An Information Perspective on Evolutionary Computation

Yossi Borenstein
University of Essex

1

"Are some classes of combinatorial optimization problems intrinsically harder than others, without regard to the algorithm one uses, or can difficulty only be assed relative to particular algorithms?" Macready, Wolpert (1996)

2

Outline

- Focusing on randomized search heuristics, this tutorial presents an answer to this question which is orthogonal to that of the NFLTs:
  - Not only intrinsically hard fitness functions exist, but the vast majority of all functions is hard
- This is shown using arguments based on the notion of Kolmogorov complexity.

3

Overview

- Introduction
- Kolmogorov Complexity
  - Introduction
  - Fitness functions as binary strings
  - Semantics (the representation issue)
  - Meaningful information
- Decomposition of fitness functions
  - Meaningful information for evolutionary algorithms
  - Comparison based selection mechanism
  - The counting argument: almost all possible problems are difficult
  - Entropy based bounds
- Conclusion

4

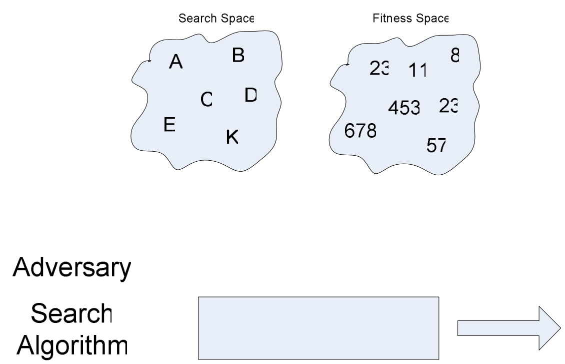
GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-698-1/07/0007.

Introduction

Preliminaries

- Let $f : X \to Y$ where *X*,*Y* are finite sets.
- We consider maximization problems where the objective is to find $x_{opt} \in X$
- Small number of global optima

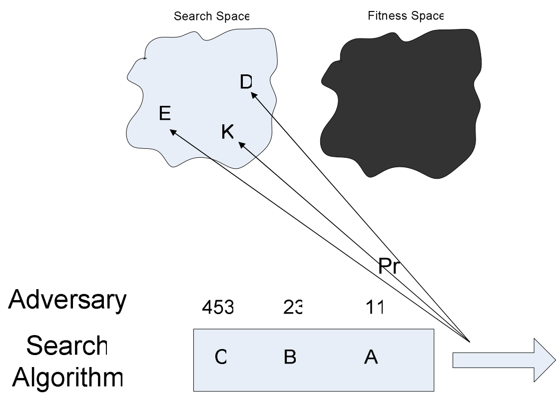
No Free Lunch -- The *adversary* argument --



No Free Lunch -- consequences --

- *"All search algorithms are equivalent when compared over all possible discrete functions."* Wolpert, Macready (1995)
- Considering a single function – "regardless of *f* it is always possible to construct the optimal algorithm." Macready, Wolpert (1996)
  - Simply: generate the optimum in step 1.

More Realistic Scenario



9

More Realistic Questions…

(for the black box scenario)

- The feasibility question:
  - Given a sample of the search space:
    - e.g., { {C,453}, {B,23}, {A,11} }
  - Is it possible to select in a ***rational*** way a new point from the search space?
    - e.g., {E, K, D}
- Rational:
  - the sample can be obtained using a regular expression
  - Assuming that the same rules apply to the whole space, sample a new point such that fitness is maximized

10

Related questions…

- Can the notion of regular vs. non-regular samples be formalized?
- If so, is it possible to *quantify* the level of regularity of a particular object?
- Is it possible to define accordingly an intrinsic notion of hardness?
- How this is related to evolutionary algorithms?

11

Kolmogorov Complexity

“Measuring the randomness of a single object”

12

Kolmogorov complexity -- a single object --

- KC is a function, K : {0,1}* → *N,* which represents the size of the minimal program that can generate a string and halts.
  - LOW: *000000000 :* print 9 times '0'
  - HIGH: *011011010* : print '011011010'
- Cannot be computed!

