Coláiste na hOllscoile Corcaigh, Éire
University College Cork, Ireland

CS4618 Artificial Intelligence I

Today: Assessment of Randomised Search Heuristics: Almost No Free Lunch Black-Box Complexity

Thomas Jansen

November 7th

Plans for Today

1. NFL/ANFL
   - Almost No Free Lunch
2. Black Box Complexity
   - Motivation
3. Summary
   - Summary & Take Home Message

Almost No Free Lunch

Almost No Free Lunch

ANFL Theorem

Let $S = \{0,1\}^n$, $R = \{0, 1, \ldots, N-1\}$, $f\colon S \to R$, A a randomised BBA.

Almost No Free Lunch

ANFL Theorem

Let $S = \{0,1\}^n$, $R = \{0,1,\ldots,N-1\}$, $f\colon S \to R$, A a randomised BBA.
The number of functions $f'\colon S \to R \cup \{N\}$ sucht that A does not find an optimum of f' within $2^{n/3}$ f-evaluations with probability at least $1 - 2^{-n/3}$ is bounded below by $N^{2^{n/3}-1}$.

Almost No Free Lunch

ANFL Theorem

Let $S = \{0,1\}^n$, $R = \{0,1,\ldots,N-1\}$, $f\colon S \to R$, A a randomised BBA.
The number of functions $f'\colon S \to R \cup \{N\}$ sucht that A does not find an optimum of f' within $2^{n/3}$ f-evaluations with probability at least $1 - 2^{-n/3}$ is bounded below by $N^{2^{n/3}-1}$.
Of these exponentially many have the additional property that their complexity (measured by evaluation time, circuit size, or Kolmogoroff complexity) is by $O(n)$ larger than that of f.

Almost No Free Lunch

ANFL Theorem

Let $S = \{0,1\}^n$, $R = \{0,1,\ldots,N-1\}$, $f\colon S \to R$, A a randomised BBA.
The number of functions $f'\colon S \to R \cup \{N\}$ sucht that A does not find an optimum of f' within $2^{n/3}$ f-evaluations with probability at least $1 - 2^{-n/3}$ is bounded below by $N^{2^{n/3}-1}$.
Of these exponentially many have the additional property that their complexity (measured by evaluation time, circuit size, or Kolmogoroff complexity) is by $O(n)$ larger than that of f.

Consequence If your algorithm is efficient on some function f it is necessarily inefficient for very many other functions, many of those not too different from f.

Proving the ANFL

Proving the ANFL

W.l.o.g. A eventually evaluates any $s \in \{0,1\}^n$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$q(x) = \text{Prob}\left(A \text{ evaluates } x \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$$q(x) = \text{Prob}\Big(A \text{ evaluates } x \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\Big)$$

Obvious $\sum_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$$q(x) = \text{Prob}\left(A \text{ evaluates } x \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$$

Obvious $$\sum_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$$

Define for $b \in \{0,1\}^{2n/3}$

$$S_b := \left\{x \in \{0,1\}^n \mid x \in b*^{n/3}\right\}$$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$q(x) = \mathrm{Prob}\left(A \text{ evaluates } x \text{ in} \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Obvious $\sum_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$

Define for $b \in \{0,1\}^{2n/3}$

$S_b := \left\{x \in \{0,1\}^n \mid x \in b{*}^{n/3}\right\}$

$q'(b) := \mathrm{Prob}\left(A \text{ evaluates } x \in S_b \text{ in} \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$$q(x) = \text{Prob}\left(A \text{ evaluates } x \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$$

Obvious $\sum_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$

Define for $b \in \{0,1\}^{2n/3}$

$$S_b := \left\{x \in \{0,1\}^n \mid x \in b{*}^{n/3}\right\}$$

$$q'(b) := \text{Prob}\left(A \text{ evaluates } x \in S_b \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$$

