INFORMS Journal on Computing
Vol. 19, No. 2, Spring 2007, pp. 273–279
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1902 | 0273

DOI 10.1287/ijoc.1060.0194

Conditions that Obviate the No-Free-Lunch Theorems for Optimization

Gary J. Koehler
Department of Decision and Information Sciences, 351 Stuzin Hall,
The Warrington College of Business Administration, University of Florida,
Gainesville, Florida 32611, USA, koehler@ufl.edu

Roughly speaking, the no-free-lunch (NFL) theorems state that any blackbox algorithm has the same average performance as random search. These results have largely been ignored by algorithm researchers. This paper looks more closely at the NFL results and focuses on their implications for combinatorial problems typically faced by many researchers and practitioners. We derive necessary conditions for the NFL results to hold based on common problem structures. Often it is simple to verify that these conditions are *not* present in the class of problems under investigation, thus providing a theoretical basis for ignoring the doleful implications of NFL giving justification for believing there might be a dominant algorithm for the problem class under study. We apply our results to three common classes of problems. We find that only trivial subclasses of these problems fall under the NFL implications.

Key words: blackbox search; no free lunch
History: Accepted by Michel Gendreau, Area Editor for Heuristic Search and Learning; submitted May 2004; revised February 2005; accepted November 2005.

1. Introduction

Roughly speaking, the no-free-lunch (NFL) theorem for optimization by Wolpert and Macready (1995, 1997) states that any two blackbox algorithms (deterministic or stochastic) have the same average performance over all problems. Random search is one such blackbox algorithm, so all blackbox algorithms have the same average performance as random search over all problems. What is meant by "blackbox algorithm," "performance," and "all problems" will be made more precise below. One immediate implication of this result is that when a new algorithm is shown to perform better than others over some test-bed of problems, then, disconcertedly, what one has demonstrated is that the new algorithm will perform worse, on average, over all remaining problems.

In reviewing a new blackbox search method proposed by a colleague for a class of problems, I mentioned the NFL results and asked how he was avoiding their implications. His answer was basically "reviewers don't seem to know or care about the NFL results" and "anyway, recent results seem to indicate they apply to only a small set of problems" (we discuss this after Theorem 3 below). Whitley and Watson (2004, p. 2) draw similar conclusions noting "researchers had two general reactions to the No Free Lunch results. Many researchers simply dismissed No Free Lunch arguing that results concerning the set of all possible discrete functions ... is not representative of real-world problems. The other reaction ... was ... that researchers ... need to leverage extensive problem-specific knowledge" in developing algorithms.

This paper looks more closely at the NFL results and focuses on their implications for many common combinatorial optimization problems typically faced by researchers and practitioners. More specifically, we derive necessary conditions for the NFL results to hold based on common problem structures. Often it is simple to verify that these conditions are *not* present in the class of problems under investigation, thus providing a theoretical basis for ignoring the doleful implications of NFL, at the least, and giving justification for believing there might be a dominant algorithm for the problem class at the best.

We warn the reader that the absence of the NFL implications itself does not provide any justification that one algorithm is better than another. The burden of explicitly stating what kind of problems one's heuristic aims at and why they have reasons to believe that their newly designed heuristic is better suited than others still remains.

Even for classes where the NFL theorems hold, it may be the case that one algorithm "beats" another badly over a number of functions but that the second barely beats the first over a larger number of functions. Since NFL holds, the two algorithms have the same average performance over all functions, but the first has better "head-to-head" minimax behavior (Wolpert and Macready 1995, 1997).

We start with a literature review of the NFL theorems for optimization in Section 2. In Section 3 we give our main results providing necessary conditions for NFL to hold on some important classes of problems long studied by researchers that include many combinatorial problems. Our goal is to be able to verify that these conditions do not exist, thus freeing the way for hoping that a dominant algorithm might exist. Section 4 applies these results to three problem classes: partitioning problems, the knapsack problem, and the traveling-salesman problem. Results are often able to be sharpened when specific problem structures, such as these, are known. We discuss the findings vis-à-vis NP-completeness. Finally, in Section 5 we summarize our results and discuss future research areas.

Throughout the paper we use the vector e for the sum vector (the vector of ones), 1_i for the ith unit vector, x' for the transpose of column vector x, $B = \{0, 1\}$, $\mathbb{Z}$ for the set of integers, $\mathbb{Z}_{\geq 0}$ for nonnegative integers, $\mathbb{Z}_{>0}$ for positive integers, $\mathbb{Q}$ for rational numbers, and $\Re$ for the set of real numbers. $\delta()$ is the usual Kronecker delta function.