13

Incompressible strings

- For each *n* there are:
  - 2^n possible binary strings
  - $\sum_{i=0}^{n-1} 2^i = 2^n - 1$ shorter descriptions
- At least one string cannot be compressed at all!
- For every constant *c* we call a string *x* *c-incompressible if* $K(x) \geq l(x) - c$

14

Almost all strings are incompressible

- When *c* is small we call the string simply incompressible.
- By a simple counting argument:
  - At least 1 string is *0-incpomressible*
  - *At least one-half(!) are 1-incpomressible*
  - *At least three-fourth are 2-incompressible*
  - *At* least $(1-1/2^c)th$ are *c-incompressible*

15

Kolmogorov complexity -- functions --

- **Assuming a priori order of X**, Kolmogorov complexity can be defined for functions:

x	$f(x)$	$f(x)$ (Binary)
00000000000000	3	011
00000000000001	4	100
...	...	...
01010101010101	6	110
...	...	...
11111111111111	2	010

Binary Representation of f(x): 011100...110...010

16

Kolmogorov Complexity -- hardness --

- If the KC of a fitness function is high, it contains no regularities (otherwise it could have been compressed). Therefore, no *rational* inference is possible.
- Almost all functions are incompressible therefore almost all functions are intrinsically hard.
- Is this that simple?

17

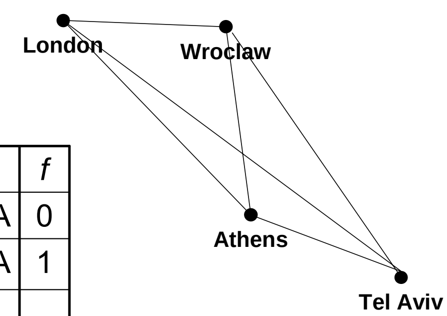
Alas, things are more complicated…

18

Semantics

19

Travelling Salesperson Problem



Solution	*f*
LO-WR-AT-TA	0
AT-LO-WR-TA	1
….	…
WR-TA-AT-LO	0

Binary representation: 01…0

20

Representation or genotype-phenotype mappings

Representation (genotype)		Search space (phenotype)
0010101010		LO-WR-AT-TA
1101001010		AT-LO-WR-TA
….	Π ⇒	….
0001001111		WR-TA-AT-LO

21

KC – Semantics (representing a problem in a computer)

Arbitrary (independent)

00	London
01	Tel Aviv
10	Wroclaw
11	Athens

⇒

Search Space (natural order)

00011011	f(LTWA)=0
00100111	f(LWTA)=1
…	…
11100100	f(AWTL)=1

Binary representation of *f*

010010010101010101001001

\+

00011011 => LO-TA-WR-AT ??

n (K)

Search Space (a possible representation)

01101100	f(LTWA)=0
11100100	f(WATL)=0
…	…
10001101	f(ATWL)=1

Binary representation of π*f*

000000000000011111111

\+

11100100 => WR-AT-TA-LO ??

n! (4K)

22

KC - Functions

- As long as the representation is not chosen Kolmogorov complexity is not defined.
- Once a representation is chosen, it is possible to distinguish between compressible functions to incompressible ones.
- Only a random (i.e., incompressible) permutation of the search space can transform a random function to a regular one.
- Real world problem usually have a *natural* representation which can be used as a reference to any other

23

Kolmogorov complexity -- hardness --

- So, as long as a natural representation exists KC is well defined!
- Does this imply that the vast majority of all fitness functions (or alternatively, all representation of the same fitness function) are intrinsically hard?

24

Alas, things are more complicated…

25

Meaningful Information

Does high KC imply hardness?

