(End of GA Theory Slides)

There were some substititions (approximations) made in the argument by Goldberg that do not change the problem with the theorem.

assumed $(1-p_m)^{o(H)} \approx (1-o(H)p_m)$

and $\left[1-p_c\frac{\delta(H)}{n-1}\right]\cdot\left[1-o(H)\cdot p_m\right]$

$\cong \left[1-p_c\frac{\delta(H)}{n-1}-o(H)p_m\right]$

Handout 10-31-11

Equations from Goldberg book on Genetic Algorithms with these substitutions

Effect with only crossover

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{n-1} \right]$$

adding mutation

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{n-1} - o(H) p_m \right]$$

- Even with these approximations the same problem occurs in the asymptotic limit as t goes to infinity

Significance

- Although the argument does not prove convergence asymtoptically, it does give some explanation about why GA works when you have a finite number of generations.
- It is possible for a fixed number of generations K, that the presence of highly fit schema will be increased and hence good solutions will be found.

Significane

- It also shows that the closeness of decision variables in a binary string can be important.
- For example if I think decision variable A and decision variable B are correlated in the benefit they give to the objective function , should I put them close together or far apart?

Convergence of Tabu Search

- There is no Theory of Convergence for (deterministic) Tabu Search.
- Hence we cannot say that Tabu Search converges asymptotically.
- Convergence of global optimization methods for discrete problems seems to be more possible with stochastic search methods.
- Tabu Search is a deterministic method. The inventor of Tabu Search feels it is better to control the search method deterministically rather than to subject it to stochastic search.

“No Free Lunch” Theorem

- There is a famous theorem in Computer Science called the No Free Lunch Theorem
- The main point of this Theorem is that you cannot guarantee that any one algorithm A is “better” than another algorithm B if you apply both A and B to <u>all possible problems</u>.

References

Original Paper:

Wolpert, D.H., W.G. Macready, "No Free Lunch Theorems for Optimization,"

IEEE Transactions on Evolutionary Computation, 1 (1) 67-82,1997

Commentaries on Original Paper:

Ho, Y.-C. and D. L. Pepyne, "Simple Explanation of the No Free Lunch Theorem of Optimization", *Cybernetics and System Analysis,* 38 (2), 2002

You are not expected to read these papers and there will be no homework or exam questions on material that is not presented in class. However, if you are interested I recommend the first paper.

No Free Lunch Theorem Assumes Discrete Decision Variables

- The "No Free Lunch" Theorem is applied to optimization of a function f(x), where x is a vector of discrete variables (e.g. binary string, vector of integers, permutation, etc.). So the decision space S is discrete.
- The decision space S is bounded (necessary for global optimization)
- Y is the "range", which is the bounded set of values of f(x)

Impact of No Free Lunch Theorem

- The results of this theorem have a major impact on how we view heuristic algorithms for combinatorial (discrete) problems.

- Let us call J the objection function (cost = J in text.) Example of J would include: a) a cost function for a cellular network problem with a specific number of channels and interference matrix <u>or</u> b) a cost function for a specific satisfiability problem.
-