2. Literature Review

A particularly good review of NFL concepts and results can be found in Whitley and Watson (2004). Here we summarize four relevant results for our purposes. Let $\mathscr{X}$ be a finite search space of n-dimensional vectors, $\mathscr{Y}$ a finite set of objective values and $\mathscr{F} = \mathscr{Y}^{\mathscr{X}}$ be the set of all possible discrete objective functions for this problem class. Technically speaking, the "finite" restrictions are realistic for all search problems where the algorithms will be implemented on finite precision computers. We study nonrepeating, blackbox algorithms (henceforth, called just algorithms) that, after m steps, choose a new point, $x_{m+1} \in \mathscr{X}$, from the search space depending on the history of prior points examined, $x_1, \ldots, x_m$ and their objective values. This excludes algorithms that work with partial solutions like branch and bound. Popular examples of blackbox algorithms are genetic algorithms (Goldberg 1989), simulated annealing (Kirkpatrick et al. 1983), and tabu search (Glover 1989, 1990).

Algorithms that can revisit already-observed points in the search space (like genetic algorithms) can be made "nonrepeating" by adding a database to record all discovered points and outputting only newly discovered ones (Wolpert and Macready 1995, 1997). These authors also show how to handle stochastic algorithms. However, Radcliffe and Surry (1995) simply note that in practice stochastic algorithms use pseudorandom number generators so including these with an initial seed converts a stochastic algorithm to a deterministic one. Finally, algorithms that stop before m steps can be viewed as restarting with a randomly chosen new point in the search space.

The Wolpert and Macready paradigm does not deal with the computational cost of finding an element in the search space, but just with the sequence of discovered points. Culberson (1998) explores the relationship of NFL with standard algorithmic theory. We do not focus directly on this aspect here but have some comments relating our results to some complexity issues in Section 4.

Let α represent an algorithm, m the number of iterations or steps so far, $f \in \mathscr{F}$ the particular objective function being optimized and a sample of points and their $f()$ values as

$$d_m \in \{(d_m^x(1), d_m^y(1)), \ldots, (d_m^x(m), d_m^y(m))\},$$

where $d_m^x(i)$ is the ith point from $\mathscr{X}$ in a sample of size m and $d_m^y(i) = f(d_m^x(i))$. A performance measure, $\Phi(d_m^y)$, maps d_m^y to the real numbers. Typical measures are max value so far, min value so far, average objective value so far, etc. Wolpert and Macready (1995, 1997) proved the following NFL theorem (restated to comply with subsequent results given in Theorems 2 and 3).

Theorem 1 (NFL-1). *For any two algorithms α_1 and α_2, any $m \in \{1, \ldots, |\mathscr{X}|\}$, any $z \in \Re$, and any performance measure, Φ,*

$$\sum_{f \in \mathscr{F}} \delta(z, \Phi(d_m^y \mid f, m, \alpha_1)) = \sum_{f \in \mathscr{F}} \delta(z, \Phi(d_m^y \mid f, m, \alpha_2)).$$

Roughly speaking, this states that any two algorithms (deterministic or stochastic) have the same average performance over all objective functions $f \in \mathscr{F}$. Random search is one such algorithm, so all algorithms have the same average performance as random search over all objective functions. This result has been sharpened recently by Schuhmacher et al. (2001). Let π be a permutation of $\mathscr{X}$ (i.e., π: $\mathscr{X} \to \mathscr{X}$) and $\Pi(\mathscr{X})$ be the set of all permutation mappings of $\mathscr{X}$. That is, if one considers $\mathscr{X}$ as an ordered set, π is a reordering. A set $F \subseteq \mathscr{F}$ is "closed under permutation" (c.u.p.) of $\mathscr{X}$ if for any $\pi \in \Pi(\mathscr{X})$ and any $f \in F$ then $f \circ \pi \equiv g \in F$.

Theorem 2 (NFL-2). *For any two algorithms α_1 and α_2, any $m \in \{1, \ldots, |\mathscr{X}|\}$, any $z \in \Re$, any $F \subseteq \mathscr{F}$, and any performance measure, Φ,*

$$\sum_{f \in F} \delta(z, \Phi(d_m^y \mid f, m, \alpha_1)) = \sum_{f \in F} \delta(z, \Phi(d_m^y \mid f, m, \alpha_2))$$

iff F is c.u.p.

