MIT Open Access Articles

Lower and upper bounds for linkage discovery

Citation	Sung-Soon Choi, Kyomin Jung, and Byung-Ro Moon. “Lower and Upper Bounds for Linkage Discovery.” Evolutionary Computation, IEEE Transactions on 13.2 (2009): 201-216. © 2009 Institute of Electrical and Electronics Engineers
As Published	http://dx.doi.org/10.1109/TEVC.2008.928499
Publisher	Institute of Electrical and Electronics Engineers
Version	Final published version
Accessed	Thu Jan 27 03:48:02 EST 2011
Citable Link	http://hdl.handle.net/1721.1/52340
Terms of Use	Article is made available in accordance with the publisher’s policy and may be subject to US copyright law. Please refer to the publisher’s site for terms of use.
Detailed Terms	

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

DSpace@MIT

Lower and Upper Bounds for Linkage Discovery

Sung-Soon Choi, Kyomin Jung, and Byung-Ro Moon, *Member, IEEE*

***Abstract*—For a real-valued function f defined on $\{0,1\}^n$, the linkage graph of f is a hypergraph that represents the interactions among the input variables with respect to f. In this paper, lower and upper bounds for the number of function evaluations required to discover the linkage graph are rigorously analyzed in the black box scenario. First, a lower bound for discovering linkage graph is presented. To the best of our knowledge, this is the first result on the lower bound for linkage discovery. The investigation on the lower bound is based on Yao's minimax principle. For the upper bounds, a simple randomized algorithm for linkage discovery is analyzed. Based on the Kruskal–Katona theorem, we present an upper bound for discovering the linkage graph. As a corollary, we rigorously prove that $\mathcal{O}(n^2 \log n)$ function evaluations are enough for bounded functions when the number of hyperedges is $\mathcal{O}(n)$, which was suggested but not proven in previous works. To see the typical behavior of the algorithm for linkage discovery, three random models of fitness functions are considered. Using probabilistic methods, we prove that the number of function evaluations on the random models is generally smaller than the bound for the arbitrary case. Finally, from the relation between the linkage graph and the Walsh coefficients, it is shown that, for bounded functions, the proposed bounds are eventually the bounds for finding the Walsh coefficients.**

***Index Terms*—Black box scenario, complexity analysis, linkage discovery, lower and upper bounds, linkage graph, Walsh analysis.**

I. Introduction

A. Linkage and Evolutionary Algorithms

For nontrivial combinatorial optimization problems, the encoding of solutions usually contains interactions between bits (or genes), i.e., the contribution of a bit to the fitness function depends on other bits. Such an interaction between bits is called *linkage* or *epistasis*. Incorporating the process of discovering the linkage structure into evolutionary algorithms is now considered as a standard approach to improve the performance of evolutionary algorithms.

According to the building block hypothesis [1], [2], a genetic algorithm (GA) implicitly gives favor to low-order, high-quality schemata and, over time, it generates higher order high-quality schemata from low-order schemata via crossover. In order that high-quality schemata are well preserved, it is thus important to identify the bits of strong linkage and reorder the bit positions such that those bits stay close together on the chromosome. For this reason, there were proposed a number of reordering methods. They may be classified into two categories, static reordering methods and dynamic reordering methods. Static reordering methods identify the bits of strong linkage and perform the reordering of bit positions before the genetic process. Once the genetic process starts, the representation is statically used all through the algorithm. They were successfully applied to the graph/circuit partitioning problems [3]–[5]. On the other hand, dynamic reordering methods change or evolve the representation dynamically in the genetic process. Examples of the GAs with dynamic reordering methods include the messy GA [6]–[8], the gene expression messy GA (GEMGA) [9], and the linkage learning GA (LLGA) [10].

Estimation-of-distribution algorithms (EDAs) utilize linkages among bits to learn the probability distribution of high-quality solutions and generate promising solutions from the distribution. In fact, linkage structures were not used in the early versions of EDAs such as the population-based incremental learning algorithm (PBIL) [11] and the compact genetic algorithm (CGA) [12]. They evolve the marginal distribution for each bit, assuming that there is no dependency between bits. The linkage information for bit pairs was first used in the mutual information maximization for input clustering (MIMIC) [13] and later used in the bivariate marginal distribution algorithm (BMDA) [14]. The linkage structure for bit subsets of higher order was evolved and exploited in recent EDAs including the Bayesian optimization algorithm (BOA) [15], the hierarchical BOA (hBOA) [16], and the factorized distribution algorithm (FDA) [17].

Recently, Streeter [18] presented an efficient algorithm, which combines a linkage detection algorithm with a local search heuristic, to optimize bounded fitness functions with nonoverlapping subfunctions. Wright and Pulavarty [19] showed that exact factorization can be constructed for a class of fitness functions by incorporating linkage detection algorithm into the factorization process.

B. Linkage Discovery in the Black Box Scenario

A real-valued function f is *pseudo-Boolean* if it is defined on $\{0,1\}^n$. The *support set* of a pseudo-Boolean function f is the set of bits that f depends on and it is denoted by $\mathsf{supp}(f)$. For a pseudo-Boolean function f, an *additive expression* of f is a finite sum of subfunctions of the following form:

$$f = \sum_i f_i$$

Manuscript received March 29, 2007; revised January 27, 2008; accepted February 03, 2008. First published February 03, 2009; current version published April 01, 2009. This work was supported in part by the Brain Korea 21 Project and in part by ICT at Seoul National University, which provided research facilities for this study. The work of S.-S. Choi was supported in part by the Korea Science and Engineering Foundation (KOSEF) under Grant R16-2007-075-01000-0 funded by the Korea government (MOST).

S.-S. Choi is with Random Graph Research Center, Yonsei University, Seoul 120-749, Korea (e-mail: ss.choi@yonsei.ac.kr).

K. Jung is with the Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: kmjung@mit.edu).

B.-R. Moon is with the School of Computer Science and Engineering, Seoul National University, Seoul 151-744, Korea (e-mail: moon@soar.snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2008.928499

1089-778X/$25.00 © 2009 IEEE

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

where f_i's are pseudo-Boolean functions. If f can be represented as an additive expression in which $f = \sum_i f_i$ and the size of the support set of f_i is at most k for all i, then f is called *k-bounded*. Many combinatorial optimization problems induce bounded fitness functions. Examples include the NK landscape problem, the MAX k-CUT problem, the MAX k-SAT problem [20], and the constraint satisfaction problem [21]. Denote by $[n]$ the set of positive integers from 1 to n, i.e., $[n] = \{1, 2, \dots, n\}$. Given a subset H of $[n]$, we say that there is a *linkage* (or an *epistasis*) among the bits in H if, for any additive expression of f, $f = \sum_i f_i$, there exists i' such that $H \subseteq \text{supp}(f_{i'})$. The *linkage graph* of f is a hypergraph $G_f = ([n], E)$, where each bit in $[n]$ represents a vertex and a subset H of $[n]$ belongs to the edge set E if and only if there is a linkage among the bits in H. For a subset H in E, H is called a *hyperedge* of G_f. If a hyperedge H is not contained in any other hyperedge of G_f, H is called a *maximal hyperedge*.

For example, consider the following simple function:

$$f(x_1, x_2, x_3, x_4) = x_1x_2 + x_2x_3x_4.$$

If we let $f_1(x_1, x_2) = x_1x_2$ and $f_2(x_2, x_3, x_4) = x_2x_3x_4$, f can be represented as an additive expression $f = f_1 + f_2$. In this expression, each subfunction of f has a support set of which size is at most 3 and so f is 3-bounded. It can be shown that the support sets of f_1 and f_2, $\{1, 2\}$ and $\{2, 3, 4\}$, are hyperedges of G_f. By definition of linkage, the nonempty subsets of $\{1, 2\}$ and $\{2, 3, 4\}$ are also hyperedges of G_f. The linkage graph G_f consists of the vertex set $\{1, 2, 3, 4\}$ and the edge set $E = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{2, 3, 4\}\}$. There are two maximal hyperedges in G_f, $\{1, 2\}$ and $\{2, 3, 4\}$.

We consider the problem to discover the linkage graph of a bounded pseudo-Boolean function f in the black box scenario. In the black box scenario, only the minimal prior knowledge of f is given such as the number of bits that f depends on, n and the boundedness information of f, k. To know the function value for an input string, we should query the black box. The black box scenario has been used for the analysis of randomized search heuristics [22]. The main interest of this paper is to analyze the number of queries, i.e., the number of function evaluations required in discovering the linkage graph of a k-bounded pseudo-Boolean function. In particular, we put emphasis on the k-bounded functions such that k is a constant independent of n. As described in [23] and [24], once the linkage graph of a given function f is discovered, the Walsh coefficients of f can be found efficiently (by using additional function evaluations). Thus, the algorithm of discovering the linkage graph is useful for the problem of finding the Walsh coefficients.

There were a few rigorous analyses on the problem to discover the linkage graph. For a k-bounded pseudo-Boolean function with constant k, there were algorithms proposed using a polynomial number of function evaluations. Kargupta and Park [25] first provided a deterministic algorithm based on the relation between Walsh coefficients requiring $\mathcal{O}(n^k)$ function evaluations. Later, Heckendorn and Wright [23], [24] extended the work of Kargupta and Park to propose an efficient randomized algorithm, the hyperedge-candidate-based linkage discovery algorithm (HCA), which finds the hyperedges in a bottom-up fashion (from low- to higher orders).[1] Under a random model of fitness functions with $\mathcal{O}(n)$ maximal hyperedges, they showed that the algorithm terminates after $\mathcal{O}(n^2 \log n)$ function evaluations on average. For a pseudo-Boolean function with nonoverlapping subfunctions, Streeter [18] presented an efficient randomized algorithm to find the connected components of the linkage graph based on binary search. It terminates only after $\mathcal{O}(n \log n)$ function evaluations.

C. Contributions: Arbitrary Case

In this paper, we rigorously analyze the lower and upper bounds for the number of function evaluations required to discover the linkage structure of a given pseudo-Boolean function. When f is a k-bounded function defined on $\{0, 1\}^n$, the bounds for discovering the linkage structure of f are basically provided in terms of n and k. If we focus on the situation that k is a constant independent of n, our contributions become more clear.

First, we investigate the lower bound for the problem to discover the linkage graph. Let $\mathcal{B}(n, m, k)$ be the set of k-bounded pseudo-Boolean functions defined on $\{0, 1\}^n$ whose linkage graphs have m maximal hyperedges.

Theorem 1: Suppose that k is a constant and $m = \mathcal{O}(n^{k-\delta})$ for some constant $\delta > 0$. Given a constant $0 < \varepsilon < 1$, any randomized algorithm that, for any $f \in \mathcal{B}(n, m, k)$, finds G_f with error probability at most ε requires $\Omega(m \log n / \log m)$ function evaluations.

The investigation on the lower bound mainly depends on *Yao's minimax principle* [26]. To the best of our knowledge, this is the first result on the lower bound for linkage discovery. This result would serve as the basis in analyzing the performance of evolutionary algorithms for the problems such as concatenated k-deceptive trap functions.

To investigate the upper bounds for discovering the linkage graph, we analyze the randomized algorithm HCA to obtain the following result.

Theorem 2: Suppose that f is a k-bounded pseudo-Boolean function defined on $\{0, 1\}^n$, k is a constant independent of n, and the number of maximal hyperedges in G_f is $m = \Theta(n^{h+\alpha})$ for a nonnegative integer h and $0 \leq \alpha < 1$. Then, for any constant $\varepsilon > 0$, HCA finds the linkage graph G_f in $\mathcal{O}(n^{\widehat{q}} \log n)$ function evaluations with error probability at most ε, where $\widehat{q} = \max\{(h+\alpha)(h+2)/(h+1), h+1\}$.

Theorem 2 implies that $\mathcal{O}(n^2 \log n)$ function evaluations are enough for arbitrary k-bounded functions with $\mathcal{O}(n)$ maximal hyperedges for constant k. This is stronger than the result of Heckendorn and Wright [23], [24], in that they achieved the upper bound only in terms of the expected number of function evaluations on a random model of fitness functions. Our analysis is based on the *Kruskal–Katona theorem* [27], [28], which helps to effectively bound the number of bit subsets that the algorithm checks for a hyperedge in finding the linkage graph.

[1] In this paper, we call the algorithm of Heckendorn and Wright HCA instead of its original name. The reason of calling it HCA and its description are provided in Section II-A.

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

D. Contributions: Random Models

To see the typical behavior of the algorithm for linkage discovery HCA, we consider three random models of fitness functions: the *uniform probability model*, the *uniform density model without replacement*, and the *uniform density model with replacement*. The uniform probability model $\mathcal{F}(n,p,k)$ generates a fitness function f in the following steps. First, it chooses each k-order subset of $[n]$ with probability p and independently of the other subsets. Then, it sets a function f_i for each subset S_i chosen in the previous step and generates the fitness values of f_i for the bit strings defined on S_i in an arbitrary way. Finally, a fitness function f is constructed by summing up f_i's, i.e., $f = \sum_i f_i$. The uniform density model without replacement $\tilde{\mathcal{F}}(n,\tilde{m},k)$ chooses $\tilde{m}$ k-order subsets of $[n]$ uniformly at random and without replacement from the possible $\binom{n}{k}$ k-order subsets of $[n]$. The uniform density model with replacement $\tilde{\mathcal{F}}'(n,\tilde{m},k)$ chooses $\tilde{m}k$-order subsets of $[n]$ uniformly at random and with replacement from the possible $\binom{n}{k}k$-order subsets of $[n]$. Then, they generate a fitness function in the same way as in the uniform probability model.

