In short, I see from my angle that on inference to best explanation, M & D are NOT presenting an algorithm or program or output from a run, but they are explaining a concept, then they go on to use the evident ratcheting as given above for Weasel 1986, to show its mathematical consequences.

Runs from 1987 are irrelevant to the 1986 result which do show no reversions among the run of champions, in a sufficiently large sample that with the gushing commentary on cumulative selection, one may reasonably conclude that the program runs showcased did latch or ratchet, manifesting partitioning explicitly or — more likely on balance of evidence and testimony — implicitly achieved.

That is the M & D Analysis is about what searches that show cumulative, ratcheted, latching selection do, and is thought to be relevant to the 1986 Weasel runs as showcased for reasons of the size of sample [as measured in letters] and the lack of reversions therein. Such ratcheting can be achieved by explicit or implicit latching effects, and on balance of evidence the later is most likely.

In any case the underlying analysis is relevant and applicable as CRD's commentary in BW reveals — inadvertently — that the active information supplied by use of a preloaded target and a proximity hotter-colder metric and filter, makes the program outperform random walks by a wide margin.

201

DiEb

08/30/2009

8:19 am

Your links above and quotes are, unfortunately, not helping us resolve the evident differences between the showcased runs of 1986, and the videotape of 1987.
The differences can be easily explained by the size of the population used in both examples:
Using a mutations probability of 4%, a population of 10 takes 1305 generations on average, a population of 100 takes 78 generations.
Obviously, Dawkins wanted to have the program running for the length of his talk, so he chose a smaller population than for the book.
I'm sure that Dawkins would answer along these lines – if he was asked.

202

DiEb

08/30/2009

8:21 am

BTW, these numbers are calculable, you don't have to take my word for them…

203

Indium

08/30/2009

8:24 am

kf, you are so fixated on the latching stuff and the imagined differences between 1986 and 1987 it is honestlyn amazing. And of course you still avoid my questions! Hilarious!

You are more and more retreating to the strange position that the algorithm that D+M describe is not really Weasel (admitting what we say all the time) but is still relevant even if the math is completely incorrect. Could you elaborate? How much active info does the real Weasel then have? Why don´t D+M discuss Weasel when say reference it and boast about their critic on this website?

So, have fun discussing this with Dr. Dembski, who seems to think that he criticized Weasel and not something else.

204

Tomato Addict

08/30/2009

8:35 am

@kairosfocus, #185

By all means, the behavior you call latching exists. You have twice demonstrated such and ignored the question. It appears the definition of latching/ratcheting in effect (explicit, implicit, or quasi-) is that if the search moves toward the target, then "it must be latching because it looks like latching". I suggested a more formal mathematical definition, and I am unaware of any other definition.

I have to conclude that the concept of latching and ratcheting is arbitrary, or that it is a property of all search algorithms other than blind random search. Either way, it seems to be a useless concept.

205

DiEb

08/30/2009

8:47 am

–kf,
I just wanted to add my voice to the choir:

> Which algorithm from Atoms suite corresponds to Weasel and which one corresponds tpo the Partitioned Search discussed by D+M?

206

BillB

08/30/2009

9:23 am

Tomato Addict:

> I have to conclude that the concept of latching and ratcheting is arbitrary, or that it is a property of all search algorithms other than blind random search. Either way, it seems to be a useless concept.

I think KF has even tried to imply that a blind random search has a latching mechanism here:

> One of the peculiarities of Weasel algors, is that they halt when they hit home; of course halting being a key property of algorithms. As a result, if they hit home before a generation number G, they do not get to G. Thus, if Q means number of mutants to date, then if size of generation is S, Q = G*S.
>
> Immediately, if a run completes [all L letters correct] at gen G, it cannot have been complete before G.
>
> So, the paper's p.1055 discussion would on this reading of Q would be of a ratcheted run that shows a march of champions with latching up to G, when it halts: completion AT G. (In an explicitly latched Weasel this would be automatic, in a version that on being tuned and giving a good run latches — as observed [our first swallowed camel . . .] — this implies that when metric falls to distance to target = 0, there is a latching action imposed.)

KF is trying to make an art out of semantic mud wrestling. He is claiming that if an algorithm has a halting condition then this is a latching mechanism and because it is assumed that both Dawkins and D&M's algorithms are assumed to halt when they find the target, they both contain a latching mechanism. KF presumably doesn't regard the fact that this latching has no affect on the search as important.

Dawkins also doesn't specify a halting condition and one is not actually required, although it would seem daft not to write one in. You can leave WEASEL running after it has found the target and it will continue to produce new generations, and the fittest will sometimes show incorrect letters. This becomes vital if you change the target to a new phrase because WEASEL will start to converge on this new target.

Contrast this with Dembski and Mark's algorithm. They appear to specify a halting condition "The search for these letters is over" but it is slightly ambiguous. I realise that you could interpret and code it either way as follows:

1

207

BillB

08/30/2009

9:40 am

(Hmm, for some reason it just posted my comment for me before I finished – here is the rest)

1 You code it so that there is a running list of incorrect letters. You work though this list on each iteration of the software and randomise letters on it. When the list is zero the program implicitly halts (or loops for ever doing nothing)

2 With each programme loop you check each letter, if it matches you do nothing, if it doesn't you randomise it.

Option 2 is simpler but functionally equivalent as far as I can see EXCEPT that option two will re-converge on a new target if one is set. The latching mechanism in option 1 is permanent but in option 2 it is only permanent (i.e. non reverting) if the target does not move. I think both could be argued as legitimate interpretations?

Of course there is still an explicit mechanism required for D&M's compared to Dawkins algorithms.

The issue over halting conditions is a rather sad but amusing attempt to stretch the idea of latching behaviour even further – it would apply to any algorithm that can achieve its goal and renders the idea of latching, in the context of D and M's representation of WEASEL, meaningless.

When looked at clearly – without KF's will-full attempts at obfuscation, his handy portable goal posts and the bulldozer he is using to try and re-shape the playing field – there is no way you can follow both algorithm descriptions and come up with the same code.

Onlookers may also have noted how he laces all his comments with poisonous accusations of dishonesty. An upstanding example of civil discourse for sure!

208

yakky d

08/30/2009

9:47 am

KF,

I think after reading the comments at Pharyngula, virtually everyone will agree that Dawkins' WEASEL did *not* use partitioned search.

That's really the relevant point here, not whether the algorithm involved quasi or implicit latching, concepts which don't appear in the paper and which D&M don't discuss.

209

BillB

08/30/2009

10:10 am

Its worth repeating this:

D&M's process:
–
With each programme loop you check each letter against the target, if it matches you do nothing, if it doesn't you pick a letter at random to replace it.

Repeat.
–

Dawkins method:
–
With each programme loop copy each letter of the previous winning phrase, with each letter copied having a probability (P) of being randomly replaced. Now add up the number of complete letters to get the fitness score. Perform this G times (where G is the number of generations). Now look through the list of scores and pick the highest, or any one of the highest if there are more than one.

Repeat.
–

So should we consider these two processes to be the same? They can’t be written the same way as code, they don’t function the same way and they don’t produce the same results.

Can they be considered the same for D&M’s purposes (according to KF). Well I don’t see how – the question about the amount of active info is very pertinent here – but this is also irrelevant to the main topic under debate, one which it is easy to loose sight of what with KF’s army of oily burning sixty foot straw men marching across these pages. D&M describe a different process than Dawkins does, the differences are not trivial, particularly to anyone familiar to the topic of search algorithms. Because of this their reference to Dawkins work is incorrect and inappropriate; it should be corrected – **and this will have no impact, as far as I can see, on the content of their paper.**

The surreal and farcical attempts to avoid this quite simple issue begs the question – is this really about search algorithms at all or is it about poking Dawkins with a stick.

210

Indium

08/30/2009

10:32 am

BillB, as far as I can see only kf continues with this marathon of distractions. I am quite sure that Dr. Dembski will of course correct his misrepresentatiojn soon, it´s easy enough! For a man with his integrity it will only take a few days or 1-2 weeks to prepare a correction. I would bet a (very small) amount of money that he has so far not been made aware of these problems.

So, may I ask again: Instead of waiting for Dawkins to show up here, which is very unlikely, why doesn´t Dr. Dembski explain the issue on his own blog? Can somebody ask him for a clarification? On the other hand he closed comments on the thread where he announced the paper. Thanks Clive and O´Leary for opening new ones, btw!

211

nephmon

08/30/2009

12:16 pm

Oops. Sorry, I didn’t realize before that kf is actually suffering from a pathology. I’m astonished at the stamina of those who are arguing with him, but somewhat perplexed as to why they bother. Is someone who is so hung up up on the specifics of a program written 20+ years ago, and whose algorithm has been shown time and time again to exhibit just the behavior that was claimed for it, really worth arguing with?

The really funny thing about all this is that RD’s claims for the program were so tiny, and his acknowledgment of its limitations so clear, you really have to sympathize with someone whom it causes to spill so much virtual ink.

Good luck with the debate, all!

(Yes, I don’t mention the Dembski algorithm because it is, of course, a huge irrelevancy.)

212

Excession

08/30/2009

12:26 pm

I’ve just been observing with a mix of amusement and astonishment. It is a vary minor but clear issue, these are two different algorithms, Dembski and Marks made a mistake in their citation, they should issue a correction, Kariosfocus can stop making a mockery of reason (and ID), everyone can move on.

213

Indium

08/30/2009

12:35 pm

nephmon,
observing how kf tries to defend some truly obvious errors with more and more distractions/obfuscations provides a certain amount of amusement to some people.