Obvious $q'(b) \leq \sum_{x \in S_b} q(x)$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$q(x) = \text{Prob}\left(A \text{ evaluates } x \text{ in} \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Obvious $\sum\limits_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$

Define for $b \in \{0,1\}^{2n/3}$

$S_b := \left\{x \in \{0,1\}^n \mid x \in b{*}^{n/3}\right\}$

$q'(b) := \text{Prob}\left(A \text{ evaluates } x \in S_b \text{ in} \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Obvious $q'(b) \leq \sum\limits_{x \in S_b} q(x)$

Obvious S_b pairwise disjoint for different b

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$

$$q(x) = \text{Prob}\left(A \text{ evaluates } x \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$$

Obvious $\sum_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$

Define for $b \in \{0,1\}^{2n/3}$

$$S_b := \left\{x \in \{0,1\}^n \mid x \in b*^{n/3}\right\}$$

$$q'(b) := \text{Prob}\left(A \text{ evaluates } x \in S_b \text{ in } \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$$

Obvious $q'(b) \leq \sum_{x \in S_b} q(x)$

Obvious S_b pairwise disjoint for different b

Thus $\sum_{b \in \{0,1\}^{2n/3}} q'(b) \leq \sum_{x \in \{0,1\}^n} q(x)$

Proving the ANFL

W. l. o. g. A eventually evaluates any $s \in \{0,1\}^n$

Consider first $2^{n/3}$ points in $\{0,1\}^n$ evaluated by A

Define for $x \in \{0,1\}^n$
$q(x) = \text{Prob}\left(A \text{ evaluates } x \text{ in} \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Obvious $\sum\limits_{x \in \{0,1\}^n} q(x) \leq 2^{n/3}$

Define for $b \in \{0,1\}^{2n/3}$
$S_b := \left\{x \in \{0,1\}^n \mid x \in b{*}^{n/3}\right\}$
$q'(b) := \text{Prob}\left(A \text{ evaluates } x \in S_b \text{ in} \leq 2^{n/3} \text{ first } f\text{-evaluations}\right)$

Obvious $q'(b) \leq \sum\limits_{x \in S_b} q(x)$

Obvious S_b pairwise disjoint for different b

Thus $\sum\limits_{b \in \{0,1\}^{2n/3}} q'(b) \leq \sum\limits_{x \in \{0,1\}^n} q(x)$

Thus $\exists b^* \in \{0,1\}^{2n/3} \colon q'(b^*) \leq 2^{n/3}/2^{2n/3} = 2^{-n/3}$ ✔

Defining the f'

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{`almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{`almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{'almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

NFL Summary

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{'almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

NFL Summary

- Statements about efficiency of search heuristics need by restricted to function classes.

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{`almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

NFL Summary

- Statements about efficiency of search heuristics need by restricted to function classes.
- For most function classes NFL does not hold.

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{`almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

NFL Summary

- Statements about efficiency of search heuristics need by restricted to function classes.
- For most function classes NFL does not hold.
- For 'natural' function classes NFL does not hold.

Defining the f'

Define $f' \colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{'almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

NFL Summary

- Statements about efficiency of search heuristics need by restricted to function classes.
- For most function classes NFL does not hold.
- For 'natural' function classes NFL does not hold.
- NFL tells you nothing about actual computation times.

Defining the f'

Define $f'\colon S \to R \cup \{N\}$

by $f'(x) := \begin{cases} f(x) & \text{if } x \notin S_{b^*} \\ \text{`almost arbitrary'} & \text{otherwise} \end{cases}$

(but with a $x' \in S_{b^*}$ with $f'(x') = N$)

Observation counting ✓

□

NFL Summary

- Statements about efficiency of search heuristics need by restricted to function classes.
- For most function classes NFL does not hold.
- For 'natural' function classes NFL does not hold.
- NFL tells you nothing about actual computation times.
- There are no 'generally efficient' search heuristics.