Both NFL-1 and NFL-2 implicitly assume that all objective functions are equally likely. Igel and Toussaint (2003b) remove this assumption to give the sharpest NFL result. Let $P(f)$ be an arbitrary probability function over $\mathscr{F}$. A mapping h from $\mathscr{Y}$ to the nonnegative integers such that $\sum_{y\in\mathcal{Y}} h(y) = |\mathcal{X}|$ is called a $\mathcal{Y}$-histogram (or simply a histogram). For any $f \in \mathcal{F}$, we have a histogram $h_f(y) = |\{x\colon f(x) = y\}|$. For a given histogram, h, the equivalence class, $B_h \subseteq \mathcal{F}$, is the set of all $f \in \mathcal{F}$ having histogram h.

Theorem 3 (NFL-3). *For any two algorithms α_1 and α_2, any $m \in \{1, \ldots, |\mathcal{X}|\}$, any $z \in \Re$, and any performance measure, Φ,*

$$\sum_{f\in\mathcal{F}} P(f)\delta(z, \Phi(d_m^y \mid f, m, \alpha_1)) = \sum_{f\in\mathcal{F}} P(f)\delta(z, \Phi(d_m^y \mid f, m, \alpha_2))$$

iff for all histograms, h, we have that $f, g \in B_h \Rightarrow P(f) = P(g)$.

They then show that such conditions are unlikely to occur for randomly chosen subsets of functions.

In the absence of evidence to the contrary, it is hard to imagine how one can justify anything other than a uniform prior distribution over $\mathcal{F}$ without some emphatic statement somehow constraining the set of problems over which the intended algorithm is to be applied. For this reason we will focus on NFL-1 and NFL-2.

In general, many papers that propose new algorithms resort to comparing their algorithm with other algorithms on a test bed of problems under the belief they are providing a benchmark for comparisons. Hooker (1995) points-out that this often breeds algorithms that are so specialized that they have little value beyond the problems in the test-bed. Whitley and Watson (2004, p. 16) discuss what it usually means to compare algorithms on a test-bed of problems. They state "Given the NFL theorems, comparison is meaningless unless we prove (which virtually never happens) or assume (an assumption which is rarely made explicit) that the benchmarks used in a comparison are somehow representative of a particular subclass of problems." This paper addresses the "prove" part for several common problem classes of wide interest. Since NFL-1 addresses the class of all problems, we are left with NFL-2 as our main focal point since it offers tools for studying subclasses of problems.

Droste et al. (1999) frame a related goal. They pose several scenarios where optimization techniques might have different efficiencies. Ultimately they resort to empirical means by studying ten classes of functions with four optimization algorithms on relatively small search-space instances. They completely enumerate the different function assignments so are able to compute exact algorithm performance. Although no algorithm dominates, some perform better than others.

Igel and Toussaint (2003a) give a new theoretical result that parallels our goal. They characterize general constraints on problem structure that lead to violations of c.u.p. More formally, they consider a neighborhood relation that maps $\mathcal{X} \times \mathcal{X} \to \{0, 1\}$ where the value 1 signifies that two points are neighbors in some sense. A neighborhood is called nontrivial if there are distinct points that are neighbors and distinct points that are not neighbors. They then show:

Theorem 4. *A nontrivial neighborhood on $\mathcal{X}$ is not invariant under permutations of $\mathcal{X}$.*

They then explore some choices for neighborhood relations. In particular, they explore a constraint on steepness and on the number of local minima.

Some researchers believe that NFL research has obtained a level of completeness that leaves little else for exploration. For example, Wegener's (2004, p. 23) tutorial notes claim "The NFL theorem is fundamental and everything has been said on it" and "It is time to stop the discussion on NFL." Indeed, the four theorems reviewed above speak to this. However, whether specific problem classes fall under NFL is still unknown in large part. Shedding light on this point for some common problem classes is the focus of this paper.

In the next section we speak to the relationship between the NFL results (via NFL-2) and the structure of $\mathcal{X}$ and $F \subseteq \mathcal{F}$ rather than to the probability of choosing an $f \in F$ (as in Theorem NFL-3) or by concocting a neighborhood function (as needed in Theorem 4). Our intent is to determine necessary conditions for NFL to hold on some important classes of problems. Our goal is to be able to verify these conditions do not exist, thus obviating the NFL results.

