Free Lunches on the Discrete Lipschitz Class

Pei Jiang
Ying-ping Chen

NCLab Report No. NCL-TR-2009001
January 2009

Natural Computing Laboratory (NCLab)
Department of Computer Science
National Chiao Tung University
329 Engineering Building C
1001 Ta Hsueh Road
HsinChu City 300, TAIWAN
http://nclab.tw/

Free Lunches on the Discrete Lipschitz Class

Pei Jiang and Ying-ping Chen
Department of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
{pjiang, ypchen}@nclab.tw

January 14, 2009

Abstract

The No-Free-Lunch theorem states that there does not exist a genuine general-purpose optimizer because all algorithms have the identical performance on average over all functions. However, such a result does not imply that search heuristics or optimization algorithms are futile if we are more cautious with the applicability of these methods and the search space. In this paper, within the No-Free-Lunch framework, we firstly introduce the discrete Lipschitz class by transferring the Lipschitz functions, i.e., functions with bounded slope, as a measure to fulfill the notion of continuity in discrete functions. We then investigate the properties of the discrete Lipschitz class, generalize an algorithm called subthreshold-seeker for optimization, and show that the generalized subthreshold-seeker outperforms random search on this class. Finally, we propose a tractable sampling-test scheme to empirically demonstrate the superiority of the generalized subthreshold-seeker under practical configurations. This study concludes that there exists algorithms outperforming random search on the discrete Lipschitz class in both theoretical and practical aspects and indicates that the effectiveness of search heuristics may not be universal but still general in some broad sense.

1 Introduction

Back to 1980s, in the field of evolutionary computation, there is a belief that while evolutionary algorithms may not perform as well as the specialized algorithm for a specific optimization problem, they are more widely applicable and have superior overall performance. However, in 1995, Wolpert and Macready proposed the No-Free-Lunch (NFL) theorem [1, 2] which formally states that every algorithm performs equally well on average over all functions. A direct implication of NFL is that, given any performance measure, the better performance of an algorithm on some problems always accompanies with the worse performance on others. The number of problems on which the algorithm performs well is exactly the number of those on which it does not perform well. In other words, there is no such thing as robustness under the NFL framework, or all algorithms are considered robust. Therefore, it is no surprise that the proposition of the NFL theorem causes a great deal of controversy in the optimization and heuristic search community [3], as the NFL theorem sets a limitation on the pursuit of general-purpose optimizers.

Indeed, the implications of the NFL theorem seem to disagree with empirical observations of the effectiveness of optimization algorithms and search heuristics, since general-purpose optimizers such as gradient-based methods, simulated-annealing, and biologically inspired algorithms do have their share of significance in real-world applications. On the other hand, the NFL theorem is a mathematical theorem, which means that it is absolutely true when all the hypotheses are given. As a consequence, previous studies intending to address the incoherence between

theoretical results and empirical observations are mostly aiming at the hypotheses of the NFL theorem, especially the notion of "all functions". Droste et al. [4, 5] systematically described a few scenarios of functions and claimed that the scope of the NFL theorem is too enormous to be realistic. Streeter [6] proved that the NFL theorem does not hold over the problems with sufficiently bounded description length. Beyond identifying a subset of problems to which the NFL result can not be applied, Christensen and Oppacher [7] started with a more direct standpoint by proposing the submedian-seeker and demonstrated such an algorithm can outperform random search on certain types of functions. Thereafter, Whitley and Rowe [8] simplified and extended Christensen and Oppacher's work and showed that a more generic subthreshold-seeker can outperform random search on uniformly sampled polynomials in the sense of the number of subthreshold points visited in a given time span.

In the aforementioned studies, the topics may be different, but a common goal is shared – addressing the issue of how general optimization algorithms and search heuristics can be. This study serves the same purpose. Borrowing the notion of Lipschitz functions in real analysis, we introduce the discrete Lipschitz class as an attempt to capture the continuity of a discrete search space. The property of similarities in objective values within a neighborhood is possessed by many real-world problems, and such a problem structure does facilitate the search process. In particular, we prove that a generalized subthreshold-seeker indeed outperforms random search on the discrete Lipschitz class in theory as well as demonstrate the theoretical result can be carried over into practice by proposing a sampling-test scheme and conducting numerical experiments with comparisons.

The remainder of this paper is organized as follows: In section 2, we briefly review the NFL framework to establish and unify the terminology and definitions as preliminaries. Section 3 introduces the discrete Lipschitz class and describes the relationship between the class and the theorem with a focus on the condition under which the NFL theorem holds over the discrete Lipschitz class. In section 4, we generalize the subthreshold-seeker and discuss its performance on the discrete Lipschitz class in comparison with random search. In section 5, we propose a sampling-test scheme as an alternative way to examine the effectiveness of optimizers in practice, followed by the conclusions in section 6.

2 A brief review of NFL

The No-Free-Lunch (NFL) theorem, in short, states that all algorithms have the same overall performance. As plain as this statement may seem, there are several aspects to be clarified. Firstly, "algorithms" in the realm of NFL are restricted to the scope of "non-repeating black-box algorithms". The term "black-box algorithm", referred to as "blind search" in some literatures, is used to describe the class of algorithms only employing the result of function evaluations as information. The requirement of non-repeating ensures that the search process can be viewed as a permutation of the elements in search space, and revisiting points merely increases the running time without rendering any assistance for identifying the optimum. In fact, when the performance is averaged over all functions, based on NFL, the best an algorithm can do is try not to re-sample.

The concept of "all functions" is another intriguing point for its inherent vagueness. One of the fundamental results in computability is that the set of problems is uncountably infinite. If we consider the the collection of feasible regions of optimization problems as a language, we can easily use the diagonalization method to show that such a language is not recursive. The NFL framework takes a more practical stand here and bypasses this difficulty by considering those functions defined on a finite domain with a finite codomain.

Within the NFL framework, the concepts of optimization problems and search algorithms can be formalized in the following definition:

Definition 1 (NFL framework)**.** *Given two finite sets $\mathcal{X}$ and $\mathcal{Y}$,*

1. *The set of all functions $\mathcal{F}_{\mathcal{X},\mathcal{Y}}$, with respect to $\mathcal{X}$ and $\mathcal{Y}$, is defined as $\mathcal{F}_{\mathcal{X},\mathcal{Y}} := \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$.*

2. *A* trace *of length m is a sequence $T_m := ((x_i, y_i))_1^m = ((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ with distinct x_i's. "$x \in T_m$" denotes that $x = x_i$ for some $i \in \{1, 2, \ldots, m\}$. Let T_0 be the empty sequence and $\mathcal{T}^\ell$ be the set containing all the traces of a length smaller than or equal to ℓ.*

3. *Let A_T, where $T \in \mathcal{T}^{|\mathcal{X}|-1}$, be a random variable over $\mathcal{X}$ satisfying that $Prob\{A_T = x\} = 0$ for all $x \in T$. An* algorithm *A is a collection of such random variables, i.e., $A = \{A_T \mid T \in \mathcal{T}^{|\mathcal{X}|-1}\}$.*

4. *The* search process *of A on f, $S(A, f)$, is the stochastic process $(X_i, Y_i := f(X_i))$ over $\mathcal{X} \times \mathcal{Y}$ defined by $X_1 \sim A_{T_0}$ and $X_{k+1} \sim A_{((X_i,Y_i))_1^k}$. Let $S(A, f, k) := ((X_i, Y_i))_1^k$, and $S_y(A, f, k) := (Y_i)_1^k$ is called the* performance vector*.*

5. *Let $\mathcal{V} := \bigcup_{i=1}^{|\mathcal{X}|} \mathcal{Y}^i$ be the set containing all possible performance vectors. A* performance measure *is any function mapping $\mathcal{V}$ to $\mathbb{R}$.*

The terminology in Definition 1 mostly follows those adopted in [2] and [9] with a few slight modifications applied to avoid the situation that an algorithm is undefinable on a complete trace and to make search processes able to be expressed in a naturally stochastic way.