RSH and NFL

We know

- several randomised search heuristics

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms
- No Free Lunch
  - ‘On average all RSH perform equal.’
  - NFL holds iff $\mathcal{F}$ is c. u. p.

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms
- No Free Lunch
  - ‘On average all RSH perform equal.’
  - NFL holds iff $\mathcal{F}$ is c. u. p.

Consequences

- $\nexists$ general best RSH

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms
- No Free Lunch
  - 'On average all RSH perform equal.'
  - NFL holds iff $\mathcal{F}$ is c. u. p.

Consequences

- $\nexists$ general best RSH
- RSH can only be good for specific $\mathcal{F}$

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms
- No Free Lunch
  - 'On average all RSH perform equal.'
  - NFL holds iff $\mathcal{F}$ is c. u. p.

Consequences

- $\nexists$ general best RSH
- RSH can only be good for specific $\mathcal{F}$

Can we find limitations for all RSH for specific $\mathcal{F}$?

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms
- No Free Lunch
  - 'On average all RSH perform equal.'
  - NFL holds iff $\mathcal{F}$ is c. u. p.

Consequences

- $\nexists$ general best RSH
- RSH can only be good for specific $\mathcal{F}$

Can we find limitations for all RSH for specific $\mathcal{F}$?

If our RSH performs poorly
is it our fault or is the problem intrinsically hard?

RSH and NFL

We know

- several randomised search heuristics
  - local search
  - simulated annealing, Metropolis algorithm
  - evolutionary algorithms
- No Free Lunch
  - 'On average all RSH perform equal.'
  - NFL holds iff $\mathcal{F}$ is c. u. p.

Consequences

- $\nexists$ general best RSH
- RSH can only be good for specific $\mathcal{F}$

Can we find limitations for all RSH for specific $\mathcal{F}$?

If our RSH performs poorly
is it our fault or is the problem intrinsically hard?

$\rightsquigarrow$ complexity theory

Black Box Optimisation

Known complexity theory for 'classical algorithms'

Black Box Optimisation

Known complexity theory for 'classical algorithms'

classical algorithms	black box algorithms

Black Box Optimisation

Known complexity theory for 'classical algorithms'

classical algorithms	black box algorithms
problem class known	problem class known

Black Box Optimisation

Known complexity theory for 'classical algorithms'

classical algorithms	black box algorithms
problem class known	problem class known
problem instance known	problem instance unknown

Black Box Optimisation

Known complexity theory for 'classical algorithms'

classical algorithms	black box algorithms
problem class known	problem class known
problem instance known	problem instance unknown
problem-specific	(often) general

Black Box Optimisation

Known complexity theory for 'classical algorithms'

classical algorithms	black box algorithms
problem class known	problem class known
problem instance known	problem instance unknown
problem-specific	(often) general

Observation different optimisation scenario requires different complexity theory

Black Box Optimisation

Known complexity theory for 'classical algorithms'

classical algorithms	black box algorithms
problem class known	problem class known
problem instance known	problem instance unknown
problem-specific	(often) general

Observation different optimisation scenario requires different complexity theory

Now black box complexity
$\rightsquigarrow$ general lower bound for all black box algorithms

Notation

Let $\mathcal{F} \subseteq \{f\colon S \to V\}$ be a class of functions, A a black box algorithm for $\mathcal{F}$, x_t the t-th search point sampled by A.

Notation

Let $\mathcal{F} \subseteq \{f \colon S \to V\}$ be a class of functions, A a black box algorithm for $\mathcal{F}$, x_t the t-th search point sampled by A.

optimisation time of A on $f \in \mathcal{F}$
$T_{A,f} = \min\{t \mid f(x_t) = \max\{f(x) \in S\}\}$

Notation

Let $\mathcal{F} \subseteq \{f \colon S \to V\}$ be a class of functions, A a black box algorithm for $\mathcal{F}$, x_t the t-th search point sampled by A.