214

Cabal

08/30/2009

12:36 pm

Kairosfocus,

> In short, the cites from Mr Dawkins raise — or miss — more questions than they answer. [And the ad hominems are unfortunately of a piece with his longstanding contempt for those who differ with him: ignorant, stupid, insane or wicked. Mr Dawkins: FYI, I have what seem to me to be some serious questions, and I am not simply setting out to smear and distort, I would like to have them resolved in a positive fashion. In that light I do not find off- the- cuff prejudice and dismissal civil; but instead to sound far too much like the incivility and regrettably ill-informed commentary that I and others, e.g, start here, find in your The God Delusion.

I suppose your words are heartfelt, but wouldn’t it have been easier and more conducive to the matter at hand, once and for all to offer a couple of words just to answer that simple question:

> I ask again: Which algorithm from Atoms suite corresponds to Weasel and which one corresponds tpo the Partitioned Search discussed by D+M?

or

> Which of Atoms algorithms is the partitioned search and which one Weasel?

Since you have made a lot of references to Atom’s program:

> Nor is this theory, there are actual runs that do that. (HT: Atom and EIL.)
>
> using replication of results as a good cross-check, e.g through Atom’s adjustable Weasel, here.
>
> j] As already noted and linked, these patterns have been demonstrated through actual runs of Atom’s adjustable Weasel, from EIL

I think it would be a good idea to answer the question, or at least tell us why you don’t consider it relevant? I believe a lot of people are interested but I am afraid your performance so far has not convinced many of them.

215

Indium

08/30/2009

12:49 pm

To be honest, Cabal, the answers to these questions will probably no longer be of much interest now that kf has more or less admitted that D+M discuss something that is NOT Weasel. I bet he will still avoid them anyway, otherwise he will have to contradict himself even more.

In fact, all questions have now more or less been answered:
- the explicit/implicit latching obfuscation has been exposed (about 100 times now)
- it is obvious that Dembski and Marks don´t use a proper Weasel and therefore misrepresent Dawkins

The only one left on the sinking ship is kf, so who cares except for people interested in the argument regarding design?

216

Cabal

08/30/2009

12:56 pm

With computers being binary devices, how can 'implicit latching' be implemented?

To me, implicit latching have a ring like 'implicitly pregnant'?

217

R0b

08/30/2009

12:59 pm

kairosfocus:

> In short, more word twisting and strawmanising.

No, it's a simple statement of uncontroversial fact. The EIL page presents math (the same math as the IEEE paper) that purportedly apply's to *Dawkins*' algorithm. The simple fact is that it does not apply to Dawkins' algorithm. This is made abundantly clear by the fact that the math does not take into account population size or mutation rate. But if you still don't believe it, run the numbers yourself, as I did above. There is no controversy here — the numbers don't lie.

218

nephmon

08/30/2009

2:28 pm

Cabal: in this case, "implicitly" simple means "having the appearance of". If you look at the output of an unlatched weasel program using a suitable offspring population size and mutation probability, you will see that once a letter has matched the target, it will never be corrupted again in the next generation, giving the appearance that it's latched.

Of course, amongst the 200 (say) candidate offspring, many of them WILL have a corruption of already correct positions, but these ones never (or with a vanishingly small p, anyway) get selected to go forward to the next generation.

If you make the offspring population small enough (or the chance of mutation high enough), then you WILL see letters flipping between correct and incorrect, and indeed it will never converge on the target string.

I'm sure someone will say "Aha! So you have to TUNE the values to make it work!" to which I would reply, evolution is quite capable of tuning such meta properties of the process itself.

219

Tomato Addict

08/30/2009

3:11 pm

BillB writes> I think KF has even tried to imply that a blind random search has a latching mechanism here:

AND

KF writes> One of the peculiarities of Weasel algors, is that they halt when they hit home; of course halting being a key property of algorithms.

I think BillB is correct: This seems to be saying that any algorithm that halts is latching, and that Weasel is peculiar for doing so?

Therefore even a blind random walk that happens to find the target – and halts – is latching.

I state as a given that halting is a universally accepted property of good coding practice. However, since good practice is not requirement, any search in an infinite loop might be considered non-latching. If halting is latching, I could be forced to admit this meets my request (in #146) for an example of a non-latching search algorithm. It would be a really stupid non-useful example though.

Usefulness aside, by this definition I can now claim that all algorithms that halt, or might halt, exhibit latching. The very concept of latching and ratcheting is now even less meaningful than before.

@KF: You wrote a lot, and I am willing to give you the benefit of the doubt and allow the quote above is perhaps taken out of context. I don't think it matters. The point I would repeat is that latching and ratcheting are either undefined, arbitrarily defined, or so broadly defined (pick one), as to be useless.

220

tigerhawkvok

08/31/2009

1:40 am

Well, here's one that I wrote up. Non-latching, highly configurable, and doesn't require the start string to be 28 characters (it can be any length). You can furthermore choose penalty weighting, etc.

Should that work, I'd go with Dawkin's book. In any case, the program is really quite simple, and Panda's Thumb had a nice breakdown of the convergence times of latching and non-latching programs (though all with 28 character starting strings).

Direct program link here.

221

kairosfocus

08/31/2009

3:41 am

Mrs O'Leary (and onlookers):

Let us first draw attention to the issue for this thread: *absent someone out there having a copy that is currently unknown, no copy of Weasel c 1986 code will be forthcoming.*

It therefore remains to correct a few mis-impressions and to underscore the real central issue and achievement of the recent IEEE paper (and its stable-mates).

For, the clutch of papers introduce an extension to the achievements of design theory: ***active information*** *as an explanatory factor for the superior performance of intelligent search over blind, random search in large configuration spaces.* And, we may again underscore that Mr Dawkins, in 1986, inadvertently pointed the way to that conclusion when he developed Weasel as a targetted, proximity increment rewarding search for a preloaded target:

____

>> [Weasel] . . . begins by choosing a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. *The computer* ***examines*** *the mutant* ***nonsense phrases****, the 'progeny' of the original phrase, and* ***chooses*** *the one which, however slightly,* ***most resembles the target phrase***, METHINKS IT IS LIKE A WEASEL . . . . **What matters is the difference between the time taken by cumulative selection, and the time which the same computer, working flat out at the same rate, would take to reach the target phrase if it were forced to use the other procedure of single-step selection**: about a million million million million million years. This is more than a million million million times as long as the universe has so far existed . . . .

Although the monkey/Shakespeare model is useful for explaining the distinction between single-step selection and cumulative selection, **it is misleading in important ways**. *One of these is that, in each generation of selective 'breeding',* ***the mutant 'progeny' phrases were judged according to the criterion of resemblance to a distant ideal target***, the phrase METHINKS IT IS LIKE A WEASEL. Life isn't like that. **Evolution has no long-term *goal*** . . . In real life, the criterion for selection is always short-term, either simple survival or, more generally, reproductive success. [TBW, Ch 3, as cited by Wikipedia, various emphases added] >>

____

1 –> Now, clearly, Dawkins accounts for the advantages of his "cumulative selection" that rewards mere increments in proximity to "a distant ideal target" by "nonsense phrases" over the "single step selection" — i.e. random mutation that has to achieve significant function before it can hill-climb — he would dismiss by the active information contained in the target, the comparison to target and reward of hotter as opposed to colder NON-FUNCTIONAL phrases.

2 –> So, **the core point of the current series of papers was inadvertently acknowledged from 1986.** By Mr Dawkins.

3 –> Q: So, why has there been a debate over a plainly *secondary* (and often simply contentious and distractive) issue — latching and/or ratcheting — since December last?

____

ANS: i] Of course, the answer has long been accessible one click away, from App 7, the always linked.

ii] And, yes, there *is* a bit of Darwinist revisionism of history at work here, projectively shifting the locus of "obsession" to those who have pointed to the reason why explicit and implicit latching are valid interpretations of the description and showcased printoffs from Weasel c 1986.

iii] In the main, this is because the issue was raised again and again and again by Darwinist objectors to those who pointed to the evident ratcheting action published by Dawkins in 1986; objectors who to often insist on mischaracterisations and resist the most reasonable and simple corrections.

iv] Presumably, they raised it in the main because it is a point where they think they can discredit those who they object to, while focussing on what they want to highlight rather than the more embarrassing — for them — details of Mr Dawkins' statements as cited above.

v] Unfortunately for them, they thereby show for the discerning reader several of the standard Darwinist rhetorical tactics of misdirection, mischaracterisation (of both arguments and persons) and dismissal.

vi] For instance, they are unwilling to acknowledge that the plain fact is that in the published, showcased — presumably "good" but representative — print runs of 1986, the excerpted runs of champions show as a matter of fact that in 200+ cases of letters that if reversions from correct were frequent we would see a few reversions, we see NONE. (Cf. on the law of large numbers and observability of low-probability phenomena here in App 7. [Do we not find it interesting how adverse comments are routinely being advanced by the Darwinists without seeking to fairly represent those being criticised.])

vii] Equally, they do not wish to acknowledge that in his gushing comments on the prowess of Weasel's ***cumulative** selection* in BW, CRD used a term that normally means: "Increasing or enlarging by successive addition," which when multiplied by the highlighted runs strongly supports the conclusion that at minimum, "good" runs of weasel produced lines of generational champions that routinely behaved like the published excerpts.

viii] So, CRD has the dilemma of either using misleading examples/excerpts, or using misleading words. (And that in a context of acknowledging that the is being in key parts misleading — Weasel as described is intelligently designed, targetted search, not a good stand-in for chance variation and natural selection based on differential FUNCTIONALITY.)

ix] Similarly, there has been a campaign to discredit the concept of ratcheting or of latching. In fact these are quite simple: based on what we see in the 1986 runs of champions, when Weasel is working "right" it will hold on to correct letters, and it will add more until the target is achieved, generation by generation. As in what seems to be clearly going on in the 1986 runs — at least if we are unprepared to swallow the *"don't believe yer lyin' eyes"* rhetoric.

