

Extended Probe Method for Linkage Discovery over High-cardinality Alphabets

Shude Zhou
Department of Computer
Science and Technology
Tsinghua University
Beijing, 100084, China
zsd03@mails.tsinghua.edu.cn

Zengqi Sun
Department of Computer
Science and Technology
Tsinghua University
Beijing, 100084, China

Robert B. Heckendorn
Department of Computer
Science
University of Idaho
Moscow, ID83844-1010, USA
heckendo@uidaho.edu

ABSTRACT

The work addresses the problem of identifying the epistatic linkage of a function from high cardinality alphabets to the real numbers. It is a generalization of Heckendorn and Wright's work that restricts problem representation into the binary-string domain. Discrete Fourier transform is used to analyze underlying structure in high-cardinality alphabets space. Boolean operators are replaced with new operators such as \oplus, \ominus, \otimes and so on in high cardinality alphabets. The "probe" formulation is redesigned to determine epistatic properties of non-binary function. Theoretical analysis shows the close relationship between probe value and problem structure. A deterministic and a stochastic algorithm based on properties of probes are proposed to completely identify the linkage structure and rigorously compute all Fourier coefficients. Some discussion about linkage detection for continuous problems is given.

Categories and Subject Descriptors: I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, Search; F.2 [Analysis of Algorithms and Problem Complexity]: General

General Terms: Algorithms, Theory

Keywords: epistasis, linkage detection, high-cardinality probe, Fourier transform

1. BACKGROUND AND MOTIVATION

The importance of linkage information (nonlinear interaction relationship between variables) in optimization algorithms has been studied in the evolutionary computation field [1, 3, 11, 10, 14]. Linkage information is critical for designing scalable and competent evolutionary algorithms [16]. For many real-world optimization problems, the interaction structure is actually unknown, and identifying these unknown interaction structure is an important and interesting research topic. Messy GA [2], gene expression messy GA [7] and LLGA [3] manipulate the representation of solutions

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

to make the interacting components of partial solutions less likely to be disrupted. EDAs [11, 10, 14] deal with linkage information by the estimated probabilistic graphical model. Another research line as described in details in next paragraph is on methods to directly detect the linkage information before evolutionary search.

There has been extensive previous work that address the linkage detection problem in binary-coded domain. Munetomo and Goldberg proposed LINC and LIMD algorithms to identify linkage structure directly by performing perturbations between a pair of loci to detect monotonicity/non-monotonicity of fitness changes [12]. Kargupta et al. demonstrated that for an epistatically bounded function, i.e. the size of the epistatic subsets is bounded, all the Walsh coefficients could be computed in a polynomial number of function evaluations [8]. In 2002, Heckendorn introduced embedded landscapes as extension of *NK* landscapes and *MAXSAT* problems [4] and gave theoretical analysis about the relationship between Walsh coefficients and epistases. In 2004, Heckendorn and Wright further built the mathematical foundations for predicting epistasis of binary functions [5]. Walsh coefficients completely describe the function and so completely characterize the epistatic linkage. Probe methods are randomized algorithms for linkage detection. They extended the previous work by Munetomo et al. and Kargupta et al. by proposing a framework of probe theory and algorithms [5].

The paper addresses the problem of detecting epistasis in a **high-cardinality** domain. It is a generalization of Heckendorn and Wright's work [5] which restricts problem representation into the binary strings. After introducing some basic mathematical notations in Section 2, a general theoretical framework for probe methods is developed based on discrete Fourier transform of high-cardinality function in Section 3 and 4. Section 5, 6 and 7 propose a top-down deterministic and a bottom-up stochastic algorithm to determine the linkage structure and accurately calculate the Fourier coefficients. An artificial Needle-in-Haystack function is used to verify the algorithms in Section 8. The contributions and shortcomings of the work are discussed at last, and some promising research directions are given.

2. NOTATIONS

Below are some basic notations and definitions.

1. $\widetilde{M} = \{0, 1, \dots, M - 1\}$, where M is a positive integer that $M \geq 2$. The space \widetilde{M}^L is the space of L dimen-

sional vectors with each component selected from \widetilde{M} . Specially, $B = \{0, 1\}$.

2. $x, y \in \widetilde{M}^L$, the **supplement** of x is denoted by \bar{x} : $\forall k = 1, \dots, L, \bar{x}_k = \text{mod}_M(M - x_k)$, the operator \oplus is defined $x \oplus y = (\text{mod}_M(x_1 + y_1), \dots, \text{mod}_M(x_L + y_L))$, the operator \ominus is defined as $x \ominus y = x \oplus \bar{y}$, the operator \otimes is defined $x \otimes y = (\text{mod}_M(x_1 y_1), \dots, \text{mod}_M(x_L y_L))$, dot product is defined as $x \cdot y = \text{mod}_M(\bigoplus_i (x_i \otimes y_i))$. $\|x\|$ denotes the number of positions with nonzero value. We say $x \subseteq y$, if and only if: $\forall x_i, x_i = y_i$ if $y_i \neq 0$; $x \subset y$, iff $x \subseteq y$ and $x \neq y$.
3. **Template set** $S(x)$ is defined over $x \in \widetilde{M}^L$ such that: $S(x) = \{y | y \in \widetilde{M}^L \wedge (x_k = 0 \rightarrow y_k = 0)\}$. For $\forall x \in \widetilde{M}^L, \|x\| \neq 0$, there are $M^{\|x\|}$ elements in $S(x)$; if $\|x\| = 0$, then $S(x) = \emptyset$.
4. A function $\text{pack}(x, y)$ is defined as: $\widetilde{M}^L \times \widetilde{M}^L \mapsto \widetilde{M}^H$, where H, L are positive integers, and H is the number of nonzero elements in y , $H \leq L$. $\text{pack}(x, y)$ is the vector composed of elements in x that are in the same positions as the elements of y that are nonzero. Order is preserved. For example, $\text{pack}((0, 2, 3, 0, 1, 2), (0, 1, 0, 1, 1, 0)) = (2, 0, 1)$.
5. A function $\text{zero}(x): \widetilde{M}^L \mapsto B^L$ is defined as: if $y = \text{zero}(x)$, then $y_i = 1$ if $x_i = 0$ and $y_i = 0$ if $x_i \neq 0$.