optimisation time of A on $f \in \mathcal{F}$

$T_{A,f} = \min\{t \mid f(x_t) = \max\{f(x) \in S\}\}$

worst case expected optimisation time of A on $\mathcal{F}$

$T_{A,\mathcal{F}} = \max\{\mathrm{E}(T_{A,f}) \mid f \in \mathcal{F}\}$

Notation

Let $\mathcal{F} \subseteq \{f : S \to V\}$ be a class of functions, A a black box algorithm for $\mathcal{F}$, x_t the t-th search point sampled by A.

optimisation time of A on $f \in \mathcal{F}$
$T_{A,f} = \min\{t \mid f(x_t) = \max\{f(x) \in S\}\}$

worst case expected optimisation time of A on $\mathcal{F}$
$T_{A,\mathcal{F}} = \max\{\mathrm{E}(T_{A,f}) \mid f \in \mathcal{F}\}$

black box complexity of $\mathcal{F}$
$B_{\mathcal{F}} = \min\{T_{A,\mathcal{F}} \mid A \text{ is black box algorithm for } \mathcal{F}\}$

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \leq i < j \leq n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \leq i < j \leq n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \leq i < j \leq n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard since MAX-2-SAT is contained in $\mathcal{F}$.

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \le i < j \le n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard
since MAX-2-SAT is contained in $\mathcal{F}$.

Theorem $B_{\mathcal{F}} = O(n^2)$

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \le i < j \le n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard since MAX-2-SAT is contained in $\mathcal{F}$.

Theorem $B_{\mathcal{F}} = O(n^2)$

Proof

$w_0 = f(0^n)$ (1 search point)

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \le i < j \le n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard
since MAX-2-SAT is contained in $\mathcal{F}$.

Theorem $B_{\mathcal{F}} = O(n^2)$

Proof

$w_0 = f(0^n)$ (1 search point)

$w_i = f(0^{i-1}10^{n-i}) - w_0$ (n search points)

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \le i < j \le n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard since MAX-2-SAT is contained in $\mathcal{F}$.

Theorem $B_{\mathcal{F}} = O(n^2)$

Proof

$w_0 = f\left(0^n\right)$ (1 search point)

$w_i = f\left(0^{i-1}10^{n-i}\right) - w_0$ (n search points)

$w_{i,j} = f\left(0^{i-1}10^{j-i-1}10^{n-j}\right) - w_i - w_j - w_0$ ($\binom{n}{2}$ search points)

Comparison With Computational Complexity

$$\mathcal{F} := \left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \le i < j \le n} w_{i,j} x_i x_j \right\}$$

with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard
since MAX-2-SAT is contained in $\mathcal{F}$.

Theorem $B_{\mathcal{F}} = O(n^2)$

Proof

$w_0 = f\left(0^n\right)$ (1 search point)

$w_i = f\left(0^{i-1}10^{n-i}\right) - w_0$ (n search points)

$w_{i,j} = f\left(0^{i-1}10^{j-i-1}10^{n-j}\right) - w_i - w_j - w_0$ ($\binom{n}{2}$ search points)

Compute optimal solution x^* without access to the oracle.

Comparison With Computational Complexity

$\mathcal{F} :=$
$$\left\{ f\colon \{0,1\}^n \to \mathbb{R} \mid f(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{1 \le i < j \le n} w_{i,j} x_i x_j \right\}$$
with w_i, $w_{i,j} \in \mathbb{R}$

known Optimisation of $\mathcal{F}$ is NP-hard
since MAX-2-SAT is contained in $\mathcal{F}$.