x] So far, we have highlighted the pattern observable/ supported by the showcased runs c 1986 and the commentary on them. The question is where did these come from?

xi] One obvious possibility is explicit latching, and this is a legitimate interpretation of the information c. 1986. However, on further commentary and claims c 2000 and currently from Dawkins and proxies, the other possibility seems to be more likely: IMPLICIT LATCHING; as is discussed in the App 7 and in earlier UD threads. (The mechanism that Darwinists seem to want to either dismiss as unreal or accept as trivial.)

xii] But in fact, since CRD has in effect accepted that Weasel type algorithms as implemented in recent years are more or less typical of what his program did, we can rely on the demonstrated and long since publicly posted fact of implicit latching shown by Atom's adjustable weasel. (I'll bet some onlookers would not realise that such is **demonstrated fact** from the objections above and elsewhere!)

xiii] Thus, we can *see* that indeed under certain conditions where the population size per generation, the mutation rate per letter and the filter work together right, the sequence of generation champions on closest yet approach to target will show latching and ratcheting. (How desperate are some objectors to dismiss the observed fact as non-existent or poorly defined!)

xiv] How this likely happens: under not too hard to achieve conditions (enough pop and a reasonably low mut rate)the mutant population form a seed will almost certainly contain at least one unchanged member — cf calculation and explanation in the App 7 on this.

xv] That means that practically, progress to date on proximity will be preserved, given the stipulation on the filter. And, if in addition, single step advances are favoured otherwise because the mut rate is fairly low [so that 1 mutation per child phrase is typical], then mutations that affectt he correct letters will tend to be retreats, and will be eliminated by the filter.

xvi] Most mutations of non-correct letters will make little or no difference to the proximity too (or if they hit a correct letter they will most likely revert it to incorrect for that member of the pop). *But when a newly correct letter arises and the presently correct ones are undisturbed for a member of a generation, we have a one-letter distance increment in proximity.* if such a member is present, it will be picked by the filter in the normal case. [Obserfve how int eh Weasel 1986 runs as showcased, we saw 40+ and 60+ gens to target for "good" runs. That is no-change won out about half the time, and single step increments in proximity dominated the rest.]

xvii] Of course, where mutation rate or population size is high enough, we will see occasional cases where double mutations pop up which may give a two-letter advance, or may substitute one correct letter for another: the latter marking a LOSS of latching behaviour.

xviii] On the other end, if the pop size per gen is too small, not enough members of population will be present to have a predictable pattern within the generation, so reversions may win, and latching vanishes again. (Hence the discussion on tuning or matching of pop size, mut rate and filter. The filter, of course, can also affect outcomes, as is mentioned in the App 7. As Apollos inadvertently demonstrated some months back, this can sometimes have a dramatic effect.)

xix] If reversions are rare, we see near-latching or as I have termed it quasi-latching with an occasionally slipping ratcheting action.

xx] Indeed, given that the above are probabilistic phenomena, do enough runs and you will see reversions in cases where latching "often" turns up. For a probabilistic barrier is not an absolute barrier.

xxi] That brings up the 1987 videotaped runs. Most likely, they showcase a detuned run, which does not latch implicitly, and the onscreen reversions may reflect that. (It is also possible that the video is highlighting the members of the population, which are not the same as the run of generational champions published in the 1986 printoffs. The two are not mutually exclusive.)

xxii] Beyond that, we may see behaviour that does not in any wise resemble latching, i.e cases with a rather high mutation rate will find it hard to close the deal, and maybe in some cases will never hit home to the target.

____

4 –> Now, *none of the above is really new*, having been put up several times — including in this thread. Not to mention, it is presented in the always linked app 7. Why then is it apparently so hard for Darwinists to see what should be obvious?

5 –> Partly, because it does not fit with what they expect to see. (It is hard to see what is really there when that does not fit with what you expect to see.)

6 –> For instance, just above we see TA @ 219: *The point I would repeat is that latching and ratcheting are either undefined, arbitrarily defined, or so broadly defined (pick one), as to be useless.*

7 –> Compare that ill-informed, counter-factual statement with the facts as again outlined. How could a reasonably informed person not see that here is a sufficiently precise definition, with publicly accessible demonstrations [starting with the showcased Weasel printooff excerpts of 1986]? And counter-examples that show contrasting cases?

> ANS: *because the actual facts cut across the expectations.* [Starting with "don't believe yer lyin' eyes" on what Weasel 1986 credibly did.]

8 –> In that context, it is easy to see as well why a focus on an imaginary flaw can distract from the actual outright acknowledgments in BW of:

> [1] use of artificial selection,
>
> [2] resulting material dis-analogy of this computer simulation to chance variation and natural selection,
>
> [3] the further resulting fundamentally "misleading" nature of the Weasel icon of evolution,
>
> [4] the actual reason for improvement in performance over blind search: active information in the form of a preset, embedded target and a hotter-colder beacon that draws in non-functional "nonsense phrases" until they hit home in a few dozen generations if the program is set up right.
>
> [5] the distraction from and dismissal of the need to de novo originate complex functional information through chance variation before hill-climbing/ warmer-colder algorithms etc can be legitimately employed.

9 –> And this last cluster extends directly to the more modern and more sophisticated Genetic Algorithms. GA's may reasonably mimic certain aspects of micro-evolution, but hey do not credibly account for the origin of complex information based functionality in the absence of intelligent action and the associated injection of active information.

10 –> And, that is a fatal defect.

GEM of TKI

PS: BillB of course — as is sadly usual — manges to excerpt out of context and mischaracterise. The context for my remark on halting was that because of halting, Weasel type algors will lock up as they hit home. that means that when we talk about probabilities of getting home AT gen x, we are talking about just that — we would not get *to* to gen X if there had been an earlier hitting of the target.

222

kairosfocus

08/31/2009

3:49 am

PPS: TH, you can see an implicitly latched run of Atom’s adjustable Weasel hit home in 31 gens here.

I reproduce, FYI, as it seems hitting on a link is not popular:

____

RUN B, 500 pop/gen, 4% per letter mut rate:

```
1. MEL LSI YHXMAJLMDGMVKTSKGW
2. MEL LSI YHXIAJLMDNMVKTSKGW
3. MEL LSI YHXISJLMDNMJKTSKGW
4. MEL LSI YHXISJLMDN JKTSKGW
5. MEL LNI YHXISJLDDN JKTSKGW
6. MEL LNI YHXISJLDDN JKTEKGW
7. MEL LNB BHXISJLDDN JKTEKGE
8. MEL LNB BHXISJLIDN JKTEKGE
9. MEL LNB BHXISJLIDN JKTEKSE
10. MEL LNB BHXISJLIDN JKTEKSEL
11. MEL LNK BHXISJLIDN JKTEKSEL
12. MEL LNK BHXIS LIDN JKTEKSEL
13. MET LNKV BHXIS LIDN JKTEKSEL
14. MET LNKV BHXIS LIDN AKTEKSEL
15. MET LNKV BHXIS LIDE AKFEKSEL
16. MET LNKV BHXIS LIKE AKFEKSEL
17. MET LNKS BHXIS LIKE AKFEKSEL
18. MET LNKS BH IS LIKE AKFEKSEL
19. MET LNKS BH IS LIKE AKFEKSEL
20. MET LNKS BH IS LIKE AKWEKSEL
21. MET INKS BH IS LIKE AKWEKSEL
22. MET INKS BH IS LIKE AKWEKSEL
23. MET INKS BH IS LIKE AKWEKSEL
24. MET INKS IH IS LIKE AKWEKSEL
25. MET INKS IH IS LIKE A WEKSEL
26. MET INKS IH IS LIKE A WEASEL
27. MET INKS IH IS LIKE A WEASEL
28. METHINKS IH IS LIKE A WEASEL
29. METHINKS IH IS LIKE A WEASEL
30. METHINKS IH IS LIKE A WEASEL
31. METHINKS IT IS LIKE A WEASEL
```

____

And this one hit home in 21 gens:

____

Case D: 999/gen [maxed out], 8%:

____

```
1. MMCJXLTPPCNATTMLKDXOBDKMBJQX
2. MMCJXL PPCNATT LKDXOBDKMAJQX
3. MMCJXL PPCNATT LKDXOB KMAJUX
4. MECJXL PPCLATT LIDXOB KMAJUX
5. MECJXL PPWPOVS LIDXOB WMAJUX
6. MECLXL PPWPOVS LIDXOY WMAJUL
7. MECLXL SPWPOVS LIDXOY WMAJUL
8. MECLXL SJWPOIS LITXOY WMAJUL
9. MECLXL SJWP IS LIZXOY WTASUL
10. MECLXL S WP IS LIZAOY WTASUL
11. MECLXL S IP IS LIZAOY WTASEL
12. MECLXL S IT IS LIZAOY WTASEL
13. MECLXL S IT IS LIKNOY WTASEL
```

```
14. MECLXL S IT IS LIKEOY WTASEL
15. MECHXL S IT IS LIKE Y WUASEL
16. METHXZ S IT IS LIKE Y WUASEL
17. METHXZ S IT IS LIKE A WUASEL
18. METHKN S IT IS LIKE A WUASEL
19. METHKN S IT IS LIKE A WEASEL
20. METHIN S IT IS LIKE A WEASEL
21. METHINKS IT IS LIKE A WEASEL
```

———————————

Here we see speeding up of run to the target.

223

kairosfocus

08/31/2009

3:59 am

PPS: On a subtlety: when mutation occurs, the same letter is a candidate, so 1 in 27 times, a mutation will go back to being itself. As an effect of that, 1 in 27 times on average, when a correct letter "mutates," it goes back to being itself. That should not be overlooked in analysis.