26

Meaningful information

- A single string sometimes represents more than one source of information…
- One has to distinguish between the relevant or meaningful information and the irrelevant one…

27

An example…

28

Subjectively meaningful

- The previous example:
  - an absolute notion of meaningfulness
- A subjective notion of meaningful information
  - The relevant information depends on the way one intends to use it.

29

An example…

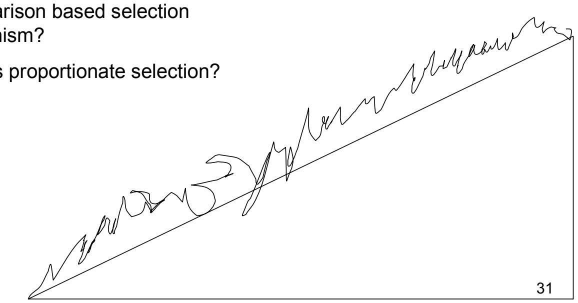
- The relevant information in an image depends on the resolution of the monitor…

30

What about functions, hardness and evolutionary computation?

Is this hard?

- Local search
- Comparison based selection mechanism?
- Fitness proportionate selection?



31

Decomposition of Fitness Functions

So, what is the meaningful information in fitness functions for genetic algorithms?

32

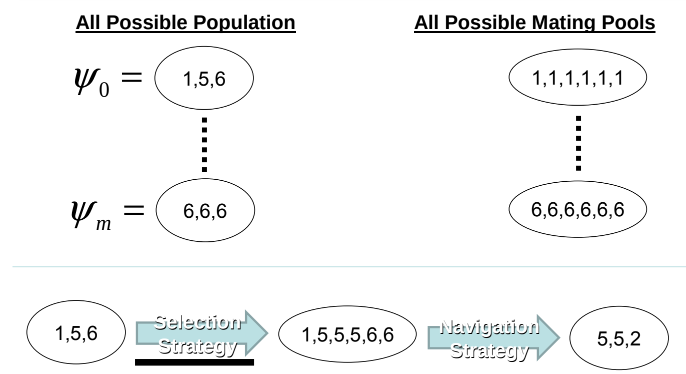
Simple genetic algorithm

- Repeat *n Times:*
  - Selection:
    - Select **two** individuals from the population
  - Crossover, Mutation
    - Apply operators to obtain a new **individual**
  - Selection Strategy:
    - Select ***2n*** individual (mating pool)
  - Navigation Strategy:
    - Apply search operators to obtain new **population**

33

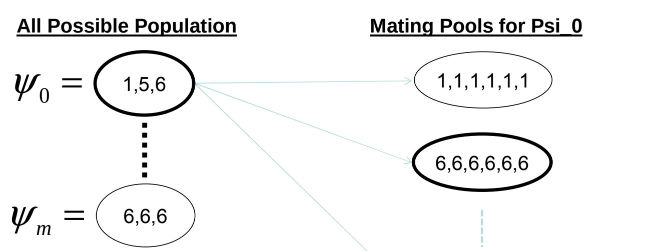
Enumerations...

Let X={1,2,3,4,5,6}, Population size: 3



34

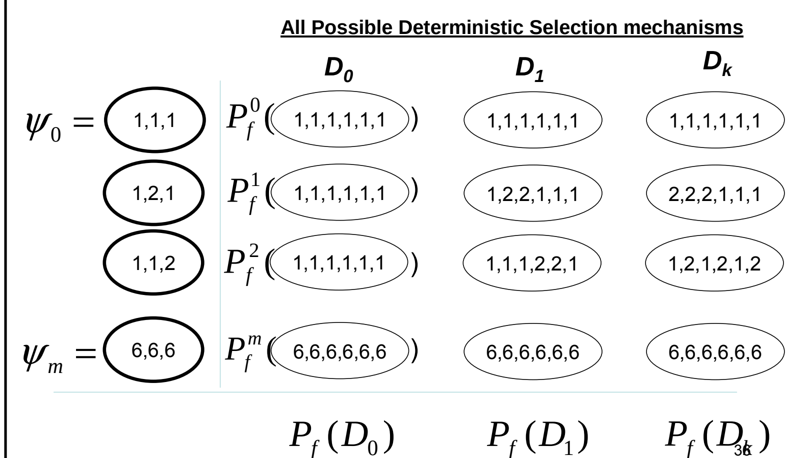
The selection strategy



$$P_f^0(\text{6,6,6,6,6,6}) = \Pr(\text{6,6,6,6,6,6} \mid \text{1,5,6}, f, a)$$

35

Some more enumerations...