Theorem $B_{\mathcal{F}} = O(n^2)$

Proof
$w_0 = f(0^n)$ (1 search point)
$w_i = f\left(0^{i-1}10^{n-i}\right) - w_0$ (n search points)
$w_{i,j} = f\left(0^{i-1}10^{j-i-1}10^{n-j}\right) - w_i - w_j - w_0$ ($\binom{n}{2}$ search points)
Compute optimal solution x^* without access to the oracle.
$f(x^*)$ (1 search point)
together $\binom{n}{2} + n + 2 = O(n^2)$ search points □

From Functions to Classes of Functions

Observation $\forall \mathcal{F}\colon B_{\mathcal{F}} \leq |\mathcal{F}|$

From Functions to Classes of Functions

Observation $\forall \mathcal{F}\colon B_{\mathcal{F}} \leq |\mathcal{F}|$

Consequence $B_{\{f\}} = 1$ for any f — pointless

From Functions to Classes of Functions

Observation $\forall \mathcal{F}\colon B_{\mathcal{F}} \leq |\mathcal{F}|$

Consequence $B_{\{f\}} = 1$ for any f — pointless

Can we still have meaningful results for our example functions?

From Functions to Classes of Functions

Observation $\forall \mathcal{F}$: $B_{\mathcal{F}} \leq |\mathcal{F}|$

Consequence $B_{\{f\}} = 1$ for any f — pointless

Can we still have meaningful results for our example functions?

Randomised search heuristics are often symmetric
with respect to 0s and 1s.

From Functions to Classes of Functions

Observation $\forall \mathcal{F}\colon B_{\mathcal{F}} \leq |\mathcal{F}|$

Consequence $B_{\{f\}} = 1$ for any f — pointless

Can we still have meaningful results for our example functions?

Randomised search heuristics are often symmetric
with respect to 0s and 1s.

Definition For $f\colon \{0,1\}^n \to \mathbb{R}$, we define $f^* := \{f_a \mid a \in \{0,1\}^n\}$
where $f_a(x) := f(a \oplus x)$.

From Functions to Classes of Functions

Observation $\forall \mathcal{F}\colon B_{\mathcal{F}} \leq |\mathcal{F}|$

Consequence $B_{\{f\}} = 1$ for any f — pointless

Can we still have meaningful results for our example functions?

Randomised search heuristics are often symmetric
with respect to 0s and 1s.

Definition For $f\colon \{0,1\}^n \rightarrow \mathbb{R}$, we define $f^* := \{f_a \mid a \in \{0,1\}^n\}$
where $f_a(x) := f(a \oplus x)$.

Clearly, such RSHs perform equal on all $f' \in f^*$.

An Example: Needle

Definition

$$\textsc{Needle}\colon \{0,1\}^n \to \{0,1\}$$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x[i]$$

An Example: NEEDLE

Definition

$$\textsc{Needle}\colon \{0,1\}^n \to \{0,1\}$$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x[i] = \begin{cases} 1 & \text{if } x = 1^n \\ 0 & \text{otherwise} \end{cases}$$

An Example: Needle

Definition $\textsc{Needle}\colon \{0,1\}^n \to \{0,1\}$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x[i] = \begin{cases} 1 & \text{if } x = 1^n \\ 0 & \text{otherwise} \end{cases}$$

Consider $\textsc{Needle}^*$

An Example: Needle

Definition

$$\textsc{Needle}\colon \{0,1\}^n \to \{0,1\}$$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x[i] = \begin{cases} 1 & \text{if } x = 1^n \\ 0 & \text{otherwise} \end{cases}$$

Consider $\textsc{Needle}^*$

Remember

$$f^* = \{f_a \mid a \in \{0,1\}^n\}$$

$$f_a(x) = f(a \oplus x)$$

An Example: Needle

Definition $\textsc{Needle}\colon \{0,1\}^n \to \{0,1\}$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x[i] = \begin{cases} 1 & \text{if } x = 1^n \\ 0 & \text{otherwise} \end{cases}$$

