This document contains full proofs of the lemmas and theorems whose proofs were omitted from the GECCO 2004 paper:

Upper Bounds on the Time and Space Complexity of Optimizing Additively Separable Functions

Matthew J. Streeter

Computer Science Department and
Center for the Neural Basis of Cognition
Carnegie Mellon University
Pittsburgh, PA 15213
matts@cs.cmu.edu

Lemma 2. If $\Delta f_i(s) \neq \Delta f_i(s')$, FINDLINKEDPOSITION($f$, s, s', i) performs no more than $4(\lceil \lg(2\ell) \rceil - 1)$ function evaluations.

Proof: The lemma follows from fact (shown below) that binary search on a set of δ positions requires at most $\lceil \lg(2\delta) \rceil$ iterations. As illustrated in the proof of lemma 1, FINDLINKEDPOSITION performs a binary search on the set of positions that differ between s and s', and the number of positions that differ can be at most ℓ, so the function can perform at most $\lceil \lg(2\ell) \rceil$ iterations before returning. FINDLINKEDPOSITION performs 4 function evaluations each iteration except the last one, where it performs no function evaluations, so the total number of function evaluations will be $4(\lceil \lg(2\ell) \rceil - 1)$ in the worst case (when $\delta = \ell$ and when all recursions are in the direction of the strings with $\lceil \delta/2 \rceil$ rather than $\lfloor \delta/2 \rfloor$ differences).

To see that the binary search requires at most $\lceil \lg(2\delta) \rceil$ iterations, let I_δ denote the number of iterations required in the worst case (when all recursions are in the direction of the strings with $\lceil \delta/2 \rceil$ differences) as a function of δ, and note that I_δ must satisfy the recurrence relation:

$$I_1 = 1$$
$$I_\delta = 1 + I_{\lceil \delta/2 \rceil}$$

The solution to this recurrence relation is $I_\delta = \lceil \lg(2\delta) \rceil$. To verify this, note that if δ is even, $\lceil \delta/2 \rceil = \delta/2$ and we have:

$$I_\delta = \lceil \lg(2\delta) \rceil = \lceil \lg(\delta) + 1 \rceil = 1 + \lceil \lg(\delta) \rceil = 1 + I_{\lceil \delta/2 \rceil}$$

As long as δ is not a power of 2, $\lceil \lg(\delta) \rceil = \lceil \lg(\delta+1) \rceil$. So in particular if δ is odd, we have:

$$I_\delta = 1 + \lceil \lg(\delta) \rceil = 1 + \lceil \lg(\delta+1) \rceil = 1 + I_{(\delta+1)/2} = 1 + I_{\lceil \delta/2 \rceil}\ .$$ ■

Lemma 5. For any t, FINDLINKAGEGROUPS(f, t) returns a partition Γ such that for any i and j, if $j \in \Gamma[i]$ then $\mathcal{G}(i, j)$.

Proof: By induction on the sequence of updates to Γ. Immediately after Γ is initialized at line 1, $\Gamma[i] = \{i\}$, so because $\mathcal{G}(i, i)$ is true by definition the lemma is trivially satisfied. The code for updating Γ consists of removing $\Gamma[i]$ and $\Gamma[j]$ from Γ and replacing them with $\Gamma[i] \cup \Gamma[j]$. Note that by lemma 3, Γ is only updated in this way when $\mathcal{L}(i, j)$ is true. Letting Γ' denote the updated Γ, we have $\Gamma'[i] = \Gamma'[j] = \Gamma[i] \cup \Gamma[j]$ and $\Gamma'[h] = \Gamma[h]$ for $h \notin \Gamma'[i]$. Thus it suffices to show that for any $i' \in \Gamma'[i]$, $\mathcal{G}(i, i')$ is true. If $i' \in \Gamma'[i]$ then either $i' \in \Gamma[i]$ or $i' \in \Gamma[j]$. If $i' \in \Gamma[i]$, then $\mathcal{G}(i, i')$ by the inductive hypothesis. If $i' \in \Gamma[j]$, then $\mathcal{G}(i', j)$ by the inductive hypothesis, and (noting that $\mathcal{L}(i, j)$ implies $\mathcal{G}(i, j)$) it follows from the transitivity of $\mathcal{G}$ that $\mathcal{G}(i', j)$. ■