3. Problems with Linear Objective Functions

In this section we look at the NFL-2 characterization when applied to $L \subseteq \mathcal{F}$ where L consists of only linear functions. We also restrict $\mathcal{Y}$ to a finite set of nonnegative real numbers. We represent L and $\mathcal{X}$ as a set of vectors where the usual inner product gives the objective function value. As usual, we restrict $\mathcal{X}$ to a finite set. Then

$$L \subseteq \{c\colon \forall x \in \mathcal{X},\ c'x \in \mathcal{Y}\} \subseteq \mathcal{F} = \mathcal{Y}^{\mathcal{X}}.$$

We wish to characterize what c.u.p. means for this class. We assume $s \equiv |\mathcal{X}| > 1$ thus avoiding unnecessary complications for the trivial case of $|\mathcal{X}| = 1$. Let matrix X consist of the s transposed vectors of $\mathcal{X}$ as its rows, so X is an $s \times n$ matrix. If L is c.u.p., then for any $f \in L$ and any $s \times s$ permutation matrix, P, there is a $g_{f,P} \in L$ such that $P'Xg_{f,P} = Xf$ (or equivalently $Xg_{f,P} = PXf$). For $f \in L$ let $[f] = \bigcup_P \{g_{f,P}\}$ be the equivalence class of f. Let $\mathcal{S} = \bigcup_P \{PXf\}$ and S be the $s \times |\mathcal{S}|$ matrix having the elements of $\mathcal{S}$ as its columns. Finally, let $z' = e'X$.

As we will see, circulant matrices play a role in understanding the implications of c.u.p. A circulant matrix, denoted circ(t) generated by an s-dimensional vector t, is

$$\operatorname{circ}(t) = \begin{bmatrix} t_1 & t_2 & \cdots & t_{s-1} & t_s \\ t_s & t_1 & \cdots & t_{s-2} & t_{s-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_3 & t_4 & \cdots & t_1 & t_2 \\ t_2 & t_3 & \cdots & t_s & t_1 \end{bmatrix}.$$

We say vector t is (strictly) monotone decreasing if $t_i \geq t_{i+1}$ ($t_i > t_{i+1}$). Let $t \in \mathcal{S}$ be a monotone decreasing vector. By construction there must be at least one such vector. Some interesting and useful properties of circulant matrices are summarized by Geller et al. (2002). We use the following.

Lemma 1 (Circulant Matrices). *Let* $V = \operatorname{circ}(t)$.

(a) *If for some integer* $d \geq 2$ *that divides* s *there are* s/d *consecutive constant blocks of* d *components of* t, *then* $\operatorname{rank}(V) \leq s+1-d$. *For example,* $t' = (2, 2, 1, 1)$ *has two blocks of size* 2.

If $t \geq 0$ *is monotone decreasing then*

(b) V *is singular iff for some integer* $d \geq 2$ *that divides* s *there are* s/d *consecutive constant blocks of* d *components of* t.

(c) *If* t *is strictly monotone decreasing, then* V *is nonsingular.*

(d) *If* s *is prime, then* V *is nonsingular if no* λ *gives* $t = \lambda e$.

We now derive some implications resulting from L being closed under permutation.

Theorem 5. *If* L *is c.u.p. and nonempty, then*

(a) *If for all* $f \in L$ *there is no integer* $d \geq 2$ *that divides* s *such that there are* s/d *consecutive constant blocks of* d *components in* Xf, *then* $\operatorname{rank}(X) = s$ *and* $s \leq n$.

(b) *If there is an* $f \in L$ *such that all components of* Xf *are distinct then* $\operatorname{rank}(X) = s$ *and* $s \leq n$.

(c) *For each* $f \in L$ *and any permutation matrix* P, $z'g_{f,p} = z'f$ *and* $z'f \geq 0$.

(d) *Let* $f \in L$. *Then* $z'f = 0$ *iff* $Xf = 0$.

(e) *For each* $f \in L$ *there is a* $\lambda_f \in \Re_{\geq 0}$ *such that* $X\sum_{g\in[f]} g = \lambda_f e$.

Proof. (a) Let $f \in L$ have Xf consisting of distinct values. Let P be such that $t = PXf$ is strictly monotone decreasing. $t \geq 0$ since $\mathcal{Y}$ has only nonnegative values. Then by Lemma 1c, $V = \operatorname{circ}(t)$ is nonsingular. However, since $Xg_{f,P} = PXf$ has a solution for each P, it must be the case that $\operatorname{rank}(X) = \operatorname{rank}(X{:}S)$ where $X{:}S$ is the matrix consisting of the columns of X and S. Then

$$s \geq \operatorname{rank}(X) = \operatorname{rank}(X{:}S) \geq \operatorname{rank}(S) \geq \operatorname{rank}(V) = s.$$

Finally, $s = \operatorname{rank}(X) \leq n$.