Example 2 (Random search in NFL)**.** *Let R_{T_m} be a random variable that $Prob\{R_{T_m} = x\} = 1/(|\mathcal{X}| - m)$ for all $x \notin T_m$. In the NFL framework, random search can be accordingly defined as $RS := \{R_T \mid T \in \mathcal{T}^{|\mathcal{X}|-1}\}$.*

Now, the NFL theorem can be given as Theorem 3. The complete proof can be found in the original NFL papers [1, 2].

Theorem 3 (NFL theorem)**.** *If $v \in \mathcal{V}$ is a performance vector with length ℓ, $\sum_f Prob\{S_y(A, f, \ell) = v\} = c$, where c is a constant independent of A.*

3 Discrete Lipschitz class

3.1 Definition of the Discrete Lipschitz class

In real analysis, Lipschitz functions refer to the functions with bounded slope. Given a set $\mathcal{C} \subseteq \mathbb{R}$, $f : \mathcal{C} \to \mathbb{R}$ is a *Lipschitz function* if there exists a constant $K > 0$ such that $|f(a) - f(b)| \leq K|a - b|$ for all $a, b \in \mathcal{C}$. The Lipschitz condition is a stronger condition than normal continuity, because any Lipschitz function is uniformly continuous. On the other hand, the functions that are not everywhere differentiable may still be Lipschitz, e.g., $f(x) = |x|$. On a closed interval, the Lipschitz class lies between continuous functions and the functions having continuous derivatives [10].

For the discrete space, there is no such thing as continuity. However, if there is some sort of distance defined in some discrete space, the Lipschitz condition can still be applied, and therefore a natural way to simulate continuity in the discrete space can be obtained. In combinatorics, the spatial structures are typically formed via graph theory. If we view the vertex set as the search space and the edge set as the specification of the geometry, the Lipschitz condition can be transferred here by restricting the difference of objective values between any two adjacent vertexes. The merit of such definition is that we do not put any constraints on the global

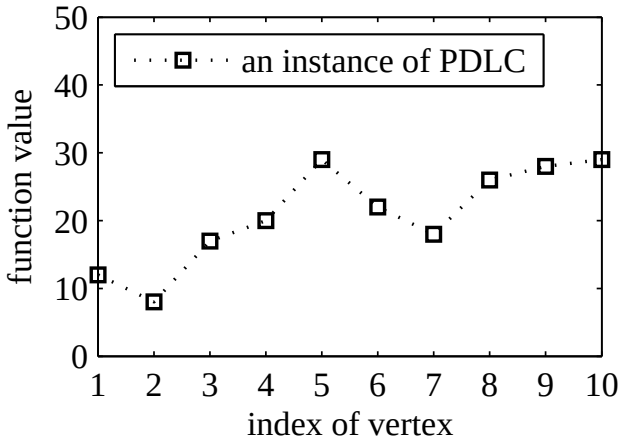

Figure 1: A function instance of PDLC with 10 vertexes and $K = 10$.

structure directly such as to demand the functions to be polynomial or the description length to be bounded. Instead, we only expect some similarities of the objective values within a neighborhood in the search space.

Since we will focus on the discrete Lipschitz class in the remainder of this paper, the domain $\mathcal{X}$ is always the vertex set $V(G)$ of a graph G, representing the spatial structure. Hence, the two notations $\mathcal{X}$ and $V(G)$ are used exchangeably. The discrete Lipschitz class (DLC) can now be introduced.

Definition 4 (Discrete Lipschitz class, DLC)**.** *Given a connected graph G and a finite set $\mathcal{Y} \subset \mathbb{R}$, the corresponding discrete Lipschitz class with Lipschitz constant K is defined as*

$$\mathcal{L}(G, \mathcal{Y}, K) := \{f : V(G) \to \mathcal{Y} \mid \forall \overline{v_1 v_2} \in E(G), |f(v_1) - f(v_2)| \leq K\} \ .$$

Such a definition provides a means to represent the intrinsically real-parameter optimization problems through discretization (for practical computing devices). For instance, if a cube $\mathcal{C} \subset \mathbb{R}^n$ is discretized uniformly into a set of grid points, $V(G) = \{x_1, x_2, \ldots, x_M\}^n \subset \mathbb{R}^n$ with $x_{i+1} - x_i = u > 0$, we can let $E(G) = \{\overline{v_i v_j} \mid v_i, v_j \in V(G) \text{ and } \|v_i - v_j\|_1 = u\}$. $\mathcal{L}(G, \mathcal{Y}, K)$ then forms a class containing all functions, defined on $\mathcal{C}$ with the absolute values of partial derivatives upper bounded by K/u, discretized over $V(G)$. Furthermore, since $\mathcal{Y}$ is bounded, this class contains all functions mapping $V(G)$ to $\mathcal{Y}$ with sufficient large K (e.g., $K = \max \mathcal{Y} - \min \mathcal{Y}$).

The simplest case of DLC is the class of functions defined on $\mathbb{R}$, in which the graph representing the spatial structure is a simple path. Figure 1 gives an illustrative example of such functions.

Definition 5 (Pathwise discrete Lipschitz class, PDLC)**.** *Given a finite set $\mathcal{Y} \subset \mathbb{R}$ and a simple path $G = \overline{v_1 v_2 \ldots v_n}$, the pathwise discrete Lipschitz class with Lipschitz constant K is defined as $\mathcal{L}(G, \mathcal{Y}, K)$.*

Throughout this paper, the property of $\mathcal{Y}$ of interest is the ordering, without loss of generality, $\mathcal{Y}$ is assumed to be a subset of $\mathbb{N}$ of the form $\{0, 1, \ldots, m\}$ unless specified otherwise. $deg(v)$ and $N(v)$ are used to denote the degree and the neighborhood of a vertex v, respectively.

3.2 DLC and NFL

In this section, we will investigate DLC within the NFL framework and derive a condition under which the the NFL theorem holds. In order to determine whether the NFL theorem holds

over a problem class, Schumacher et al. [9] provided a criterion for the NFL theorem based on permutation closure.

Definition 6 (Permutation closure)**.** *If π is a permutation on $\mathcal{X}$, define f_π as $f_\pi(x) := f(\pi(x))$ for all $x \in \mathcal{X}$. $F \subseteq \mathcal{F}_{\mathcal{X},\mathcal{Y}}$ is closed under permutation if for all $f \in F$ and for every permutation π on $\mathcal{X}$, $f_\pi \in F$.*

Lemma 7. *The NFL theorem holds over F if and only if F is closed under permutation.*

Although in [9], this criterion is proposed for deterministic algorithms, since a randomized algorithm is simply a mixed strategy, i.e., a distribution over all possible deterministic strategies [11, 5], this criterion still holds for randomized algorithms in the sense of expectation. Utilizing Lemma 7, a simple criterion for whether or not the NFL result can be applied to a DLC can be obtained.

Theorem 8 (Criterion for NFL on DLC)**.** *Let $\mathcal{L}(G, \mathcal{Y} = \{0, 1, \ldots, m\}, K)$ with $m > K$ be a DLC. NFL holds over $\mathcal{L}(G, \mathcal{Y}, K)$ if and only if G is complete.*

Proof. By Lemma 7, it is sufficient to show that $\mathcal{L}(G, \mathcal{Y}, K)$ is closed under permutation if and only if G is complete.