Theorem 1. Let f be an order-k additively separable function with linkage group partition Γ_f. The probability that FINDLINKAGEGROUPS(f, t) returns Γ_f is at least $(1-(1-2^{-k})^t)^\ell$. To find the Γ_f with probability p we must invoke FINDLINKAGEGROUPS(f, $\ln(1-p^{1/\ell})/\ln(1-2^{-k})$), which will require $O(2^k \ell \ln(\ell))$ evaluations of f.

Proof: Parts (*i*) and (*ii*) of lemma 7 together with lemma 6 imply that the probability that FINDLINKAGEGROUPS(f, t) returns Γ_f is at least $(1-(1-2^{-k})^t)^\ell$. The remainder of the theorem follows from the proof of lemma 6. ■

Lemma 8. Let γ be a linkage group of f, and let s_1 and s_2 be two strings such that for all $i \in \gamma$, $s_1.i = s_2.i$. Then $\Delta f_i(s_1) = \Delta f_i(s_2)$.

Proof: By induction on the hamming distance between s_1 and s_2.

Case δ=0. s_1=s_2, so $\Delta f_i(s_1) = \Delta f_i(s_2)$.

Case δ>0. Let j be a position such that $s_1.j \neq s_2.j$, and let $s_1' = s_1[j \rightarrow s_2.j]$. Note that the hamming distance between s_1 and s_1' is 1, while that between s_1' and s_2 is δ-1. By assumption $s_1.i = s_2.i$ for all $i \in \gamma$, so it must be that $j \notin \gamma$ which means $\mathcal{L}(i, j)$ is false. Thus by definition of $\mathcal{L}$, $\Delta f_i(s_1) = \Delta f_i(s_1')$. But by the induction hypothesis, $\Delta f_i(s_1') = \Delta f_i(s_2)$, so $\Delta f_i(s_1) = \Delta f_i(s_2)$. ■

Theorem 2. Let $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_m\}$ be the linkage group partition for f, and let $\gamma_i = \{\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,k(i)}\}$, where $k(i) = |\gamma_i|$. Let S_0 be a string consisting entirely of zeroes. Then for all s, $f(s) = f_\Sigma(s)$, where:

$$f_\Sigma(s) \equiv \sum_{i=1}^{m} f_i(s.\alpha_{i,1}, s.\alpha_{i,2}, ..., s.\alpha_{i,k(i)}) \text{ , and}$$

$$f_i(\beta_1, \beta_2, ..., \beta_{k(i)}) \equiv f(S_0[\alpha_{i,1} \rightarrow \beta_1][\alpha_{i,2} \rightarrow \beta_2]...[\alpha_{i,k(i)} \rightarrow \beta_{k(i)}]) + f(S_0)(1/m\text{-}1).$$

Proof: By induction on the hamming distance between s and S_0.

Case δ = 0. By inspection,

$$f_i(S_0) = f(S_0) + f(S_0)(1/m\text{-}1) = f(S_0)/m$$
$$f_\Sigma(S_0) = m(f_1(S_0)) = m(f(S_0)/m) = f(S_0)$$

If δ=0 then s=S_0, so $f_\Sigma(s) = f(s)$ as desired.

Case δ > 0. Let α be a position such that $s.\alpha \neq S_0.\alpha$, and let $S' = s[\alpha \rightarrow S_0.\alpha]$. The hamming distance between s and S' is 1, while the hamming distance between S' and S_0 is δ-1. Thus $f(S') = f_\Sigma(S')$ by the induction hypothesis. By definition, exactly one of the m subfunctions on the right side of equation 2 depends on position α. Let this subfunction be i. Observe that $f_j(s) = f_j(S')$ for $j \neq i$, and that:

$f_i(s) - f_i(S')$

$$= f(S_0[\alpha_{i,1}\rightarrow s.\alpha_{i,1}][\alpha_{i,2}\rightarrow s.\alpha_{i,2}]...[\alpha_{i,k(i)}\rightarrow s.\alpha_{k(i)}]) + f(S_0)(1/m\text{-}1)$$
$$- f(S_0[\alpha_{i,1}\rightarrow S'.\alpha_{i,1}][\alpha_{i,2}\rightarrow S'.\alpha_{i,2}]...[\alpha_{i,k(i)}\rightarrow S'.\alpha_{k(i)}]) + f(S_0)(1/m\text{-}1)$$
$$= f(S_0[\alpha_{i,1}\rightarrow s.\alpha_{i,1}][\alpha_{i,2}\rightarrow s.\alpha_{i,2}]...[\alpha_{i,k(i)}\rightarrow s.\alpha_{k(i)}][\alpha\rightarrow s.\alpha])$$
$$- f(S_0[\alpha_{i,1}\rightarrow s.\alpha_{i,1}][\alpha_{i,2}\rightarrow s.\alpha_{i,2}]...[\alpha_{i,k(i)}\rightarrow s.\alpha_{k(i)}][\alpha\rightarrow S_0.\alpha])$$
$$= \Delta f_\alpha(S_0[\alpha_{i,1}\rightarrow s.\alpha_{i,1}][\alpha_{i,2}\rightarrow s.\alpha_{i,2}]...[\alpha_{i,k(i)}\rightarrow s.\alpha_{k(i)}][\alpha\rightarrow S_0.\alpha])$$
$$= \Delta f_\alpha(S_0[\alpha_{i,1}\rightarrow S'.\alpha_{i,1}][\alpha_{i,2}\rightarrow S'.\alpha_{i,2}]...[\alpha_{i,k(i)}\rightarrow S'.\alpha_{k(i)}])$$
$$= \Delta f_\alpha(S')$$

Where in the last step we have used lemma 8. Thus:

$$f_\Sigma(s) - f_\Sigma(S') = f_i(s) - f_i(S') = \Delta f_\alpha(S') = f(s) - f(S') = f(s) - f_\Sigma(S').$$

So $f_\Sigma(s) = f(s)$, as desired. ■

Corollary 1. A string s is globally optimal w.r.t. a function f iff. s is optimal w.r.t. each linkage group of f.

Proof: Suppose s is optimal w.r.t. each linkage group γ of f. By theorem 2, $f(s)$ can be expressed as the sum of subfunctions f_i for $1 \leq i \leq |\Gamma_f|$, where Γ_f is the linkage group partition of f. By definition 6, s maximizes each of these subfunctions, so it must also maximize f. Thus s is globally optimal w.r.t. f if it is optimal w.r.t. each of f's linkage groups.

For the converse, suppose s is globally optimal w.r.t. f. If s were not globally optimum w.r.t. some linkage group γ_i, then we could alter s at the positions in γ_i to obtain an s' that achieves a higher value of f_i, where f_i is the subfunction that covers the positions in γ_i. But by lemma 8, s' will achieve the same value as s on every subfunction other than f_i, so $f(s')$ will exceed $f(s)$, in contradiction to the assumption that s is globally optimal. ■

Corollary 2. If γ is linkage group for f, then OPTIMIZEWRTGROUP(f, s, γ) returns a string s' that is optimal w.r.t. γ.

Proof: Let $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_m\}$ be the linkage group partition of f, and let $\gamma = \gamma_i$. By theorem 2, $f(s)$ can be expressed as the sum of a function $f_i(s.\alpha_{i,1}, s.\alpha_{i,2}, ..., s.\alpha_{i,k(i)})$ and m-1 other functions that do not depend on the values of the positions in γ_i. Thus an assignment of values to positions $\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,k(i)}$ that maximizes f also maximizes f_i. The loop beginning on line 2 of OPTIMIZEWRTLINKAGEGROUP evaluates all $2^{k(i)}$ assignments of $\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,k(i)}$ to find one that maximizes f and returns a string s' with that assignment. Therefore s' also maximizes f_i. ■