The three models specify only the bits that subfunctions may depend on and do not make any assumption for the values that the subfunctions have. There are many problems whose instances are usually generated in the framework of these models. Examples are the MAX k-SAT problem [29], the MAX CUT problem [30], the problem of maximizing NK landscapes [31], etc. The random models have served as a test bed for the theoretical studies on evolutionary algorithms. Heckendorn and Wright [23], [24] analyzed HCA on the uniform density model with replacement for a specific parameter. The uniform density model without replacement was used by Gao and Culberson [32] to investigate the space complexity of EDAs.

In the following, "a sequence of events $\{A_n\}$ *almost always* occurs" means that $\lim_{n\to\infty} \Pr[A_n] = 1$.

Theorem 3: Let k be a constant independent of n. For a nonnegative integer h and $0 \le \alpha < 1$, suppose that f is generated from one of the following models:

1) $\mathcal{F}(n,p,k)$ with $\binom{n}{k}p = \Theta(n^{h+\alpha})$;
2) $\tilde{\mathcal{F}}(n,\tilde{m},k)$ with $\tilde{m} = \Theta(n^{h+\alpha})$;
3) $\tilde{\mathcal{F}}'(n,\tilde{m},k)$ with $\tilde{m} = \Theta(n^{h+\alpha})$.

Then, for any constant $\varepsilon > 0$, the number of function evaluations of HCA to guarantee the error probability at most ε is almost always $\mathcal{O}(n^{\tilde{q}} \log n)$, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

To prove Theorem 3, the second moment method [33] and coupling arguments among the models are crucially used. Theorem 3 describes the behaviors of HCA on the three random models in the sense of "high" probability. For a random function f from $\mathcal{F}(n,p,k)$ with $\binom{n}{k}p = \Theta(n^{h+\alpha})$, the number of subfunctions f_i's generated by the random model is almost always $\Theta(n^{h+\alpha})$, provided that α is a positive constant. For a random function f from $\tilde{\mathcal{F}}(n,\tilde{m},k)$ or $\tilde{\mathcal{F}}'(n,\tilde{m},k)$ with $\tilde{m} = \Theta(n^{h+\alpha})$, the number of subfunctions f_i's generated by the random models is $\Theta(n^{h+\alpha})$. Because the subfunctions may be arbitrary, for a random function f in Theorem 3, we may assume that the number of maximal hyperedges in G_f is (almost always) $\Theta(n^{h+\alpha})$. In this regard, Theorem 3 (compared with Theorem 2) implies that the number of function evaluations of HCA on the random models is generally smaller than the upper bound for the arbitrary case, which is described in more detail in the following.

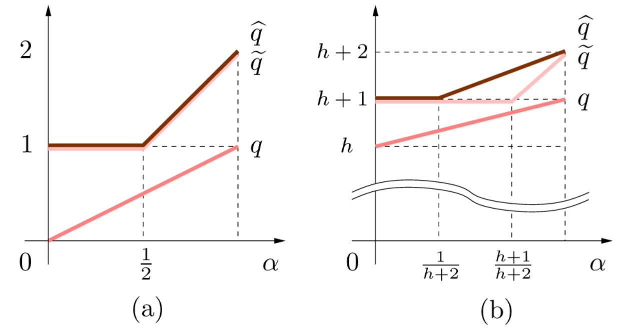


Fig. 1. Lower and upper bounds for linkage discovery (a) $h = 0$ and (b) $h \ge 1$.

Let $q = h + \alpha$. Then, for $m = \Theta(n^{h+\alpha})$, the lower bound of $\Omega(n^q)$ follows from Theorem 1 (provided that $m = \mathcal{O}(n^{k-\delta})$ for some constant $\delta > 0$). Fig. 1 compares the lower bound, the upper bound for the arbitrary case, and the upper bounds on the random models from Theorems 1, 2, and 3, respectively. It shows the change of the bounds in terms of $q, \hat{q}$, and $\tilde{q}$ as the number of maximal hyperedges increases, i.e., as h and α increase. The difference between q and $\hat{q}$ is at most one, which means that the lower and upper bounds are within a factor of $n \log n$. When $h \ge 1$ and $1/(h+2) < \alpha < 1$, the upper bounds on the random models are strictly less than the upper bound for the arbitrary case. In this case, HCA uses less function evaluations on the random models than the bound for the worst case. For example, when $h = 1$ and $\alpha = 2/3$, the upper bounds on the random models are less than the upper bound for the arbitrary case by a factor of $n^{1/2}$ [$\tilde{q} = 2$ and $\hat{q} = 5/2$].

E. Organization

The rest of this paper is organized as follows. Previous works for linkage discovery are reviewed in Section II. In Section III, we briefly introduce Yao's minimax principle and present a lower bound for discovering linkage graph based on Yao's minimax principle. In Section IV, the algorithm HCA is analyzed to provide an upper bound for discovering linkage graph. In Section V, the number of function evaluations by HCA on the three random models is given. Based on the relation between linkage graph and Walsh coefficients, implications of our results to finding Walsh coefficients are presented in Section VI. Finally, Section VII concludes this paper with suggestions for future work.

II. Previous Works

This section describes previous works for linkage discovery mainly focusing on the results of Heckendorn and Wright [23], [24].

A. Algorithm for Linkage Discovery

The linkage identification by nonlinearity check (LINC) [34] presents a perturbation method to determine whether a second-order subset H of $[n]$ has a linkage with respect to a function f. Given a string x, it checks the existence of nonlinearity within

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

```
HC-based linkage discovery($f$, $\varepsilon$)
// $f$ :   $a$ $k$-bounded pseudo-Boolean function defined on $\{0,1\}^n$
// $\varepsilon$ :   a positive constant
    $V \leftarrow [n]$ and $E \leftarrow \emptyset$;
    for each 1-order subset $H$ of $[n]$
        repeat $\lceil 2^k \log \frac{(2n)^k}{\varepsilon} \rceil$ times
            choose a string $x$ uniformly at random from $\{0,1\}^n$;
            if $\mathfrak{L}(f, H, x) \neq 0$, then $E \leftarrow E \cup \{H\}$ and break;
    for $j$ from 2 to $k$
        for each $j$-order hyperedge candidate $H$
            repeat $\lceil 2^k \log \frac{(2n)^k}{\varepsilon} \rceil$ times
                choose a string $x$ uniformly at random from $\{0,1\}^n$;
                if $\mathfrak{L}(f, H, x) \neq 0$, then $E \leftarrow E \cup \{H\}$ and break;
    return $G = (V, E)$;
```

Fig. 2. Pseudocode of the HC-based linkage discovery algorithm (HCA).

the two bits in H by flipping the bits of x in H individually and simultaneously and adding/subtracting the function values at the flipped strings. Heckendorn and Wright [23], [24] generalized the method to detect linkage for higher order subsets. Let f be a pseudo-Boolean function, H be a subset of $[n]$, and x be a string in $\{0,1\}^n$. They defined the *linkage test function* $\mathfrak{L}$ of f, H, and x as follows[2]:

$$\mathfrak{L}(f, H, x) = \sum_{A \subseteq H} (-1)^{|A|} f(x \oplus 1_A)$$

where 1_A represents the string consisting of ones in the bit positions of A and zeros in the rest and, for two strings $x, y \in \{0,1\}^n, x \oplus y$ means the bitwise addition modulo 2 of x and y. The linkage test function $\mathfrak{L}$ performs a series of function evaluations at x and the strings obtained by flipping x to detect the existence of the linkage in H. The following lemma describes the usefulness of the linkage test function in finding hyperedges of G_f.

Lemma 1: Suppose that f is a k-bounded pseudo-Boolean function defined on $\{0,1\}^n$. Then, the following hold.

a) A subset H of $[n]$ is a hyperedge of G_f if and only if $\mathfrak{L}(f, H, x) \neq 0$ for some string $x \in \{0,1\}^n$.

b) For a j-order hyperedge H of G_f, the probability that $\mathfrak{L}(f, H, x) \neq 0$ for a string x chosen uniformly at random from $\{0,1\}^n$ is at least $(1/2^{k-j})$.

c) For a hyperedge H of G_f, every j-order subset of H is also a hyperedge for $j \geq 2$.

The proof of Lemma 1 was provided explicitly or implicitly in [24].[3]

Lemma 1 a) and b) indicate that the linkage test function detects a hyperedge with one-sided error. Thus, by repeatedly evaluating the test function for randomly chosen strings, the error can be made arbitrarily small. In particular, when k is a constant, this implies that a constant number of linkage tests is enough for detecting any hyperedge. Hence, from Lemma 1 a) and b), it is straightforward to design a randomized algorithm requiring a polynomial number of function evaluations. Testing linkage independently for each subset of order at most k induces a randomized algorithm requiring $\mathcal{O}(n^k \log n)$ function evaluations, where the factor $\mathcal{O}(n^k)$ bounds the number of subsets to be tested and the factor $\mathcal{O}(\log n)$ guarantees to bound a given constant error probability.

Lemma 1 c) describes an important property that a linkage graph should have: a subset H of $[n]$ cannot be a hyperedge if there exists a nonempty subset of H that is not a hyperedge. Suppose that f is a pseudo-Boolean function and H is a j-order subset of $[n]$ for $j \geq 2$. We call H a *j-order hyperedge candidate* of G_f if every $(j-1)$-order subset of H is a hyperedge of G_f. Lemma 1 c) implies that H cannot be a hyperedge if H is not a hyperedge candidate and it is thus enough to investigate only the hyperedge candidates in detecting hyperedges. Based on this observation, Heckendorn and Wright [24] proposed a randomized algorithm that performs linkage test only for the hyperedge candidates. Fig. 2 describes the main idea of the algorithm of Heckendorn and Wright. In this paper, we call it the hyperedge-candidate-based linkage discovery algorithm (HCA) in order to emphasize the role of the hyperedge candidates in the algorithm.

HCA first detects the first-order hyperedges by investigating all the first-order subsets of $[n]$. Then, in order to discover the hyperedges of higher order, it performs linkage test for the hyperedge candidates that have been identified from the information of the hyperedges of lower order. To analyze the performance of HCA, Heckendorn and Wright considered the uniform density model with replacement $\tilde{\mathcal{F}}'(n, \tilde{m}, k)$. Under the model $\tilde{\mathcal{F}}'(n, \tilde{m}, k)$ with $\tilde{m} = \mathcal{O}(n)$ and constant k, they showed that, for any constant $\varepsilon > 0$, HCA finds the linkage graph of a randomly generated function in $\mathcal{O}(n^2 \log n)$ function evaluations on average with error probability at most ε.

[2] In fact, Heckendorn and Wright suggested the notion of *probe*, which is the linkage test function multiplied by a positive constant. Because the constant does not affect our result, it is omitted for simplicity.

[3] If Lemma 1 a) is true, Lemma 1 b) and c) are easily implied from the theorems in [24]. Lemma 1 a) also can be proved by using the theorems in [24] without much difficulty and we omit the proof.

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

B. Finding Walsh Coefficients

Walsh transform is a Fourier transform for the space of pseudo-Boolean functions in which a pseudo-Boolean function is represented as a linear combination of 2^n basis functions called *Walsh functions* [35]. For each subset H of $[n]$, the Walsh function corresponding to H, $\psi_H : \{0,1\}^n \to \mathbb{R}$, is defined as

$$\psi_H(x) = (-1)^{\sum_{i \in H} x[i]}$$

where $x[i]$ represents the ith bit value in x. If we define an inner product of two pseudo-Boolean functions f and g as

$$\langle f, g \rangle = \sum_{x \in \{0,1\}^n} \frac{f(x) \cdot g(x)}{2^n}$$

the set of Walsh functions $\{\psi_H | H \subseteq [n]\}$ becomes an orthonormal basis of the space of pseudo-Boolean functions. Hence, a pseudo-Boolean function f can be represented as

$$f = \sum_{H \subseteq [n]} \hat{f}(H) \cdot \psi_H$$

where $\hat{f}(H) = \langle f, \psi_H \rangle$ is called the *Walsh coefficient* corresponding to H. Specifically, if $\hat{f}(H) \neq 0$ and $\hat{f}(H') = 0$ for any $H' \supsetneq H$, $\hat{f}(H)$ is called a *maximal nonzero Walsh coefficient* of f. We refer to [36] for surveys of the properties of Walsh functions and Walsh transform in the space of pseudo-Boolean functions.

Heckendorn and Wright [24] provided a number of results to show the relation between the linkage test function and Walsh coefficients. Some of them are summarized in the following lemma.

Lemma 2: Suppose that f is a pseudo-Boolean function defined on $\{0,1\}^n$. Then, the following holds.

a) For a subset H of $[n]$, $\hat{f}(H)$ is a maximal nonzero Walsh coefficient of f if and only if H is a maximal hyperedge of G_f.

b) For a maximal hyperedge $H \subseteq [n]$
$$\hat{f}(H) = \frac{\mathfrak{L}(f, H, 0^n)}{2^{|H|}}.$$

c) For a subset H of $[n]$
$$\hat{f}(H) = \frac{\mathfrak{L}(f, H, 0^n)}{2^{|H|}} - \sum_{H \subsetneq H'} \hat{f}(H').$$

d) For subsets H and H' of $[n]$ with $H \subseteq H'$
$$\mathfrak{L}(f, H', 0^n) = \sum_{A \subseteq H' \setminus H} (-1)^{|A|} \mathfrak{L}(f, H, 1_A).$$

The proof of Lemma 2 can be found in [24].