- If G is complete, for every $f \in \mathcal{L}(G, \mathcal{Y}, K)$, we have $|f(v_i) - f(v_j)| \leq K$ for all v_i and $v_j \in V(G)$. For any permutation π on $\mathcal{X}$ and for all v_i and $v_j \in V(G)$, $|f_\pi(v_i) - f_\pi(v_j)| = |f(\pi(v_i)) - f(\pi(v_j))| \leq K$. Therefore, $f_\pi \in \mathcal{L}(G, \mathcal{Y}, K)$.
- If $\mathcal{L}(G, \mathcal{Y}, K)$ is closed under permutation, suppose for contradiction that G is not complete. The incompleteness and connectivity of G imply that there exist v_i and $v_j \in V(G)$ with $\overline{v_i v_j} \notin E(G)$. Select $v_k \in N(v_i)$, where $N(v_i)$ is the neighborhood of v_i. Obviously, $v_k \neq v_j$. Consider the function $f \in \mathcal{L}(G, \mathcal{Y}, K)$:

$$f(v) = \begin{cases} 0 & \text{if } v = v_i \ ; \\ K+1 & \text{if } v = v_j \ ; \\ K & \text{otherwise.} \end{cases}$$

  and the permutation π:

$$\pi(v) = \begin{cases} v_k & \text{if } v = v_j \ ; \\ v_j & \text{if } v = v_k \ ; \\ v & \text{otherwise.} \end{cases}$$

  $|f_\pi(v_k) - f_\pi(v_i)| = |f(\pi(v_k)) - f(\pi(v_i))| = |f(v_j) - f(v_i)| = K + 1$, so $f_\pi \notin \mathcal{L}(G, \mathcal{Y}, K)$, a contradiction.

□

The completeness of a graph implies that the entire search space is in the same neighborhood. However, such a case is rare in real-world applications, because the degree can be viewed as an indication of dimensions, and the size of search space typically surpasses the number of dimensions significantly. For example, for discretized real-parameter optimization problems, the cardinality of the domain is usually notably larger than the number of dimensions, and hence the corresponding graphs are not complete in most cases. Taking PDLC as an example, when $m > K$, the NFL theorem sustains over a PDLC only if there are merely two vertexes in the problem.

4 DLC and subthreshold-seeker

The subthreshold-seeker (STS), introduced by Whitley and Rowe [8] and proved to outperform random search on uniformly sampled polynomials of one variable, is a metaheuristic that employs the threshold as a switch of local search. In essence, it is a selective local search method as it conducts local search if a given condition is satisfied. In this section, a generalization of subthreshold-seeker is firstly presented, and we will demonstrate that the generalized subthreshold-seeker can outperform random search on DLC.

4.1 Generalized subthreshold-seeker

In Whitley and Rowe's work, the subthreshold-seeker is an optimization algorithm aiming at functions with a one-dimensional domain, i.e., functions defined on a subset $\mathcal{C} \subseteq \mathbb{R}$. The subthreshold-seeker will successively select a point from the search space uniformly at random (u.a.r.) until a subthreshold point is encountered. Once encountering a subthreshold point, the subthreshold-seeker will search through the quasi-basin where that subthreshold point resides. In Whitley and Rowe's definition, a quasi-basin is a set of contiguous points with objective values below the threshold. In other words, the threshold is used to determine whether the subthreshold-seeker enters the local search phase, and the subthreshold-seeker can be viewed as an optimizer with an exhaustively local search operator.

According to this point of view, we generalize the subthreshold-seeker to the extent that it is applicable to any function of which the domain possesses a neighborhood structure as in Algorithm 1.

Algorithm 1 (Generalized subthreshold-seeker)**.**
procedure SUBTHRESHOLD-SEEKER*($\mathcal{X}$, $\mathcal{Y}$, $N : \mathcal{X} \to 2^{\mathcal{X}}$, $f : \mathcal{X} \to \mathcal{Y}$)*
  while *the stopping criterion is not satisfied* ***do***
    if *Queue is not empty* ***then***
      $x \leftarrow$ *Queue.pop();*
    else
      Select x from $\mathcal{X}$ u.a.r.
    end if
    if $f(x) \leq \theta$ ***then***
      Queue.push($N(x)$)
    end if
  end while
end procedure

Following the NFL framework, the parts of selecting and pushing are both restricted to unvisited points. Such a task can be achieved by a bookkeeping manner. Since the performance of an algorithm is judged by the performance vector, all overheads other than function evaluations will not count under the NFL framework.

The only control parameter of the subthreshold-seeker is the threshold. The elegance of the subthreshold-seeker is that it comprises the two fundamental operations of search heuristics, local search and global restart, and yet still stays in a simple form.

4.2 Subthreshold-seeker on DLC

Christensen and Oppacher [7] defined the performance measure as the number of submedian points visited by an algorithm, and Whitley and Rowe [8] generalized this notion to any threshold less than or equal to the median. That is, given a predefined stopping time L and $\alpha \in (0, 1/2]$,

the performance measure is the number of points visited in the first L function evaluations with the top $\alpha|\mathcal{X}|$ values in the objective space.

This performance measure may seem odd at the first glance, for typically the performance of an optimizer is measured in terms of the time in which the optimum is located. However, even focusing on functions as simple as unimodal functions that are monotone with respect to the distance from the optimum, the time complexity analysis is still a difficult task. For instance, to the best of our limited knowledge, the time complexity of (1+1)-ES [12] on such functions has not been analyzed until recently [13]. Hence, it seems unlikely to analyze the runtime of an algorithm that is more sophisticated than random search over a broad class of problems. Furthermore, as mentioned in section 2, within the NFL framework, the performance measure can be any function defined on the set containing all performance vectors, and roughly speaking, with more subthreshold points visited, it is more likely to identify a point with a satisfiable objective value. Therefore, Whitley and Rowe's notion appears in between theoretically analyzable and practically meaningful.

For any function f, we define $\beta_\alpha(f)$ be the maximum objective value below the performance threshold, i.e.,

$$\beta_\alpha(f) := \max\left\{ y \in \mathcal{Y} \;\middle|\; \sum_{i=0}^{y} |\{x \in \mathcal{X} \mid f(x) = i\}| \leq \alpha|\mathcal{X}| \right\} .$$

If the set following the "max" notation is empty, then $\beta_\alpha(f)$ is defined to be $-\infty$. Let $\Psi_{\alpha,f}(v)$ be a performance measure that maps a performance vector v to the number of components of v below performance threshold, i.e., $\Psi_{\alpha,f}\left((v_1, v_2, \ldots, v_L)\right) = |\{v_i \mid v_i \leq \beta_\alpha(f)\}|$. It is noteworthy that the performance threshold should be distinguished from the algorithmic threshold. The latter should be regarded as a control parameter of the algorithm and hence is not related to the performance measure.

Whitley and Rowe showed that if f is a uniformly sampled polynomials of one variable, and $\beta_\alpha(f)$ is known in advance, setting $\theta = \beta_\alpha(f)$, under certain conditions the subthreshold-seeker outperforms random search on f. In this study, we will show that the subthreshold-seeker with θ within some range of codomain, rather than a specific value, will outperform random search in the sense that for all functions in the DLC, the expected number of points below the performance threshold visited by the subthreshold-seeker is greater than or equal to that by random search, and there does exist a function such that the inequality is strict.

Theorem 9 (Equal or better performance of STS on DLC)**.** *Let $\mathcal{L}(G, \mathcal{Y} = \{0, 1, \ldots, m\}, K)$ with $m > K$ be a DLC. For all $f \in \mathcal{L}(G, \mathcal{Y}, K)$ if the algorithmic threshold θ of a subthreshold-seeker satisfies $\theta \leq \beta_\alpha(f) - K$, then $E[\Psi_{\alpha,f}(S_y(STS, f, L))] \geq E[\Psi_{\alpha,f}(S_y(RS, f, L))]$ for all L with $1 \leq L \leq |\mathcal{X}|$.*

Proof. Let f be any function belonging to $\mathcal{L}(G, \mathcal{Y}, K)$. Suppose $S(STS, f, L) = ((X_{si}, Y_{si}))_{i=1}^{L}$ and $S(RS, f, L) = ((X_{ri}, Y_{ri}))_{i=1}^{L}$. Define the indicator variable I_{si} as $I_{si} = 1$ when $Y_{si} \leq \beta_\alpha(f)$ and $I_{si} = 0$ otherwise, and I_{ri} is defined in a similar way for random search. We can obtain that $\Psi_{\alpha,f}(S_y(STS, f, L)) = \sum_{i=1}^{L} I_{si}$ and $\Psi_{\alpha,f}(S_y(RS, f, L)) = \sum_{i=1}^{L} I_{ri}$.