224

nephmon

08/31/2009

4:45 am

I've often wonder if the fact that the weasel program is performing directed a search at "a distant goal" is as troublesome as even RD states.

If you think of the phrase "methinks it is a weasel" as representing the environment in which the mutated, generated phrases are "living", then what the program is doing is optimizing the progeny of the random seed phrase for fitness in that environment.

Of course, there are many simplifications involved here: the "environment" happens to also be a specific value of the "genome" that's evolving; there is no analogue of a phenotype resulting from simple genotype that's being modeled (and on which selection would operate in reality); and the "environment" never varies. (This last point is easily addressed by also mutating the target phrase across generations, so that the program will now be chasing a moving target, which is closer to the idea of organisms adapting to slow-changing environments. Of course in this case, it's unlikely the simulation would halt, just like evolution.)

There are many other enhancements that could be made to the program to better model real-world evolution, and I'm sure many such simulations exist. But I still think that as a very simple-to-understand illustration of the vast difference between random mutation and the selection of random mutations according to some selective pressure, it holds up pretty well.

225

nephmon

08/31/2009

4:55 am

kf@222: can you explain why you're reproducing that? I think almost everyone here has written an implementation of the weasel algorithm and has observed the fact that with many progeny to choose from in each generation, a letter that is already correct is not going to change in the next generation (unless p(mutation) is extremely high).

So what are you revealing by listing a result that's already very well known? Is it that you're trying to prove that RD's original program would have behaved just as he claimed in the book without having to fix letters in place once they'd matched, and therefore we don't need to see the source code?

Thanks anyway, but I think we know that already.

226

DiEb

08/31/2009

5:51 am

–kf
sorry, I couldn't spot where in your variably emphasized and enumerated post you answered the question:

> Which algorithm from Atoms suite corresponds to Weasel and which one corresponds tpo the Partitioned Search discussed by D+M?

And for your *subtlety*:

> when mutation occurs, the same letter is a candidate, so 1 in 27 times, a mutation will go back to being itself.

Why? That's just a question of implementation. Doing it this way, a possibility of 4% of change for a letter would translate into a de-facto-mutation rate of 3.85% – but you can also achieve a mutation rate of 4% if you chose uniformly only from the other 26 symbols.
For practical purposes, it's not much of a difference…

227

Indium

08/31/2009

6:26 am

kf, everybody here has understood the latching issue. The main point is that we all agree that there is no proof or evidence that explicit latching is involved. This was the original question of this thread and it has been solved a long time ago. The programm DOES let the characters vary, the subsequent filtering eliminates bad mutations most of the time. Your letter counting ("as a matter of fact that in 200+ cases of letters that if reversions from correct were frequent") is of course wrong because you should a) count only correct letters and b) take into acount that you only see every 10th generation so that wrong letters have a good chance to be restored anyway. How often do we have to go through this?

The discussion has moved on. It has now been shown that this is by far not the only misrepresentation in the D+M paper. In fact they use a completely different algorithm while referencing Weasel. This is most easily demonstrated by checking Eq 22 which is incorrect for Weasel. Even Joseph has acknowledged this.

And finally, you continue to evade my questions which in turn continues to amuse me. I will repeat them later for your convenience!

228

DNA_Jock

08/31/2009

6:28 am

nephmon – you and I may already understand how a program that has no latching **mechanism** may still display latching **behavior**, but it is nice of kf to explain it to this audience, even if he is a bit long-winded about it.

Let us remind ourselves that kfocus has defined Q as

> kf @185
> Q is — on second thought — number of generations multiplied by size of generations

And he has told us (@163) that Weasel is the same as the "Proximity Reward" search (this, at least, is true).

I am surprised that kf (@222) would then showcase Weasel runs with Q = 999 x 21 and 500 x 31

As shown in the D&M paper, Q for a Partitioned Search has a median of 98, and should rarely go above 160.
But according to kf, his didactic examples of Weasel runs have values of Q of 20,979 and 15,500.

And he still wants to argue that D&M's use of eqn22 to describe Weasel is accurate.

229

kairosfocus

08/31/2009

6:34 am

NM:

Plainly, you represent the "it doesn't matter . . . " faction!

Again: the *primary* issue is that Weasel's superiority over "lucky noise" comes from its active information, manifested in targetted, proximity rewarding search that picks and promotes non-functioning "nonsense phrases." (This was actually inadvertently acknowledged and/or implied by CRD in BW.)

The *secondary* debates over Weasel and explicit or implicit latching have to do with [a] how the results showcased in 1986 were best explained, [b] responses to attempts to dismiss that. Latterly [c] to the claim that here is a radical divergence between what Weasel 1986 did and the description ion Marks and Dembski's IEEE paper that the showcased results reflect partitioning due to ratcheting action [which enfolds latching of correct letters].

Unfortunately, we have also had to deal with a lot of Darwinist rhetorical games that drag distractive red herrings, lead off to strawman misrepresentations soaked in ad hominem mischaracterisations, which are then ignited to cloud, confuse, poison and polarise the atmosphere. (Apart from this factor, we probably would not have had several threads at UD since about March running over 1,000 comments on a plainly SECONDARY matter. There are threads where I have actually been rebuked for pointing out the primary issue, as being distractive from the "main" point!)

I trust this sets the matter in clear enough context.

GEM of TKI

PS: Lest we forget, here arte the showcased runs of Weasel from 1986:

We may conveniently begin by inspecting the published o/p patterns circa 1986, thusly [being derived from Dawkins, R, The Blind Watchmaker , pp 48 ff, and New Scientist, 34, Sept. 25, 1986; p. 34 HT: Dembski, Truman]:

```
1 WDL*MNLT*DTJBKWIRZREZLMQCO*P
2? WDLTMNLT*DTJBSWIRZREZLMQCO*P
10 MDLDMNLS*ITJISWHRZREZ*MECS*P
20 MELDINLS*IT*ISWPRKE*Z*WECSEL
30 METHINGS*IT*ISWLIKE*B*WECSEL
40 METHINKS*IT*IS*LIKE*I*WEASEL
43 METHINKS*IT*IS*LIKE*A*WEASEL
```

```
1 Y*YVMQKZPFJXWVHGLAWFVCHQXYPY
10 Y*YVMQKSPFTXWSHLIKEFV*HQYSPY
20 YETHINKSPITXISHLIKEFA*WQYSEY
30 METHINKS*IT*ISSLIKE*A*WEFSEY
```

```
40 METHINKS*IT*ISBLIKE*A*WEASES
50 METHINKS*IT*ISJLIKE*A*WEASEO
60 METHINKS*IT*IS*LIKE*A*WEASEP
64 METHINKS*IT*IS*LIKE*A*WEASEL
```

——————————

These are runs of generational champions, used to seed the next successive generation. For over 200 cases of letters that go correct, there are no reversions thereafter that are seen in the sample — and since we can easily see how persistent incorrect letters often are [E.g. cf the initially incorrect W, C Z and M in the first run and the Y, P , F and Q in the second] — this strongly argues that there are none; especially given the gushing description of "cumulative selection" in BW.

230

BillB

08/31/2009

6:40 am

KF

Yet again you need to be reminded of the basis for this debate, and O'Leary's question.

The issue is whether Dembski and Marks describe the same algorithm as Dawkins in the section of their paper where they reference The Blind Watchmaker. It is a question of good scholarship, of correcting your errors when they are pointed out.

The issue of WEASEL's biological plausibility, of whether it explains how life began, is totally irrelevant to this issue of scholarship – WEASEL could be a total failure at whatever Dawkins was trying to achieve, but it is still wrong to misrepresent it in a peer reviewed paper.

Please can you try and stay on topic. If you want to discuss the interesting and worthwhile question of how WEASEL applies to biology, or the technicalities of Dembski and Marks paper then I'm sure Clive would oblige with a separate thread. The question we are trying to get a clear answer on is purely about why Dembski and Marks cite the WEASEL algorithm in The Blind Watchmaker, but then describe a substantially different algorithm.

You are jumping backwards through oily hoops to try and make out that the differences between these two clearly different things don't amount to anything. It is both tragic, and very entertaining.

Moving on…

When you said this:

> this implies that when metric falls to distance to target = 0, there is a latching action imposed.

You appeared to be trying to claim halting as a latching mechanism relevant to the debate on latching mechanisms, I'm glad that this was just poor communication on your part, but why mention halting with reference to latching at all? – it seems irrelevant to the debate but on closer inspection it highlights the differences between WEASEL and a partitioned search.

Without an explicit stop condition for the software Dembski and Marks algorithm will stop searching when it hits the target and loop forever doing nothing. WEASEL will continue to produce generations, each with mutated members and, depending on mutation rates and population sizes, the winning phrase will continue to show occasional reversions even after the target is found. So when you say this:

> Weasel type algors will lock up as they hit home.

You are in error.

I also agree with nephmon about your comment at 222 – what was the point, you are just demonstrating what we already know – that a latching mechanism, an explicit piece of code to lock correct letters out of the search, is not required. You seem to have a blind spot about this issue and the way it impacts the two algorithms, I'll try and explain (again):

Dembski and Marks describe a series of searches for individual letters – each letter is ranomised and checked against a target letter, if it matched then a halting condition is achieved. Proximity to target with repect to the whole phrase is determined by the number of letters being searched for – when there are zero the target is found.

Dawkins describes a process where the number of correct letters are summed to produce a score, a proximity, for the phrase as a whole, not for individual letters. The search is NOT partitioned up into individual letter searches, each with its own halting condition. The search mechanism does NOT halt for an individual letter when it matches, it halts when the sum of matching letters is equal to the number of letters in the phrase.