Note that, if we restrict $\widetilde{M} = B$, the operator \otimes is equivalent to boolean operator *and* \wedge , \oplus and \ominus are both equivalent to *exclusive-or*. It should be also noted that the definition of **supplement** in high-cardinality domain is very different from that in binary domain, and can not be directly reduced to the definition in binary domain. In binary domain, the **supplement** of $x, x \in B$ is usually defined as: $\bar{x} = \text{zero}(x)$; clearly this type of definition does not apply to the general discrete domain.

3. HIGH-CARDINALITY FOURIER ANALYSIS

Original Walsh function[5, 8, 4] can not be applied to analyze the high-cardinality domain, so we have to extend it to discrete Fourier function. Some useful properties of Fourier functions are given.

Discrete Fourier basis function in domain \widetilde{M}^L is defined[13, 6, 9, 17]

$$\psi_i^{(M)}(x) = v_M(i \cdot x) \quad (1)$$

where $i, x \in \widetilde{M}^L$, and $v_M(a) = e^{\frac{2\pi\sqrt{-1}}{M}a}$ where $a \in \widetilde{M}$. Its complex conjugate is denoted by $\bar{\psi}_i^{(M)}(x) = v_M(\bar{i} \cdot x)$. The following properties about discrete Fourier functions are important. Theorem 1~4 are obvious or have been proved recently[17, 6]; Theorem 5 is a new theorem.

THEOREM 1. (Sum of Power of Roots of One)

$$\sum_{b=0}^{M-1} v_M(a \otimes b) = \begin{cases} M & \text{if } a = 0 \\ 0 & \text{if } a \neq 0 \end{cases} \quad (2)$$

THEOREM 2. (Basic Properties of Discrete Fourier Function) For $p, q, x \in \widetilde{M}^L$,

$$\psi_p^{(M)}(x) = \psi_x^{(M)}(p) \quad (3)$$

$$\psi_p^{(M)}(x)\psi_q^{(M)}(x) = \psi_{p \oplus q}^{(M)}(x) \quad (4)$$

$$\psi_{\bar{0}}^{(M)}(x) = \psi_x^{(M)}(\bar{0}) = 1 \quad (5)$$

THEOREM 3. (Inverse Fourier) For $p, q, x \in \widetilde{M}^L$,

$$\sum_{x \in \widetilde{M}^L} \bar{\psi}_p^{(M)}(x)\psi_q^{(M)}(x) = \begin{cases} M^L & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases} \quad (6)$$

THEOREM 4. (Balance Sum Theorem for High-cardinality Hyperplane) For $x, i, m \in \widetilde{M}^L, \|m\| = H, j \in \widetilde{M}^H$,

$$\sum_{x: \text{pack}(x, m) = j} \psi_x^{(M)}(i) = \begin{cases} M^{L-H} \psi_{\text{pack}(i, m)}^{(M)}(j) & \text{if } i \in S(m) \\ 0 & \text{if } i \notin S(m) \end{cases} \quad (7)$$

THEOREM 5. (Inverse Fourier for High-cardinality Hyperplanes) For $p, q, x, m \in \widetilde{M}^L, \|m\| = H$,

$$\sum_{x \in S(m)} \bar{\psi}_p^{(M)}(x)\psi_q^{(M)}(x) = \begin{cases} M^H & \text{if } \text{pack}(p, m) = \text{pack}(q, m) \\ 0 & \text{if } \text{pack}(p, m) \neq \text{pack}(q, m) \end{cases} \quad (8)$$

PROOF.

$$\begin{aligned} & \sum_{x \in S(m)} \bar{\psi}_p^{(M)}(x)\psi_q^{(M)}(x) \\ &= \sum_{x \in S(m)} \overline{v_M(p \cdot x)} v_M(q \cdot x) \\ &= \sum_{x \in S(m)} \overline{v_M(\text{pack}(p, m) \cdot \text{pack}(x, m))} v_M(\text{pack}(q, m) \cdot \text{pack}(x, m)) \\ &= \sum_{u \in \widetilde{M}^H} \overline{v_M(\text{pack}(p, m) \cdot u)} v_M(\text{pack}(q, m) \cdot u) \\ &= \sum_{u \in \widetilde{M}^H} \bar{\psi}_{\text{pack}(p, m)}^{(M)}(u)\psi_{\text{pack}(q, m)}^{(M)}(u) \text{ Note: } \psi\text{s here are in } \widetilde{M}^H \\ &= \begin{cases} M^H & \text{if } \text{pack}(p, m) = \text{pack}(q, m) \\ 0 & \text{if } \text{pack}(p, m) \neq \text{pack}(q, m) \end{cases} \quad \text{By Theorem 3} \end{aligned}$$

□

Any function $f: \widetilde{M}^L \mapsto \mathbb{R}$ can be written as a linear combination of discrete Fourier functions:

$$f(x) = \sum_{i \in \widetilde{M}^L} \omega_i \psi_i(x) \quad (9)$$

where ω_i is the **Fourier coefficients**. The Fourier coefficients can be calculated:

$$\omega_i = \frac{1}{M^L} \sum_{x \in \widetilde{M}^L} f(x) \bar{\psi}_i(x) \quad (10)$$

The **generalized embedded landscape (GEL)** is a function $f: \widetilde{M}^L \mapsto \mathbb{R}$ which can be written as a sum of subfunctions each of which depends only on a small number of variables. A GEL is **k -bounded epistatic** if it can be written as the sum of subfunction each of whose number of variables is at most k . It has recently shown the close relationship between Fourier coefficients and the structure of k -bounded GEL [17, 18].

THEOREM 6. (*Generalized Embedding Theorem*) A function $f : \widetilde{M}^L \mapsto \mathbb{R}$ has k -bounded epistasis, if and only if $\omega_j = 0 \forall \|j\| > k$.

4. HIGH-CARDINALITY PROBES

In binary domain, a probe is a way of determining epistatic properties of a function $f : B^L \mapsto \mathbb{R}$ by performing a series of specific function evaluations[5]. Here, we would like to re-describe the definition of ‘‘probe’’ in the language of section 2’s notations, so that we can extend it to high-cardinality domain more naturally:

$$P(f, m, c) = \frac{1}{2^{\|m\|}} \sum_{i \in S(m)} (-1)^{\|i\|} f(i \oplus c) \quad (11)$$

where $m \in B^L$ and $c \in S(\text{zero}(m))$.