We prove the theorem by induction on L. Let $U := |\{x \in V(G) \mid f(x) \leq \beta_\alpha(f)\}$ be the total number of points below the performance threshold. When $L = 1$, since both strategies select a point u.a.r. from $\mathcal{X}$ in the first move, clearly $E[I_{s1}] = U/|\mathcal{X}| = E[I_{r1}]$. Suppose

$E[\sum_{i=1}^{L} I_{si}] \geq E[\sum_{i=1}^{L} I_{ri}]$ for $1 \leq L < |\mathcal{X}|$. Then,

$$
\begin{aligned}
& E[\sum_{i=1}^{L+1} I_{si}] \\
&= E[\sum_{i=1}^{L} I_{si}] + E[I_{sL+1}] \\
&= E[\sum_{i=1}^{L} I_{si}] + \sum_{(x_i)_{i=1}^{L} \in \mathcal{X}^L} E\left[I_{sL+1} \mid (X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\right] Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\}
\end{aligned} \tag{1}
$$

If X_{si} is popped out from the queue, $f(X_{si}) \leq \theta + K \leq \beta_\alpha(f) - K + K = \beta_\alpha(f)$, and hence, $I_{si} = 1$. Otherwise, if X_{si} is selected from $\mathcal{X}$ u.a.r., then $Prob\{I_{si} = 1\} = (U-k)/(|\mathcal{X}| - i + 1)$, where k is the number of points visited in the first $i-1$ steps with objective values smaller than or equal to $\beta_\alpha(f)$. Let C_L be the set collecting all $(x_i)_{i=1}^{L} \in \mathcal{X}^L$ such that if $(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}$, the queue will be nonempty in the $(L+1)$-th move. Therefore,

$$
\begin{aligned}
& \sum_{(x_i)_{i=1}^{L} \in \mathcal{X}^L} E\left[I_{sL+1} \mid (X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\right] Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\} \\
&= \sum_{(x_i)_{i=1}^{L} \in C_L} E[I_{sL+1} \mid (X_{si})_{i=1}^{L} \in C_L] Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\} + \\
& \quad \sum_{(x_i)_{i=1}^{L} \notin C_L} E[I_{sL+1} \mid (X_{si})_{i=1}^{L} \notin C_L] Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\} \\
&= \sum_{(x_i)_{i=1}^{L} \in C_L} 1 \cdot Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\} + \\
& \quad \sum_{(x_i)_{i=1}^{L} \notin C_L} \frac{U - \left|\{x_i \in (x_i)_{i=1}^{L} \mid f(x_i) \leq \beta_\alpha(f)\}\right|}{|\mathcal{X}| - L} Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\} \\
&\geq \sum_{(x_i)_{i=1}^{L} \in \mathcal{X}^L} \frac{U - \left|\{x_i \in (x_i)_{i=1}^{L} \mid f(x_i) \leq \beta_\alpha(f)\}\right|}{|\mathcal{X}| - L} Prob\{(X_{si})_{i=1}^{L} = (x_i)_{i=1}^{L}\} \\
&= \sum_{k=0}^{L} \frac{U-k}{|\mathcal{X}| - L} Prob\{\sum_{i=1}^{L} I_{si} = k\}
\end{aligned} \tag{2}
$$

Substituting into (1),

$$
\begin{aligned}
E[\sum_{i=1}^{L+1} I_{si}] &\geq \sum_{k=0}^{L} kProb\{\sum_{i=1}^{L} I_{si} = k\} + \sum_{k=0}^{L} \frac{U-k}{|\mathcal{X}|-L} Prob\{\sum_{i=1}^{L} I_{si} = k\} \\
&= \frac{U}{|\mathcal{X}|-L} + \frac{|\mathcal{X}|-L-1}{|\mathcal{X}|-L} \sum_{k=0}^{L} kProb\{\sum_{i=1}^{L} I_{si} = k\} \\
&= \frac{U}{|\mathcal{X}|-L} + \frac{|\mathcal{X}|-L-1}{|\mathcal{X}|-L} E[\sum_{i=1}^{L} I_{si}] \\
&\geq \frac{U}{|\mathcal{X}|-L} + \frac{|\mathcal{X}|-L-1}{|\mathcal{X}|-L} E[\sum_{i=1}^{L} I_{ri}] \qquad (3) \\
&= \sum_{k=0}^{L} kProb\{\sum_{i=1}^{L} I_{ri} = k\} + \sum_{k=0}^{L} \frac{U-k}{|\mathcal{X}|-L} Prob\{\sum_{i=1}^{L} I_{ri} = k\} \\
&= E[\sum_{i=1}^{L} I_{ri}] + \sum_{k=0}^{L} E[I_{rL+1} \mid \sum_{i=1}^{L} I_{ri} = k] Prob\{\sum_{i=1}^{L} I_{ri} = k\} \\
&= E[\sum_{i=1}^{L+1} I_{ri}]
\end{aligned}
$$

Inequality (3) follows from the induction hypothesis. □

Furthermore, next theorem guarantees that for any $f \in \mathcal{L}(G, \mathcal{Y}, K)$, if there exists a point above performance threshold and the subthreshold-seeker ever enters the local search phase, the subthreshold-seeker will outperform random search strictly in expectation according to the performance measure $\Psi_{\alpha,f}$.

Theorem 10 (Strictly better performance of STS on DLC)**.** *Let $\mathcal{L}(G, \mathcal{Y} = \{0, 1, \ldots, m\}, K)$ with $m > K$ be a DLC. For all $f \in \mathcal{L}(G, \mathcal{Y}, K)$ and for every subthreshold-seeker STS with $\theta \leq \beta_\alpha(f) - K$ satisfy:*

1. *$\exists v \in V(G)$ with $f(v) > \beta_\alpha(f)$, and*
2. *$\exists v \in V(G)$ with $f(v) \leq \theta$,*

$E[\Psi_{\alpha,f}(S_y(STS, f, L))] > E[\Psi_{\alpha,f}(S_y(RS, f, L))]$ for all $L \in [2, |\mathcal{X}| - 1]$.

Proof. If there are no such functions in $\mathcal{L}(G, \mathcal{Y}, K)$, the theorem holds vacuously. Otherwise, let f be any function satisfying the two conditions and define $((X_{si}, Y_{si}))_{i=1}^{L}$, $((X_{ri}, Y_{ri}))_{i=1}^{L}$, I_{si}, I_{ri}, U, and C_L in the same way as in Theorem 9. We prove by induction on L. When $L = 2$, since in the second step, the queue is nonempty if and only if $f(X_{si}) \leq \theta$, $C_1 = \{v \in V(G) \mid f(v) \leq$

$\theta\} \neq \emptyset$ by Condition (2). Therefore,

$$
\begin{aligned}
&E[I_{s1} + I_{s2}] \\
=&E[I_{s1}] + \sum_{x\in\mathcal{X}} E[I_{s2} \mid X_{s1} = x] Prob\{X_{s1} = x\} \\
=&E[I_{s1}] + \sum_{x:f(x)\leq\theta} 1 \cdot Prob\{X_{s1} = x\} + \sum_{x:\theta<f(x)\leq\beta_\alpha(f)} \frac{U-1}{|\mathcal{X}|-1} Prob\{X_{s1} = x\} \\
&+ \sum_{x:f(x)>\beta_\alpha(f)} \frac{U}{|\mathcal{X}|-1} Prob\{X_{s1} = x\} \\
>&E[I_{s1}] + \sum_{x:f(x)\leq\beta_\alpha(f)} \frac{U-1}{|\mathcal{X}|-1} Prob\{X_{s1} = x\} + \sum_{x:f(x)>\beta_\alpha(f)} \frac{U}{|\mathcal{X}|-1} Prob\{X_{s1} = x\} \\
=&E[I_{s1}] + \sum_{k\in\{0,1\}} \frac{U-k}{|\mathcal{X}|-1} Prob\{I_{s1} = k\} \\
=&E[I_{r1}] + \sum_{k\in\{0,1\}} \frac{U-k}{|\mathcal{X}|-1} Prob\{I_{r1} = k\} \\
=&E[I_{r1} + I_{r2}]
\end{aligned}
$$