These are important differences between the two, even if they sometimes produce similar behaviour. Glossing over this by trying to pretend that it is irrelevant to the content of the paper misses the point. It trivialises the whole subject of search algorithms. **These are two different processes at work, it is inappropriate to describe them as being the same.**

231

Joseph

08/31/2009

6:47 am

I would say they are using the same algorithm just using it differently.

It appears that they (D/M) are combining the outputs to get one.

Ya see the intention is different- as I said above.

Dawkins was trying to show how the ratcheting properties of cumulative selection will find a target much sooner than a random step-wise search.

D/M were trying to show how that type of search does its job.

232

Indium

08/31/2009

7:03 am

Joseph,

> It appears that they (D/M) are combining the outputs to get one.

If it means what I think it means (combine the correct letters of several parallel partitioned searches to get to the next phrase) then this is wrong. Just look at Eq22 again. Or at the text, which explains it will enough. And even IF it would be the way D+M were constructing the next parent string it would still be very different from Weasel. I thought you understood the difference earlier in this thread but it seems I was mistaken.

How can one use the same algorithm differently anyway without using in fact a different algorithm? You could use different parameters (population sizes or mutation rates) for Weasel but you have no such choices for the partitioned search!

233

kairosfocus

08/31/2009

7:14 am

PPS: I need correct Indium and DJ, that:

1 –> The observed Weasel 1986 behaviour [o/p] requires a mechanism [i/p & processing] — i.e. the old I–> P –> O elements of a program — that explains its dynamics; of which explicit and implicit mechanisms are the two best candidates. [Is this a case where StephenB's "causeless events" problem for Darwinists is surfacing? I firmly believe -- on good grounds -- that *unless necessary causal*

factors are present an event CANNOT happen, and unless sufficient ones are present it WILL not happen. So if it happens, we can identify causal factors and how they work -- i.e mechanisms. And given the issue of synergy -- effects due to interaction -- mechanisms do not have to be explicitly built and labelled as such. I do not believe in magical poofery!]

2 –> In terms of Atom’s adjustable weasel, explicit latching corresponds to “partitioned search” and implicit latching will show up with cases under his “proximity reward search.”

3 –> The decision that implicit mechanisms best explain the observed showcased runs of champions c 1986 is based in large part on Mr Dawkins’ subsequent testimony; understood as denying that Weasel explicitly latched.

4 –> Once ratcheting behaviour exists, the eqn 22 from p. 1055 of the IEEE paper, suitably understood, applies. For instance — and here I took a correction on the interpretation of Q — with generational clustering, queries [= mutant pop members] come in generational bunches of size S: Q = G*S. And the latching-ratcheting o/p will apply to the run of generational champions; an observable behaviour.

5 –> So, for instance with pop size = 50, queries to date will go 50, 100, 150 etc. corresponding to 1, 2, 3 etc times 50.

6 –> Interesting, onlookers, how a concession that implicit latching is real can be made to look like a claim of victory for the Darwinists! (Hint: rhetoric is the art of persuasion, and too often works by making he weaker case seem stronger than it is. So, let us attend to the merits and facts, and follow back to the points where it was being ever so stoutly insisted that here was no evidence of latching in the Weasel o/p c 1986, or that latching only meant explicit latching, or that implicit latching did not happen or that implicit latching is a triviality etc etc.)

234

Joseph

08/31/2009

7:19 am

Indium,

D/M are explaining something that Dawkins’ wasn’t.

So yes two different sets of people can use one thing to explain two different things.

The confusion comes from yet another set of people who can’t understand that.

And here you are…

235

BillB

08/31/2009

7:36 am

> Interesting, onlookers, how a concession that implicit latching is real can be made to look like a claim of victory for the Darwinists!

KF, the onlookers aren’t stupid, they can see that this whole *“lets define latching to mean whatever we need it to mean in order to claim victory”* is your own tactic, one you keep employing again and again.

The onlookers who have been following this long enough will remember how you invented the ideas of implicit/quasi/semi-latching in order to get around the fact that al the published output of WEASEL can be explained without it requiring an **EXPLICITLY DEFINED LATCHING MECHANISM** – something Dembski and marks explicitly define for their algorithm but which Dawkins doesn’t.

This concession over implicit latching is a fantasy of yours, in Dawkins algorithm the fittest member *appears* to latch some of the time without ever needing a latching MECHANISM. We all knew this from the start and pointed it out to you, you then invented the term “implicit latching” to describe this behaviour, and claimed that you were right about there being a latching mechanism all along.

You are making a fool out of yourself, and of this website.

236

Joseph

08/31/2009

7:45 am

The facts are that Dawkins used cumulative selection as a ratcheting process.

He descirbes it as “slight improvements”.

The way cumulative selection is described and illustrated in TBW it is a ratcheting process.

And the only way around that is to contort what Dawkins actually wrote.

237

Joseph

08/31/2009

7:47 am

I challenge anyone to find a passage or passages in TBW that would show that cumulative selection is not a ratcheting process. That reversals can happen and be selected.

From the book only.

Good luck…

238

BillB

08/31/2009

8:01 am

Joseph,

I think you are confused. Dembski and Marks are using a different thing to explain a different thing but they cite Dawkins algorithm and make out it is the same, when as we have seen, it is actually different.

They should have referenced a similar thing when explaining their thing rather than referencing a different thing than the one they explain.

It is all very simple really!

239

kairosfocus

08/31/2009

8:05 am

Onlookers:

Re BillB: *This concession over implicit latching is* ***a fantasy of yours****, in Dawkins algorithm the fittest member appears to latch some of the time without ever needing a latching MECHANISM . . .*

Sadly, this reveals that BillB is till well out beyond the pale of civil discourse:

1 –> In the App 7, the always linked [dating to about April] we may see from point 12 on, after a presentation on what is to be explained, Weasel 1986:

> 12 –> To explain the latching more realistically, we may have an explicit latching algorithm based on letterwise search . . . .
>
> 13 –> Letterwise partitioned search is also a very natural way to understand the Weasel o/p in light of Mr Dawkins' cited remarks about cumulative selection and rewarding the slightest increment to target of mutant nonsense phrases. As such, it has long been and remains a legitimate interpretation of Weasel. **However, on recently and indirectly received reports from Mr Dawkins, we are led to understand that he did not in fact explicitly latch the o/p of Weasel, but used a phrasewise search.**
>
> 14 –> Q: Can that be consistent with an evidently latched o/p?
>
> **ANS: yes, for IMPLICIT latching is possible as well.**
>
> **15 –> Namely, (i) the mutation rate per letter acts with (ii) the size of population per generation and (iii) the proximity to target filter to (iv) strongly select for champions that will preserve currently correct letters and/or add new ones, with sufficient probability that we will see a latched o/p. (This effect has in fact been demonstrated through runs of the EIL's recreation of Weasel.)**
>
> 16 –> in a slightly weaker manifestation, the implicit mechanism will have more or less infrequent cases of letters that will revert to incorrect status; which has been termed implicit quasi-latching. This too has been demonstrated, and it occurs because an implicit latching mechanism is a probabilistic barrier not an absolute one. So, as the parameters are sufficiently detuned to make reversions occur, we will see quasi-latched cases. **Sometimes, under the same set of parameters, we will see some runs that latch and some that quasi-latch.**
>
> 17 –> In a litle more detail, we will see reversions in a case where the odds of mutation per letter are sufficiently low that in a reasonable generation of size N, there will be a significant fraction of no-change members, and so the proximity filter will select no-change to be next champion, or a single step increment in proximity; or, in some cases a substitution where one correct letter reverts and one incorrect letter advances . . . .
>
> 19 –> In short, there is almost no chance that such a mutant generation population will have no unchanged members. In that context, with the proximity filter at work, no-change cases or substitutions or single step advances will dominate the run of champions. Indeed, the Weasel 1986 o/p samples show that runs to target took 40+ and 60+ generations, i.e. no-change won about half the time and single step changes dominated the rest. No substitutions were observed in the samples, suggesting strongly that there were none in the showcased runs. (Double step advances etc or substitutions plus advances will be much less probable. But in principle, per sheer "luck," we could see the very first random variant going right to the target. Just, the odds are astronomically against it. As, probabilistic barriers may be stiff, but are not absolute roadblocks.) . . . .
>
> 24 –> In turn that means that a further key issue on champion selection per generation is the specific action of the proximity filter, in a context where — continuing the concrete example — the expected number of zero change cases in a generation is about 17, that for single step advances is about 1, and that for substitutions ranges from 1/2 down to 5/1000, depending on degree of proximity already achieved. That turns out to be a fairly complex issue:

2 –> Given subsequent debates, I would adjust phrasing for clarity, but the point is plain: explicit mechanisms latch letters explicitly, and implicit ones do so out of the interaction of population size, mutation rate and filter characteristics. remember o/p is what we directly observe, and mechanism is based on the i/p's and processing that give rise to it.

3 –> That is, the issue of observed behaviour and its explanation by alternative mechanisms was long since addressed. It is not a latterday "concession," but an alternative EXPLANATORY MECHANISM, one that — per inference to best, empirically anchored explanation — is in fact preferred on the balance of the evidence; ever since some time in March or so, when the information came to our attention at UD. (Remember, it is a DEMONSTRATED mechanism, as is for instance reproduced and/or linked above.)

4 –> Therefore, it seems that BillB has regrettably failed to do his homework before criticising, and has indulged in pejorative mischaracterisation and misrepresentation.

5 –> This pattern of uncivil rhetorical conduct is to be corrected on his part.

GEM of TKI

240

BillB

08/31/2009

8:06 am

Joseph:

> [Weasel] . . . begins by choosing a random sequence of 28 letters … **it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying.** The computer examines the mutant nonsense phrases, the 'progeny' of the original phrase, and chooses the one which, however slightly, most resembles the target phrase, METHINKS IT IS LIKE A WEASEL

Challenge met – no mention of a rule that prevents letters reverting. When you implement this you get a mechanism that works towards the target, revisions occasionally occur, as KF has demonstrated, and all published data is consistent with this.