(b) Follows directly from (a).

(c) Since L is c.u.p., for any $f \in L$ and any permutation matrix, P, there is a $g_{f,P} \in L$ such that $P'Xg_{f,P} = Xf$. Thus $Xg_{f,P} = PXf$ and then $e'Xg_{f,P} = e'PXf = z'f$. Also, $Xf \geq 0$ since $\mathcal{Y}$ has only nonnegative values so $z'f \geq 0$.

(d) Follows directly from (c).

(e) We have

$$X\sum_P g_{f,P} = \Big(\sum_P P\Big)Xf = ee'Xf(s-1)!$$
$$= ez'f(s-1)! = (z'f(s-1)!)e.$$

Let $\lambda_f = z'f(s-1)!$. □

If L is c.u.p. and $s > n$ then all $f \in L$ will assign multiple points of $\mathcal{X}$ with the same objective value. Each equivalence class of L lies on a hyperplane.

In many combinatorial problems we have more structure. For example, requiring $\mathcal{X} \subseteq \mathbb{Z}^n_{\geq 0}$, $\mathcal{Y} = [l, \mu] \subseteq \mathbb{Z}_{>0}$ and $C = \{c \in \mathbb{Z}^n_{>0}\colon \forall x \in \mathcal{X},\ c'x \in \mathcal{Y}\}$ is not uncommon. Note that, implicitly, we are excluding the zero vector as a point of $\mathcal{X}$. Several results can be tightened.

Theorem 6. *If* C *is c.u.p. and nonempty, then*

(a) *Either* $\operatorname{rank}(X) = s$ *or* $Xf = \gamma_f e$, *where* $\gamma_f \in \mathbb{Z}_{>0}$ *for each* $f \in C$.

(b) *For each* $f \in C$ *and any permutation matrix* P, $z'g_{f,p} = z'f$ *and* $z'f > 0$.

Proof. (a) Suppose $\operatorname{rank}(X) < s$. Let $f \in C$. In order for $Xg_{f,P} = PXf$ to have a solution for each permutation matrix, it must be the case that $V = \operatorname{circ}(PXf)$ is singular for each permutation matrix P (or else $\operatorname{rank}(X, V) > \operatorname{rank}(X)$, a contradiction). Let $g \in [f]$ so that Xg is monotone decreasing. Thus (Lemma 1, Case b) there is some integer $d \geq 2$ that divides s such that there are s/d consecutive constant blocks of d components of Xg. Suppose $d < s$ for some f. The rows of X can thus be permuted without loss of generality so that

$$Xg = \begin{bmatrix} A \\ BA \end{bmatrix} \qquad g = \begin{bmatrix} t \\ q \end{bmatrix},$$

where A has row rank r equal to the rank of X. There must be at least one row in BA since $\operatorname{rank}(X) < s$ and it must be nonzero since $0 \notin \mathcal{X}$. Write this row as $b'A$ with the last component of q denoted by w. Suppose $t_i = w$ for some $d+1 \leq i \leq r$. Consider a permutation P that moves the right-hand side (RHS) of row i to row 1's RHS and vice-versa. Then

$$Ag_P = t + (w - t_1)1_1 + (t_1 - w)1_i$$
$$b'Ag_P = w$$

so

$$w = b'Ag_P = b't + (w - t_1)b_1 + (t_1 - w)b_i,$$

and then

$$(w - t_1)b_1 = (w - t_1)b_i$$

so $b_1 = b_i$. For any i such that $t_i > w$, let P be a permutation such that the RHS is permuted so that the last row's RHS becomes t_i and row i's becomes w. Then

$$Ag_{P_i} = t + (w - t_i)1_i$$
$$b'Ag_P = t_i$$

so

$$t_i = b'Ag_P = b't + (w - t_i)b'1_i = w + (w - t_i)b_i$$

and

$$(w - t_i)b_i = t_i - w.$$

Thus $t_i > w \Rightarrow b_i < 0$ and $b_i < 0$. Thus $b < 0$. However, since $b'A \geq 1'_j$ for some j, we reach a contradiction. Thus if $\text{rank}(X) < s$ then $d = s$ meaning that $Xf = \gamma_f e$ for some $\gamma_f \in \mathbb{Z}_{>0}$.

