Problem difficulty

Fernando Lobo

University of Algarve

Outline

- Problem difficulty
- Deception
- Problems of bounded difficulty
- The No Free Lunch Theorem

Problem difficulty

- Some problems are easier than others.
- We have seen the onemax problem at the beginning of the course.
  - Do you think it's easy or hard for an EA?
  - Why?
- Can you think of a harder problem?

Problem difficulty (cont.)

- Consider this function:

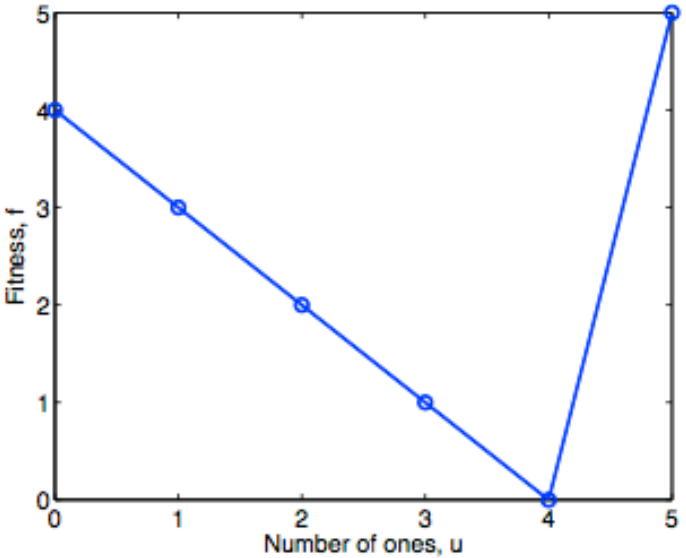
$$f(x) = \begin{cases} 5 & \text{if } x = x^* \\ 3 & \text{otherwise} \end{cases}$$

- What will happen if you run an EA on this problem?

Problem difficulty (cont.)

- The previous problem is often referred as a *Needle In A Haystack* (NIAH).
- An EA does no better than random search.

Trap function



$$f_{trap}(u) = \begin{cases} k & \text{if } u = k \\ k - 1 - u & \text{otherwise,} \end{cases}$$

- Global optimum is 11111, with fitness 5.
- Solutions close to 11111 (like 11110 or 11011 are very bad, have fitness 0).
- Second best solution (with fitness 4) is 00000, which is maximally away from 11111.

Trap function

- The trap function has two things that cause difficulties to EAs:
  - The global optimum is isolated.
  - The search is mislead to a local optimum that is far away from the desired solution.

- If k is large, EAs get trapped in the 000…0 solution.

- The trap function is an example of a deceptive function. (we will look at the formal definition later on.)

Bounded difficult problems

- If k=l there's no hope that an EA can get to the global optimum.

- A commonly used test function is the concatenation of m sub-functions, each being a small sized trap function defined over k bits.

$$f_{mk}(X) = \sum_{i=1}^{m} f_{trap}(x_{I_i}),$$

Example with *m*=6, *k*=4

- x = 111101110000010110111000
  - 1st trap: bits 1-4
  - 2nd trap: bits 5-8
  - …
  - 6^{th} trap: bits 21-24

$$\underbrace{1111}_{4}\underbrace{0111}_{0}\underbrace{0000}_{3}\underbrace{0101}_{1}\underbrace{1011}_{0}\underbrace{1000}_{2}$$

- f(x) = 4 + 0 + 3 + 1 + 0 + 2 = 10

m-k trap function is harder than onemax

- Compare 100 bit onemax with 20-5 trap function.
- Both have 100 bit length.
- The size of the search space is the same, but one is harder than the other.
- Try running a GA and see if you can solve it.

Tight and loose linkage

- Previous m-k trap is tougher than onemax, but a simple GA can still solve it quickly.
  - with properly set parameters, more on this later in the course.
- But if the bits that define the traps are far away from each other, the problem becomes much more difficult.
  - In fact, simple GAs cannot solve such problems efficiently.
  - But other more sophisticated GAs can.

No Free Lunch Theorem

- An informal definition:
  - Averaged over all possible problems, all non-revisiting black-box optimization algorithms perform the same.

- This means that we cannot make general statements saying that algorithm A is better than algorithm B for all possible problems.

No Free Lunch Theorem: implications for EC

- Real world problems are not representative of all possible problems (i.e., they are not random problems).
- Most of them have some sort of structure that can be exploited (or learned) by an optimization algorithm.
- Empirical success of EAs shows that they work reasonably well for many difficult real world optimization problems.