It is not straightforward to generalize the above formulation to a domain of high-cardinality alphabets. But we find that in binary domain $\forall i \in S(m)$, $(-1)^{\|i\|} = (-1)^{\|i \oplus m\|} = \psi_m^{(B)}(i) = \bar{\psi}_m^{(B)}(i)$. So, we define **probe** in high-cardinality domain as:

$$P(f, m, c) = \frac{1}{M^{\|m\|}} \sum_{i \in S(m)} \bar{\psi}_m^{(M)}(i) f(i \oplus c) \quad (12)$$

where $m \in \widetilde{M}^L$ and $c \in S(\text{zero}(m))$. c is called the **background** of the probe. **Complex conjugate of a probe** is defined below and we will use **conjugate probe** for short.

$$\bar{P}(f, m, c) = P(f, \bar{m}, c) = \frac{1}{M^{\|m\|}} \sum_{i \in S(m)} \psi_m^{(M)}(i) f(i \oplus c) \quad (13)$$

The **order of the probe** is number of nonzeros in m . The direct computation of the value of a probe requires $M^{\|m\|}$ function evaluations. Later theoretical induction about the properties of the high-cardinality probe will show the validity of this extension.

4.1 Properties of Probes

THEOREM 7. (*Fourier Function Probing*) For any $j, m \in \widetilde{M}^L$ and $c \in S(\text{zero}(m))$

$$P(\psi_j, m, c) = \begin{cases} \psi_j(c) & \text{if } j \subseteq m \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

PROOF.

$$\begin{aligned} P(\psi_j, m, c) &= \frac{1}{M^{\|m\|}} \sum_{i \in S(m)} \bar{\psi}_m^{(M)}(i) \psi_j^{(M)}(i \oplus c) \\ &= \frac{1}{M^{\|m\|}} \sum_{i \in S(m)} \bar{\psi}_m^{(M)}(i) \psi_j^{(M)}(i) \psi_j^{(M)}(c) \\ &= \frac{1}{M^{\|m\|}} \psi_j^{(M)}(c) \sum_{i \in S(m)} \bar{\psi}_m^{(M)}(i) \psi_j^{(M)}(i) \end{aligned}$$

By Theorem of Inverse Fourier for High-cardinality Hyperplanes, we know that the sum is $M^{\|m\|}$ if $\text{pack}(m, m) = \text{pack}(j, m)$, that is $j \subseteq m$, and is 0 otherwise. \square

THEOREM 8. (*High-cardinality Probe Subset*) For any $m \in \widetilde{M}^L$ and $c \in S(\text{zero}(m))$

$$P(f, m, c) = \sum_{j \subseteq m} \omega_j \psi_j(c) \quad (15)$$

PROOF. Recall that $f(x) = \sum_{j \in \widetilde{M}^L} \omega_j \psi_j(x)$. Thus,

$$\begin{aligned} P(f, m, c) &= \sum_{j \in \widetilde{M}^L} \omega_j P(\psi_j(x), m, c) \\ &= \sum_{j \subseteq m} \omega_j \psi_j(c) \quad \text{By Theorem 7} \end{aligned}$$

\square

A **maximal nonzero Fourier coefficient** is a Fourier coefficient ω_m such that $\omega_m \neq 0$ and $\omega_j = 0 \forall j \subset m$.

THEOREM 9. (*Maximal Probe*) If ω_m is a maximal nonzero Fourier coefficient, then for any $c \in S(\text{zero}(m))$,

$$P(f, m, c) = \omega_m \quad (16)$$

PROOF. It follows from High-cardinality Probe Subset Theorem that

$$\begin{aligned} P(f, m, c) &= \sum_{j \subseteq m} \omega_j \psi_j(c) \\ &= \omega_m \psi_m(c) + \sum_{j \subset m} 0 \psi_j(c) = \omega_m \psi_m(c) \end{aligned}$$

And $\psi_m(c) = v_M(m \cdot c) = v_M(0) = 1$. \square

In binary domain, a probe can be written as a sum of lower-order probes. We check whether this property still holds in high-cardinality domain.

THEOREM 10. (*High-cardinality Probe Recursion*) For any function $f : \widetilde{M}^L \mapsto \mathbb{R}$, and $m, n \in \widetilde{M}^L$ with $m \subseteq n$, and any $c \in S(\text{zero}(m))$:

$$P(f, m, c) = \frac{1}{M^{\|n\|}} \sum_{i \in S(n)} \bar{\psi}_n^{(M)}(i) P(f, m \oplus n, i \oplus c) \quad (17)$$

PROOF. Any $j \in S(m)$ can be written uniquely as $j = i \oplus u$ where $i \in S(n)$ and $u \in S(m \oplus n)$. Thus:

$$\begin{aligned} P(f, m, c) &= \frac{1}{M^{\|m\|}} \sum_{j \in S(m)} \bar{\psi}_m^{(M)}(j) f(j \oplus c) \\ &= \frac{1}{M^{\|m\|}} \sum_{(i \oplus u) \in S(m)} \bar{\psi}_m^{(M)}(i \oplus u) f(i \oplus u \oplus c) \\ &= \frac{1}{M^{\|n\|} M^{\|m \oplus n\|}} \sum_{i \in S(n)} \sum_{u \in S(m \oplus n)} \bar{\psi}_m^{(M)}(i) \bar{\psi}_m^{(M)}(u) f(i \oplus u \oplus c) \\ &= \frac{1}{M^{\|n\|}} \sum_{i \in S(n)} \bar{\psi}_m^{(M)}(i) \frac{1}{M^{\|m \oplus n\|}} \sum_{u \in S(m \oplus n)} \bar{\psi}_m^{(M)}(u) f(i \oplus u \oplus c) \\ &= \frac{1}{M^{\|n\|}} \sum_{i \in S(n)} \bar{\psi}_m^{(M)}(i) \left(\frac{1}{M^{\|m \oplus n\|}} \sum_{u \in S(m \oplus n)} \bar{\psi}_{(m \oplus n)}^{(M)}(u) f(u \oplus (i \oplus c)) \right) \end{aligned}$$

(Since $(i \oplus c) \in S(\text{zero}(m \oplus n))$, by definition of probe:)

$$\begin{aligned} &= \frac{1}{M^{\|n\|}} \sum_{i \in S(n)} \bar{\psi}_m^{(M)}(i) P(f, m \oplus n, i \oplus c) \\ &= \frac{1}{M^{\|n\|}} \sum_{i \in S(n)} \bar{\psi}_n^{(M)}(i) P(f, m \oplus n, i \oplus c) \end{aligned}$$

\square

Note that the above extension of Probe Recursion Theorem from binary domain to high-cardinality domain is not straightforward. The key point is that the operator \ominus is brought into to replace the original operator “exclusive or” somewhere. The special design of n that $m \subseteq n$ makes it possible that $\forall j \in S(m)$ can be written uniquely as $j = i \oplus u$ where $i \in S(n)$ and $u \in S(m \ominus n)$, so that we can decompose the probe into additive low-order probes. Further we can derive the following theorem. The description is also different from that in binary domain.