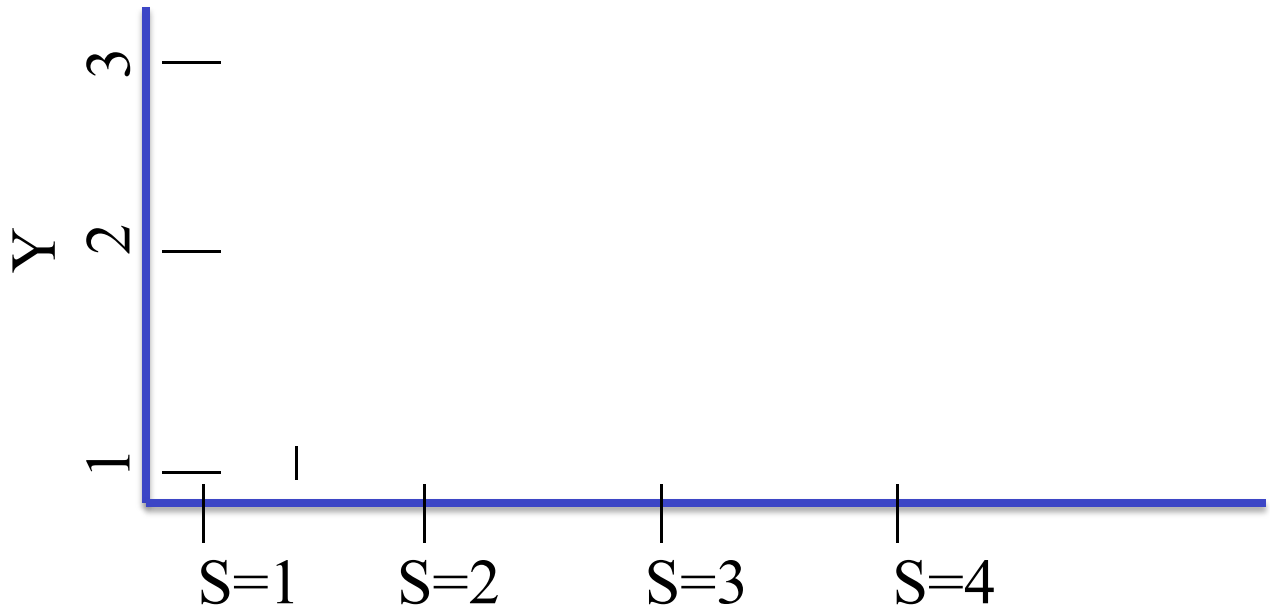
Computing the Number of Possible Cost Functions

- Assume J is an objective function with a domain S and range Y where S and Y are finite, e.g. the number of elements in S, called |S|, is finite and the number of elements in Y, called |Y|, is finite. (For example, S can be the set of all binary strings in length N, which has 2^N elements.)

- As a result, there are finite number of possibilities for the function J. Let Z be the set of functions J.

- Number of possible functions J = $|Z|$=_____???

Function J from S to Y . Let's Assume |S|=4 and |Y|=3



- Number of possible functions J = |*Z*|=_____*???*

Defining a Search Algorithm

- Let F_i denote a search algorithm such that the points are never repeated, i.e. if it evaluates J (x*) once it does not evaluate J (x*) in a later iteration.

The No Free Lunch Theorem says that

- The performance of any two search algorithms F_1 and F_2 is the same when averaged over all possible values of the functions J contained in the set Z (For a given S and Y).

- Hence if F_1 does better than F_2 on some subset of objective functions M ε Z , then F_2 does better than F_1 on another subset ε Z .

- More precisely, assume an algorithm F_i does n steps for a given objective function. Let M be the minimum objective function value that has been found during the search. Hence M depends upon J, n, F; and M is a random variable, so we can characterize it as
- Let $P(M|J,n,F_i)$=probability that M is the minimum solution of problem J found in n evaluations of J using the algorithm F_i .

- *(Notes for those of you who want to read the paper: In the paper $M=\Phi(d_m^y)$ on page 69, and $a=F_i$, $m=n$, $f=J$. In the previous slides I had discussed a "histogram" which refers to all the elements of the vector d_m^y, but since this is harder to explain, I focused on the $M=\Phi(d_m^y)$ in the revised slides.)*

Equation for NFL

- Then the No Free Lunch Theorem states for any F_1 and F_2 functions on same domain:

$$\sum_{J \varepsilon Z} P(M \mid J, n, F_1) = \sum_{J \varepsilon Z} P(M \mid J, n, F_2)$$

- It follows from this that if a search algorithm F1 is better than F2 on some subset in Z, then there is another subset of in Z where the search algorithm F2 must perform better than F1.
- The general theorem is on page 69 and the specific case for the minimum is discussed in the following paragraph in the paper (optional reading).

Example of Significance of NFL for Algorithms we have studied

- F_1 = simulated annealing and
- F_2 = random Search
- Then if Simulated annealing is better than random search on one problem J, then random search will be better than simulated annealing on another problem given that both of the algorithms evaluates the cost function m times (not counting returning to any previously evaluated points).

How could this be true?

Remember that there can be some pretty pathological objective functions J.

For example consider the case where J(s) is a random number independent of s.

What does this tell you about the kinds of problems that we would like to solve, which are based on a search for real solutions for real problems?

Contradiction

- How does this relate to our statistical analysis when we compare algorithms?

- What is the meaning of the statistical analysis given the No Free Lunch Theorem?

- Does the Statistical Analysis contradict the No Free Lunch Theorem?