Suppose that the linkage graph of f is given. Lemma 2 a) says that the subsets of $[n]$ with maximal nonzero Walsh coefficients are the maximal hyperedges. Thus, from Lemma 2 b), the maximal nonzero Walsh coefficients of f are found by evaluating the linkage test function at the zero string for each maximal hyperedge. Once the maximal nonzero Walsh coefficients are found, the Walsh coefficients corresponding to the subsets of lower orders can be found by successively applying Lemma 2 c). Lemma 2 d) shows that the function evaluations for the Walsh coefficients corresponding to the subsets of lower orders have already been performed in the process of computing the maximal nonzero Walsh coefficients. Hence, if f is k-bounded and m is the number of maximal hyperedges in G_f, $\mathcal{O}(2^k m)$ additional function evaluations are enough to find the Walsh coefficients of f. In the case that k is a constant independent of n, this implies that an upper bound for discovering the linkage graph is valid as an upper bound for finding Walsh coefficients if the bound for discovering linkage graph is $\Omega(m)$. Based on these arguments combined with the results for linkage discovery, Heckendorn and Wright showed that, under the uniform density model with replacement $\tilde{\mathcal{F}}'(n, \tilde{m}, k)$ for $\tilde{m} = \mathcal{O}(n)$ and constant k, the Walsh coefficients of a randomly generated function f can be found in $\mathcal{O}(n^2 \log n)$ function evaluations on average.

III. Lower Bound for Linkage Discovery

In this section, we prove Theorem 1. The main tool for the analysis is Yao's minimax principle [26], [33], which is stated as follows.

Proposition 1 (Yao's Minimax Principle): Consider a complexity model for computing a function F. Let $R_\varepsilon(F)$ be the minimum complexity over all randomized algorithms that, for all input x, compute $F(x)$ with error probability at most ε. Given a distribution μ on the inputs, let $D_\varepsilon^\mu(F)$ be the minimum complexity over all deterministic algorithms that correctly compute F on a fraction of at least $1 - \varepsilon$ of all inputs with respect to μ. Then

$$R_\varepsilon(F) = \max_\mu D_\varepsilon^\mu(F).$$

In the context of linkage discovery, Yao's minimax principle may be restated as follows.

Corollary 1: Suppose that $\mathcal{B}$ is a set of pseudo-Boolean functions. Let $R_\varepsilon[\mathcal{B}]$ be the minimum number of function evaluations by a randomized algorithm that, for any $f \in \mathcal{B}$, finds G_f with error probability at most ε. Given a distribution μ on $\mathcal{B}$, let $D_\varepsilon^\mu[\mathcal{B}]$ be the minimum number of function evaluations by a deterministic algorithm that, for a function f sampled according to μ, finds G_f with error probability at most ε. Then

$$R_\varepsilon[\mathcal{B}] = \max_\mu D_\varepsilon^\mu[\mathcal{B}].$$

Recall that $\mathcal{B}(n, m, k)$ is the set of k-bounded pseudo-Boolean functions defined on $\{0,1\}^n$ whose linkage graphs have m maximal hyperedges. We use Corollary 1 to obtain a lower bound for linkage discovery.

Lemma 3: Given $0 < \varepsilon < 1$, any randomized algorithm that, for any $f \in \mathcal{B}(n, m, k)$, finds G_f with error probability at most ε requires at least

$$\frac{\log_2 (1 - \varepsilon) \binom{\binom{n}{k}}{m}}{\log_2 (m + 1)}$$

function evaluations provided that $m \leq \binom{n}{k}$.

Proof: To prove the lemma, we define a probability distribution μ on the set $\mathcal{B}(n, m, k)$. Then, when a function f is given according to μ, it is shown that any deterministic algorithm requires at least

$$\frac{\log_2 (1 - \varepsilon) \binom{\binom{n}{k}}{m}}{\log_2 (m + 1)}$$

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

function evaluations to find G_f with error probability at most ε. Lemma 3 follows from Corollary 1.

Let $\mathcal{S}$ be the set of all P's of the form $P = \{H_1, H_2 \ldots, H_m\}$, where the H_i's $(1 \leq i \leq m)$ are distinct k-order subsets of $[n]$. For each $P \in \mathcal{S}$, we assign a k-bounded pseudo-Boolean function f_P so that P forms exactly the set of maximal hyperedges of G_{f_P}. Suppose that $P = \{H_1, H_2, \ldots, H_m\}$ and $H_i = \{i_1, i_2, \ldots, i_k\}$ for $1 \leq i \leq m$. Define f_i as $f_i(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) = x_{i_1} x_{i_2} \cdots x_{i_k}$ and let

$$f_P = \sum_{i=1}^{m} f_i.$$

The definition of the linkage test function $\mathfrak{L}$ implies that, for any $x \in \{0,1\}^n$

$$\mathfrak{L}(f_P, H_i, x) = \mathfrak{L}\left(\sum_{l=1}^{m} f_l, H_i, x\right) = \sum_{l=1}^{m} \mathfrak{L}(f_l, H_i, x).$$

For any $x \in \{0,1\}^n, \mathfrak{L}(f_l, H_i, x) = 0$ for $l \neq i$ and thus $\mathfrak{L}(f_P, H_i, x) = \mathfrak{L}(f_i, H_i, x)$. Note that $\mathfrak{L}(f_i, H_i, 1^n) = 1 \neq 0$ for all i, where 1^n denotes the string consisting of n ones. Also, for all $x \in \{0,1\}^n$, $\mathfrak{L}(f_i, H, x) = 0$ for any $H \subseteq [n]$ such that $H \not\subseteq H_i$ for all i. Thus, by Lemma 1, we see that $P = \{H_1, H_2, \ldots, H_m\}$ forms exactly the set of maximal hyperedges of G_{f_P}.

Now we define μ as the uniform distribution over the set $\{f_P \mid P \in \mathcal{S}\}$. Then, consider a deterministic algorithm $\mathfrak{A}$ that takes a pseudo-Boolean function f according to μ as an input and outputs G_f. Suppose that $\mathfrak{A}$ performs at most τ function evaluations. Because a function in $\{f_P \mid P \in \mathcal{S}\}$ has at most $m{+}1$ values that are between 0 and m, there are at most $(m{+}1)^\tau$ different combinations of function values that $\mathfrak{A}$ gets from the functions in $\{f_P \mid P \in \mathcal{S}\}$. Also, $\mathfrak{A}$ is a deterministic algorithm and thus the output of $\mathfrak{A}$ is uniquely determined by a combination of function values. Hence, $\mathfrak{A}$ has at most $(m{+}1)^\tau$ different outputs. Because f_P's for $P \in \mathcal{S}$ have different linkage graphs from one another, the output of $\mathfrak{A}$ may be correct for at most $(m+1)^\tau$ inputs from $\{f_P \mid P \in \mathcal{S}\}$. From the fact that μ is the uniform distribution over $\{f_P \mid P \in \mathcal{S}\}$, we have

$$\Pr[\mathfrak{A} \text{ outputs correctly}] \leq \frac{(m+1)^\tau}{|\mathcal{S}|}.$$

Because

$$|\mathcal{S}| = \binom{\binom{n}{k}}{m}$$

the probability over μ that $\mathfrak{A}$ outputs correctly is less than $1 - \varepsilon$ unless

$$\tau \geq \frac{\log_2 (1-\varepsilon)\binom{\binom{n}{k}}{m}}{\log_2(m+1)}.$$

This completes the proof. □

Now, suppose that k is a constant independent of n, and $m = \mathcal{O}(n^{k-\delta})$ for some constant $\delta > 0$. Then, Stirling's formula [37] implies that

$$\binom{\binom{n}{k}}{m} = n^{\Omega(m)}.$$

For a constant $0 < \varepsilon < 1$

$$\frac{\log_2 (1-\varepsilon)\binom{\binom{n}{k}}{m}}{\log_2(m+1)} = \Omega\left(\frac{m \log n}{\log m}\right)$$

and Theorem 1 follows from Lemma 3.

IV. Upper Bound for Linkage Discovery

The number of function evaluations by HCA is proportional to the number of hyperedge candidates appearing in the process of running HCA. Thus, to obtain a good upper bound for the number of function evaluations by HCA, it is critical to count the number of such hyperedge candidates as tightly as possible. The Kruskal–Katona theorem, which was proved independently by Kruskal [27] and Katona [28], is useful for the following purpose. Given a collection of j-order subsets, it gives a tight lower bound for the number of $(j-1)$-order subsets included in the j-order subsets. The original version of the theorem is somewhat complicated, however, we use the version of Lovász [38], which is slightly weaker but easier to handle. In the following, the generalized binomial coefficient $\binom{s}{j}$ for a real number s and a positive integer j is defined as

$$\binom{s}{j} = \prod_{i=1}^{j} \left(\frac{s-j+i}{i}\right).$$

Let $\binom{s}{0} = 1$ for any real number s by convention.

Theorem 4 (Kruskal–Katona): Suppose that A is a collection of j-order subsets of $[n]$ and $|A| = \binom{s}{j}$, where s is a real number with $s \geq j-1$. Then, the number of $(j-1)$-order subsets of $[n]$ that are included in a j-order subset in A is at least $\binom{s}{j-1}$.

Lemma 4: Suppose that a hypergraph G has $r (j-1)$-order hyperedges for $j \geq 2$. Then, the number of j-order hyperedge candidates in G is at most $((j-1)/j) r^{j/(j-1)}$.

Proof: Let $r = \binom{s}{j-1}$, where s is a real number with $s \geq j-2$, and denote by $c_j(G)$ the number of j-order hyperedge candidates in G. We first prove that $c_j(G) \leq \binom{s}{j}$ by contradiction. Suppose that $c_j(G) > \binom{s}{j}$. The Kruskal–Katona theorem implies that there are more than $\binom{s}{j-1}$ $(j-1)$-order subsets of $[n]$ that are included in a j-order hyperedge candidate of G. Because every $(j-1)$-order subset of a j-order hyperedge candidate is a $(j-1)$-order hyperedge by the definition of a hyperedge candidate, G has more than $\binom{s}{j-1} (j-1)$-order hyperedges, which is a contradiction.

From the fact that $(s/(j-1))^{j-1} \leq \binom{s}{j-1} = r, s \leq (j-1) r^{1/(j-1)}$ and so

$$\begin{aligned} c_j(G) &\leq \binom{s}{j} \\ &= \left(\frac{s-j+1}{j}\right)\binom{s}{j-1} \\ &= \left(\frac{s-j+1}{j}\right) r \\ &\leq \left(\frac{(j-1) r^{\frac{1}{j-1}} - j + 1}{j}\right) r \\ &\leq \left(\frac{j-1}{j}\right) r^{\frac{j}{j-1}}. \end{aligned}$$

□

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

Lemma 5: Suppose that f is a k-bounded pseudo-Boolean function defined on $\{0,1\}^n$ and the number of maximal hyperedges in G_f is m. Then, HCA finds the linkage graph G_f in

$$\mathcal{O}\left(2^{2k}\left(\log\left(\frac{1}{\varepsilon}\right)+k\log n\right)\left(n+\sum_{j=2}^{k}\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)^{\frac{j}{j-1}}\right)\right)$$

function evaluations with error probability at most ε.

Proof: We first bound the number of function evaluations by HCA. Let

$$t=\left\lceil 2^k\log\frac{(2n)^k}{\varepsilon}\right\rceil.$$

Computing the value of $\mathfrak{L}(f,H,x)$ for a first-order subset H requires two function evaluations. Because there are n first-order subsets of $[n]$, the number of function evaluations in identifying the first-order hyperedges is $2tn$. Consider the number of hyperedge candidates appearing in the process of running HCA. For $2\le j\le k$, the number of $(j-1)$-order hyperedges is at most $\binom{n}{j-1}$, which is the number of all possible $(j-1)$-order subsets of $[n]$. Because the number of maximal hyperedges is m and every $(j-1)$-order hyperedge is contained in a maximal hyperedge, the number of $(j-1)$-order hyperedges is at most $m\binom{k}{j-1}$ as well. Hence, the number of $(j-1)$-order hyperedges is

$$\mathcal{O}\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)$$

and, by Lemma 4, the number of j-order hyperedge candidates is

$$\mathcal{O}\left(\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)^{j/(j-1)}\right).$$

Because computing the value of $\mathfrak{L}(f,H,x)$ for a j-order subset H requires 2^j function evaluations, the number of function evaluations in identifying the j-order hyperedges is thus

$$\mathcal{O}\left(2^j t\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)^{j/(j-1)}\right).$$

Hence, the total number of function evaluations by HCA is

$$\mathcal{O}\left(2^k t\left(n+\sum_{j=2}^{k}\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)^{j/(j-1)}\right)\right). \quad (1)$$

From the fact that $t=\lceil 2^k\log((2n)^k)/\varepsilon\rceil=\lceil 2^k(\log(1/\varepsilon)+k\log n+k)\rceil$, the proof for the number of function evaluations is completed.