(b) From Lemma 1 we have that $z'f \geq 0$. If $z'f = 0$ then $e'X \geq 0$ and $f > 0$ imply $X = 0$ which contradicts $|\mathscr{X}| > 1$. □

For example, if

$$\mathscr{X} = \left\{ \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}, \qquad \mathscr{Y} = \{2, 3, 4\},$$

then $\text{rank}(X) = 2 < s = 3$ and $C = \{\binom{1}{1}, \binom{2}{2}\}$.

For combinatorial problems fitting the structure explored above, the NFL results apply only if either $\text{rank}(X) = s \leq n$ or all feasible points have the same objective value. In the former case where $\text{rank}(X) = s$ we have search spaces that have no more feasible points than problem variables, a relatively rare situation for most problems we have encountered. In other words, these problems have search spaces that grow in size no faster than n, the number of variables. These search spaces might be easily enumerated. In the latter case, all search space points have the same objective value, an uninteresting problem.

In the following, we look at several types of combinatorial problems and determine characteristics that are required for the NFL results to hold.

4. Applications

In this section we apply the results of the last section to some particular problem classes. We start with partitioning problems where we use the constraint set structure to glean insights into problems that are not c.u.p. This is followed by a look at the binary knapsack problem where both the constraint-set structure and some knowledge of part of the solution space give even more insight into when a problem of this class is c.u.p. Last we look at the traveling-salesman problem.

4.1. Partitioning Problems

Suppose $\mathscr{X} = \{x\colon Ax = b,\ x \in \mathbb{Z}^n_{\geq 0}\}$, where A (an $m \times n$ matrix) and b are both integer-valued. We assume $0 \notin \mathscr{X}$. A typical problem that fits this model is the set-partitioning problem where $b = e$ and $A \in B^{m,n}$ (thus providing the motivation for calling this class of problems "partitioning problems"). Then we have $AX' = be'$. If $\text{rank}(X) < s$ and C is c.u.p., then by Theorem 6 all feasible solutions have the same objective value—so all are optimal—which gives an uninteresting problem.

Consider the case where $\text{rank}(X) = s$. Then if $n = s$

$$A = be'(X')^+ = be'X(X'X)^{-1} = bz'(X'X)^{-1},$$

where $(X')^+$ is the usual Moore-Penrose generalized inverse (see, for example, Schott 1997). Hence the rows of A will either be zero-valued (if that row of b is zero) or be multiples of each other. For the actual set-partitioning problem (where $b = e$), the NFL results would apply only when all rows of A are the same, an uninteresting problem class.

If $\text{rank}(X) = s$ and $n > s$, then c.u.p. might apply. For example, for

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

we have

$$X' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and if $\mathscr{Y} = \{2, 3\}$ we have

$$C = \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} \right\}.$$

Then

$$\{Xf\colon f \in C\} = \left\{ \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \right\},$$

and C is c.u.p.

Summarizing, for partitioning problems, if the NFL results pertain, then either

(a) All feasible solutions have the same objective value, so all are optimal; or

(b) $\text{rank}(X) = s = n$ and each row of A will either be zero-valued (if that row of b is zero) or multiples of each other; or

(c) $\text{rank}(X) = s < n$.

Cases (a) and (b) result in trivial problems. Case (c) may not be completely trivial, but enumeration might be a reasonable search choice since there are fewer search points than dimensions.

4.2. Binary Knapsack Problems

In this section we look at the well-known binary knapsack problem in the context of the NFL theorems. Let $\mathscr{X} = \{x \in B^n\colon\ a'x \le b\}\backslash\{0\}$, where $0 < a_1 \le \cdots \le a_n \le b$, $n > 1$ and $\mathscr{Y} = [l, \mu] \subseteq \mathbb{Z}_{>0}$. As is usual, we consider maximizing a linear objective having all positive coefficients. We remove the zero-vector from the search space since it will never be optimal and its inclusion unnecessarily complicates the presentation. The set of all objective vectors, given $\mathscr{Y}$, is $K \subseteq \mathscr{F} = \mathscr{Y}^{\mathscr{X}}$ defined by

$$K \equiv \{c \in \mathbb{Z}^n_{>0}\colon\ l \le c'x \le \mu,\ \forall x \in \mathscr{X}\}.$$

As is readily observed, this is a special case of the combinatorial problem discussed in Section 3, and, as expected, stronger statements can be made as shown in the next result.