THEOREM 11. (Nonzero High-cardinality Probe Existence) *Given a maximal nonzero Fourier coefficient ω_m , then $\forall a : m \subseteq a$, there exists an $i \in S(m \ominus a)$ such that*

$$P(f, a, i \oplus c) \neq 0 \quad \forall c \in S(\text{zero}(m)) \quad (18)$$

PROOF. By the Maximal Probe Theorem, $P(f, m, c) = \omega_m \neq 0$ for any $c \in S(\text{zeros}(m))$. We know that $\forall a : m \subseteq a$, there exists unique n such that $(m \subseteq n) \wedge (a = m \ominus n)$. By High-cardinality Probe Recursion Theorem, concerning the n , we have:

$$P(f, m, c) = \frac{1}{M^{\|n\|}} \sum_{i \in S(n)} \bar{\psi}_m^{(M)}(i) P(f, m \ominus n, i \oplus c)$$

Thus, there must exist an $i \in S(n)$, $S(n) = S(m \ominus a)$, such that $P(f, m \ominus n, i \oplus c) = P(f, a, i \oplus c) \neq 0$. \square

Nonzero High-cardinality Probe Existence Theorem implies the close relationship between the problem linkage structure and the probe values, which are summarized in the following corollary.

COROLLARY 12. (Relationship Between Linkage Structure and Probes) *Given any function $f : \widetilde{M}^L \mapsto \mathbb{R}$ and $m \in \widetilde{M}^L$, the variable positions whose values are nonzero in m are linkage epistatic if and only if:*

$$\exists c \in S(\text{zero}(m)), P(f, m, c) \neq 0 \quad (19)$$

5. LINKAGE HYPERGRAPH AND NON-ZERO FOURIER COEFFICIENTS

For the convenience of description, we use the concept hypergraph[5] to represent the interaction relationship between variables. Generally, a **hypergraph** G is defined as a pair (V, E) , where V is a set of **vertices**, and $E \subseteq 2^V$ is a set of **hyperedges** between the vertices. The **linkage hypergraph** of a fitness function f is denoted by $G_f = G_f(V, E)$ where the vertex set $V = \{x_1, x_2, \dots, x_L\}$ corresponds the set of variables (alphabet string positions), and $e \in E$ if and only if there are linkage relationship between those variables included in e . The **order of a hyperedge** e is the number of vertices in e ; the **order of a hypergraph** G_f is maximum of the orders of e 's. For example, considering a function $f(x_1, x_2, x_3) = g_1(x_1, x_2) + g_2(x_3)$, where g_1 and g_2 are maximum epistasis, the vertices set of G_f is $V = \{x_1, x_2, x_3\}$, hyperedge set $E = \{\Phi, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}\}$, and the order of hyperedge $\{x_1\}$ is 1, the order of G_f is 2. Below are the linkage hypergraphs of some specified functions.

THEOREM 13. (Linkage Hypergraph of Some Specified Functions)

1. If $f : \widetilde{M}^L \mapsto \{c\}$, where $c \in \mathbb{R}$ is constant, then $G_f = (V, \Phi)$.

2. If $f : \widetilde{M}^L \mapsto \mathbb{R}$ is linear function (i.e. order-1 epistatic), then $G_f = (V, E)$, where $E = \{\Phi, \{x_1\}, \{x_2\}, \dots, \{x_L\}\}$.

3. If function $f : \widetilde{M}^L \mapsto \mathbb{R}$ is maximal epistasis, then $G_f = (V, E)$, where $E = 2^V$.

4. If $f : \widetilde{M}^L \mapsto \mathbb{R}$ is k -bounded epistatic function, then $G_f = (V, E)$, where the order of G_f is k .

From the Theorem of Relationship Between Linkage Structure and Probes, we should notice that: a hyperedge e corresponds to at least one mask string $m \in \widetilde{M}^L$ in which $m_i \neq 0$ if $x_i \in e$ and $m_i = 0$ if $x_i \notin e$, and m has the property $\exists c \in S(\text{zero}(m)), P(f, m, c) \neq 0$. Thus, by Theorem of Nonzero High-cardinality Probe Existence, we have the following corollary.

COROLLARY 14. (Linkage Hypergraph Property) *Given $f : \widetilde{M}^L \mapsto \mathbb{R}$ and its linkage hypergraph $G_f = G_f(V, E)$, if $a \in 2^V$ is a hyperedge (i.e. $a \in E$), then $\forall e \in 2^a, e \in E$.*

In binary domain, each $e \in E$ corresponds a unique mask string $m \in B^L$; but in high-cardinality ($M > 2$) alphabets domain, each e might correspond many strings, and those strings have the property: if m_1 and m_2 are two different M -cardinality strings corresponding to e , then $\text{zero}(m_1) = \text{zero}(m_2)$. We should also note the observation that for any $m \in \widetilde{M}^L$ and $c \in S(\text{zero}(m))$, if $p(f, m, c) \neq 0$, then $p(f, \bar{m}, c) \neq 0$ (This is easily seen from the definition of probe). Therefore, in high-cardinality domain, if $m \neq \bar{m}$, one non-empty hyperedge corresponds at least two mask strings of nonzero probe value.

Now, we consider how to identify the order- j hyperedges of a given function. In the following description, we will not distinguish the denotations of a hyperedge and its corresponding **binary** mask string, because there exists one-to-one mapping between them. The Order- j Linkage Detection Algorithm in Figure 1 can construct the set of order- j hyperedges of the linkage hypergraph for problems in high-cardinality domain.