Now, we complete the proof of the theorem by bounding the probability that HCA works incorrectly. Note that

$$\Pr[\text{HCA is incorrect}]=\Pr\left[\bigcup_{H\in E(G_f)}(H\text{ is not detected})\right]. \quad (2)$$

For a j-order hyperedge $H, \Pr[H\text{ is not detected}]\le(1-2^{j-k})^t\le(1-2^{-k})^t$ by Lemma 1 and so applying union bound to (2) gives

$$\begin{aligned}\Pr[\text{HCA is incorrect}]&\le\sum_{H\in E(G_f)}\Pr[H\text{ is not detected}]\\&\le\sum_{H\in E(G_f)}(1-2^{-k})^t.\end{aligned}$$

From the facts that $|E(G_f)|\le m\sum_{j=1}^{k}\binom{k}{j}\le 2^k m$ and $m\le\sum_{j=1}^{k}\binom{n}{j}\le n^k$

$$\begin{aligned}\Pr[\text{HCA is incorrect}]&\le|E(G_f)|(1-2^{-k})^t\\&\le(2n)^k(1-2^{-k})^t. \quad (3)\end{aligned}$$

By Taylor expansion, $-\log(1-2^{-k})\ge 2^{-k}$ and so $t=\lceil 2^k\log((2n)^k/\varepsilon)\rceil\ge-\log((2n)^k/\varepsilon)/\log(1-2^{-k})$. Hence

$$(1-2^{-k})^t\le(1-2^{-k})^{\frac{\log\frac{\varepsilon}{(2n)^k}}{\log(1-2^{-k})}}=\frac{\varepsilon}{(2n)^k}$$

and, from inequality (3)

$$\Pr[\text{HCA is incorrect}]\le(2n)^k(1-2^{-k})^t\le\varepsilon. \qquad \square$$

Now, we prove Theorem 2. Suppose that k is a constant independent of n, and $m=\Theta(n^{h+\alpha})$ for a nonnegative integer h and $0\le\alpha<1$. Note that

$$\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}=\mathcal{O}(n^{j-1})$$

for $j\le h+1$. The conditions that $m=n^{h+\alpha}$ and k is constant imply that

$$\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}=\mathcal{O}(n^{h+\alpha})$$

for $j\ge h+2$. From these facts, we see that

$$\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)^{\frac{j}{j-1}}=\begin{cases}\mathcal{O}(n^{h+1}), & \text{if } j\le h+1\\ \mathcal{O}\left(n^{\frac{(h+\alpha)(h+2)}{h+1}}\right), & \text{if } j\ge h+2.\end{cases}$$

Hence

$$\max_{2\le j\le k}\left(\min\left\{\binom{n}{j-1},m\binom{k}{j-1}\right\}\right)^{\frac{j}{j-1}}=\mathcal{O}\left(n^{\max\left\{\frac{(h+\alpha)(h+2)}{h+1},h+1\right\}}\right). \quad (4)$$

Because $t=\mathcal{O}(\log n)$ for constant k, from (1) and (4), the total number of function evaluations by HCA is $\mathcal{O}(n^{\hat{q}}\log n)$ with

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

$\hat{q} = \max\{(h+\alpha)(h+2)/(h+1), h+1\}$. This completes the proof of Theorem 2.

V. Behaviors on Random Models

If a fitness function f is generated from $\mathcal{F}(n,p,k), \tilde{\mathcal{F}}(n,\tilde{m},k)$, or $\tilde{\mathcal{F}}'(n,\tilde{m},k)$, it is clear that f is k-bounded. Thus, Lemma 5 guarantees that HCA finds the linkage graph of f with a bounded error. In the following sections, we show that the number of function evaluations by HCA on the random models is generally smaller than the bound for the worst case.

A. Uniform Probability Model

Lemma 6: Suppose that f is generated from $\mathcal{F}(n,p,k)$ and $j \geq 2$. For a j-order subset H of $[n]$, the probability that H is a hyperedge candidate of G_f is at most

$$\pi_j(n,p,k) = 1-(1-p)^{\binom{n-j}{k-j}} + \left(1-(1-p)^{\binom{n-j}{k-j+1}}\right)^j.$$

Proof: We see that H is a hyperedge candidate if and only if H is a hyperedge or H is not a hyperedge but a hyperedge candidate. First, consider the case that H is a hyperedge. Denote by $C(H)$ the collection of k-order subsets of $[n]$ including H. Then, the probability that H is a hyperedge is at most the probability that at least one k-order subset in $C(H)$ is chosen. Because the size of $C(H)$ is $\binom{n-j}{k-j}$ and each k-order subset is chosen with probability p and independently of the others, the probability that H is a hyperedge is at most

$$1-(1-p)^{\binom{n-j}{k-j}}. \tag{5}$$

Now, consider the condition that H is not a hyperedge. For each $(j-1)$-order subset H_i of H, denote by $C'(H_i)$ the collection of k-order subsets of $[n]$ including H' but not H. Then, the probability that H is a hyperedge candidate, conditioned that H is not a hyperedge, is at most the probability that at least one k-order subset in $C'(H_i)$ is chosen for each H_i. Because $C'(H_i)$'s are disjoint, the conditional probability is at most

$$\left(1-(1-p)^{\binom{n-j}{k-j+1}}\right)^j. \tag{6}$$

The probability that H is not a hyperedge is at most one and so the probability that H is a hyperedge candidate is at most

$$1-(1-p)^{\binom{n-j}{k-j}} + \left(1-(1-p)^{\binom{n-j}{k-j+1}}\right)^j$$

from (5) and (6). □

Lemma 7: Suppose that f is generated from $\mathcal{F}(n,p,k)$. Then, HCA finds the linkage graph G_f in

$$\mathcal{O}\left(2^{2k}\left(\log\left(\frac{1}{\varepsilon}\right)+k\log n\right)\left(n+\sum_{j=2}^{k}\binom{n}{j}\pi_j(n,p,k)\right)\right)$$

function evaluations on average with error probability at most ε.

Proof: Letting

$$t = \left\lceil 2^k \log\frac{(2n)^k}{\varepsilon}\right\rceil$$

the expected number of function evaluations by HCA is

$$\mathcal{O}\left(2^k t\left(n+\sum_{j=2}^{k}\binom{n}{j}\pi_j(n,p,k)\right)\right)$$

by using Lemma 6 and linearity of expectation. The proof follows by the fact that $t = \mathcal{O}(2^k(\log(1/\varepsilon)+k\log n))$. □

Proposition 2: Suppose that f is generated from $\mathcal{F}(n,p,k)$, k is a constant independent of n, and $\binom{n}{k}p = \Theta(n^{h+\alpha})$ for a nonnegative integer h and $0 \leq \alpha < 1$. Then, HCA finds the linkage graph G_f in $\mathcal{O}(n^{\tilde{q}}\log n)$ function evaluations on average with error probability at most ε, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

Proof: For $2 \leq j \leq h+1$, it is trivially true that $\pi_j(n,p,k) \leq 1$ and $\binom{n}{j}\pi_j(n,p,k) = \mathcal{O}(n^{h+1})$. Because k is fixed, for $h+2 \leq j \leq k$

$$1-(1-p)^{\binom{n-j}{k-j}} = \mathcal{O}(n^{h+\alpha-j})$$

and

$$\left(1-(1-p)^{\binom{n-j}{k-j+1}}\right)^j = \mathcal{O}(n^{(h+\alpha+1-j)j})$$

by plugging into

$$p = \Theta\left(\frac{n^{h+\alpha}}{\binom{n}{k}}\right)$$

and using the fact that $1-(1-x)^a \leq ax$ when $0 \leq x \leq 1$ and $a \geq 1$. Thus

$$\pi_j(n,p,k) = \mathcal{O}(n^{(h+\alpha+1-j)j} + n^{h+\alpha-j})$$

and

$$\binom{n}{j}\pi_j(n,p,k) = \mathcal{O}(n^{(h+\alpha+2-j)j} + n^{h+\alpha}).$$

Hence, from Lemma 7, the expected number of function evaluations by HCA is

$$\mathcal{O}\left(t\left(n+n^{h+1}+\max_{h+2\leq j\leq k} n^{(h+\alpha+2-j)j}+n^{h+\alpha}\right)\right).$$

The proof follows by the facts that

$$\max_{h+2\leq j\leq k}(h+\alpha+2-j)j = \alpha(h+2)$$

$n^{h+\alpha} = \mathcal{O}(n^{h+1})$ and $t = \mathcal{O}(\log n)$. □

Let $g(n)$ be any positive valued function such that $\lim_{n\to\infty} g(n) = \infty$. When k is a constant independent of n, Proposition 2 combined with Markov inequality [33] implies that the number of function evaluations of HCA to guarantee the error probability at most ε is almost always $\mathcal{O}(g(n)n^{\tilde{q}}\log n)$, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$. By applying the second moment method [33], it is possible to obtain a stronger result in which the term $g(n)$ is removed.

Theorem 5 (Theorem 3 for the Uniform Probability Model): Suppose that f is generated from $\mathcal{F}(n,p,k)$, k is a constant independent of n, and $\binom{n}{k}p = \Theta(n^{h+\alpha})$ for a nonnegative in-

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

teger h and $0 \leq \alpha < 1$. Then, for any constant $\varepsilon > 0$, the number of function evaluations of HCA to guarantee the error probability at most ε is almost always $\mathcal{O}(n^{\tilde{q}} \log n)$, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

Proof: See the Appendix. □

B. Uniform Density Models

We first consider the number of function evaluations by HCA for a fitness function generated from the uniform density model without replacement. As in the uniform probability model, we start with the lemma for the probability that a hyperedge candidate occurs. To prove the lemma, the relation between the uniform probability model and the uniform density model without replacement is exploited.

Lemma 8: Suppose that f is generated from $\tilde{\mathcal{F}}(n, \tilde{m}, k)$ and $j \geq 2$. For a j-order subset H of $[n]$, the probability that H is a hyperedge candidate of G_f is $\mathcal{O}(\tilde{\pi}_j(n, \tilde{m}, k, \delta))$ for any constant $\delta > 0$, where

$$\tilde{\pi}_j(n, \tilde{m}, k, \delta) = \frac{(1+\delta)\binom{k}{j}\tilde{m}}{\binom{n}{j}} + \left(\frac{(1+\delta)\binom{k}{j-1}\tilde{m}}{\binom{n}{j-1}}\right)^j + \exp\left(-\frac{\delta^2 \tilde{m}}{2(1+\delta)^2}\right).$$

Proof: Let $\tilde{E}_H$ be the event that a k-order subset including H is chosen or, for each $(j-1)$-order subset H' of H, a k-order subset including H' is chosen when f is generated from $\tilde{\mathcal{F}}(n, \tilde{m}, k)$. Because the probability that H is a hyperedge candidate is at most the probability that the event $\tilde{E}_H$ occurs, it is enough to show that

$$\Pr[\tilde{E}_H] = \mathcal{O}(\tilde{\pi}_j(n, \tilde{m}, k, \delta)).$$

Let

$$p = \frac{(1+\delta)\tilde{m}}{\binom{n}{k}}$$

and consider the uniform probability model $\mathcal{F}(n, p, k)$. Let E_H be the event that a k-order subset including H is chosen or, for each $(j-1)$-order subset H' of H, a k-order subset including H' is chosen from $\mathcal{F}(n, p, k)$ and let U be the number of k-order subsets of $[n]$ chosen from $\mathcal{F}(n, p, k)$. Then

$$\begin{aligned}\Pr[E_H] &= \sum_{i=0}^{\binom{n}{k}} \Pr[U = i] \cdot \Pr[E_H | U = i] \\ &\geq \sum_{i \geq \tilde{m}} \Pr[U = i] \cdot \Pr[E_H | U = i].\end{aligned}$$

Because $\Pr[E_H | U = \tilde{m}] \leq \Pr[E_H | U = i]$ for $i \geq \tilde{m}$ and $\Pr[E_H | U = \tilde{m}] = \Pr[\tilde{E}_H]$

$$\begin{aligned}\Pr[E_H] &\geq \sum_{i \geq \tilde{m}} \Pr[U = i] \cdot \Pr[E_H | U = \tilde{m}] \\ &= \Pr[U \geq \tilde{m}] \cdot \Pr[\tilde{E}_H] \\ &= (1 - \Pr[U < \tilde{m}]) \cdot \Pr[\tilde{E}_H] \\ &\geq \Pr[\tilde{E}_H] - \Pr[U < \tilde{m}].\end{aligned}$$

Hence

$$\Pr[\tilde{E}_H] \leq \Pr[E_H] + \Pr[U < \tilde{m}]. \tag{7}$$

As shown in the proof of Lemma 6

$$\begin{aligned}\Pr[E_H] &\leq \pi_j(n, p, k) \\ &= 1 - (1-p)^{\binom{n-j}{k-j}} + \left(1 - (1-p)^{\binom{n-j}{k-j+1}}\right)^j.\end{aligned}$$

By plugging into

$$p = \frac{(1+\delta)\tilde{m}}{\binom{n}{k}}$$

and using the fact that $1 - (1-x)^a \leq ax$ when $0 \leq x \leq 1$ and $a \geq 1$, we see that

$$1 - (1-p)^{\binom{n-j}{k-j}} = \mathcal{O}\left(\frac{(1+\delta)\binom{k}{j}\tilde{m}}{\binom{n}{j}}\right)$$

and

$$\left(1 - (1-p)^{\binom{n-j}{k-j+1}}\right)^j = \mathcal{O}\left(\left(\frac{(1+\delta)\binom{k}{j-1}\tilde{m}}{\binom{n}{j-1}}\right)^j\right).$$

Thus

$$\Pr[E_H] = \mathcal{O}\left(\frac{(1+\delta)\binom{k}{j}\tilde{m}}{\binom{n}{j}} + \left(\frac{(1+\delta)\binom{k}{j-1}\tilde{m}}{\binom{n}{j-1}}\right)^j\right). \tag{8}$$

Because U has the binomial distribution with parameters $\binom{n}{k}$ and

$$p = \frac{(1+\delta)\tilde{m}}{\binom{n}{k}}$$

then

$$\begin{aligned}\Pr[U < \tilde{m}] &= \Pr\left[U < \left(1 - \frac{\delta}{1+\delta}\right)(1+\delta)\tilde{m}\right] \\ &\leq \exp\left(-\frac{\delta^2 \tilde{m}}{2(1+\delta)^2}\right)\end{aligned} \tag{9}$$

by Chernoff bound [33]. Therefore, $\Pr[\tilde{E}_H] = \mathcal{O}(\tilde{\pi}_j(n, \tilde{m}, k, \delta))$ from (7)–(9). □

Lemma 9: Suppose that f is generated from $\tilde{\mathcal{F}}(n, \tilde{m}, k)$. Then, for an arbitrarily small constant $\delta > 0$, HCA finds the linkage graph G_f in

$$\mathcal{O}\left(2^{2k}\left(\log\left(\frac{1}{\varepsilon}\right) + k \log n\right) \times \left(n + \sum_{j=2}^{k} \binom{n}{j} \tilde{\pi}_j(n, \tilde{m}, k, \delta)\right)\right)$$

function evaluations on average with error probability at most ε.