The inequality follows from $C_1 \neq \emptyset$ and $(U-1)/(|\mathcal{X}|-1) < 1$, for Condition (1) implies $U < |\mathcal{X}|$. For induction hypothesis, suppose $E[\sum_{i=1}^{L} I_{si}] > E[\sum_{i=1}^{L} I_{ri}]$ for L with $2 \leq L < |\mathcal{X}| - 1$. In the $(L+1)$-th step, from the proof of Theorem 9, we always have

$$
\begin{aligned}
E[\sum_{i=1}^{L+1} I_{si}] &\geq \frac{U}{|\mathcal{X}|-L} + \frac{|\mathcal{X}|-L-1}{|\mathcal{X}|-L} E[\sum_{i=1}^{L} I_{si}] \\
&> \frac{U}{|\mathcal{X}|-L} + \frac{|\mathcal{X}|-L-1}{|\mathcal{X}|-L} E[\sum_{i=1}^{L} I_{ri}] \qquad (4) \\
&= E[\sum_{i=1}^{L+1} I_{ri}]
\end{aligned}
$$

Since $(|\mathcal{X}|-L-1)/(|\mathcal{X}|-L) > 0$ when $L < |\mathcal{X}| - 1$, and $E[\sum_{i=1}^{L} I_{si}] > E[\sum_{i=1}^{L} I_{ri}]$ from the induction hypothesis, Inequality (4) is strict. □

Let $d := \max\{deg(v) \mid v \in V(G)\}$ be the maximum degree of the graph and $dis(u, v)$ be the length of the shortest path from u to v. For any subthreshold-seeker, if we are able to set its θ within some interval, the following corollary gives a sufficient condition of the existence of functions on which the subthreshold-seeker strictly outperforms random search.

Corollary 11. *Let $\mathcal{L}(G, \mathcal{Y} = \{0, 1, \ldots, m\}, K)$ be a DLC. Given $\alpha \in (0, 1/2]$ and an integer $C > 1$ with $CK + 1 \leq m$, if*

$$\alpha|V(G)| > \frac{d(d-1)^C - 2}{d-2},$$

then there exists a function $f \in \mathcal{L}(G, \mathcal{Y}, K)$ such that $E[\Psi_{\alpha,f}(S_y(STS, f, L))] > E[\Psi_{\alpha,f}(S_y(RS, f, L))]$ for all L with $2 \leq L \leq |\mathcal{X}| - 1$, where STS is a subthreshold-seeker with $\theta \in \beta_\alpha(f) - [K, CK]$.

Proof. We prove this corollary constructively. Select a vertex v_0 from $V(G)$ arbitrarily. Consider

the function f defined as

$$f(v) = \begin{cases} 0 & \text{if } v = v_0 \ ; \\ dis(v, v_0)K & \text{if } 1 \leq dis(v, v_0) \leq C \ ; \\ CK + 1 & \text{otherwise.} \end{cases}$$

Since

$$\begin{aligned} &|v_o| + |\{v \in V(G) \mid 1 \leq dis(v, v_0) \leq C\}| \\ &\leq 1 + \left(d + d(d-1) + d(d-1)^2 + \ldots + d(d-1)^{C-1}\right) \\ &= 1 + \frac{d\left[(d-1)^C - 1\right]}{(d-1) - 1} \\ &= \frac{d(d-1)^C - 2}{d-2} < \alpha|V(G)| \ , \end{aligned}$$

from the definition of $\beta_\alpha(f)$, $\beta_\alpha(f) = CK$. Furthermore, there must exist $v_1 \in V(G)$ that $f(v_1) = CK + 1$, for $|v_o| + |\{v \in V(G) \mid 1 \leq dis(v, v_0) \leq C\}| < |V(G)|$. Therefore, we have $f(v_0) \leq \theta$ and $f(v_1) > \beta_\alpha(f)$. Thereby Theorem 10 can be applied. □

Combining Theorem 9 and Theorem 10, if we manage to set $\theta \leq \beta_\alpha(f) - K$, the subthreshold-seeker will perform at least as good as random search on a DLC. If the subthreshold-seeker has a chance to conduct local search, it will strictly outperform random search. Estimating a θ within some range should be more practical than gauging a specific value such as $\beta_\alpha(f)$. In next section, we will explore this possibility and empirically confirm the theoretical results obtained in this section by proposing and adopting a sampling-test scheme.

5 Sampling-test scheme

Conventionally, the effectiveness of an optimizer is examined via experiments on a suite of test functions that serves as a benchmark. These test functions are selected according to some prior knowledge of the importance thereof. Here we propose and adopt a different approach in order to confirm the theoretical results obtained in the previous section from an empirical aspect. We draw a sample of functions randomly from PDLC in a manner similar to select respondents in a campaign survey and conduct experiments on these sampled functions. There is no bias in favor of which functions should be selected. We expect the arbitrariness delivers information about the composition of the problem class.

A uniform sampler for PDLC is firstly given in section 5.1. Experiments are then presented to summarize this section and demonstrate how the Lipschitz condition facilitates the search process in a practical standpoint.

5.1 A uniform sampler for PDLC

In order to conduct the sampling test, we need a uniform sampler in the first place. The following algorithm generates problem instances of PDLC with Lipschitz constant K uniformly at random (u.a.r.)

Algorithm 2 (Uniform PDLC Sampler)**.**
procedure UNIFORM PDLC SAMPLER$(\overline{v_1 v_2 \ldots v_n}, \mathcal{Y} = \{0, 1, \ldots, m\}, K)$
$f(v_1) \leftarrow Uniform([0, m])$
$i \leftarrow 2$

while $i \leq n$ ***do***
$f(v_i) \leftarrow f(v_{i-1}) + Uniform([-K, K])$
$i \leftarrow i + 1$
if $f(v_i) > m$ *or* $f(v_i) < 0$ ***then***
$f(v_1) \leftarrow Uniform([0, m])$
$i \leftarrow 2$;
end if
end while
return f
end procedure

Here $Uniform([a, b])$ denotes the function that selects an integer u.a.r. from the closed interval $[a, b]$. Such a sampler belongs to the category of accept-reject algorithms [14]. It generates a problem instance with bounded difference between any two successive vertexes u.a.r., and if the instance at hand exceeds the range of the codomain, the sampler rejects the instance. The accept-reject mechanism guarantees the uniformity. Once the sampler halts, the output is always an instance of the PDLC.

Since this sampler is Las Vegas, we need to address its time complexity for the practicality. For each candidate instance, the sampler will go through at most $|\mathcal{X}|$ steps to assign all the vertex objective values, so it remains to show how many candidate instances it takes to generate a legit instance successfully. The accept-reject process is geometrically distributed, and therefore the expected number of instances consumed is the inverse of the acceptance probability. The following theorem provides an upper bound for the rejection probability.