Theorem 7. *Suppose K is nonempty. K is c.u.p. iff $|\mathscr{X}| = n$.*

Proof. ($\Rightarrow$) We are given that K is c.u.p. and nonempty. Since every unit vector is in $\mathscr{X}$, $|\mathscr{X}| \ge n$. From Theorem 6(a) either $\text{rank}(X) = s$ or $Xf = \gamma_f e$, where $\gamma_f \in \mathbb{Z}_{>0}$ for each $f \in K$. If $\text{rank}(X) = s$ then $n \ge \text{rank}(X) = s = |\mathscr{X}| \ge n$. Suppose $\text{rank}(X) < s$. Let $f \in K$. Then $Xf = \gamma_f e$ with $\gamma_f \ge l > 0$. However, if $|\mathscr{X}| > n$ at least one $x \in \mathscr{X}$ has two or more nonzero components so $x'f \ge 2\gamma_f$ which is a contradiction. Thus $|\mathscr{X}| = n$.

($\Leftarrow$) Suppose $|\mathscr{X}| = n$. Then $\mathscr{X} = \{1_1, \ldots, 1_n\}$ and K is trivially shown to be c.u.p. Note, this case occurs if $a = e$ and $b = 1$. □

Note that $|\mathscr{X}| = n$ if and only if $a_1 + a_2 > b$. To see this, since $0 < a_1 \le \cdots \le a_n \le b$, we have that all the unit vectors are feasible, i.e., $\{1_1, \ldots, 1_n\} \subseteq \mathscr{X}$. However, if $a_1 + a_2 \le b$ then $1_1 + 1_2 \in \mathscr{X}$ which would give $|\mathscr{X}| \ge n + 1$. This condition yields trivial knapsack problems. Thus, no nontrivial binary knapsack problem is c.u.p.

4.3. Symmetric Traveling-Salesman Problem

Consider a complete graph with vertices V and $n = |V|(|V| - 1)/2$ edges. A tour is a cycle that contains all vertices. For $x \in B^n$ we set a component to 1 if the corresponding edge is a part of the tour, and set it at 0 otherwise. Let $\mathscr{X}$ be the set of all such incidence vectors for tours. For the symmetric traveling-salesman problem, a tour length is $c'x$ where component i of c is the cost to traverse edge i. Many integer programming formulations have been provided for representing the symmetric traveling-salesman problem (see Langevin et al. 1990) but one given by Dantzig et al. (1954) provides a formulation that remains stronger than others proposed since then (i.e., its linear relaxation is contained in the linear relaxations of others).

Let $\mathscr{Y} = [l, \mu] \subseteq \mathbb{Z}_{>0}$ and $C = \{c \in \mathbb{Z}^n_{>0}\colon\ \forall x \in \mathscr{X},\ c'x \in \mathscr{Y}\}$. Maurras (1975) has shown that the convex hull of $\mathscr{X}$ has dimension $|V|(|V| - 3)/2$. This implies that X has rank $|V|(|V| - 3)/2$. From Theorem 6, the symmetric traveling-salesman problem is c.u.p. only if either $|V|(|V| - 3)/2 = \text{rank}(X) = |\mathscr{X}|$ or all tours have the same length (a trivial problem). In the former case, $|\mathscr{X}| = (|V| - 1)!/2$ so $|V|(|V| - 3)/2 = (|V| - 1)!/2$ is never satisfied. Hence the only case where the symmetric traveling-salesman problem is c.u.p. is the trivial case where all tours have the same length.

4.4. Discussion

The problems of Section 4 are known to be NP-hard when viewed as optimization problems (Garey and Johnson 1979) or NP-complete when viewed as decision problems. Whitley and Watson (2004, p. 10) note "No Free Lunch has not been proven to hold over the set of problems in the complexity class NP" and that to do so would "prove that P ≠ NP." Further "The existence of ratio bounds for certain NP-complete problems also shows that NFL theorems do not hold for some specific NP-complete problems." We add to this body of evidence showing that only trivial subclasses of partitioning problems, binary knapsack, and traveling-salesman problems fall under NFL. Paradoxically, NFL holds on subclasses of these problems, but only ones that are utterly trivial or easy to enumerate.

5. Summary and Future Research

Theorems 1–3 provide necessary and sufficient conditions for NFL results. When NFL applies, all algorithms are equal, on average. The conditions, especially those in Theorem 2, are not easily tested without enumeration. This work gives necessary conditions that can be tested on important problems that arise often in optimization. Theorem 5 provides general conditions that are sharpened in Theorem 6 to usual conditions found in combinatorial problems with linear objective functions. We apply Theorem 6 to three combinatorial problem classes and are able to sharpen the conditions further based upon additional problem structure inherent in these problems.