Proof: Let

$$t = \left\lceil 2^k \log \frac{(2n)^k}{\varepsilon} \right\rceil.$$

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

For arbitrarily small $\delta > 0$, the expected number of function evaluations by HCA is

$$\mathcal{O}\left(2^k t\left(n+\sum_{j=2}^{k}\binom{n}{j}\tilde{\pi}_j(n,\tilde{m},k,\delta)\right)\right)$$

by using Lemma 8 and linearity of expectation. The proof follows by the fact that $t=\mathcal{O}(2^k(\log(1/\varepsilon)+k\log n))$. □

Proposition 3: Suppose that f is generated from $\tilde{\mathcal{F}}(n,\tilde{m},k), k$ is a constant independent of n, and $\tilde{m} = \Theta(n^{h+\alpha})$ for a nonnegative integer h and $0 \le \alpha < 1$. Then, HCA finds the linkage graph G_f in $\mathcal{O}(n^{\tilde{q}}\log n)$ function evaluations on average with error probability at most ε, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

Proof: Suppose that $h = 0$. Because $\tilde{m} = \Theta(n^{\alpha})$ and k is constant, the number of maximal hyperedges in G_f is $\mathcal{O}(n^{\alpha})$ with probability 1. From Theorem 2, the number of function evaluations by HCA is $\mathcal{O}(n^{\max\{2\alpha,1\}}\log n)$ with Probability 1 and the proposition follows.

Now, suppose that $h \ge 1$ and δ is an arbitrary positive constant. For $2 \le j \le h+1$, it is trivially true that $\tilde{\pi}_j(n,\tilde{m},k,\delta) = \mathcal{O}(1)$ and $\binom{n}{j}\tilde{\pi}_j(n,\tilde{m},k,\delta) = \mathcal{O}(n^{h+1})$ for constant k. Because $\tilde{m} = n^{h+\alpha}$ with $h \ge 1$ and k is constant, $\exp(-\delta^2\tilde{m}/(2(1+\delta)^2))$ is exponentially small in n and so

$$\tilde{\pi}_j(n,\tilde{m},k,\delta) = \mathcal{O}(n^{(h+\alpha+1-j)j} + n^{h+\alpha-j})$$

for $h+2 \le j \le k$. Hence

$$\binom{n}{j}\tilde{\pi}_j(n,\tilde{m},k,\delta) = \mathcal{O}(n^{(h+\alpha+2-j)j} + n^{h+\alpha}).$$

From Lemma 9, the expected number of function evaluations by HCA is

$$\mathcal{O}\left(t\left(n^{h+1} + \max_{h+2\le j\le k} n^{(h+\alpha+2-j)j} + n^{h+\alpha}\right)\right).$$

The proof follows by the facts that

$$\max_{h+2\le j\le k}(h+\alpha+2-j)j = \alpha(h+2)$$

$n^{h+\alpha} = \mathcal{O}(n^{h+1})$ and $t = \mathcal{O}(\log n)$. □

By coupling with the result for $\mathcal{F}(n,p,k)$, Theorem 3 for the uniform density model without replacement is obtained.

Theorem 6 (Theorem 3 for the Uniform Density Model Without Replacement): Suppose that f is generated from $\tilde{\mathcal{F}}(n,\tilde{m},k), k$ is a constant independent of n, and $\tilde{m} = \Theta(n^{h+\alpha})$ for a nonnegative integer h and $0 \le \alpha < 1$. Then, for any constant $\varepsilon > 0$, the number of function evaluations of HCA to guarantee the error probability at most ε is almost always $\mathcal{O}(n^{\tilde{q}}\log n)$, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

Proof: See the Appendix. □

Now, we consider the number of function evaluations by HCA on the uniform density model with replacement. Based on the relation between the uniform density models, the probability that a given subset is a hyperedge candidate is bounded as follows.

Lemma 10: Suppose that f is generated from $\tilde{\mathcal{F}}'(n,\tilde{m},k)$ and $j \ge 2$. For a j-order subset H of $[n]$, the probability that H is a hyperedge candidate of G_f is $\mathcal{O}(\tilde{\pi}_j(n,\tilde{m},k,\delta))$ for any constant $\delta > 0$.

Proof: Let $\tilde{E}_H$ and $\tilde{E}'_H$ be the events that a k-order subset including H is chosen or, for each $(j-1)$-order subset H' of H, a k-order subset including H' is chosen from $\tilde{\mathcal{F}}(n,\tilde{m},k)$ and $\tilde{\mathcal{F}}'(n,\tilde{m},k)$, respectively. The probability that H is a hyperedge candidate when f is generated from $\tilde{\mathcal{F}}'(n,\tilde{m},k)$ is at most the value of $\Pr[\tilde{E}'_H]$. Because $\Pr[\tilde{E}_H] = \mathcal{O}(\tilde{\pi}_j(n,\tilde{m},k,\delta))$ as shown in the proof of Lemma 8, it is enough to show that $\Pr[\tilde{E}'_H] \le \Pr[\tilde{E}_H]$.

Let $\tilde{U}'$ be the number of distinct k-order subsets of $[n]$ chosen from $\tilde{\mathcal{F}}'(n,\tilde{m},k)$. Because $\Pr[\tilde{E}'_H|\tilde{U}' = i] \le \Pr[\tilde{E}'_H|\tilde{U}' = \tilde{m}]$ for $i \le \tilde{m}$ and $\Pr[\tilde{E}'_H|\tilde{U}' = \tilde{m}] = \Pr[\tilde{E}_H]$

$$\begin{aligned}\Pr[\tilde{E}'_H] &= \sum_{i=1}^{\tilde{m}}\Pr[\tilde{U}' = i]\cdot\Pr[\tilde{E}'_H|\tilde{U}' = i]\\ &\le \sum_{i=1}^{\tilde{m}}\Pr[\tilde{U}' = i]\cdot\Pr[\tilde{E}'_H|\tilde{U}' = \tilde{m}]\\ &= \sum_{i=1}^{\tilde{m}}\Pr[\tilde{U}' = i]\cdot\Pr[\tilde{E}_H]\\ &= \Pr[\tilde{E}_H].\end{aligned}$$

Using Lemma 10, we have the following theorems on $\tilde{\mathcal{F}}'(n,\tilde{m},k)$. The proofs are omitted, which are analogous to those of Lemma 9 and Proposition 3.

Lemma 11: Suppose that f is generated from $\tilde{\mathcal{F}}'(n,\tilde{m},k)$. Then, for an arbitrarily small constant $\delta > 0$, HCA finds the linkage graph G_f in

$$\mathcal{O}\left(2^{2k}\left(\log\left(\frac{1}{\varepsilon}\right)+k\log n\right)\left(n+\sum_{j=2}^{k}\binom{n}{j}\tilde{\pi}_j(n,\tilde{m},k,\delta)\right)\right)$$

function evaluations on average with error probability at most ε.

Proposition 4: Suppose that f is generated from $\tilde{\mathcal{F}}'(n,\tilde{m},k), k$ is a constant independent of n, and $\tilde{m} = \Theta(n^{h+\alpha})$ for a nonnegative integer h and $0 \le \alpha < 1$. Then, HCA finds the linkage graph G_f in $\mathcal{O}(n^{\tilde{q}}\log n)$ function evaluations on average with error probability at most ε, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

By coupling with the result for $\tilde{\mathcal{F}}(n,\tilde{m},k)$, Theorem 3 for the uniform density model with replacement is obtained.

Theorem 7 (Theorem 3 for the Uniform Density Model With Replacement): Suppose that f is generated from $\tilde{\mathcal{F}}'(n,\tilde{m},k), k$ is a constant independent of n, and $\tilde{m} = \Theta(n^{h+\alpha})$ for a nonnegative integer h and $0 \le \alpha < 1$. Then, for any constant $\varepsilon > 0$, the number of function evaluations of HCA to guarantee the error probability at most ε is almost always $\mathcal{O}(n^{\tilde{q}}\log n)$, where $\tilde{q} = \max\{\alpha(h+2), h+1\}$.

Proof: See the Appendix. □

VI. Remarks on Finding Walsh Coefficients

Suppose that f is a pseudo-Boolean function and m is the number of maximal hyperedges in G_f. As shown in Section II, once G_f is discovered, it is possible to find the Walsh coefficients of f with $\mathcal{O}(2^k m)$ additional function evaluations from Lemma 2. In the case that k is a constant independent of n,

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

this means that the proposed upper bounds for discovering the linkage graph (Theorems 2 and 3) are also valid as the upper bounds for finding Walsh coefficients. Conversely, suppose that all the Walsh coefficients are known for a pseudo-Boolean function f. Then, from the maximal nonzero Walsh coefficients of f, maximal hyperedges of G_f are automatically identified by Lemma 2 a). The remaining hyperedges are discovered from the maximal hyperedges by Lemma 1 c). This implies that the number of function evaluations required for finding Walsh coefficients is lower bounded by the number of function evaluations required for discovering the linkage graph. Thus, we see that the proposed lower bound for discovering the linkage graph (Theorem 1) is also valid as the lower bound for finding Walsh coefficients.

VII. Discussion and Future Work

The problem of discovering linkage structures was investigated in the black box scenario. Based on a formal definition of linkage, the lower and upper bounds for linkage discovery were rigorously analyzed. The lower bounds that we obtained are the first results for linkage discovery. They may serve as a basis for analyzing and evaluating evolutionary algorithms based on linkage discovery. The upper bounds imply that linkage discovery and even Walsh analysis can be accomplished efficiently for fitness functions of many NP-hard problems. Through the investigation on random models, it was shown that linkage discovery and Walsh analysis are achieved more efficiently in typical situations.

Although the lower and upper bounds are within a reasonable factor, we consider that there remains room for further improvement, in particular, for the upper bounds. Recently, Choi *et al.* [39] showed that the problem for checking the linear separability of a two-bounded function is solved in a constant number of function evaluations by a method perturbing a group of bits simultaneously. For bounded functions, the effectiveness and efficiency of a group-perturbation method were empirically verified in [40]. Generalizing the linkage test so as to permit group perturbations seems to be a promising approach for more efficient linkage discovery. We are currently working on improving the upper bounds for linkage discovery in this direction.

Appendix

Proof of Theorem 5: Suppose that $h = 0$. If $\alpha \leq 0.5$, the number of maximal hyperedges in G_f is almost always $\mathcal{O}(n^{0.5})$ by Chernoff bound [33]. From Theorem 2, the number of function evaluations by HCA is almost always $\mathcal{O}(n \log n)$ and the theorem follows. If $0.5 < \alpha < 1$, the number of maximal hyperedges in G_f is almost always $\mathcal{O}(n^{\alpha})$ by Chernoff bound. From Theorem 2, the number of function evaluations by HCA is almost always $\mathcal{O}(n^{2\alpha} \log n)$ and the theorem follows.

Now, we assume that $h \geq 1$ in the rest of the proof. Let X, Y, and Z denote the numbers of hyperedge candidates, hyperedge candidates that are indeed hyperedges, and hyperedge candidates that are not hyperedges, respectively, in G_f. It is clear that $X = Y + Z$.

Lemma 12: Suppose that $\binom{n}{k}p = \Theta(n^{h+\alpha})$ for $h \geq 1$. Then, there is a constant c such that $\Pr[Z > cn^{\tilde{q}}] = o(1)$.

Proof: For each j-order subset H_{ji} of $[n]$, set an indicator random variable Z_{ji} such that $Z_{ji} = 1$ if H_{ji} is not a hyperedge but a hyperedge candidate of G_f and $Z_{ji} = 0$ otherwise. If we let Z_j be the number of j-order hyperedge candidates of G_f that are not hyperedges, then $Z_j = \sum_i Z_{ji}$ and $Z = \sum_{j=2}^{k} Z_j$. We prove that there exists a positive constant c_j such that $\Pr[Z_j > c_j n^{\tilde{q}}] = o(1)$ for each $2 \leq j \leq k$. Then, by letting $c = \max_{2 \leq j \leq k} c_j$, the proof of the lemma follows.