36

The meaningful information of the fitness function

- Depending on the population size:
  - The number of all possible deterministic selection mechanisms is finite.
- Given a fitness function: *f*
- The GA defines the distribution: P_f
  - Which is (finally) the meaningful information in the fitness function!!!!
- But how the KC of P_f is connected with performance?

37

Comparison based Selection Mechanism

A concrete example: from information to performance

38

An alternative GA (deterministic selection mechanism)

- Select with probability P_f the deterministic selection mechanism *D*
  - Initialize population
  - Repeat until stopping criteria is met
    - Select mating pool according to *D* (deterministic step)
    - Use selection + crossover to generate next population
  - restart

39

Comparison based selection mechanism

Fitness Function

X	f(x)
000	6
001	5
010	5
011	3
100	2
101	2
110	2
111	7

Decomposition (the P_f^i's)

X	000	001	010	011	100	101	110	111
000		1	1	1	1	1	1	0
001	0		0.5	1	1	1	1	0
010	0	0.5		1	1	1	1	0
011	0	0	0		1	1	1	0
100	0	0	0	0		0.5	0.5	0
101	0	0	0	0	0.5		0.5	0
110	0	0	0	0	0.5	0.5		0
111	1	1	1	1	1	1	1	

Possible Deterministic selection mechanism

X	000	001	010	011	100	101	110	111
000		L	L	L	L	L	L	U
001			U	L	L	L	L	U
010				L	L	L	L	U
011					L	L	L	U
100						U	L	U
101							L	U
110								U
111								

40

Meaningful information

- Some of the elements in the matrix are defined by the partial order of the fitness function (gray area)
- The other elements depend on the realization of the probability distribution defined over the *D*'s. (white area)

$$t(x,y) = \begin{cases} 1 & \text{if } f(x) > f(y) \\ 0.5 & \text{if } f(x) = f(y) \\ 0 & \text{if } f(y) > f(x) \end{cases}$$

X	000	001	010	011	100	101	110	111
000		L	L	L	L	L	L	U
001			U	L	L	L	L	U
010				L	L	L	L	U
011					L	L	L	U
100						U	L	U
101							L	U
110								U
111								

41

Kolmogorov complexity and performance (I)

- Let us assume matrix with no 0.5's.
- The maximum KC is: $\log(2^n!) \approx O(2^n \log 2^n)$
- When *D* is incompressible the function contains no regularities and hence, the performance cannot be better than random search.
- The counting argument implies that the vast majority of all such *D's* is incompressible.

42

Kolmogorov complexity and performance (II)

- Let us assume matrix with **almost only** 0.5's.
- The KC is: $O(\log n) << O(2^n \log 2^n)$
- Does it implies that the function will be easy?
- The needle-in-a-haystack is clearly a counter example!