Detect-Linkage(j, N)

```

V ← {0, 1, ..., L - 1}, E ← Φ
for each possible hyperedge e ∈ BL with ||e|| = j
  if e ∉ E then
    sign ← 0
    for each m ∈  $\widetilde{M}^L$  that zero(m) = zero(e)
      for i ← 1 to N do
        c ← a random string in S(zero(m))
        if P(f, m, c) ≠ 0 then
          E ← E ∪ {e}
          sign ← 1
          break
      endif
    endfor
  endif
endfor
return E.

```

Figure 1: Order- j linkage detection algorithm

Next, we give the computation complexity of detecting all order- j mask strings that correspond to order- j hyperedges. The reason that the complexity analysis is focused on mask strings instead of hyperedges is that in Section 7 we will propose a stochastic algorithm that can compute all Fourier coefficients, which need to detect order- j mask strings of nonzero probe values. The Detect-Nonzero-Probe-Algorithm is shown in Figure 2. Considering a class of k -epistatically bounded fitness functions where the number of hyperedges is $O(L)$, how many function evaluations are required? Theoretical results about the question in binary domain are given in [5]. Their results can be extended to the high-cardinality domain.

```

Detect-Nonzero-Probe( $j, N$ )
   $T \leftarrow \Phi$ 
  for each mask strings  $m \in \widetilde{M}^L$  with  $\|m\| = j$ 
    if  $m \notin T$  then
      for  $i \leftarrow 1$  to  $N$  do
         $c \leftarrow$  a random string in  $S(\text{zero}(m))$ 
        if  $P(f, m, c) \neq 0$  then
           $T.add(m)$ 
          break
        endif
      endif
    endif
  endfor
return  $T$ .

```

Figure 2: Detect order- j mask strings with nonzero probe value that correspond to hyperedges

THEOREM 15. (Nonzero Probe Probability) Let $f : \widetilde{M}^L \mapsto \mathbb{R}$ be k -epistatically bounded and let $m \in \widetilde{M}^L$ have the property: $\|m\| = j, j \leq k$. If c is a randomly chosen string in $S(\text{zero}(m))$, then the probability that $P(f, m, c) \neq 0$ is at least M^{j-k} .

PROOF. Since $P(f, m, c) \neq 0$, by the Theorem of Nonzero High-cardinality Probe Existence, there is a u such that $u \subseteq m$ and $\omega_u \neq 0$. Without loss of generality we can assume that u has the property that $v \subset u \Rightarrow \omega_v = 0$. By assumption, $\|u\| \leq k$. The Nonzero High-cardinality Probe Existence Theorem shows that there is at least one $i \in S(u \oplus m)$ such that $P(f, m, i \oplus b) \neq 0$ for any $b \in S(\text{zero}(u))$. The probability that the randomly selected background c matches some such i on the positions masked by $u \oplus m$ is at least $M^{-\|u \oplus m\|} = M^{\|m\| - \|u\|} \geq M^{j-k}$. \square

THEOREM 16. Let $f : \widetilde{M}^L \mapsto \mathbb{R}$ be k -epistatically bounded and let J be the number of order- j mask strings with nonzero probe value. If the number of iterations N in the Algorithm of Figure 2 is chosen so that either

$$N \geq \begin{cases} \frac{\ln(1 - \delta^{1/J})}{\ln(1 - M^{j-k})} & \text{if } j < k \\ 1 & \text{if } j = k \end{cases} \quad (20)$$

or

$$N \geq \begin{cases} M^{k-j} \ln(1 - \delta^{1/J}) & \text{if } j < k \\ 1 & \text{if } j = k \end{cases}$$

then the probability that all order- j mask strings of nonzero probe value are detected is at least δ .

PROOF. Theorem of Nonzero Probe Probability shows that the probability of failure for one probe on one trial is at most $1 - M^{j-k}$. Thus, the probability of failure on N independent trials is at most $(1 - M^{j-k})^N$, and the probability of success on N trials is at least $1 - (1 - M^{j-k})^N$. The probability of success on all J hyperedges is at least $(1 - (1 - M^{j-k})^N)^J$. Thus, we want to choose N so that

$$\begin{aligned} (1 - (1 - M^{j-k})^N)^J &\geq \delta \\ 1 - \delta^{1/J} &\geq (1 - M^{j-k})^N \\ \ln(1 - \delta^{1/J}) &\geq N \ln(1 - M^{j-k}) \\ \frac{\ln(1 - \delta^{1/J})}{\ln(1 - M^{j-k})} &\leq N \end{aligned}$$

To prove the second formula, we need to show that

$$\begin{aligned} -M^{k-j} \ln(1 - \delta^{1/J}) &\geq \frac{\ln(1 - \delta^{1/J})}{\ln(1 - M^{j-k})} \\ \Leftrightarrow M^{k-j} &\geq -\frac{1}{\ln(1 - M^{j-k})} \\ \Leftrightarrow M^{j-k} &\leq -\ln(1 - M^{j-k}) \end{aligned}$$

\square

Note the difference from the binary domain: in \widetilde{M}^L domain, for each order- j hyperedge, we have to do at most $(M - 1)^j N$ probes, so that the probability that all order- j strings of nonzero probe value are detected is at least δ ; in binary domain, we only need to do at most N probes. We next consider how the number N of iterations increases as the strings length increases for a class of high-cardinality fitness functions.

THEOREM 17. Assume a class of k -epistatically bounded fitness function $\widetilde{M}^L \mapsto \mathbb{R}$ where the number of maximal hyperedges is $O(L)$. If δ is constant, the number of function evaluations required by Algorithm of Figure 2 is $O(M^k \binom{L}{j} (\ln L + \ln \binom{k}{j}))$. If j is constant, then the number of function evaluations is $O(M^k L^j \ln L)$.

PROOF. We set N to be $-M^{k-j} \ln(1 - \delta^{1/J})$. For each of the N iterations of the inner loop a probe needs M^j function evaluations. The outer loop is executed $\binom{L}{j}$ times, so the total number of function evaluations is $-M^k \binom{L}{j} \ln(1 - \delta^{1/J})$, and by Lemma 10[5], this is $O(M^k \binom{L}{j} \ln J)$.