For $2 \leq j \leq h+1$, it is trivially true that $Z_j \leq \binom{n}{h+1} = \mathcal{O}(n^{\tilde{q}})$ and so we may choose a positive constant c_j such that $\Pr[Z_j > c_j n^{\tilde{q}}] = o(1)$. Consider the case that $h + 2 \leq j \leq k$. For each Z_{ji}, we have

$$\begin{aligned} \mathrm{E}[Z_{ji}] &= \Pr[Z_{ji} = 1] \\ &\leq (1-p)^{\binom{n-j}{k-j}} \left(1 - (1-p)^{\binom{n-j}{k-(j-1)}}\right)^j \\ &= \mathcal{O}(n^{(h+\alpha+1-j)j}) \end{aligned} \tag{10}$$

and

$$\begin{aligned} \mathrm{E}[Z_j] &= \sum_i \mathrm{E}[Z_{ji}] \\ &\leq \binom{n}{j} (1-p)^{\binom{n-j}{k-j}} \left(1 - (1-p)^{\binom{n-j}{k-(j-1)}}\right)^j \\ &= \mathcal{O}(n^{(h+\alpha+2-j)j}). \end{aligned}$$

From the facts that $\max_{h+2 \leq j \leq k}(h+\alpha+2-j)j = \alpha(h+2)$ and $n^{\alpha(h+2)} = \mathcal{O}(n^{\tilde{q}}), \mathrm{E}[Z_j] = \mathcal{O}(n^{\tilde{q}})$ and thus we may choose a positive constant c_0 such that $\mathrm{E}[Z_j] \leq c_0 n^{\tilde{q}}$ for sufficiently large n. Letting $c_j = 2c_0$, by Chebyshev inequality [33]

$$\begin{aligned} \Pr[Z_j > c_j n^{\tilde{q}}] &\leq \Pr[|Z_j - \mathrm{E}[Z_j]| > c_0 n^{\tilde{q}}] \\ &\leq \frac{\mathrm{Var}[Z_j]}{c_0^2 n^{2\tilde{q}}}. \end{aligned} \tag{11}$$

Because $Z_j = \sum_i Z_{ji}$, the following holds:

$$\mathrm{Var}[Z_j] = \sum_i \mathrm{Var}[Z_{ji}] + \sum_{i \neq i'} \mathrm{Cov}[Z_{ji}, Z_{ji'}]. \tag{12}$$

We first consider the variance of Z_{ji}.

Fact 1: Suppose that $j \geq h+2$. For each j-order subset H_{ji}

$$\mathrm{Var}[Z_{ji}] = \mathcal{O}(n^{(h+\alpha+1-j)j}).$$

Proof: From (10)

$$\mathrm{Var}[Z_{ji}] = \mathrm{E}[Z_{ji}] - (\mathrm{E}[Z_{ji}])^2 = \mathcal{O}(n^{(h+\alpha+1-j)j}).$$

□

Now, consider the covariance of Z_{ji} and $Z_{ji'}$. Suppose that A and A' are $(j-1)$-order subsets of H_{ji} and $H_{ji'}$, respectively. Then, (A, A') is called a *dependent pair* in H_{ji} and $H_{ji'}$ if $A \cup A' \not\supseteq H_{ji}$ and $A \cup A' \not\supseteq H_{ji'}$. More specifically, a dependent pair (A, A') is called of type 1 if $A \cup A' \supseteq H_{ji} \cap H_{ji'}$ and it is called of type 2 otherwise.

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

Fact 2: Suppose that $j \geq h+2$ and $0 \leq s < j-1$. For two j-order subsets H_{ji} and $H_{ji'}$ of $[n]$ such that $|H_{ji} \cap H_{ji'}| = s$

$$\mathrm{Cov}[Z_{ji}, Z_{ji'}] = \mathcal{O}\left(n^{v(j,s)}\right)$$

where $v(j,s) = \max\{v_1(j,s), v_2(j,s)\}$ with $v_1(j,s) = (h+\alpha+1-j)(2j-1) + \max_{1 \leq r \leq j-1}(s-h-\alpha)r$ and $v_2(j,s) = (h+\alpha-(2j-s-1))j + \min\{(j-s)^2, j\}$.

Proof: Note that there are $(j-s)^2+s$ dependent pairs in H_{ji} and $H_{ji'}$ among which $(j-s)^2$ pairs are of type 1 and s pairs are of type 2. Label the $(j-s)^2+s$ dependent pairs, say (A_l, A'_l) for $1 \leq l \leq (j-s)^2+s$. For each dependent pair (A_l, A'_l), set an indicator random variable D_l such that $D_l = 1$ if $A_l \cup A'_l$ is included in a k-order subset chosen in the course of generating f and $D_l = 0$ otherwise. Let $D = \sum_l D_l$.

The value of $\Pr[Z_{ji} = 1 \text{ and } Z_{ji'} = 1]$ is decomposed in terms of the value of D as follows:

$$\begin{aligned}\Pr[Z_{ji} = 1 \text{ and } Z_{ji'} = 1]\\ = \sum_{r=0}^{(j-s)^2+s} \Pr[D = r, Z_{ji} = 1, \text{and } Z_{ji'} = 1].\end{aligned} \tag{13}$$

Let $\mathcal{I}_r$ be the collection of the r-order subsets of $[j] = \{1, 2, \ldots, j\}$. Suppose that $1 \leq r \leq j-1$. We have (14), shown at the bottom of the page. Let $\mathcal{J}_I$ be the collection of the $(j-1)$-order subsets, B's, of H_{ji} or $H_{ji'}$ such that $B \nsubseteq A_l \cup A'_l$ for all $l \in I$. Because there are at least $2j-1$ $(j-1)$-order subsets that are included in H_{ji} or $H_{ji'}$ and at most two $(j-1)$-order subsets are concerned with each dependent pair, we see that $|\mathcal{J}_I| \geq 2j-1-2|I| = 2(j-r)-1$. For $B \in \mathcal{J}_I$, set an indicator random variable W_B such that $W_B = 1$ if a k-order subset, which includes B but includes neither H_{ji} nor $H_{ji'}$, is chosen in the course of generating f and $W_B = 0$ otherwise. Because D_l's with $l \in I$ and W_B's with $B \in \mathcal{J}_I$ are probabilistically independent

$$\begin{aligned}&\Pr[D_l = 1 \quad \forall\, l \in I, Z_{ji} = 1, \text{and } Z_{ji'} = 1]\\ &\quad \leq \Pr[D_l = 1 \quad \forall l \in I \text{ and } W_B = 1 \quad \forall B \in \mathcal{J}_I]\\ &\quad = \prod_{l \in I} \Pr[D_l = 1] \prod_{B \in \mathcal{J}_I} \Pr[W_B = 1].\end{aligned}$$

Because $|H_{ji} \cup H_{ji'}| = 2j-s$, for each $l \in I$

$$\begin{aligned}\Pr[D_l = 1] &\leq 1-(1-p)^{\binom{n-|H_{ji} \cup H_{ji'}|}{k-|A_l \cup A'_l|}}\\ &= 1-(1-p)^{\binom{n-(2j-s)}{k-|A_l \cup A'_l|}}.\end{aligned}$$

If (A_l, A'_l) is of type 1, $|A_l \cup A'_l| = 2j-s-2$ and

$$\begin{aligned}\Pr[D_l = 1] &\leq 1-(1-p)^{\binom{n-(2j-s)}{k-(2j-s-2)}}\\ &= \mathcal{O}(n^{h+\alpha-(2j-s-2)}).\end{aligned}$$

If (A_l, A'_l) is of type 2, $|A_l \cup A'_l| = 2j-s-1$ and

$$\begin{aligned}\Pr[D_l = 1] &\leq 1-(1-p)^{\binom{n-(2j-s)}{k-(2j-s-1)}}\\ &= \mathcal{O}(n^{h+\alpha-(2j-s-1)}).\end{aligned}$$

For $B \in \mathcal{J}_I$, consider the case that $B \supseteq H_{ji} \cap H_{ji'}$. There are $\binom{n-|H_{ji} \cup H_{ji'}|}{k-|B|}$ k-order subsets that include B but include neither H_{ji} nor $H_{ji'}$. Thus

$$\begin{aligned}\Pr[W_B = 1] &\leq 1-(1-p)^{\binom{n-|H_{ji} \cup H_{ji'}|}{k-|B|}}\\ &= 1-(1-p)^{\binom{n-(2j-s)}{k-(j-1)}} = \mathcal{O}(n^{h+\alpha-j+1}).\end{aligned}$$

For the case that $B \nsupseteq H_{ji} \cap H_{ji'}$, we have $\Pr[W_B = 1] = \mathcal{O}(n^{h+\alpha-j+1})$ in a similar way. Let $u = \min\{(j-s)^2, r\}$. Because there are at most u dependent pairs of type 1 among the r pairs, (A_l, A'_l)'s with $l \in I$, and $|\mathcal{J}_I| \geq 2(j-r)-1$, we have (15), shown at the bottom of the page. For the exponent of n in (15), because $u \leq r$, we see that

$$\begin{aligned}&(h+\alpha-(2j-s-2))u + (h+\alpha-(2j-s-1))(r-u)\\ &\quad + (h+\alpha-j+1)(2(j-r)-1)\\ &\quad = (h+\alpha-(2j-s-1))r + u\\ &\qquad + (h+\alpha-j+1)(2(j-r)-1)\\ &\quad \leq (h+\alpha-(2j-s-1))r + r\\ &\qquad + (h+\alpha-j+1)(2(j-r)-1)\\ &\quad = (h+\alpha+1-j)(2j-1) + (s-h-\alpha)r.\end{aligned}$$

Thus

$$\begin{aligned}&\Pr[D_l = 1 \quad \forall l \in I, Z_{ji} = 1, \text{and } Z_{ji'} = 1]\\ &\qquad = \mathcal{O}(n^{(h+\alpha+1-j)(2j-1)+(s-h-\alpha)r})\end{aligned}$$

for each $I \in \mathcal{I}_r$ and, from (14), we have

$$\begin{aligned}&\Pr[D = r, Z_{ji} = 1, \text{and } Z_{ji'} = 1]\\ &\qquad = \mathcal{O}(n^{(h+\alpha+1-j)(2j-1)+(s-h-\alpha)r})\end{aligned} \tag{16}$$

for $1 \leq r \leq j-1$.

Now, suppose that $r \geq j$. In this case

$$\begin{aligned}&\Pr[D_l = 1 \quad \forall\, l \in I, Z_{ji} = 1, \text{and } Z_{ji'} = 1]\\ &\quad \leq \Pr[D_l = 1 \quad \forall\, l \in I]\\ &\quad = \prod_{l \in I} \Pr[D_l = 1]\\ &\quad = \mathcal{O}(n^{(h+\alpha-(2j-s-2))u+(h+\alpha-(2j-s-1))(r-u)}).\end{aligned}$$

$$\begin{aligned}\Pr[D = r, Z_{ji} = 1, \text{and } Z_{ji'} = 1] &= \sum_{I \in \mathcal{I}_r} \Pr[D_l = 1 \quad \forall l \in I, D_l = 0\, \forall l \notin I, Z_{ji} = 1, \text{and } Z_{ji'} = 1]\\ &\leq \sum_{I \in \mathcal{I}_r} \Pr[D_l = 1 \quad \forall l\ \in I, Z_{ji} = 1, \text{and } Z_{ji'} = 1].\end{aligned} \tag{14}$$

$$\Pr[D_l = 1\, \forall l \in I, Z_{ji} = 1, \text{and } Z_{ji'} = 1] = \mathcal{O}\left(n^{(h+\alpha-(2j-s-2))u+(h+\alpha-(2j-s-1))(r-u)+(h+\alpha-j+1)(2(j-r)-1)}\right). \tag{15}$$

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

From the fact that

$$\begin{aligned}&(h+\alpha-(2j-s-2))u+(h+\alpha-(2j-s-1))(r-u)\\ &\quad=(h+\alpha-(2j-s-1))r+u\\ &\quad\le(h+\alpha-(2j-s-1))j+u\\ &\quad=(h+\alpha-(2j-s-1))j+\min\{(j-s)^2,j\}\end{aligned}$$

when $r\ge j, 0\le s\le j-1$, and $j\ge h+2$, we have

$$\begin{aligned}&\Pr[D_l=1\quad\forall l\in I, Z_{ji}=1,\text{and } Z_{ji'}=1]\\ &\qquad=\mathcal{O}\left(n^{(h+\alpha-(2j-s-1))j+\min\{(j-s)^2,j\}}\right)\end{aligned}$$

for each $I\in\mathcal{I}_j$ and, from (14)

$$\begin{aligned}&\Pr[D=r, Z_{ji}=1,\text{and } Z_{ji'}=1]\\ &\qquad=\mathcal{O}\left(n^{(h+\alpha-(2j-s-1))j+\min\{(j-s)^2,j\}}\right)\end{aligned}\tag{17}$$

for $r\ge j$.