For example, we find that the NFL results do not apply to any reasonable nontrivial binary knapsack problem, partitioning problem, or traveling-salesman problem.

Theorem 4 by Igel and Toussaint (2003a) and our Theorems 5 and 6 are useful for deciding whether NFL results do not hold for problem classes. Our approach appears easier (at least to us) to apply to traditional optimization problems while the approach by Igel and Toussaint is useful in other settings. More problem classes need to be systematically examined to decide whether NFL applies.

When NFL does not apply, there still remains the issue of deciding whether one algorithm dominates another. Just testing one's algorithm on a common test-bed is likely insufficient as Hooker (1995) argues. The NFL results provide a statement about algorithm equivalence. Less well structured are ways to declare that one algorithm dominates another on a class of problems. Wolpert and Macready (1997) give some benchmark performance measures that allow a "programmatic (rather than ad hoc) assessment of the efficacy of any individual optimization algorithm and principled comparisons between algorithms" but this depends on knowing or computing values over all objective functions for the problem class under study. It is not clear how this helps in general. Future research on this issue is needed.

References

Culberson, J. C. 1998. On the futility of blind search: An algorithmic view of "no free lunch." *Evol. Comput.* **6**(2) 109–127.

Dantzig, G., D. Fulkerson, S. Johnson. 1954. Solution of a large-scale traveling-salesman problem. *Oper. Res.* **2** 393–410.

Droste, S., T. Jansen, I. Wegener. 1999. Perhaps not a free lunch but at least a free appetizer. *Genetic and Evol. Comput. Conf. GECCO-99*, Morgan Kaufmann, San Francisco, CA, 833–839.

Garey, M. R., D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA.

Geller, D., I. Kra, S. Popescu, S. Simanca. 2002. On circulant matrices. State University of New York at Stony Brook, Stony Brook, NY, http://www.math.sunysb.edu/~sorin/eprints/circulant.pdf.

Glover, F. 1989. Tabu search—part i. *ORSA J. Comput.* **1** 190–260.

Glover, F. 1990. Tabu search—part ii. *ORSA J. Comput.* **2** 4–32.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Hooker, J. N. 1995. Testing heuristics: We have it all wrong. *J. Heuristics* **1** 33–42.

Igel, C., M. Toussaint. 2003a. On classes of functions for which no free lunch results hold. *Inform. Processing Lett.* **86** 317–321.

Igel, C., M. Toussaint. 2003b. Recent results on no-free-lunch theorems for optimization. Institut für Neuroinformatik and Ruhr-Universität Bochum, Germany.

Kirkpatrick, S., D. C. Gelatt, M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.

Langevin, A., F. Soumis, J. Desrosiers. 1990. Classification of travelling salesman problem formulations. *Oper. Res. Lett.* **9** 127–132.

Maurras, J. F. 1975. Some results on the convex hull of Hamiltonian cycles of symmetric complete graphs. B. Roy, ed. *Combinatorial Programming: Methods and Applications*. Reidel Publishing Company, Dordrecht, Holland, 179–190.

Radcliffe, N. J., P. D. Surry. 1995. Fundamental limitations on search algorithms: Evolutionary computing in perspective. J. van Leeuwen, ed. *Lecture Notes in Computer Science*, Vol. 1000. Springer-Verlag, Berlin, Germany.

Schott, J. R. 1997. *Matrix Analysis for Statistics*. John Wiley and Sons, New York.

Schuhmacher, C., M. D. Vose, L. D. Whitley. 2001. The no free lunch and description length. L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, E. Burke, eds. *Genetic and Evol. Comput. Conf.* (*GECCO 2001*), Morgan Kaufmann, San Francisco, CA, 565–570.

Wegener, I. 2004. Tutorial: Computational complexity and EC. *Genetic and Evol. Comput. Conf.* (*GECCO 2004*), Morgan Kaufmann, San Francisco, CA.

Whitley, D., J. Watson. 2004. A "no free lunch" tutorial. Department of Computer-Science, Colorado State University, Fort Collins, CO.

Wolpert, D. H., W. G. Macready. 1995. No free lunch theorems for search. Working Paper sfi-tr-95-02-010, Santa Fe Institute, Santa Fe, NM.

Wolpert, D. H., W. G. Macready. 1997. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1** 67–82.