The number of order- j hyperedges in a single maximal Fourier coefficient of order k is bounded by $\binom{k}{j}$, and we have assumed that the number of maximal order- k hyperedges is $O(L)$ (that is, the number of nonzero Fourier coefficients is at most $O((M - 1)^k L)$), so J is bounded by $O(\binom{k}{j} (M - 1)^k L)$. Since we assume that M and k are fixed number independent of L , thus, $\ln J$ is $O(\ln L + \ln \binom{k}{j})$. If j is constant, the number of function evaluations is $O(M^k L^j \ln L)$. \square

The above analysis gives us an upper bound on the amount of work to guarantee that all order- j mask strings of nonzero probe value are detected with at least probability δ , under the assumption that $f : \widetilde{M}^L \mapsto \mathbb{R}$ is k -epistatically bounded and the number of order- k hyperedges is $O(L)$. The scalability in L seems similar to that in binary domain[5]. Note that, for the same L, k, j but different M , bigger M will need more

function evaluations. For example, when $M = 2$, the function evaluations $O(2^k \binom{L}{j} (\ln L + \ln \binom{k}{j}))$, when $M > 2$, the function evaluations $O(M^k \binom{L}{j} (\ln L + \ln \binom{k}{j} + k \ln(M - 1)))$.

6. COMPUTING FOURIER COEFFICIENTS USING DETERMINISTIC ALGORITHM

Given a black-box high-cardinality function $f : \widetilde{M}^L \mapsto \mathbb{R}$ with k -bounded epistasis, we design a deterministic algorithm to compute its Fourier coefficients based on the properties of probe.

Since f is k -bounded epistatic, by Generalized Embedding Theorem, let $\omega_m, \|m\| = k$ be a maximal nonzero Fourier coefficient. The Maximal Probe Theorem shows that $P(f, m, c) = \omega_m$ for any $c \in S(\text{zero}(m))$. Thus, we do the probe $P(f, m, 0)$, and the result will be ω_m . By Equ.(12), all the order- k Fourier coefficients can be computed by doing $\binom{L}{k}$ probes, each of which uses M^k function evaluations.

Then, concerning the calculation of ω_j that $j = k - 1$, by High-cardinality Probe Subset Theorem, we have:

$$\begin{aligned} P(f, j, 0) &= \sum_{u \subseteq j} \omega_u \psi_u(0) \\ &= \sum_{u \subseteq j} \omega_u \psi_u(0) + \omega_j \psi_j(0) \\ &= \sum_{u \subseteq j} \omega_u + \omega_j \end{aligned} \quad (21)$$

The potential nonzero Fourier coefficients in the summation are all of order k and has been computed. Thus ω_j can be calculated by $\omega_j = P(f, j, 0) - \sum_{u \subseteq j} \omega_u$.

1. $\forall k' > k, \omega_{j: \|j\|=k'} = 0$.
2. $k' = k, \omega_{j: \|j\|=k'} = P(f, j, 0)$.
3. $k' = k' - 1$, if $k' < 0$, all the Fourier Coefficients have been computed, and exit.
4. $\omega_{j: \|j\|=k'} = P(f, j, 0) - \sum_{u: u \subseteq j} \omega_u$, go to step 3.

Figure 3: Deterministic Algorithm to Compute Fourier Coefficients

The above algorithm is an extension of probe-based top-down algorithm from binary [5] to high-cardinality domain. We next consider the function evaluations needed for calculating all Fourier coefficients. Note that all function evaluations necessary to compute $P(f, j, 0)$ that $m \subset j$ have already been done in the computation of $P(f, m, 0)$. So, for problems with bounded order k , calculating all the Fourier coefficients need $M^k \binom{L}{k}$ function evaluations.

7. STOCHASTIC LINKAGE DETECTION AND FOURIER COMPUTATION BASED ON PROBE METHOD

A stochastic algorithm is proposed here to detect linkage hypergraph and compute Fourier coefficients. We will show that the stochastic probe algorithm for binary problems developed by Heckendorn and Wright[5] can be extended to a high-cardinality domain.

Because an order- j hyperedge of high-cardinality function might correspond to at most $(M - 1)^j$ mask strings of

nonzero probe value, in the following description, we have to concentrate on mask strings instead of hyperedges. This is different from that in binary domain.

TestByProbe Algorithm in Figure 4 is to see whether a high-cardinality mask string corresponds a hyperedge. If variables of m does not nonlinearly interacts, then TestByProbe algorithm returns a NULL probe value; else it will return the probe value $P(f, m, c)$ where c is an all-zero background. $N(m)$ depends on $\|m\|$ and is determined by Theorem 16. $N(m)$ guarantees that we correctly detect the hyperedges with probability δ .

```

TestByProbe ( $m, N(m)$ )
  ProbeValue = null
  for  $i = 1 : N$ 
     $c \leftarrow$  a random string in  $S(\text{zero}(m))$ 
     $temp = P(f, m, c)$ 
    if  $temp \neq 0$  then
       $\text{zero}(\text{zero}(m))$  is a hyperedge
      ProbeValue =  $P(f, m, 0)$ 
    break
  endif
  return ProbeValue
endfor

```

Figure 4: Algorithm of TestByProbe

The bottom-up part of stochastic algorithm in Figure 5 is aimed to detect all the mask strings that corresponds to hyperedges. It traverses strings from low order to high order due to the following observation: considering a mask string $m \in \widetilde{M}^L$, by Corollary of Linkage Hypergraph Property, if there exists any $m' \supset m$ that m' does not correspond a hyperedge, then the mask m does not correspond to a hyperedge. It does a breadth-first traversal of the lattice of mask strings, starting with empty mask, then looking at the order-1 masks, order-2 masks, etc. Before seeing whether a new mask string m corresponds a hyperedge by TestByProbe Algorithm, we first test that whether all masks $m' \supset m$ correspond a hyperedge, i.e. whether m' 's are in the *maskList*. If all of them are in the *maskList*, then we do TestByProbe; otherwise, we need not do probing any more, because we can surely say that it does not correspond to a hyperedge.

In the algorithm, *quene* is used for breadth-first traversal, *hyperedgeList* is to store the hyperedges, *maskList* is used to store the those mask strings of nonzero probe value which correspond to hyperedges, and *maskSet* stores the probe values (all-zeros string background) of those mask strings in *maskList*. *Superset - List*(m) is a list of mask strings a such that $\|a\| = \|m\| + 1$ and $a \subset m$.