This is also consistent with the use of cumulative as paying attention to movement in one direction and not in the other, for example cumulative height or accumulated wealth.

241

DNA_Jock

08/31/2009

8:07 am

BillB – It has been pointed out to kf that latching is irrelevant to the original question, which is "Are D&M accurate when they call Weasel a partitioned search, and use eqn 22 to describe it?"
We have been explaining this to kf in simple, plain english since post 34, where I said

> Forget about latching, you cannot have a partitioned search that changes only one element in its first iteration, and reaches a solution within 43…

kf, OTOH, has resorted to re-defining the word "mechanism" (@232) in his increasingly desperate attempts to avoid admitting any mistake.
Humpty Dumpty would be so proud.

242

Joseph

08/31/2009

8:10 am

BillB,

There isn't anything in your quote that satisfies my challenge.

No mention that letters, once matched, can be changed.

That is what I am looking for and I made that clear.

Also I have said that intellignet agencies can use "cumulative" in a different way.

So if you are saying that evolution is guided by an intelligent agency then you would have a point.

Otherwise "cumulative" along with the description of "slight improvements" only means ratcheting is taking place.

243

Joseph

08/31/2009

8:12 am

I challenge anyone to find a passage or passages in TBW that would show that cumulative selection is not a ratcheting process. That reversals can happen and be selected.

From the book only.

244

Joseph

08/31/2009

8:14 am

DNA Jock,

If we go by what Dawkins wrote in TBW then "weasel" is a partitioned search- ie ratcheting occurs.

And we know ratchets can move in both directions but force is only applied in one.

245

BillB

08/31/2009

8:19 am

KF,

Please could you at least try and understand what the debate is about.

When Dawkins describes a mechanism that does not lock letters, and Dembski and Marks describe one that does, and both produce different behaviour, and Dawkins published behaviour is consistent with the algorithm he describes … and Dembskis is inconsistent with Dawkins output … and Dawkins specifies a population … and so on and so on …

Why would any sane person still try and maintain that these are the same algorithm and that the citation is appropriate? Playing semantic games with words like latching doesn't help your case at all, it just smacks of arrogant desperation and a total inability to acknowledge your own mistakes.

246

BillB

08/31/2009

8:22 am

I challenge anyone to find a passage or passages in TBW that would show that cumulative selection is a ratcheting process. That reversals can not happen and will not be selected.

247

Indium

08/31/2009

8:48 am

Joseph,
ratcheting, latching, implicit, explicit, so many arguments are just based on semantics. What do you mean by ratcheting? A ratchet can only go in one direction, the algorithm Dawkins describes can go in both (just try it yourself with Atoms suite).

In the TBW Dawkins says that the strings are copied with random errors/mutations. Since he does not limit this mutations to incorrect letters correct letters also mutate -> No ratcheting or latching at the mutation level. Agreed?

This is of course in line with what Dawkins says, what is demonstrated in a video and with biological reality. And even with kairosfocus allways linked appendix 7. And with the EIL software.

248

Joseph

08/31/2009

8:56 am

Indium,

The way Dawkins describes cumulative selection as “slight improvements” can only mean it goes in one direcetion- towards the goal.

Reversals as an improvement only work with agency involvement because agencies can plan ahead.

249

Joseph

08/31/2009

8:57 am

BillB:

> I challenge anyone to find a passage or passages in TBW that would show that cumulative selection is a ratcheting process.

That part has been presented already.

It has to do with Dawkins talking about “slight improvements”.

As soon as I get the book back I will provide the quotes.

250

CJYman

08/31/2009

8:58 am

my two cents:

Cumulative selection — increasing by successive additions — is indeed a ratcheting process so long as the string, chosen from the present generation, most closely representing the target phrase is closer to the target phrase than the parent string. The chances of reversal would depend on the size of the population and the mutation rates. Thus, within specific parameters, Dawkins algorithm will provide a ratcheting process (”implicit” latching since it is based on probability of reversion to a string further from the target) and within other parameters it will not provide a ratcheting process.

So long as we see a steady closing in on the target, we are viewing a ratcheting process (whether it happens as a result of probability or explicit programming to achieve ratcheting behavior).

251

Joseph

08/31/2009

8:59 am

It looks like my challenge cannot be met.

No surprise there.

I challenge anyone to find a passage or passages in TBW that would show that cumulative selection is not a ratcheting process. That reversals can happen and be selected.

Without that all you anti-IDists are blowing smoke.

Not only that you are accusing D/M of something that they didn’t do.

252

DNA_Jock

08/31/2009

9:16 am

No, Joseph, you are wrong.
Dawkins’ *narrative* is clear to 99% of the people who have read it — “Random mutation” means random mutation as it is observed to occur in nature. In particular Dawkins refers to selecting the best “phrase” which makes little sense for a latching mechanism, and none for a partitioned search. “Cumulative” is used in contrast to the alternative “single-step”, or perhaps “instantaneous”. It is as telic as “In April, water accumulated in my wheel-barrow” which does not (to me anyway) deny the existence of evaporation.
You are asking (I hope) a rhetorical question – “Where in TBW does it say **There is no latching** ?”. I might as well ask “Where is the KJV Bible does it say ‘Adam did not have three arms’ ?”.
But, and this is the really important point, even if there were an explicit latching mechanism in Weasel, that would not make it a partitioned search as described by D&M. The first reported Weasel run in TBW **CANNOT** be the result of a partitioned search. Only two letters change in generation 2.

Latching is irrelevant to the accuracy of D&M’s citation. I am accusing D&M of mis-citation when they claim that eqn22 describes Weasel. I encourage everyone to go to EIL and play with Atom’s adjustable Weasel, specifically the Proximity Reward Search, and contrast its behavior with that of the Partitioned Search.

253

Indium

08/31/2009

9:17 am

Joseph, it is not our fault that you seem to think that a cumulative process can´t also accumulate negative values.

Weasel is a cumulative because the results from a search are the basis for the next search step.

I have never seen such a desperate semantic argument before.

So, maybe you can give me a straight answer to the following question, we can then proceed from there:
In the TBW Dawkins says that the strings are copied with random errors/mutations. Since he does not limit these mutations in any way to incorrect letters this means correct letters also mutate -> No ratcheting or latching at the mutation level. Agreed?

254

Joseph

08/31/2009

9:22 am

Indium,

It is very telling that you cannot meet my challenge.

It is also very telling tat all you have are desperate semantics.

Again "cumulative" and reversals only make sense in an agency involved scenario.

Non-telic processes would not see reversals as an improvement over the parent.

255

Joseph

08/31/2009

9:28 am

DNA Jock,

A partioned search is a latching/ ratcheting process.

256

Indium

08/31/2009

9:42 am

Joseph,

I am in the process of explaining why you are wrong.
So, can you give me a straight answer to my question:

In the TBW Dawkins says that the strings are copied with random errors/mutations. Since he does not limit these mutations in any way to incorrect letters this means correct letters also mutate -> No ratcheting or latching at the mutation level. Agreed?

257

DNA_Jock

08/31/2009

10:00 am

Joseph,

A partioned search is *[insert]* one type of *[/insert]* ~~a~~ latching/ ratcheting process.

Fixed that for you.

Perhaps you meant to say
“A latching process is (necessarily) a partitioned search (as modeled in eqn22)”
But that would be wrong.

Or you meant to say “Weasel is a partitioned search (as modeled in eqn22)”
But that would be obviously wrong.

258

nephmon

08/31/2009

1:02 pm

I like Joseph’s demand for proof from the text of TBW that it explicitly precludes ratcheting. Somewhat akin to asking for proof that God (sorry, an intelligent designer) doesn’t exist. And in both cases the response is the same: instead of demanding proof for the non-existence of something, why not provide proof for its existence? In the case of WEASEL, please do what BillB asked for in #245.

I suspect that TBW doesn’t explicitly say “positions aren’t latched once they have their correct values” because such an outrageous hack would never have occurred to RD. It just doesn’t represent what he was trying to model.

Joseph, please just confirm that you understand this: letters don’t revert to incorrect values because the candidates that *do* are never the fittest amongst the generated offspring. This is true even for small values of n (about 50 from my experiments, depending on p(mutation)). Make n smaller, and yes, you WILL see reversions, and in the limit, as n -> 1, you’re basically making a random mutation to a single offspring, and off course you’ll never get closer to matching the environment string (unless you employ ratcheting, a la D/M). You do understand all this, right?

On this issue of “the program contains information that allows it to solve the problem” as raised by kf and others, I disagree: this is just an implementation detail. Instead of the target string being harcoded into the program (or input at the start), let’s read it from an input stream instead. Moreover, let’s read it at the start of every generation, allowing it to change over time. The program will still work just as well, even though its only knowledge of the target string is garnered at the beginning of each generation, making it similar to an “environment” in which the offspring are generated that I mentioned in a previous post; the progeny that goes forward to the next generation is the one that fits best the current environment, not some predetermined “distant target”.

259

DiEb

08/31/2009

5:16 pm

At least I can put the onus that no one answers to my edits on the fact that these stay in moderation for a half a day and not on a lack of quality of the edits themselves 😐

260

kairosfocus

08/31/2009

11:55 pm

Joseph and CJY:

Thanks. Appreciated.

It seems the Darwinists — oh, the irony! — are in denial.

Maybe, they need to read this classical philosophical story, and then "wheel and tun and come again."

I think a spiritual issue is at the root. I suggest meditation and prayer on the lines of this classical text from the Sermon on the Mount.

G'day.