43

Entropy

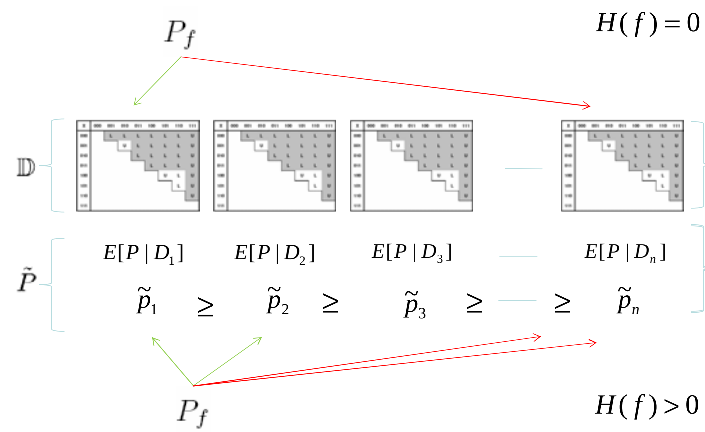
44

Properties of P_f

- For each 0.5 entry the algorithm chooses randomly one of two possible values
- It follows that P_f is uniformly distributed over $2^{\#(0.5)}$ possible deterministic selection mechanisms.
- The entropy of P_f is directly associated with the number of 0.5 in the matrix.

45

Upper and lower bounds



46

Kolmogorov complexity and performance (III)

- The KC of a typical D is:

$$2^{n-1}(2^n-1) > 2^n \log 2^n >> \log n$$

- The performance over the vast majority of D's should be equal to that of a random search.
- The "average" performance value should be equal to that of a random search.

47

Conclusion

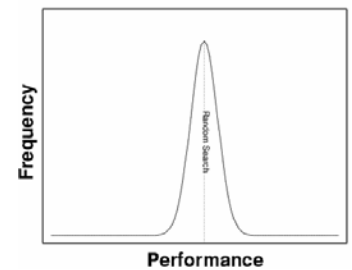
48

"Are some classes of combinatorial optimization problems intrinsically harder than others, without regard to the algorithm one uses, or can difficulty only be assed relative to particular algorithms?" Macready, Wolpert (1996)

For randomized search heuristic…

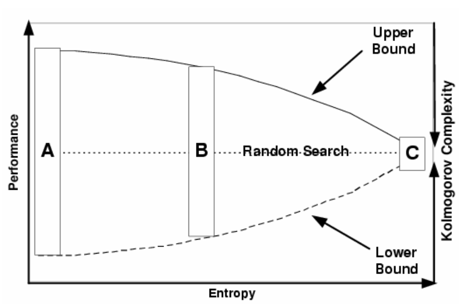
Conclusion I

- *Almost all the performance values are identical and equals* the expected performance of a random search.*



Conclusion II

- The higher the entropy of P_f the closer the performance to that of a random search.



Concluding remarks

- This tutorial focused only on two aspects of difficulty (KC and entropy)
  - Naturally, more criteria (including that of the NFLTs) exist.
- The relation to KC and hardness is not straight forward. Any interpretation based on the KC of a fitness function should be very cautious.

References/Further reading

- D.H. Wolpert and W.G. Macready. *No free lunch theorems for optimization*. IEEE Trans Evolutionary Computation, 4:67–82, 1997.
- C. Schumacher and M. Vose and D. Whitley. *The No Free Lunch and Problem Description Length*. GECCO 2001.
- W.G. Macready and D.H. Wolpert. *What makes an optimization problem hard?.* Complex , 1:5:40--46, 1996.
- P. Grünwald and P. Vitanyi. *Algorithmic Complexity*. Handbook on the Philosophy of Information. To appear.
- T. English. *Optimization is Easy and Learning is Hard In the Typical Function*. Proc. 2000 Congress on Evolutionary Computation (CEC 2000) , pages 924--931, 2000.
- H. Buhrman, M. Li, J. Tromp, P. Vitanyi . *Kolmogorov Random Graphs And The Incompressibility Method*. SIAM Journal on Computing, 29:590—599.
- Y. Borenstein, R. Poli. *Kolmogorov complexity, Optimization and Hardness*, IEEE CEC 2006
- Y. Borenstein, R. Poli. *Information Perspective of Optimization*. PPSN 2006: 102-111.
- Y. Borenstein. *What Makes an Optimization Problem Hard? An Information-Theoretic Perspective*. PhD Thesis, Chapter 3.

53