Lemma 12. *Suppose* $|\mathcal{Y}| = 2m + 1$*, where* m *is an integer, and* $|\mathcal{X}| = n$*. If*

$$m > \sqrt{\frac{(n-1)(K^2+K)}{3}} \geq 2 \,,$$

then the rejection probability is less than

$$\frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}|\mathcal{Y}|} - \frac{4(n-1)(K^2+K)}{3|\mathcal{Y}|^2} + \frac{5}{|\mathcal{Y}|} \,.$$

Proof. Without loss of generality, suppose $\mathcal{Y} = \{-m, -m+1, \ldots, m\}$. Let (K_i) be a sequence of i.i.d. random variables that $K_i = j$ with probability $1/(2K+1)$ for $j \in \{-K, -K+1, \ldots, K\}$ and $S_j := \sum_{i=1}^{j} K_i$. When $f(v_1) = i$, the instance is rejected if and only if $i + S_j \geq m + 1$ or $i + S_j \leq -m - 1$ for some $1 \leq j \leq n - 1$, so the occurrence of rejection always implies $\max_{1 \leq j \leq n-1} |S_j| \geq \min\{|m + 1 - i|, | - m - 1 - i|\}$. Moreover, the symmetry indicates that $Prob\{\text{rejection} \mid f(v_1) = i\} = Prob\{\text{rejection} \mid f(v_1) = -i\}$ for $|i| \leq m$. Therefore,

$$\begin{aligned}
&Prob\{\text{rejection}\} \\
&= \sum_{i=-m}^{m} Prob\{\text{rejection} \mid f(v_1) = i\} Prob\{f(v_1) = i\} \\
&= \frac{\sum_{i=-m}^{m} Prob\{\text{rejection} \mid f(v_1) = i\}}{2m+1} \\
&\leq \frac{Prob\{\max_{1 \leq j \leq n-1} |S_j| \geq m+1\} + 2\sum_{i=1}^{m} Prob\{\max_{1 \leq j \leq n-1} |S_j| \geq m+1-i\}}{2m+1} \\
&= \frac{Prob\{\max_{1 \leq j \leq n-1} |S_j| \geq m+1\} + 2\sum_{i=1}^{m} Prob\{\max_{1 \leq j \leq n-1} |S_j| \geq i\}}{2m+1} \,.
\end{aligned}$$

Using Kolmogorov's inequality [15], we can get

$$Prob\{\max_{1\le j\le n-1}|S_j|\ge i\}\le \min\left\{\frac{Var[S_{n-1}]}{i^2}\ ,\ 1\right\}\ .$$

Since $Var[K_i]=2(1^2+2^2+\ldots+K^2)/(2K+1)=(K^2+K)/3$, $Var[S_{n-1}]=(n-1)Var[K_i]=(n-1)(K^2+K)/3$. Moreover, $Var[S_{n-1}]/i^2\le 1$ if and only if $i\ge\sqrt{Var[S_{n-1}]}$, we have

$$\begin{aligned}
&Prob\{\text{rejection}\}\\
&\le \frac{\frac{Var[S_{n-1}]}{(m+1)^2}+2\left(\sum_{i=1}^{\left\lceil\sqrt{Var[S_{n-1}]}\right\rceil-1}1+\sum_{i=\left\lceil\sqrt{Var[S_{n-1}]}\right\rceil}^{m}\frac{Var[S_{n-1}]}{i^2}\right)}{2m+1}\\
&\le \frac{\frac{Var[S_{n-1}]}{(m+1)^2}+2\left(\left\lceil\sqrt{Var[S_{n-1}]}\right\rceil-1+Var[S_{n-1}]\int_{x=\left\lceil\sqrt{Var[S_{n-1}]}\right\rceil-1}^{m}x^{-2}dx\right)}{2m+1}\\
&\le \frac{\frac{Var[S_{n-1}]}{(m+1)^2}+2\left(\sqrt{Var[S_{n-1}]}-\frac{Var[S_{n-1}]}{m}+\frac{Var[S_{n-1}]}{\left\lceil\sqrt{Var[S_{n-1}]}\right\rceil-1}\right)}{2m+1}\ .
\end{aligned}$$

Since $x/(x-1)$ decreases when $x>1$, we obtain

$$\begin{aligned}
\frac{Var[S_{n-1}]}{\left\lceil\sqrt{Var[S_{n-1}]}\right\rceil-1}&\le\frac{Var[S_{n-1}]}{\sqrt{Var[S_{n-1}]}-1}\\
&=\sqrt{Var[S_{n-1}]}+\frac{\sqrt{Var[S_{n-1}]}}{\sqrt{Var[S_{n-1}]}-1}\\
&\le\sqrt{Var[S_{n-1}]}+2\ .
\end{aligned}$$

According to the hypothesis that $Var[S_{n-1}]/(m+1)^2<1$,

$$\begin{aligned}
Prob\{\text{rejection}\}&<\frac{4\sqrt{Var[S_{n-1}]}-\frac{2Var[S_{n-1}]}{m}+5}{2m+1}\\
&=\frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}(2m+1)}-\frac{2(n-1)(K^2+K)}{3m(2m+1)}+\frac{5}{2m+1}\\
&<\frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}|\mathcal{Y}|}-\frac{4(n-1)(K^2+K)}{3|\mathcal{Y}|^2}+\frac{5}{|\mathcal{Y}|}\ .
\end{aligned}$$

□

Theorem 13 (Upper bound for the rejection probability)**.** *Define $m:=\lfloor(|\mathcal{Y}|-1)/2\rfloor$. If $m>\sqrt{(n-1)(K^2+K)/3}\ge 2$, then the rejection probability is less than*

$$\frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}|\mathcal{Y}|}-\frac{4(n-1)(K^2+K)}{3|\mathcal{Y}|^2}+O\left(|\mathcal{Y}|^{-1}\right)\ .$$

Proof. If $|\mathcal{Y}|=2m+1$, then we are done by the previous lemma. Otherwise, if $|\mathcal{Y}|=2m+2$, without loss of generality, suppose that $\mathcal{Y}=\{-m,-m+1,\ldots,m+1\}$ and let $\mathcal{Y}'=\{-m,-m+$

$1, \ldots, m\}$. Therefore,

$$\begin{aligned}
&Prob\{\text{rejection}\} \\
=&Prob\{f(v_1) \in \mathcal{Y}'\} Prob\{\text{rejection} \mid f(v_1) \in \mathcal{Y}'\} \\
&+ Prob\{f(v_1) \notin \mathcal{Y}'\} Prob\{\text{rejection} \mid f(v_1) \notin \mathcal{Y}'\} \\
=&\left(\frac{2m+1}{2m+2}\right) Prob\{\text{rejection} \mid f(v_1) \in \mathcal{Y}'\} \\
&+ \left(\frac{1}{2m+2}\right) Prob\{\text{rejection} \mid f(v_1) = m+1\} \ .
\end{aligned}$$

When $f(v_1) \in \mathcal{Y}'$, if f exceeds the range of $\mathcal{Y}$, then f also exceeds the range of $\mathcal{Y}'$, so from the previous lemma we have

$$Prob\{\text{rejection} \mid f(v_1) \in \mathcal{Y}'\} < \frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}(2m+1)} - \frac{4(n-1)(K^2+K)}{3(2m+1)^2} + \frac{5}{2m+1} \ .$$

As a result,

$$\begin{aligned}
&Prob\{\text{rejection}\} \\
&\leq \left(\frac{2m+1}{2m+2}\right) Prob\{\text{rejection} \mid f(v_1) \in \mathcal{Y}'\} + \left(\frac{1}{2m+2}\right) \\
&< \frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}(2m+2)} - \frac{4(n-1)(K^2+K)}{3(2m+1)(2m+2)} + \frac{6}{2m+2} \\
&< \frac{4\sqrt{(n-1)(K^2+K)}}{\sqrt{3}|\mathcal{Y}|} - \frac{4(n-1)(K^2+K)}{3|\mathcal{Y}|^2} + O\left(|\mathcal{Y}|^{-1}\right)
\end{aligned}$$