From (13)

$$\begin{aligned}&\operatorname{Cov}[Z_{ji},Z_{ji'}]\\ &\quad=\Pr[Z_{ji}=1\text{ and }Z_{ji'}=1]\\ &\qquad-\Pr[Z_{ji}=1]\Pr[Z_{ji'}=1]\\ &\quad=\sum_{r=0}^{(j-s)^2+s}\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1]\\ &\qquad-\Pr[Z_{ji}=1]\Pr[Z_{ji'}=1]\\ &\quad=\Pr[D=0,Z_{ji}=1,\text{and }Z_{ji'}=1]\\ &\qquad-\Pr[Z_{ji}=1]\Pr[Z_{ji'}=1]\\ &\qquad+\sum_{r=1}^{(j-s)^2+s}\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1].\end{aligned}$$

Because

$$\begin{aligned}&\Pr[D=0,Z_{ji}=1,\text{and }Z_{ji'}=1]\\ &\quad=\Pr[D=0]\Pr[Z_{ji}=1\text{ and }Z_{ji'}=1|D=0]\\ &\quad\le\Pr[Z_{ji}=1\text{ and }Z_{ji'}=1|D=0]\\ &\quad=\Pr[Z_{ji}=1]\Pr[Z_{ji'}=1]\end{aligned}$$

when $0\le s=|H_{ji}\cap H_{ji'}|<j-1$,

$$\operatorname{Cov}[Z_{ji},Z_{ji'}]\le\sum_{r=1}^{(j-s)^2+s}\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1]\tag{18}$$

and the lemma follows from (16)–(18). □

Fact 3: Suppose that $j\ge h+2$. For two j-order subsets H_{ji} and $H_{ji'}$ of $[n]$ such that $|H_{ji}\cap H_{ji'}|=j-1$

$$\operatorname{Cov}[Z_{ji},Z_{ji'}]=\mathcal{O}\left(n^{(h+\alpha+1-j)j}\right).$$

Proof: In the same manner as in the proof of Fact 2, we have (19), shown at the bottom of the page, where $s=|H_{ji}\cap H_{ji'}|$. Plugging $s=j-1$ into (19)

$$\begin{aligned}&\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1]\\ &\qquad=\begin{cases}\mathcal{O}\left(n^{(h+\alpha+1-j)(2j-r-1)}\right), & \text{if } 0\le r\le j-1\\ \mathcal{O}\left(n^{(h+\alpha-j)j+1}\right), & \text{if } r\ge j.\end{cases}\end{aligned}$$

When $j\ge h+2$ and $0\le r\le j-1$, $(h+\alpha+1-j)(2j-r-1)\le(h+\alpha+1-j)j$ and $(h+\alpha-j)j+1\le(h+\alpha+1-j)j$, and so

$$\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1]=\mathcal{O}\left(n^{(h+\alpha+1-j)j}\right)$$

for all $r\ge 0$. Thus

$$\begin{aligned}&\Pr[Z_{ji}=1\text{ and }Z_{ji'}=1]\\ &\quad=\sum_{r\ge0}\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1]\\ &\quad=\mathcal{O}\left(n^{(h+\alpha+1-j)j}\right)\end{aligned}$$

and

$$\begin{aligned}&\operatorname{Cov}[Z_{ji},Z_{ji'}]\\ &\quad=\Pr[Z_{ji}=1\text{ and }Z_{ji'}=1]-\Pr[Z_{ji}=1]\Pr[Z_{ji'}=1]\\ &\quad\le\Pr[Z_{ji}=1\text{ and }Z_{ji'}=1]\\ &\quad=\mathcal{O}\left(n^{(h+\alpha+1-j)j}\right).\end{aligned}$$

□

Because there are $\mathcal{O}(n^j)$ j-order subsets and there are $\mathcal{O}(n^{2j-s})$ j-order subset pairs, $(H_{ji},H_{ji'})$'s with $|H_{ji}\cap H_{ji'}|=s$, from (12) and Facts 1–3

$$\begin{aligned}\operatorname{Var}[Z_j]=\mathcal{O}\Big(&n^{(h+\alpha+2-j)j}+n^{\max_{0\le s<j-1}v(j,s)+(2j-s)}\\ &+n^{(h+\alpha+2-j)j+1}\Big).\end{aligned}$$

Because $(h+\alpha+2-j)j<(h+\alpha+2-j)j+1$

$$\operatorname{Var}[Z_j]=\mathcal{O}\left(n^{\max_{0\le s<j-1}v(j,s)+(2j-s)}+n^{(h+\alpha+2-j)j+1}\right).$$

To prove that $\Pr[Z>cn^{\tilde{q}}]=o(1)$, from (11), it is enough to show that

$$\begin{aligned}&\frac{n^{\max_{0\le s<j-1}v(j,s)+(2j-s)}+n^{(h+\alpha+2-j)j+1}}{n^{2\tilde{q}}}\\ &\quad=n^{\max_{0\le s<j-1}v(j,s)+(2j-s)-2\tilde{q}}\\ &\qquad+n^{(h+\alpha+2-j)j+1-2\tilde{q}}\end{aligned}$$

approaches to zero as n goes to infinity. To this end, we show that both of $\max_{0\le s<j-1}v(j,s)+(2j-s)-2\tilde{q}$ and $(h+\alpha+2-j)j+1-2\tilde{q}$ are bounded above by a negative constant when $j\ge h+2$.

$$\Pr[D=r,Z_{ji}=1,\text{and }Z_{ji'}=1]=\begin{cases}\mathcal{O}\left(n^{(h+\alpha+1-j)(2j-1)+(s-h-\alpha)r}\right), & \text{if } 0\le r\le j-1\\ \mathcal{O}\left(n^{(h+\alpha-(2j-s-1))j+\min\{(j-s)^2,j\}}\right), & \text{if } r\ge j\end{cases}\tag{19}$$

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

Consider $v_1(j,s)+(2j-s)$ with $0 \le s < j-1$. For $1 \le r \le j-1, (r-1)s < (r-1)(j-1)$ and so

$$\begin{aligned}&(h+\alpha+1-j)(2j-1)+(s-h-\alpha)r+(2j-s)\\ &\quad=(h+\alpha+1-j)(2j-1)+2j+(r-1)s+(-h-\alpha)r\\ &\quad<(h+\alpha+1-j)(2j-1)+2j+(r-1)(j-1)+(-h-\alpha)r\\ &\quad=(h+\alpha+1-j)(2j-r-1)+j+1.\end{aligned}$$

Because $h+\alpha+1-j \le 0$ when $j \ge h+2$, $(h+\alpha+1-j)(2j-r-1)+j+1$ has the maximum value $(h+\alpha+1-j)j+j+1$ at $r=j-1$. Because $(h+\alpha+1-j)j+j+1 \le \alpha(h+2)+1$ when $j \ge h+2$, we have $(h+\alpha+1-j)(2j-1)+(s-h-\alpha)r+(2j-s) \le \alpha(h+2)+1$ for all $1 \le r \le j-1$. From the fact that $\tilde{q} \ge \alpha(h+2)$ and $\tilde{q} \ge 2$, $\alpha(h+2)+1-2\tilde{q} \le \alpha(h+2)+1-\alpha(h+2)-2=-1$ and thus $(h+\alpha+1-j)(2j-1)+(s-h-\alpha)r+(2j-s)-2\tilde{q} \le -1$ for all $1 \le r \le j-1$. Therefore

$$v_1(j,s)+(2j-s)-2\tilde{q} \le -1 \tag{20}$$

for all $0 \le s < j-1$.

Now, consider $v_2(j,s)+(2j-s)$ with $0 \le s < j-1$. In the case that $(j-s)^2 > j$

$$\begin{aligned}&v_2(j,s)+(2j-s)\\ &\quad=(h+\alpha-(2j-s-1))j+\min\{(j-s)^2,j\}+(2j-s)\\ &\quad=(h+\alpha-(2j-s-1))j+j+(2j-s)\\ &\quad=(h+\alpha+4-2j)j+(j-1)s.\end{aligned}$$

Because $(h+\alpha+4-2j)j+(j-1)s < (h+\alpha+4-2j)j+(j-1)^2=(h+\alpha+1-j)j+1$ when $0 \le s < j-1$ and $(h+\alpha+1-j)j+1 \le \alpha(h+2)+1$ for $j \ge h+2$, we have $v_2(j,s)+(2j-s) \le \alpha(h+2)+1$ for $0 \le s < j-1$. As shown in the above, $\alpha(h+2)+1-2\tilde{q} \le -1$ and thus $v_2(j,s)+(2j-s)-2\tilde{q} \le -1$ for all $0 \le s < j-1$. In the case that $(j-s)^2 \le j, j-\sqrt{j} \le s < j-1$ and

$$\begin{aligned}&v_2(j,s)+(2j-s)\\ &\quad=(h+\alpha-(2j-s-1))j+(j-s)^2+(2j-s)\\ &\quad=(h+\alpha+3-j)j+s(s-j-1)\\ &\quad\le(h+\alpha+3-j)j-2(j-1)\\ &\quad\le(h+\alpha+1-j)j+2.\end{aligned}$$

Because $(h+\alpha+1-j)j+2 \le \alpha(h+2)-h$ for $j \ge h+2$ and $\alpha(h+2)-h-2\tilde{q}=-h-2 \le -3$, $v_2(j,s)+(2j-s)-2\tilde{q} \le -3$ for all $0 \le s < j-1$. Therefore, in both cases

$$v_2(j,s)+(2j-s)-2\tilde{q} \le -1 \tag{21}$$

for all $0 \le s < j-1$.

From (20) and (21), we have $v(j,s)+(2j-s)-2\tilde{q} \le -1$ for all $0 \le s < j-1$. Hence

$$\max_{0 \le s < j-1} v(j,s)+(2j-s)-2\tilde{q} \le -1.$$

Finally, when $j \ge h+2$

$$(h+\alpha+2-j)j+1-2\tilde{q} \le \alpha(h+2)+1-2\tilde{q} \le 1-\tilde{q} \le -1$$

which completes the proof of the lemma. □

In a similar way, we have the following result for Y.

Lemma 13: Suppose that $\binom{n}{k}p=\Theta(n^{h+\alpha})$ for $h \ge 1$. Then, there is a constant c such that $\Pr[Y > cn^{\tilde{q}}]=o(1)$.

Proof: The proof is analogous to that of Lemma 12 and it is omitted. □

From Lemmas 12 and 13, we have the following.

Lemma 14: Suppose that $\binom{n}{k}p=\Theta(n^{h+\alpha})$ for $h \ge 1$. Then, there is a constant c such that $\Pr[X > cn^{\tilde{q}}]=o(1)$.

Proof: From Lemmas 12 and 13, we may choose the positive constants c_1 and c_2 such that $\Pr[Y > c_1 n^{\tilde{q}}]=o(1)$ and $\Pr[Z > c_2 n^{\tilde{q}}]=o(1)$. Let $c=c_1+c_2$. Then

$$\begin{aligned}\Pr[X > cn^{\tilde{q}}]&=\Pr[Y+Z > c_1 n^{\tilde{q}}+c_2 n^{\tilde{q}}]\\ &\le \Pr[Y > c_1 n^{\tilde{q}} \text{or} Z > c_2 n^{\tilde{q}}]\\ &\le \Pr[Y > c_1 n^{\tilde{q}}]+\Pr[Z > c_2 n^{\tilde{q}}]\\ &=o(1).\end{aligned}$$ □

Lemma 14 implies that the number of function evaluations by HCA is almost always $\mathcal{O}(tn^{\tilde{q}})$ for $t=\lceil 2^k \log((2n)^k/\varepsilon)\rceil$. Because $t=\mathcal{O}(\log n)$, the theorem follows. □

Proof of Theorem 6: Suppose that $h=0$. Because $\tilde{m}=\Theta(n^{\alpha})$ and k is constant, the number of maximal hyperedges in G_f is $\mathcal{O}(n^{\alpha})$ with probability 1. From Theorem 2, the number of function evaluations by HCA is $\mathcal{O}(n^{\max\{2\alpha,1\}}\log n)$ with probability 1 and the theorem follows.

Now, we assume that $h \ge 1$ in the rest of the proof. Let $\tilde{X}$ be the number of hyperedge candidates appearing in the process of running HCA on $\tilde{\mathcal{F}}(n,\tilde{m},k)$. To prove the theorem, it is enough to show that $\tilde{X}$ is almost always at most $cn^{\tilde{q}}$ for some constant c, which is formalized as the following lemma.

Lemma 15: Suppose that $\tilde{m}=\Theta(n^{h+\alpha})$ for $h \ge 1$. Then, there is a constant c such that $\Pr[\tilde{X} > cn^{\tilde{q}}]=o(1)$.

Proof: Let

$$p=\frac{(1+\delta)\tilde{m}}{\binom{n}{k}}$$

and consider the uniform probability model $\mathcal{F}(n,p,k)$. Let U be the number of k-order subsets of $[n]$ chosen from $\mathcal{F}(n,p,k)$ and let X be the number of hyperedge candidates appearing in the process of running HCA on $\mathcal{F}(n,p,k)$. Then, for a constant c

$$\begin{aligned}\Pr[X > cn^{\tilde{q}}]&=\sum_{i=0}^{\binom{n}{k}}\Pr[U=i]\cdot\Pr[X > cn^{\tilde{q}}|U=i]\\ &\ge\sum_{i \ge \tilde{m}}\Pr[U=i]\cdot\Pr[X > cn^{\tilde{q}}|U=i].\end{aligned}$$

Because $\Pr[X > cn^{\tilde{q}}|U=\tilde{m}] \le \Pr[X > cn^{\tilde{q}}|U=i]$ for $i \ge \tilde{m}$ and $\Pr[X > cn^{\tilde{q}}|U=\tilde{m}]=\Pr[\tilde{X} > cn^{\tilde{q}}]$

$$\begin{aligned}\Pr[X > cn^{\tilde{q}}]&\ge\sum_{i \ge \tilde{m}}\Pr[U=i]\cdot\Pr[X > cn^{\tilde{q}}|U=\tilde{m}]\\ &=\Pr[U \ge \tilde{m}]\cdot\Pr[\tilde{X} > cn^{\tilde{q}}]\\ &=(1-\Pr[U < \tilde{m}])\cdot\Pr[\tilde{X} > cn^{\tilde{q}}]\\ &\ge\Pr[\tilde{X} > cn^{\tilde{q}}]-\Pr[U < \tilde{m}].\end{aligned}$$

Hence

$$\Pr[\tilde{X} > cn^{\tilde{q}}] \le \Pr[X > cn^{\tilde{q}}]+\Pr[U < \tilde{m}]. \tag{22}$$

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

From Lemma 14

$$\Pr[X > cn^{\tilde{q}}] = o(1) \tag{23}$$

for some constant c. Because U has the binomial distribution with parameters $\binom{n}{k}$ and

$$p = \frac{(1+\delta)\tilde{m}}{\binom{n}{k}}$$

and $\tilde{m} = \Omega(n)$, by Chernoff bound [33]

$$\Pr[U < \tilde{m}] \le \exp\left(-\frac{\delta^2 \tilde{m}}{2(1+\delta)^2}\right) = o(1). \tag{24}$$

Therefore, from (22)–(24)

$$\Pr[\tilde{X} > cn^{\tilde{q}}] = o(1)$$

for some constant c, which completes the proof. □□

Proof of Theorem 7: Suppose that $h = 0$. Because $\tilde{m} = \Theta(n^{\alpha})$ and k is constant, the number of maximal hyperedges in G_f is $\mathcal{O}(n^{\alpha})$ with probability 1. From Theorem 2, the number of function evaluations by HCA is $\mathcal{O}(n^{\max\{2\alpha,1\}} \log n)$ with probability 1 and the theorem follows.