Then, a top-down algorithm in Figure 6 for computing Fourier coefficients traverses the mask strings in *maskList* from higher order masks to lower order, that is in the reverse order from which they were added to *maskList*. The algorithm is based on Equ.(21). The Fourier coefficients are computed accurately using only the function evaluations done in the bottom-up part of the algorithm.

We take a simple embedded landscape as an example. Consider a function $f(x_1, x_2, x_3) = g_1(x_1, x_2) + g_2(x_3)$, where $(x_1, x_2, x_3) \in \{0, 1, 2\}^3$, there are two epistatic block $\{x_1, x_2\}$ and $\{x_3\}$. Assume that those Fourier coefficients are nonzero: $\omega_{000}, \omega_{100}, \omega_{200}, \omega_{010}, \omega_{020}, \omega_{001}, \omega_{002}, \omega_{011}, \omega_{022}$ (note

the variable position is counted from right to left); other Fourier coefficients are zero. The algorithm of Traverse-Mask-Strings traverses the lattice of mask strings as shown in Figure 7. First, TestByProbe algorithm is applied to empty mask 000, and nonzero probe value is obtained, so Φ is added in the hyperedges set *hyperedgeList* and 000 is added to *maskList*. Then, test order-1 mask strings. Take 001 as an example: because all the subset strings of 001 (the only order-0 mask string, 000) are in the *maskList*, so we calculate the probe value of 001. After all order-1 mask strings have been tested, then consider order-2 mask strings. For example, there are two substrings of 011 that are order-1 (010 and 001), both of which in the *maskList*, so we have to test 011 using TestByProbe algorithm. Then consider order-3 strings. Because for any order-3 string, there exist some of order-2 substrings that are of zero probe value and are not included in *maskList*, so all the order-3 strings can not be nonzero-probe-valued strings and we do not need to run TestByProbe algorithm on order-3 strings any more.

According to the above analysis, we can classify all M^L mask strings into three types: type-1, type-2 and type-3. The mask string that may be included in *maskList* is named **type-1** mask. Considering a mask string a with order r , if all the order $r-1$ submasks of a ($\{i : \|i\| = r-1 \& a \subset i\}$) are type-1, but a is not included in *maskList*; then a is called **type-2** mask. The other type of masks is named **type-3** masks. Note that a type-3 mask will never be tested by TestByProbe algorithm. The same definitions in binary domain have been given by Heckendorn and Wright[5]. For example, as shown in Figure 7, the strings with solid underline are type-1, the ones with dot underline are type-2 and others without underline are type-3.

8. EXPERIMENTAL VERIFICATION

The experiment aims to verify the correctness of our algorithms. We construct a “needle-in-haystack” problem over high-cardinality alphabets domain and, see whether our algorithm can correctly detect its hyperedges and accurately compute the Fourier coefficients. The dimension is 30, the alphabet cardinality is 3, the order of maximum epistasis is 3 and there are 10 subfunctions. For each subfunction, the fitness is a constant value 8 except a randomly placed needle point with a different value 8.1. The variable positions of each subfunction are randomly chosen.

For the deterministic algorithm in Section 6, we do not need to set any parameters. The algorithm run in a deterministic way, and after $3^3 \binom{30}{3}$ function evaluations all Fourier coefficients was computed accurately. The epistatic structure was obtained from those strings with nonzero coefficients.

For the stochastic linkage detection algorithm in Section 7, the only parameter in the algorithm is number of iterations N per mask string. It is determined by Theorem 16. For example, the probability that all order-2 mask strings of nonzero probe value are detected is set to 0.99. The number of such order-2 mask strings is $J = (M-1)^k L \binom{L}{k} = 120 \binom{30}{3}$. And then we bring j into Equ.(20) and get $N = 23$. The algorithm successfully detected all the hyperedges and computed the Fourier coefficients.

The simple experiment demonstrates the correctness of our theory and methods. More experiments are needed to further test the performance of the algorithms, especially the scalability of algorithms.

Traverse-Mask-Strings

```

hypergraphList.initialize()
maskList.initialize()
queue.initialize()
m = {} //Empty mask
ProbeValue = TestByProbe(m, N(m))
if ProbeValue ≠ null then
    queue.add(m)
    maskSet[m] ← ProbeValue
endif
while queue.notEmpty() do
    m ← queue.remove()
    for a ∈ SuperSet - List(m) do
        if all subsets of a of order ||m|| are in the
        maskList then
            ProbeValue ← TestByProbe(a, N(a))
            if ProbeValue ≠ null then
                queue.add(a)
                maskSet[a] ← ProbeValue
                maskList.addFirst(a)
                e = zero(zero(a))
                if e is not in hypergraphList
                    hypergraphList.addFirst(e)
                endif
            endif
        endif
    endfor
endwhile

```

Figure 5: Bottom-up part: traverse mask strings with nonzero probe value

Compute-Fourier-Coefficients(*maskList*)

```

for m ∈ maskList
    ProbeValue ← maskSet[m]
    if ω[m] ≠ null then ω[m] ← ω[m] + ProbeValue
    else ω[m] ← ProbeValue
    for each a ∈ m
        if ω[a] ≠ null then ω[m] ← ω[a] - ω[m]
        else ω[a] ← -ω[m]
    endfor
endfor

```

Figure 6: Top-down part: compute Fourier coefficients

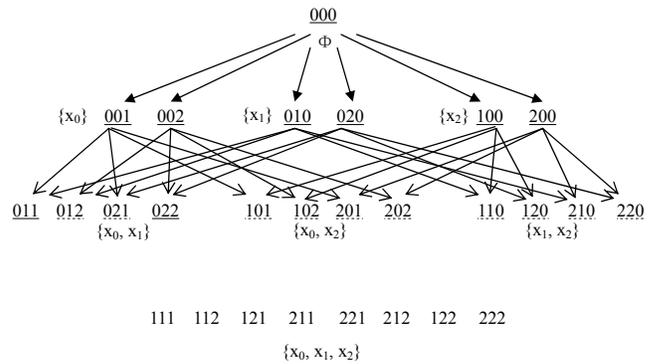


Figure 7: Breadth-first traversal of lattice masks in stochastic linkage detection algorithm

9. CONCLUSION AND DISCUSSION

The main contribution is that we design the “probe” in high-cardinality alphabets domain. The work gives a general theoretical foundation for perturbational methods for determining the epistatic structure of functions in a generalized discrete domain. It is an extension of Heckendorn and Wright’s probe theory and methods from the binary domain to a more general high-cardinality alphabets domain. Based on rigorous mathematical induction, two methods are developed to detect the epistatic structure: one is a top-down deterministic algorithm, the other is a stochastic bottom-up algorithm. Both can discover the epistatic structure of order-bounded functions, and at the same time compute the Fourier coefficients accurately in a deterministic way or under some given probability.