□

Corollary 14. *If* $|\mathcal{Y}| = C\sqrt{(n-1)(K^2+K)} > C \cdot 2\sqrt{3}$ *for some constant* $C \geq \sqrt{3}$*, then the rejection probability is less than*

$$\frac{4\sqrt{3}C - 4}{3C^2} + O\left(|\mathcal{Y}|^{-1}\right) \ .$$

Proof. If $C \geq \sqrt{3}$,

$$\begin{aligned}
m = \left\lfloor \frac{|\mathcal{Y}|-1}{2} \right\rfloor &\geq \frac{|\mathcal{Y}|}{2} - 1 \\
&\geq \frac{\sqrt{3(n-1)(K^2+K)}}{2} - 1 \\
&= \sqrt{\frac{(n-1)(K^2+K)}{3}} + \frac{\sqrt{(n-1)(K^2+K)}}{2\sqrt{3}} - 1 \\
&> \sqrt{\frac{(n-1)(K^2+K)}{3}} \ .
\end{aligned}$$

Substituting $\sqrt{(n-1)(K^2+K)}/|\mathcal{Y}|$ by $1/C$ and applying Theorem 13, the corollary is proved. □

For instance, if $C = \sqrt{3}$ and $|\mathcal{Y}|$ is so large that $O(|\mathcal{Y}|^{-1})$ is negligible, the expected number of instances consumed is no more than 9. Multiplying the time to assign all vertexes values, the expected runtime, in terms of the number of assignments, is no more than $9|\mathcal{X}|$. In other words, asymptotically speaking, if $|\mathcal{X}|$ and $|\mathcal{Y}|$ are about equal and $|\mathcal{Y}|$ is larger than K^2 to some extent, then the expected runtime is approximately linear.

5.2 Experimental settings and results

As demonstrated in section 4, the virtues of the subthreshold-seeker rely on a proper algorithmic threshold. Although the main results in section 4 hold when $\theta \leq \beta_\alpha(f) - K$, because we do not set a performance threshold literally to scrutinize how many subthreshold points are visited in real-world applications, in an experimental setting, we can examine the subthreshold-seeker more practically in terms of the time to identify the optimum. Therefore, the algorithmic threshold should be utilized for optimization, or more specifically, to minimize the objective function in this case.

We will compare the subthreshold-seeker with random search. Here we present three different subthreshold-seekers. For the theoretical purpose, the first one uses the actual median of all objective values, in the form of exterior knowledge, as θ. The second one firstly selects a 100 points u.a.r. and then employs the calculated median as θ. The third one also starts with obtaining 100 points u.a.r., but it computes the mean and the standard deviation of these points and sets θ to the mean minus the standard deviation. Moreover, the three subthreshold-seekers and random search obey the NFL framework and hence are non-repeating.

In advance of experiments, we need to determine the size of the set PDLC problems to be sampled. Suppose we want to estimate a population proportion $q \in [0, 1]$. We draw a sequence of samples uniformly and independently from the population with replacement. For each sample, we observe if it belongs to the variety of interest. With a large sample size, we expect the proportion in the sample approximates the real proportion. The following theorem depicts the relationship between the sample size and the error bound.

Theorem 15 (Sample size and error bound)**.** *Let (Z_i) be a sequence of i.i.d. indicator variables with $E[Z_i] = q$. For all $\delta, \epsilon \in (0, 1)$, if*

$$n \geq -\frac{\ln(\delta/2)}{2\epsilon^2} ,$$

then

$$Prob\left\{\left|\frac{\sum_{i=1}^{n} Z_i}{n} - q\right| > \epsilon\right\} \leq \delta .$$

Proof. Let $\overline{Z} = (\sum_{i=1}^{n} Z_i)/n$. Applying Hoeffding's inequality [16], for $0 < \epsilon < 1 - q$, we have

$$Prob\{\overline{Z} - q > \epsilon\} \leq e^{-2n\epsilon^2} ,$$

and for $0 < \epsilon < q$,

$$Prob\{\overline{Z} - q < -\epsilon\} \leq e^{-2n\epsilon^2} .$$

Moreover, if $\epsilon \geq 1 - q$,

$$Prob\{\overline{Z} - q > \epsilon\} \leq Prob\{\overline{Z} > 1\} = 0 \leq e^{-2n\epsilon^2} .$$

Similarly, if $\epsilon \geq q$,

$$Prob\{\overline{Z} - q < -\epsilon\} \leq Prob\{\overline{Z} < 0\} = 0 \leq e^{-2n\epsilon^2} .$$

Hence, we conclude that for all $\epsilon \in (0, 1)$,

$$Prob\{|\overline{Z} - q| > \epsilon\} \leq 2e^{-2n\epsilon^2} .$$

Finally,

$$n \geq -\frac{\ln(\delta/2)}{2\epsilon^2}$$

implies $2e^{-2n\epsilon^2} \leq \delta$, and we complete the proof. □

Table 1: Successful rate

θ	category	$(\lvert\mathcal{X}\rvert, \lvert\mathcal{Y}\rvert)$		
		$(10^4, 10^4)$	$(10^5, 10^5)$	$(10^6, 10^6)$
γ	$>$	0.9995 (2,049)	0.9985 (2,047)	0.9976 (2,045)
	$> .2$	0.9951 (2,040)	0.9620 (1,972)	0.9624 (1,973)
$\hat{\gamma}$	$>$	0.9995 (2,049)	0.9990 (2,048)	0.9985 (2,047)
	$> .2$	0.9937 (2,037)	0.9620 (1,972)	0.9732 (1,995)
$\hat{\mu} - \hat{\sigma}$	$>$	1.0000 (2,050)	1.0000 (2,050)	1.0000 (2,050)
	$> .2$	1.0000 (2,050)	1.0000 (2,050)	1.0000 (2,050)

γ: median. $\hat{\gamma}$: estimated median. $\hat{\mu}$: estimated mean. $\hat{\sigma}$: estimated standard deviation. ">": proportion of instances where the subthreshold-seeker outperforms random search. "$> .2$": proportion of instances where the subthreshold-seeker outperforms random search by a 20% margin.

In particular, with the conventional setting of $(\epsilon, \delta) = (0.03, 0.05)$, a sample of size $2,050$ suffices. In other words, if we draw a sample of size $2,050$, $[\overline{Z} - 0.03, \overline{Z} + 0.03]$ forms a confidence interval for q with confidence level at least 95%.

The sampler generates 2,050 instances of PDLC with $(\lvert\mathcal{X}\rvert, \lvert\mathcal{Y}\rvert) = (10^4, 10^4)$, $(10^5, 10^5)$, and $(10^6, 10^6)$, respectively. The Lipschitz constant K is set to 100 for the concern of execution time, as previously discussed. For each problem instance, we test each algorithm for 50 independent runs. If the average time of a subthreshold-seeker to find the optimum is less than that of random search, the instance is counted as a success. We also count the number of instances that a subthreshold-seeker outperforms random search by a 20% margin, i.e., the instance where the average optimization time of a subthreshold-seeker is less than 80% of that of random search. Table 1 displays the empirical results.

All three subthreshold-seekers outperform random search in most of the sampled problem instances. Furthermore, the subthreshold-seeker with $\theta = \hat{\mu} - \hat{\sigma}$ outperforms random search in all 2,050 instances sampled, even with the requirement of a 20% margin. The statistical significance of such results is obvious to see: Suppose the population proportion that the subthreshold-seeker with $\theta = \hat{\mu} - \hat{\sigma}$ outperforms random search is q. To obtain the result that random search is outperformed in all instances, the probability is q^{2050}. Even if q is as high as 0.995, the above probability is just 0.000034. To more formally rephrase, if the null hypothesis is "$q \leq 0.995$", the p-value is merely 0.000034.

Table 2 displays the averaged optimization time over the 2,050 sampled problem instances. The subthreshold-seeker with $\theta = \hat{\mu} - \hat{\sigma}$ outperforms others by a significant margin. Random search averages approximately $\lvert\mathcal{X}\rvert/2$ to find the minimum, which is expected. The subthreshold-seeker using the actual median and the one using the sample median both take about half time steps of that needed by random search to optimize the function.

