The No Free Lunch Theorems for Optimisation: An Overview

James Montgomery
PhD Student
School of Information Technology
Bond University

November 22, 2002

1 Introduction

Many algorithms have been devised for tackling combinatorial optimisation problems (COPs). Traditional Operations Research (OR) techniques such as Branch and Bound and Cutting Planes Algorithms can, given enough time, guarantee an optimal solution as they explicitly exploit features of the optimisation function they are solving. Specialised heuristics exist for most COPs that also exploit features of the optimisation function to arrive at a good, but probably not optimal, solution. There also exist a number of metaheuristics, algorithms that describe how to search the solution space without being tied to any one problem type. Some of these may be tailored to make them suit a particular problem better. However, there are a number of so-called "black-box" optimisation algorithms, which use little or no problem-specific information. Key examples of the black-box approach are Evolutionary Algorithms (EAs) and Simulated Annealing (SA). Random search is also a black-box optimisation algorithm, and so represents an important benchmark against which the performance of other search algorithms may be measured.

Given an optimisation problem (i.e. objective function) f and an algorithm a, it is important to have some measure of how well a performs on f. Moreover, given empirical evidence of a's performance on f, is it possible to make generalisations about a's performance on other functions, either of the same or different type as f? Intuition would have one believe that there are some algorithms that will perform better than others on average. However, the No Free Lunch (NFL) Theorems state that such an assertion cannot be made. That is, across all optimisation functions, the average performance of all algorithms is the same. This means that if an algorithm performs well on one set of problems then it will perform poorly (worse than random search) on all others.

To put the NFL theorems in their proper context, it is important to understand what they are and are not saying. From [2], p.69:

> ...[W]e cannot emphasize enough that **no claims whatsoever** are being made in this paper concerning how well various search algorithms work in practice. The focus . . . is on what can be said *a priori*,

without any assumptions and from mathematical principles alone, concerning the utility of a search algorithm.

2 The NFL Theorems

Wolpert and Macready [1, 2] investigated what claims can be made *a priori* about optimisation algorithms using probability theory. The reasons for choosing probability theory are given in [2]. They used two approaches in their study, the first is to analyse how a particular algorithm performs over all problems, while the second involves holding the optimisation function constant and analysing how it is dealt with over all algorithms. The former is discussed largely in [2], while the latter is discussed in more detail in [1].

The notation used and first NFL theorem are presented before a discussion of the implications of the theorem. Define $\mathcal{X}$ to be the finite search space and $\mathcal{Y}$ to be the finite space of "cost" values. Optimisation problems f are represented as mappings $f : \mathcal{X} \mapsto \mathcal{Y}$. $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$ then represents the space of all possible problems.

As an algorithm progresses, it will visit and evaluate points in the search space. The NFL analysis considers only distinct evaluations of members of $\mathcal{X}$ as this makes the analysis easier, even though this is an unrealistic given that most black-box optimisation algorithms revisit points in the search space.

A "sample" of size m is a time-ordered series of points in $\mathcal{X} \times \mathcal{Y}$, and represents the points visited by an algorithm in its search. Samples are denoted by $d_m \equiv \{(d^x_m(1), d^y_m), \ldots, (d^x_m(m), d^y_m(m))\}$. According to [2], a sample also includes the neighbours of visited solutions. For instance, a hill-climbing algorithm will have only one current solution, but at each iteration will evaluate all neighbouring solutions, which are also part of d_m. Performance measures of an algorithm are denoted by $\Phi(d^y_m)$, and may represent any evaluation of an algorithm's performance based on the costs in its current sample d_m.

The performance of an algorithm can be measured using $P(d^y_m|f, m, a)$, which is the conditional probability of obtaining the sample costs d^y_m on optimisation function f by applying algorithm a for m iterations. Given this, one can analyse how two algorithms compare across all optimisation problems by comparing the sum of conditional probabilities across all possible f for each algorithm. Through their analysis, Wolpert and Macready found that $P(d^y_m|f, m, a)$ is independent of a.

Theorem 1 *For any pair of algorithms a_1 and a_2,*

$$\sum_f P(d^y_m|f, m, a_1) = \sum_f P(d^y_m|f, m, a_2).$$

The second NFL theorem is for time-dependent (i.e. dynamic) optimisation problems, and gives the same result. The NFL results also hold for stochastic algorithms, as these are effectively deterministic given a particular random seed. It is beyond the scope of this summary to describe the proof of these theorems.

3 Implications of the NFL Theorems

The fundamental implication of the first two NFL theorems is that if an algorithm performs well on one class of problems then it must do poorly on other problems. Moreover, an algorithm that performs well on one class of problems must perform worse than random search on all remaining problems. Two immediate conclusions can be drawn from this. First, running an algorithm on a small number of problems with a small range of parameter settings may not be a good predictor of that algorithm's performance on other problems, especially problems of a different type. Second, the result of Theorem 1 holds as long as knowledge of a problem's structure is not incorporated into a search algorithm. In other words, incorporating problem-specific information into an algorithm will improve its match to that problem and hence, its performance.

4 Other Findings

Wolpert and Macready derive a number of other results from the NFL theorems, some of which are summarised here.

It is possible to consider the NFL results from a geometric perspective, which allows for the closeness of the match between an algorithm a and an optimisation problem f to be determined. However, in practice exploiting this feature of the probability functions used in the NFL theorems is extremely difficult.

While the NFL theorem states that the average performance across all problems is the same for all algorithms, in certain circumstances distinctions may be made between different algorithms. Consider two deterministic algorithms a_1 and a_2. Let it be the case that there are a number of problems for which the costs produced by a_1 are considerably better than those produced by a_2. For the NFL theorem to hold, it could be the case that there are many more problems for which a_2 performs better, but only by a small amount. Hence, if the problems we are largely concerned with are those for which a_1 performs much better than a_2, it is possible to make a distinction between the two algorithms.

5 Summary

Black-box optimisation algorithms exploit little or no information about a problem when solving it. Yet they often perform quite well (better than random search) on many of the problems we are interested in. Paradoxically, the NFL theorems offer one way to explain the success of these techniques. The first NFL theorem shows that all algorithms exhibit the same average performance taken across *all* possible problems. Considered from a geometric perspective, the performance of an algorithm is related to its *alignment* with the problem being solved. Hence, given that the space of all possible problems must contain problems for which an algorithm is well aligned as well as problems for which a problem is not well aligned, its performance will, across all problems, be no better than random search. However, in practice we are not interested in the range of all possible problems, but a much narrower group for which there is often quite close alignment between problem and algorithm. Furthermore, if an algorithm can make use of knowledge of the problem structure (or can derive such knowledge) then its alignment with that problem will be improved with a corresponding improvement in its performance on that problem.

References

[1] Wolpert, D. H., & Macready, W. G. (1995). *No Free Lunch Theorems for Search* (Technical Report SFI-TR-95-02-010). Sante Fe, NM, USA: Santa Fe Institute.

[2] Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82.