Now, we assume that $h \ge 1$ in the rest of the proof. Let $\tilde{X}'$ be the number of hyperedge candidates appearing in the process of running HCA on $\tilde{\mathcal{F}}'(n, \tilde{m}, k)$. To prove the theorem, it is enough to show that $\tilde{X}'$ is almost always at most $cn^{\tilde{q}}$ for some constant c. Let $\tilde{X}$ be the number of hyperedge candidates appearing in the process of running HCA on $\tilde{\mathcal{F}}(n, \tilde{m}, k)$ and let $\tilde{U}'$ be the number of distinct k-order subsets of $[n]$ chosen from $\tilde{\mathcal{F}}'(n, \tilde{m}, k)$. Fix a positive constant c. Note that $\Pr[\tilde{X}' > cn^{\tilde{q}} | \tilde{U}' = i] \le \Pr[\tilde{X}' > cn^{\tilde{q}} | \tilde{U}' = \tilde{m}]$ for $i \le \tilde{m}$ and $\Pr[\tilde{X}' > cn^{\tilde{q}} | \tilde{U}' = \tilde{m}] = \Pr[\tilde{X} > cn^{\tilde{q}}]$. Thus

$$\begin{aligned}
\Pr[\tilde{X}' > cn^{\tilde{q}}] &= \sum_{i=1}^{\tilde{m}} \Pr[\tilde{U}' = i] \cdot \Pr[\tilde{X}' > cn^{\tilde{q}} | \tilde{U}' = i] \\
&\le \sum_{i=1}^{\tilde{m}} \Pr[\tilde{U}' = i] \cdot \Pr[\tilde{X}' > cn^{\tilde{q}} | \tilde{U}' = \tilde{m}] \\
&= \sum_{i=1}^{\tilde{m}} \Pr[\tilde{U}' = i] \cdot \Pr[\tilde{X} > cn^{\tilde{q}}] \\
&= \Pr[\tilde{X} > cn^{\tilde{q}}].
\end{aligned}$$

Because $\Pr[\tilde{X} > c'n^{\tilde{q}}] = o(1)$ for some positive constant c' from Lemma 15, we have

$$\Pr[\tilde{X}' > c'n^{\tilde{q}}] = o(1). \quad □$$

References

[1] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: Univ. Michigan Press, 1975.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison Wesley, 1989.

[3] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," *IEEE Trans. Comput.*, vol. 45, no. 7, pp. 841–855, Jul. 1996.

[4] T. N. Bui and B. R. Moon, "GRCA: A hybrid genetic algorithm for circuit ratio-cut partitioning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 3, pp. 193–204, Mar. 1998.

[5] I. Hwang, Y. H. Kim, and B. R. Moon, "Multi-attractor gene reordering for graph bisection," in *Proc. Genetic Evolut. Comput. Conf.*, 2006, pp. 1209–1216.

[6] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Syst.*, vol. 5, no. 3, pp. 493–530, 1989.

[7] D. E. Goldberg, K. Deb, and B. Korb, "Messy genetic algorithms revisited: Studies in mixed size and scale," *Complex Syst.*, vol. 4, pp. 415–444, 1990.

[8] D. E. Goldberg, K. Deb, H. Kargupta, and G. R. Harik, "Rapid accurate optimization of difficult problems using fast messy genetic algorithms," in *Proc. 5th Int. Conf. Genetic Algorithms*, 1993, pp. 56–64.

[9] H. Kargupta, "The gene expression messy genetic algorithm," in *Proc. Int. Conf. Evolut. Comput.*, 1996, pp. 814–819.

[10] G. R. Harik and D. E. Goldberg, "Learning linkage," in *Foundations of Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 1997, vol. 4, pp. 247–262.

[11] R. P. Salustowicz and J. Schmidhuber, "Probabilistic incremental program evolution: Stochastic search through program space," in *Machine Learning: ECML-97*. New York: Springer-Verlag, 1997, vol. 1224, pp. 213–220.

[12] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," in *Proc. IEEE Conf. Evolut. Comput.*, 1998, pp. 523–528.

[13] J. S. de Bonet, C. L. Isbell, Jr., and P. Viola, "MIMIC: Finding optima by estimating probability densities," in *Advances in Neural Information Processing Systems*. Cambridge, MA: The MIT Press, 1997, vol. 9, pp. 424–430.

[14] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing—Engineering Design and Manufacturing*. New York: Springer-Verlag, 1999, pp. 521–535.

[15] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Proc. Genetic Evolut. Comput. Conf.*, 1999, pp. 525–532.

[16] M. Pelikan and D. E. Goldberg, "Hierarchical problem solving by the Bayesian optimization algorithm," in *Proc. Genetic Evolut. Comput. Conf.*, 2000, pp. 267–274.

[17] H. Mühlenbein and T. Mahnig, "FDA—A scalable evolutionary algorithm for the optimization of additively decomposed functions," *Evolut. Comput.*, vol. 7, no. 1, pp. 45–68, 1999.

[18] M. J. Streeter, "Upper bounds on the time and space complexity of optimizing additively separable functions," in *Proc. Genetic Evolut. Comput. Conf.*, 2004, pp. 186–197.

[19] A. H. Wright and S. Pulavarty, "On the convergence of an estimation of distribution algorithm based on linkage discovery and factorization," in *Proc. Genetic Evolut. Comput. Conf.*, 2005, pp. 695–702.

[20] S. Rana, R. B. Heckendorn, and D. Whitley, "A tractable walsh analysis of SAT and its implications for genetic algorithms," in *Proc. 15th Nat. Conf. Artif. Intell.*, 1998, pp. 392–397.

[21] A. Braunstein, M. Mézard, M. Weigt, and R. Zecchina, "Constraint satisfaction by survey propagation," *CoRR*, vol. cond-mat/0212451, 2002 [Online]. Available: http://arxiv.org/abs/cond-mat/0212451

[22] S. Droste, T. Jansen, and I. Wegener, "Upper and lower bounds for randomized search heuristics in black-box optimization," *Theory Comput. Syst.*, vol. 39, no. 4, pp. 525–544, 2006.

[23] R. B. Heckendorn and A. H. Wright, "Efficient linkage discovery by limited probing," in *Proc. Genetic Evolut. Comput. Conf.*, 2003, pp. 1003–1014.

[24] R. B. Heckendorn and A. H. Wright, "Efficient linkage discovery by limited probing," *Evolut. Comput.*, vol. 12, no. 4, pp. 517–545, 2004.

[25] H. Kargupta and B. Park, "Gene expression and fast construction of distributed evolutionary representation," *Evolut. Comput.*, vol. 9, no. 1, pp. 1–32, 2001.

[26] A. C. Yao, "Probabilistic computations: Toward a unified measure of complexity," in *Proc. 18th Annu. IEEE Symp. Found. Comput. Sci.*, 1977, pp. 222–227.

[27] J. B. Kruskal, , R. Bellman, Ed., "The number of simplices in a complex," in *Mathematical Optimization Techniques*. Berkeley, CA: Univ. California Press, 1963, pp. 251–278.

[28] G. O. H. Katona, "A theorem of finite sets," in *Theory of Graphs*, P. Erdős and G. O. H. Katona, Eds. New York: Academic, 1968, pp. 187–207.

[29] M. Pelikan and D. E. Goldberg, "Hierarchical BOA solves Ising spin glasses and MAXSAT," in *Proc. Genetic Evolut. Comput. Conf.*, 2003, pp. 1271–1282.

[30] S. H. Kim, Y. H. Kim, and B. R. Moon, "A hybrid genetic algorithm for the MAX CUT problem," in *Proc. Genetic Evolut. Comput. Conf.*, 2001, pp. 416–423.

[31] A. H. Wright, R. K. Thompson, and J. Zhang, "The computational complexity of NK fitness functions," *IEEE Trans. Evolut. Comput.*, vol. 4, no. 4, pp. 373–379, Nov. 2000.

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.

[32] Y. Gao and J. Culberson, "Space complexity of estimation of distribution algorithms," *Evolut. Comput.*, vol. 13, no. 1, pp. 125–143, 2005.

[33] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[34] M. Munetomo and D. E. Goldberg, "Identifying linkage groups by nonlinearity/non-monotonicity detection," in *Proc. Genetic Evolut. Comput. Conf.*, 1999, pp. 433–440.

[35] J. L. Walsh, "A closed set of orthogonal functions," *Amer. J. Math.*, vol. 55, pp. 5–24, 1923.

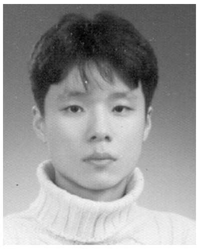
[36] R. B. Heckendorn and D. Whitley, "Predicting epistasis directly from mathematical models," *Evolut. Comput.*, vol. 7, no. 1, pp. 69–101, 1999.

[37] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd ed. Reading, MA: Addison Wesley, 1994.

[38] L. Lovász, *Combinatorial Problems and Exercises*, 2nd ed. Amsterdam, The Netherlands: North-Holland, 1993.

[39] S. S. Choi, Y. K. Kwon, and B. R. Moon, "Properties of symmetric fitness functions," *IEEE Trans. Evolut. Comput.*, vol. 11, no. 6, pp. 743–757, Dec. 2007.

[40] D. J. Coffin and C. D. Clack, "gLINC: Identifying composability using group perturbation," in *Proc. Genetic Evolut. Comput. Conf.*, 2006, pp. 1133–1140.



Sung-Soon Choi received the B.S. degree in computer science and the Ph.D. degree in computer science and engineering from Seoul National University (SNU), Seoul, Korea, in 2000 and 2006, respectively.

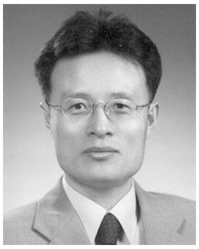
From 2006 to 2007, he was a Postdoctoral Fellow in the School of Computer Science and Engineering, SNU, and also a Research Staff Member in the Institute of Computer Technology, SNU. Since July 2007, he has been a Research Professor at Random Graph Research Center, Yonsei University, Seoul. His research interests include algorithm design/analysis, combinatorial optimization, evolutionary algorithms, probabilistic methods, random structures/algorithms, nonlinear dynamical systems, and artificial intelligence.

Dr. Choi served as a committee member at the Genetic and Evolutionary Computation Conference (GECCO 2007–2008).

Kyomin Jung received the B.S. degree in mathematics from Seoul National University, Seoul, Korea, in 2003. He is currently working towards the Ph.D. degree in mathematics at Massachusetts Institute of Technology, Cambridge, MA.

He has been a member of Laboratory for Information and Decision Systems (LIDS) since September 2004. In winter 2003–2004, he was a Research Visitor at Microsoft Research, Theory Group, Redmond, WA. In summer 2006, he was a Research Intern at Bell Labs, Department of Mathematics of Networks and Systems, Murray Hill, NJ. In summer 2007, he was a Research Intern at IBM T. J. Watson Research Center, Mathematical Sciences Department, Yorktown, NY. In summer 2008, he was a Research Intern at Microsoft Research, Machine Learning and Perception Group, Cambridge, U.K. His main research interest is in design of efficient algorithms for computationally hard problems arising in statistical inference and machine learning.

Mr. Jung won a gold medal in the International Mathematical Olympiad (IMO) in 1995.



Byung-Ro Moon (M'01) received the B.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1985, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology, Seoul, Korea, in 1987, and the Ph.D. degree in computer science from the Pennsylvania State University, University Park, in 1994.

From 1987 to 1991, he was an Associate Research Engineer with the Central Research Laboratory, LG Electronic Company, Ltd., Seoul, Korea. From November 1994 through 1995, he was a Postdoctoral Scholar with the VLSI CAD Lab, University of California, Los Angeles. From 1996 to 1997, he was a Principal Research Staff Member with the DT Research Center, LG Semicon Ltd., Seoul, Korea. Since September 1997, he has been a Professor with the School of Computer Science and Engineering, Seoul National University. He leads the Optimization Laboratory and is also the CEO of Optus, Inc., a company specialized in optimization. He has developed various optimization solutions for industry. His major interest is the theory and application of optimization methodologies (evolutionary computation, algorithm design/analysis, optimization modeling, etc.) The applications of optimization include combinatorics, financial optimization, personalized marketing, e-commerce, search, text mining, graph partitioning, scheduling, and system identification.

Dr. Moon was the recipient of a Korean Government Scholarship during the 1991–1994 academic years. He is a member of the Association for Computing Machinery (ACM). He served as the Publication Chair of the IEEE Congress on Evolutionary Computation (CEC 2000), a committee member of GP 1997–1998, the International Symposium on Intelligent Automation and Control (ISIAC 1998), and the Genetic and Evolutionary Computation Conference (GECCO 1999–2006).

Authorized licensed use limited to: MIT Libraries. Downloaded on November 16, 2009 at 15:07 from IEEE Xplore. Restrictions apply.