GEM of TKI

PS: Onlookers, it seems BillB thinks I am now "insane" and/or "stupid" — sounds familiar? — to have (months ago) analysed, identified and defended then DEMONSTRATED the mechanism that triggers implicit latching as an explanation of Weasel 1986 [probably the only reason why it is suddenly "obvious" to all and why it is held that it is my intellectual incapacity that prevents me from seeing it in his estimation!], and to point out that the Marks and Dembski analysis starts from credibly OBSERVED ratcheting (oops, he apparently denies that the cumulatively selected o/p of Weasel 1986 evidently latches!), and deduce its implications for active information. This, from one who evidently has a difficulty with explanation of program o/p's by credible causal mechanism tracing to i/p's and processing behaviour. There is something plainly deeply and sadly wrong in the state of Darwinland — and it isn't the name of the river of Egypt.

PPS: Nephmon similarly, cannot agree that targetted search — with a built-in target phrase! — using a proximity to target filter to "cumulatively select" increments of proximity of "nonsense phrases" is a case where "the program contains information that allows it to solve the problem." [He needs to read what CRD said in his own words on this, as is repeatedly cited above, e.g. last at 221, and as is explained in my online note app 7].Something is indeed deeply, sadly wrong in the state of Darwinland.

261

nephmon

09/01/2009

2:13 am

kf:

> PPS: Nephmon similarly, cannot agree that targetted search — with a built-in target phrase! — using a proximity to target filter to "cumulatively select" increments of proximity of "nonsense phrases" is a case where "the program contains information that allows it to solve the problem."

What's the conceptual difference between "proximity to target" and "fitness to environment"? In both cases some offspring fail to reproduce (all but one in the harsh world of WEASEL). In the natural world, it's the lack of reproductive success caused by poor fitness to the environment. In WEASEL this is *simulated* in a very simplistic way by representing the environment as a "perfect gene" that would be the fittest possible for the world it inhabits, and relative fitness is represented as "closeness" to that gene. What's the fundamental flaw with this?

Actually, I did mention what RD had to say about the drawback of the program containing the target phrase, and I said that I think it's an implementation detail. How "external" to the running code of the program would you need the target/environment to be before you accepted that the program itself didn't contain the "information" that it's working on? I already posited an example where the string could be read from an input stream, the other end of which could be a program running halfway across the world. Is that external enough?

If one states the problem as "show how the successive choice of the fittest of a group of randomly mutated offspring to be the sole parent of the next generation can quickly converge on the optimally 'fit' entity," you need at some point to provide a quantity to measure fitness against. For convenience, it can be hardcoded into the program itself, but as I've mentioned a few times now, that isn't necessary for the program to function, neither is it necessary for the target to be static.

Out of interest, in what way does WEASEL fail most egregiously as a simulation of natural selection as you think it's accepted by evolutionists? As I've freely admitted before, there are many simplifications implicit in it, but of the major components of "the natural selection of fittest offspring that are subject to random mutation", where to you feel WEASEL most comes up short? (Please don't concentrate on the "natural" aspect; by definition any simulation won't be "natural" – there isn't really mutating DNA in the computer.)

262

nephmon

09/01/2009

2:28 am

Also, can you explain how what you term as "implicit latching", i.e. lack of reversion of correct letters to incorrect ones, which is a result of the strategy of selecting the fittest of a large population of offspring, is fundamentally different from what's observed at in real organisms at the gene level?

If the human genome mutates at, say, 100 bases pairs per sexual reproduction (which is towards the upper bound of the estimated range), what's the chance of any of those 100 out of the 3.4 billion or so base pairs in the genome are going to mutate back again in the near (or even distant) future? Rather small, I'd imagine. Thus once a beneficial mutation has made it into the genome, it will very likely be passed down for all intents and purposes "forever", in other words, it's latched. In what way is this different from happens in WEASEL?

263

DiEb

09/01/2009

3:11 am

I try it again:
In their paper, Dembski and Mark they start with the phrase:
SCITAMROFN*IYRANOITULOVE*SAM
I calculated a next generation using Dawkins's algorithms with populations of 10,50 and 100 – and mutation rates of .04, .05 and .1. The tenth string in the list is the second generation given in the paper of Mark and Dembski. The differences with the first generation are in bold face:

1. SCITAMROFN*IYRANOI**E**ULOVE*SAM
2. SCITAMROFN*IYRANOITULO**G**E*SAM
3. **E**CITAMR**I*** N*IY**Z**ANOITULOVE*SAM
4. SCITAMROFN*IYRANOITUL*****VE*SAM
5. SCITAMROFN*IYRANOITULOVE*S**E**M
6. SCITAM**O**O**L**N**O**IYRA**M**OITULOVE*S**E**M
7. SCITA**N**ROFN*IY**Y**ANOITULOVE*SAM
8. SCIT**I**MROFN***J**YRANOITULOVE*SAM
9. SCITAMROFN*I**C**R**H**NOIT**S**LO**W**E*SA**V**
10. **OOT*DENGISEDESEHT*ERA*N**E**T**S**IL**
kf – can anyone spot a difference in the design of the strings? Could you explain why it exists?
Heck, I even calculated the probabilities for latching…
(I hope I don't have to endure another twelve hours of moderation…)

264

kairosfocus

09/01/2009

6:19 am

Nephmon:

You have asked an excellent question here, that goes to the heart of the issues at stake:

> **What's the conceptual difference between "proximity to target" and "fitness to environment"?** In both cases some offspring fail to reproduce (all but one in the harsh world of WEASEL). In the natural world, it's the lack of reproductive success caused by poor fitness to the environment. In WEASEL this is simulated in a very simplistic way by representing the environment as a "perfect gene" that would be the fittest possible for the world it inhabits, and relative fitness is represented as "closeness" to that gene. What's the fundamental flaw with this?

1 –> I first ask you to simply read the weak argument correctives above [and while you are at it you might want to take a look in a library at the Schutzenberger contributions on functional complexity dating all the way back to the Wistar high level summit of 1966], then take as look at the always linked, starting with the App 7, especially where it discussed the back-story on Weasel:

> 10 –> Back story: Mr [Fred] Hoyle had raised the issue that the origin of a first life form — such as, roughly, a bacterium — is a matter of such complexity that the odds of that happening in a prebiotic soup by chance was negligible. In a more up to date form, this is the challenge that is still raised by the design theory movement: life shows a threshold of function at about 600,000 bits of DNA storage capacity, so before one may properly apply hill climbing algorithms to scale Mt Improbable, one needs first to have a credible BLIND watchmaker mechanism to land one on the shores of Isle Improbable; e.g. drifting by reasonable random configurations of molecules in empirically justified pre biotic soups and empirically credible pre-life selection forces. (But . . . this OOL challenge is still unmet. And similarly . . . the origin of body plan level biodiversity requiring 10's – 100's or millions of bits of functional genetic information, dozens of times over, is equally unmet.)
>
> 11 –> So, it looks uncommonly like Weasel distracts attention from and begs the question. That sums up the balance on the main issue . . .

2 –> In short, there is a major question-begging — indeed, dismissal: "single-step selection" — at stake in Weasel as presented: the origin of complex, functionally specific information.

3 –> that is, until one has spontaneously created novel complex information that functions through similarly sophisticated but spontaneously originated machinery, at a realistic level, one has no right to pretend that a targetted process that uses a hotter-colder distance-to target comparison of "mutant nonsense phrases" — i.e confessedly NON-FUNCTIONAL ones — is a good analogy of the power of random variation and natural selection, which is premised on the origin of information by chance variation, and the selection of FUNCTION on superiority of some sub-populations.

4 –> that was the heart of Sir Fred's challenge and it is the heart of the design theory challenge to the claimed spontaneous origins of life and of novel body plans. That is, we do observe FSCI of an order of complexity of 1,000 or morte bits forming all the time: by DESIGN. But, we have never observed this by spontaneous forces tracing to chance + necessity without intelligent intervention. And, on search resource exhaustion grounds — as CRD actually concedes in the relevant passage — it is utterly implausible that we will ever get to such complex functionality on the gamut of our observed cosmos. [And please resist the red herrings and strawmen in the usual rebuttals of Hoyle a la Wikipedia. Remember you are in the presence of a Nobel Prize equivalent holder here. he may be wrong on points, but he is not going to be making simplistic blunders, and even his errors will be instructive. That is my experience from many years of running across his work, on a wide array of topics, starting with the Steady State universe Hypothesis. (He it is who gave the name "big bang" to the cosmogenetic theory of that name, but he did not intend it to be a positive term.)]

5 –> With these points in mind, let us look again at what CRD said in discussing Weasel:

>> It . . . begins by *choosing* a random sequence of 28 letters … it duplicates it repeatedly, but with a certain chance of random error – 'mutation' – in the copying. *The computer* ***examines*** *the mutant* ***nonsense phrases****, the 'progeny' of the original phrase, and* ***chooses*** *the one which, however slightly,* ***most resembles the target phrase***, METHINKS IT IS LIKE A WEASEL . . . . *What matters is* ***the difference*** *between the time taken by cumulative selection, and the time which the same computer, working flat out at the same rate* [i.e. search resources come in here --KF], *would take to reach the target phrase if it were forced to use the other procedure of* ***single-step selection***: about a million million million million million years. This is more than a million million million times as long as the universe has so far existed . . . .

Although the monkey/Shakespeare model is useful for explaining the distinction between single-step selection and cumulative selection, **it is misleading in important ways.** *One of these is that, in each generation of selective 'breeding',* ***the mutant 'progeny' phrases were judged according to the criterion of resemblance to a distant ideal target,*** the phrase METHINKS IT IS LIKE A WEASEL. Life isn't like that. Evolution has no long-term goal. There is no long-distance target, no final perfection to serve as a criterion for selection, although human vanity cherishes the absurd notion that our species is the final goal of evolution. In real life, the criterion for selection is always short-term, either simple survival or, more generally, reproductive success. [TBW, Ch 3, as cited by Wikipedia] >>

———————————

6 –> In short, CRD acknowledges that the so-called fitness function he uses is nothing but a measure of distance to target which ignores and even dismisses the issue of first needing to be on at least the shoreline of an island of complex function before hill-climbing through warmer-colder signals can be reasonable.