However, the work is far from complete and many unsolved problems remain. The computational complexity of the algorithms needs to be further explored in theoretical and experimental aspects. Ref.[5] shows that probe method in binary domain has polynomial computational complexity under the assumption that the problem is epistatically bounded and the number of hyperedges is $O(L)$. It is necessary to study the scalability of probe methods in non-binary domain and further understand a trade-off between computational cost and applicability of the probing method.

The meaning of nonzero Fourier coefficients should be further studied. As shown in Section 5, a hyperedge might correspond many nonzero Fourier coefficients. The relationship between those nonzero Fourier coefficients and the epistatic structure should be theoretically analyzed clear in the future. There is also much space for algorithmic enhancement. The algorithm might be enhanced if we restrict ourself only into the work of hyperedge detection: for the purpose of order- j linkage detection, we do not necessarily compute all the nonzero Fourier coefficients; once we detect ONE nonzero Fourier coefficient, then it is surely a hyperedge. Further research is needed to improve the algorithm.

Extension of the work to the continuous domain is also a promising direction. Below we will give some basic consideration about linkage detection in continuous domain. A research LINC-R based on Munetomo et al.’s previous work[12] has been proposed elsewhere[15], which tests non-linearity by order-2 perturbations for real-coded GA.

9.1 About Linkage of Continuous Problems

Without loss of generalization, we consider continuous function $f : [a, b]^L \mapsto \mathbb{R}$, where $a < b$. The linkage hypergraph is $G_f = G_f(V, E)$. Since all the continuous problems have to be considered under some granularity in digital computer, we can discretize the continuous variables into a few intervals and then discuss the epistatic structure in the discrete domain. D indicates a discretization method $D : [a, b]^L \mapsto \widetilde{M}^L$, that converts a continuous domain to a discrete domain. The linkage hypergraph of $f_D : \widetilde{M}^L \mapsto \mathbb{R}$ is denoted by $G_{f_D} = G_{f_D}(V_D, E_D)$. If f is a continuous embedded landscape with bounded epistasis, then f_D is a discrete embedded landscape and G_{f_D} can be obtained using the algorithm of the paper. So, the question is: what is the relationship between G_f and G_{f_D} , or how does G_{f_D} approximate G_f ?

We have the following suppositions: 1) Given any D , $G_{f_D} \subseteq G_f$; 2) For any continuous domain, there exists a discretization method D that make that $G_{f_D} = G_f$; 3) D_1 and

D_2 are two discretization methods that convert a continuous domain to \widetilde{M}_1^L and \widetilde{M}_2^L , if $\widetilde{M}_1^L \subseteq \widetilde{M}_2^L$, then $G_{f_{D_1}} \subseteq G_{f_{D_2}}$.

Now, the problem becomes how to discretize continuous variables so that $G_{f_D} = G_f$. The discretization is not necessarily uniform. If we know a prior that some subspace is more important or complicated than others, then we can place more intervals in this subspace. From the aspect of linkage detection, the “goodness of discretization” means that it is a discretization that leads to the correct detection of epistatic structure between variables. Given a continuous black-box problem with bounded epistatic order, if the discretization method D is sufficient to preserve the linkage information, then its epistatic linkage structure can be obtained by the method in the paper.

10. ACKNOWLEDGEMENTS

Shude Zhou would like to thank Alden Wright for helpful discussion. The authors thank the anonymous reviewers for their valuable comments. The work is supported by National Key Project for Basic Research of China G2002cb312205.

11. REFERENCES

- [1] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Boston, 2002.
- [2] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Mtivation, analysis and first results. *Complex Systems*, 3:493–530.
- [3] G. R. Harik. *Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms*. Ph.D. dissertation, University of Michigan, Ann Arbor, 1997.
- [4] R. B. Heckendorn. Embedded landscapes. *Evolutionary Computation*, 10(4):345–369, 2002.
- [5] R. B. Heckendorn and A. Wright. Efficient linkage discovery by limited probing. *Evolutionary Computation*, 12(4):517–545, 2004.
- [6] M. T. Iglesias, B. Naudts, A. Verschoren, and C. Vidal. Walsh transforms, balanced sum theorems and partition coefficients over multary alphabets. In *GECCO 2005*, pages 1303–1308.
- [7] H. Kargupta. The gene expression messy genetic algorithm. In *Proc. of the IEEE international conference on Evolutionary Computation*. Nogoya, Japan, 1996.
- [8] H. Kargupta and B. Park. Gene expression and fast construction of distributed evolutionary representation. *Evolutionary Computation*, 9(1):43–59, 2001.
- [9] G. J. Koehler, S. Bhattacharyya, and M. D. Vose. General cardinality genetic algorithms. *Evolutionary Computation*, 5(4):439–549, 1998.
- [10] P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms: A new Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston, 2002.
- [11] H. Muhlenbein and R. Hons. The estimation of distribution and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.
- [12] M. Munetomo and D. E. Goldberg. Linkage identification by non-monotonicity detection for overlapping functions. *Evolutionary Computation*, 7(4):377–398, 1999.
- [13] C. K. Oei. *Walsh Function Analysis of Genetic Algorithms of Nonbinary Strings*. Unpublished masters thesis, UIUC, 1992.
- [14] M. Pelikan. *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary algorithms*. Springer Publication, Berlin, 2005.
- [15] M. Tezuka, M. Munetomo, and K. Akama. Linkage identification by nonlinearity check for real-coded genetic algorithms. In *GECCO 2004 Proceedings, Part 2, Lecture Notes in Computer Science*, pages 222–233, 2004.
- [16] D. Thierens. Scalability problems of simple genetic algorithms. *Evolutionary Computation*, 7(4):331–352, 1999.
- [17] S. Zhou, R. B. Heckendorn, and Z. Sun. Detecting the epistatic structure of generalized embedded landscape. *Genetic Programming and Evolvable Machines*, submitted.
- [18] S. Zhou and Z. Sun. Matrix interpretation of generalized embedded landscapes. In *GECCO 2007 proceedings*.