Consider $\textsc{Needle}^*$

Remember $f^* = \{f_a \mid a \in \{0,1\}^n\}$

$f_a(x) = f(a \oplus x)$

By Definition $\textsc{Needle}^* = \{\textsc{Needle}_a \mid a \in \{0,1\}^n\}$,

$\textsc{Needle}_a(x) = \textsc{Needle}(a \oplus x)$

An Example: Needle

Definition

$$\textsc{Needle}\colon \{0,1\}^n \to \{0,1\}$$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x[i] = \begin{cases} 1 & \text{if } x = 1^n \\ 0 & \text{otherwise} \end{cases}$$

Consider $\textsc{Needle}^*$

Remember

$$f^* = \{f_a \mid a \in \{0,1\}^n\}$$

$$f_a(x) = f(a \oplus x)$$

By Definition

$$\textsc{Needle}^* = \{\textsc{Needle}_a \mid a \in \{0,1\}^n\},$$

$$\textsc{Needle}_a(x) = \textsc{Needle}(a \oplus x)$$

$$\textsc{Needle}_a(x) = \begin{cases} 1 & \text{if } x = \overline{a} \\ 0 & \text{otherwise} \end{cases}$$

A General Upper Bound

Theorem

For any $\mathcal{F} \subseteq \{f\colon \{0,1\}^n \to \mathbb{R}\}$, $B_{\mathcal{F}} \leq 2^{n-1} + 1/2$ *holds.*

A General Upper Bound

Theorem

For any $\mathcal{F} \subseteq \{f \colon \{0,1\}^n \to \mathbb{R}\}$, $B_{\mathcal{F}} \leq 2^{n-1} + 1/2$ *holds.*

Proof

Consider pure random search without re-sampling of search points.

A General Upper Bound

Theorem

For any $\mathcal{F} \subseteq \{f\colon \{0,1\}^n \to \mathbb{R}\}$, $B_{\mathcal{F}} \leq 2^{n-1} + 1/2$ *holds.*

Proof

Consider pure random search without re-sampling of search points.
For each step t, $\mathrm{Prob}\left(\text{find global optimum}\right) \geq 2^{-n}$.

A General Upper Bound

Theorem

For any $\mathcal{F} \subseteq \{f \colon \{0,1\}^n \to \mathbb{R}\}$, $B_{\mathcal{F}} \leq 2^{n-1} + 1/2$ *holds.*

Proof

Consider pure random search without re-sampling of search points.
For each step t, $\mathrm{Prob}\,(\text{find global optimum}) \geq 2^{-n}$.

$$B_{\mathcal{F}} \leq \sum_{i=1}^{2^n} i \cdot 2^n$$

A General Upper Bound

Theorem

For any $\mathcal{F} \subseteq \{f\colon \{0,1\}^n \to \mathbb{R}\}$, $B_{\mathcal{F}} \leq 2^{n-1} + 1/2$ *holds.*

Proof

Consider pure random search without re-sampling of search points.
For each step t, $\mathrm{Prob}\,(\text{find global optimum}) \geq 2^{-n}$.

$B_{\mathcal{F}} \leq \sum_{i=1}^{2^n} i \cdot 2^n$

$= \frac{2^n(2^n+1)}{2^{n+1}} = 2^{n-1} + \frac{1}{2}$ □

Summary & Take Home Message

Things to remember

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.
- black-box complexity

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.
- black-box complexity
- generalisation f^*

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.
- black-box complexity
- generalisation f^*

Take Home Message

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.
- black-box complexity
- generalisation f^*

Take Home Message

- When making too general statements about randomised search heuristics the statements are either trivial or false.

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.
- black-box complexity
- generalisation f^*

Take Home Message

- When making too general statements about randomised search heuristics the statements are either trivial or false.
- One needs to consider the class of optimisation problems to achieve good performance.

Summary & Take Home Message

Things to remember

- ANFL: Still there is no generally good search heuristic.
- black-box complexity
- generalisation f^*

Take Home Message

- When making too general statements about randomised search heuristics the statements are either trivial or false.
- One needs to consider the class of optimisation problems to achieve good performance.
- Black-box complexity allows for meaningful general lower bounds for RSHs.