7 –> Thus, he inadvertently testifies that it is the injection of artificially originated active information that makes the difference in achieving complex function within reasonable search resources.

8 –> We should note that he also realises at some level that the substitution he makes is "misleading."

9 –> Indeed.

10 –> Finally we should note that genes are in effect informaiton storage devices. Until the functional machines to interpret and carry out the information exist, they are jut polymer molecules. And, the machines in question have to be coded for and organised to function: ~ 600 – 1,000 k bits of info based on observed life. (Just 1,000 bits specifies a config space so large that the 10^80 or so atoms of our observed universe across its credible lifespan will not be able to scan through 1 in 10^150 of that, effectively the scope of feasible search is a fraction not materially different from zero.) This is the basic reason why OOL is such a challenge to evolutionary materialism.

11 –> And, when it comes to origin of body plans de novo, we are talking of not 600 – 1,000 or so k bits of information but 10's – 100's of mmega bits, all of which have to be in place before function is feasible.

GEM of TKI

265

DNA_Jock

09/01/2009

6:21 am

As I said before:

> …even if there were an explicit latching mechanism in Weasel[*and all the evidence indicates that there is not one*], that would not make it a partitioned search as described by D&M. The first reported Weasel run in TBW **CANNOT** be the result of a partitioned search. Only two letters change in generation 2.
>
> Latching is irrelevant to the accuracy of D&M's citation. I am accusing D&M of mis-citation when they claim that eqn22 describes Weasel. I encourage everyone to go to EIL and play with Atom's adjustable Weasel, specifically the Proximity Reward Search, and contrast its behavior with that of the Partitioned Search.

Now don't I deserve some of your delicious word salad, kairosfocus?
Please may I have some more?

266

Joseph

09/01/2009

6:56 am

Except that latching/ ratcheting makes it a partitioned search.

That is if you understand English…

267

DeLurker

09/01/2009

7:11 am

Joseph#265

> Except that latching/ ratcheting makes it a partitioned search.

Explicit latching makes it a form of partitioned search. Dawkins algorithm does not explicitly latch.

268

DeLurker

09/01/2009

7:14 am

I would like to publicly thank Mrs. O'Leary for getting me removed from the banned list. She has been very gracious in private email, albeit strangely fascinated with hockey. 😉

269

DNA_Jock

09/01/2009

7:21 am

Joseph, I understand that you are going for the second door:
"A latching process is (necessarily) a partitioned search (as modeled in eqn22)"
which is a much better choice than the third door:
"Weasel is a partitioned search (as modeled in eqn22)", which is blatantly wrong.

The key aspect of the partitioned search is that it is a "divide and conquer" procedure(p1055), in which the search for each character is independent of the search for the other characters. Thus for partitioned search without sample replacement, the target can be found in N-1 queries (p1056) *irrespective of the target length*. There are whole categories of search algorithms that work by step-wise comparing a short search string (one or two letters) with the target (starting at one end and moving along until a match is found.) This information is then used to infer the FOO (p1054) for the target, and subsequent short search strings use the derived FOO. Useful if the FOO is unknown.
It latches, but it ain't a partitioned search.

270

BillB

09/01/2009

7:25 am

KF:

The issue of WEASEL's biological relevance is separate and unrelated to the issue of whether the citation in Dembski and Marks paper is accurate.

You have demonstrated that WEASEL does not require an explicit latching mechanism to produce any of the observed results, as we have been saying all along. Given this fact, and Dawkins description, and his statements about latching mechanisms, your claim that the existence of an explicit latching mechanism is a reasonable interpretation is not valid.

Joseph:

Latching individual letters is a partitioned search, ratcheting towards a target is only a partitioned search if individual letters are locked out of the search when they reach their target.

Partitioning means to divide up the search into separate, independent units. WEASEL is not an algorithm that searches for individual letters, it is not partitioned.

271

DNA_Jock

09/01/2009

7:38 am

Kairosfocus-
You gave Nephmon 1,259 words of tasty word salad, changing the subject to abiogenesis and arguing from the authority of dear departed Fred Hoyle.

> 4 –> that was the heart of Sir Fred's challenge and it is the heart of the design theory challenge to the claimed spontaneous origins of life and of novel body plans. That is, we do observe FSCI of an order of complexity of 1,000 or morte bits forming all the time: by DESIGN. But, we have never observed this by spontaneous forces tracing to chance + necessity without intelligent intervention. And, on search resource exhaustion grounds — as CRD actually concedes in the relevant passage — it is utterly implausible that we will ever get to such complex functionality on the gamut of our observed cosmos. [**And please resist the red herrings and strawmen in the usual rebuttals of Hoyle a la Wikipedia. Remember you are in the presence of a Nobel Prize equivalent holder here. he may be wrong on points, but he is not going to be making simplistic blunders**, and even his errors will be instructive.
> [*Emphasis added*]

Soooo, "tornado in a junkyard makes a 747" is an appropriate analogy *because an eminent guy once said so* and we should "resist the red herrings and strawmen" that might disagree.
Now, I might be persuaded that Sir Fred was a "Nobel Prize equivalent holder", but I am certainly not in his presence, any more than I am in Sir Isaac's presence (A man with some very interesting ideas about the Philosopher's Stone and about the Holy Trinity, by the way). Much has been learnt since these eminent gents passed on, and if they were alive today, they might disagree with their previously held beliefs.

And, anyway, it is an argument from authority. Again (see post 136).

One small request though : could my word salad be on topic – whether eqn22 could possibly describe TBW Weasel? Thanks.

272

Gaz

09/01/2009

8:27 am

What a fascinating discussion! Most interesting, for me, is the parallels with the philosophies of the participants.

KF and Joseph, being creatures of Faith, clearly need there to be latching, even if it doesn't exist, because certain prophets say it exists in the original Weasel. Now, they can't say that the prophets are wrong, despite the objective evidence, otherwise the rest of the philosophy as espoused by the prophets is questionable. So they go through all kinds of linguistic contortions to claim latching exists, even to the point of defining "latching" as something it isn't.

As in this thread, so in their religious philosophies. For "latching" read "God".

273

nephmon

09/01/2009

12:29 pm

LOK Gaz@272. So in other words, “Latchingdidit” 😀

274

nephmon

09/01/2009

12:30 pm

Did I really manage to mistype “LOL”? Shurely shome mishtake!

275

nephmon

09/01/2009

2:33 pm

kf: I’m very impressed by your ability to cite background material, use long words, and deflect the topic in order to suit your purposes. You ability to answer simple questions clearly and succinctly, not so much.

I asked about the differences between the mechanism of natural selection and its first-order approximation in WEASEL. You answered by talking about Hoyle’s views on the origin of life.

(Aside: Creationists love to harp on about Darwin’s “racism” as though [if it were true] it invalidates everything else he has to say. But they don’t mention much about the astonishing sexism in some of Hoyle’s novels. Oh wait,I guess Hoyle was just a product of his time.)

Anyway, by definition, WEASEL is “picking up the story” some way down the line from its beginning. You can hark back to the start of it all if you like, but it’s not really salient to this discussion.

In any simulation and discussion of it, there have to be some “givens”. Unfortunately, you don’t seem to be prepared to “give” anything, so it makes it hard to have a reasoned debate with you.

I totally disagree with your point (3) in 264. The distance of a given letter sequence from the target in WEASEL is a *model* of a gene’s fitness to its environment. There fore it *is* “a good analogy of the power of random variation and natural selection, which is premised on the origin of information by chance variation, and the selection of FUNCTION on superiority of some sub-populations,” because that distance is the way we happen to modeling functionality. You could easily imagine a much more sophisticated model where the letter-sequence is transformed into an object (in the computer science sense) whose attributes allow it to perform better or worse in a virtual world.

Simple example to make it more concrete: the first letter of sequence corresponds to length of the neck of the virtual creature. One of the attributes of the environment is the average and SD of the height of foliage on the virtual world’s trees. Part of the candidate sequence’s “fitness” is how well its phenotype’s neck length enables it to feed. This is just a more involved simulation than “the Hamming distance between the first letter of the sequence and the “m” of “methinks”, but doesn’t alter the fundamental nature of it.

276

BillB

09/01/2009

4:40 pm

nephmon:
I’m afraid KF will probably regale you with his conspiracy theory about Lewontin and how evolution is all a materialist plot that depends on a material origin of life, which means OOL is relevant to everything, apparently.

I suspect that if a step by step origin of life is ever established, KF will most likely say that it begs the question of where matter came from, and that that was the issue all along, and by concentrating on OOL we were simply distracting with clouds of oil of ad hominem burning straw men meant to poison and confuse …

I’m tempted to see what would happen if you gave one group of computer science students Dawkins WEASEL description, gave another group Dembski and marks’ description, and asked them both to implement the algorithm as described without any other prompting or background.

I can’t imagine either group ever producing anything close to functionally equivalent programs. There is just no way you can read those two descriptions, implement them, and come up with the same piece of software.

277

nephmon

09/01/2009

5:55 pm

Agreed, Bill. Actually, I’ve been thinking it would be an interesting interview question for a software developer candidate (implementing WEASEL, not necessarily doing a comparison with D/M). If they start talking about specified complexity, I’ll know to lower my expectations 😀

Post a Response

You can use these tags: <a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>

You must be logged in to post a comment.

RSS: Entries | Comments |

Design by Zeit Studios
Hosted by Network Omega