The subthreshold-seekers with $\theta = \hat{\mu} - \hat{\sigma}$ and $\theta = \hat{\gamma}$ are indeed black-box algorithms, for there is no exterior knowledge exerted and the only information they can use are function evaluations, but they outperform random search by a remarkable difference.

The performance difference between $\theta = \hat{\gamma}$ and $\theta = \gamma$ is insignificant. Such a result suggests that in this case, an estimation of median may be adequate. Suppose that P with $\lvert P\rvert = N$ is a subset of real numbers, and for all $i \in P$, $R(i)$ is defined to be the rank (i.e., ordering) of i in P. For instance, $R(\min P) = 1$ and $R(\max P) = N$. For simplicity, we assume that N is odd and hence the median of P is the element i with $R(i) = \lceil N/2 \rceil$. Now we want to estimate the median of P. If a point sample S of size n, where n is assumed odd, is drawn by successively selecting an element u.a.r. from P with replacement, the estimated median, γ, is presumed to

Table 2: Mean time steps to locate the minimum

algorithm	$(\lvert\mathcal{X}\rvert, \lvert\mathcal{Y}\rvert)$		
	$(10^4, 10^4)$	$(10^5, 10^5)$	$(10^6, 10^6)$
STS, $\theta = \gamma$	2037.58	22913.23	229232.26
STS, $\theta = \hat{\gamma}$	2221.44	23170.58	229532.04
STS, $\theta = \hat{\mu} - \hat{\sigma}$	918.29	8095.78	80322.92
random search	4972.50	49724.74	496912.49

γ: median. $\hat{\gamma}$: estimated median. $\hat{\mu}$: estimated mean. $\hat{\sigma}$: estimated standard deviation.

be the sampled median, and we want the error is bounded by $\epsilon > 0$, i.e., $|R(\gamma) - \lceil N/2 \rceil| \leq \epsilon N$.

If $R(\gamma) < \lceil N/2 \rceil - \epsilon N$, there are at least $\lceil n/2 \rceil$ selections with ranks less than $\lceil N/2 \rceil - \epsilon N$. Let X_i be the indicator variable that indicates if the i-th selection is less than $\lceil N/2 \rceil - \epsilon N$, $X_i = 1$ with probability $p := (\lceil N/2 \rceil - \lfloor \epsilon N \rfloor - 1)/N$. $R(\gamma) < \lceil N/2 \rceil - \epsilon N$ if and only if $\sum_{i=1}^{n} X_i \geq \lceil n/2 \rceil$. Since $E[\sum_{i=1}^{n} X_i] = np$, applying another form of Hoeffding's inequality [16], we have

$$
\begin{aligned}
Prob\left\{R(\gamma) < \left\lceil \frac{N}{2} \right\rceil - \epsilon N\right\} &= Prob\left\{\sum_{i=1}^{n} X_i \geq \left\lceil \frac{n}{2} \right\rceil\right\} \\
&\leq Prob\left\{\sum_{i=1}^{n} X_i \geq \frac{n}{2}\right\} \\
&= Prob\left\{\frac{1}{n}\sum_{i=1}^{n} X_i \geq p + (\frac{1}{2} - p)\right\} \\
&\leq \left[\left(\frac{p}{p + \frac{1}{2} - p}\right)^{p+\frac{1}{2}-p} \left(\frac{1-p}{1-p-(\frac{1}{2}-p)}\right)^{1-p-(\frac{1}{2}-p)}\right]^{n} \\
&= \left[4p(1-p)\right]^{\frac{n}{2}} \ .
\end{aligned}
$$

Moreover, the symmetry implies that

$$
Prob\left\{R(\gamma) > \left\lceil \frac{N}{2} \right\rceil + \epsilon N\right\} \leq \left[4p(1-p)\right]^{\frac{n}{2}} \ .
$$

Therefore,

$$
Prob\left\{\left|R(\gamma) - \left\lceil \frac{N}{2} \right\rceil\right| > \epsilon N\right\} \leq 2\left[4p(1-p)\right]^{\frac{n}{2}} \ .
$$

Now the only quantity left is p. By definition,

$$
p = \frac{\lceil \frac{N}{2} \rceil - \lfloor \epsilon N \rfloor - 1}{N} \approx \frac{1}{2} - \epsilon \ .
$$

For instance, if we set $\epsilon = 0.1$ and $n = 100$, the probability of exceeding the error bound is less than 0.26. If the sample size n increases to $2,000$, even with a small $\epsilon = 0.03$, the probability reduces to just 0.054. It is noteworthy that the effect of the population size N is negligible. Therefore, the required number of samples remains the same, even if the search space is immense. Although in real-world applications P is usually a multiset, if the multiplicities of P are not too large, such a gauge should not diverge significantly.

6 Conclusions

In this study, we introduced and investigated the properties of the discrete Lipschitz class. A generalized subthreshold-seeker was then proposed and shown to outperform random search on this broad function class. Finally, we proposed a tractable sampling-test scheme to empirically demonstrate the performance of the generalized subthreshold-seeker under practical configurations. We showed that optimization algorithms outperforming random search on the discrete Lipschitz class do exist from both theoretical and practical aspects.

As controversial as it may be, the NFL theorem provides an alternative standpoint to review the position of optimization algorithms and search heuristics. The NFL theorem expels the false hope to conquer all possible functions with only limited information available, as it points out the expectation to find a universally black-box optimizer is definitely over-optimistic. However, the NFL theorem does not imply the utter infertility in the land of search heuristics by any means if our goals are appropriately placed. In this paper, the discrete Lipschitz class, as a simulation of continuous functions in a discrete space, is shown to be a class of problems on which black-box optimizers have performance advantages in both theory and practice. The only constraint imposed on the search space is bounded differences within a neighborhood. Under such a minor condition, black-box optimizers can still be effective over a broad, meaningful, and practical problem class as suggested by this study.

Acknowledgments

The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

References

[1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," Santa Fe Institute, Tech. Rep. SFI-TR-95-02-010, 1995.

[2] ——, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 67–82, 1997.

[3] J. C. Culberson, "On the futility of blind search: An algorithmic view of "no free lunch"," *Evolutionary Computation*, vol. 6, pp. 109–127, 1998.

[4] S. Droste, T. Jansen, and I. Wegener, "Perhaps not a free lunch but at least a free appetizer," Department of Computer Science, University of Dortmund, Tech. Rep. ISSN 1433-3325, 1998.

[5] ——, "Optimization with randomized search heuristics – the (a)nfl theorem, realistic scenarios, and difficult functions," *Theoretical Computer Science*, vol. 287, pp. 131–144, 2002.

[6] M. J. Streeter, "Two broad classes of functions for which a no free lunch result does not hold," in *Proceedings of the Genetic and Evolutionary Computation Conference 2003*, 2003, pp. 1418–1430.

[7] S. Christensen and F. Oppacher, "What can we learn from no free lunch? a first attempt to characterize the concept of a searchable function," in *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, 2001, pp. 1219–1226.

[8] D. Whitley and J. Rowe, "Subthreshold-seeking local search," *Theoretical Computer Science*, vol. 361, pp. 2–17, 2006.

[9] C. Schumacher, M. D. Vose, and L. D. Whitley, "The no free lunch and problem description length," in *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, 2001, pp. 565–570.

[10] R. Courant and F. John, *Introduction to Calculus and Analysis, Vol. 1.* Springer-Verlag, 1989, vol. 1.

[11] R. Motwani and P. Raghavan, *Randomized Algorithms.* Cambridge University Press, 1995.

[12] I. Rechenberg, *Evolutionsstrategie '94.* Frommann Holzboog, 1994.

[13] J. Jägersküpper, "Algorithmic analysis of a basic evolutionary algorithm for continuous optimization," *Theoretical Computer Science*, vol. 379, no. 3, pp. 329–347, 2007.

[14] C. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer-Verlag, 1999.

[15] Y. S. Chow and H. Teicher, *Probability theory: independence, interchangeability, martingales*, 3rd ed. Springer, 1997.

[16] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.