A RECURSIVE METHOD FOR SOLVING HAPLOTYPE FREQUENCIES IN MULTIPLE LOCI LINKAGE ANALYSIS

MICHAEL K. NG *

Department of Mathematics, Hong Kong Baptist University
Kowloon Tong, Hong Kong, China
E-mail: mng@math.hkbu.edu.hk

ERIC S. FUNG WAI-KI CHING † YIU-FAI LEE

Department of Mathematics, The University of Hong Kong
Pokfulam Road, Hong Kong, China
E-mails: ericfung@hkusua.hku,hk, wkc@maths.hku.hk, ahyiu@graduate.hku.hk

Multiple loci analysis has become popular with the advanced development in biological experiments. A lot of studies have been focused on the biological and the statistical properties of such multiple loci analysis. In this paper, we study one of the important computational problems: solving the probabilities of haplotype classes from a large linear system $Ax = b$ derived from the recombination events in multiple loci analysis. Since the size of the recombination matrix A increases exponentially with respect to the number of loci, fast solvers are required to deal with a large number of loci in the analysis. By exploiting the nice structure of the matrix A, we develop an efficient recursive algorithm for solving such structured linear systems. In particular, the complexity of the proposed algorithm is of $O(m \log m)$ operations and the memory requirement is of $O(m)$ locations where m is the size of the matrix A. Numerical examples are given to demonstrate the effectiveness of our efficient solver.

1. Introduction

Linkage analysis is an important tool for the mapping of genetic loci. With the availability of numerous DNA markers throughout the human genome, linkage analysis has demonstrated its usefulness in mapping the mutations responsible for hundreds of Mendelian diseases (Kruglyak and Lander, 1995). The genotype of an individual at loci X and Y is formed by the haplotypes of two gametes $X_f Y_f$ inherited from the father, and $X_m Y_m$ inherited from the mother. The haplotype of a gamete produced by the individual consists of a mixture of paternal and maternal alleles. A gamete contains two alleles from the same parental gamete (non-recombinant), i.e., $X_f Y_f$ or $X_m Y_m$, or one allele from each parental gamete (recombinant), i.e., $X_f Y_m$ or $X_m Y_f$. The recombination fraction between the two loci is defined as the probability that a gamete is recombinant.

*Work partially supported by RGC Grant Nos. 7130/02P, 7046/03P, 7035/04P, 7035/05P, and FRG04-05/II-51
†Work partially supported by RGC Grant No. HKU 7126/02P

When the number of loci is large, a haplotype almost certainly constitutes a new combination of alleles, different from the parental and the maternal haplotypes (Sham, 1998). If n loci are involved, there are $(n-1)$ intervals between adjacent loci, each of which can either have an even or an odd number of crossovers. This produces 2^{n-1} classes of gametic haplotypes, and therefore $(2^{n-1}-1)$ independent gametic frequencies (the 2^{n-1} classes of gametic frequencies must sum up to one). The frequency of a gametic haplotype is equal to the joint probability of the co-occurrence of a set of recombination events. Liberman and Karlin (1984) applied the concept of recombination values to establish the relationship between recombination fractions and haplotype frequencies for $n > 3$. The recombination value of a set of intervals (not necessary contiguous), is the probability of an odd number of crossovers occurring in the intervals. Since each of the $(n-1)$ intervals can be included or excluded in a set of intervals, there are $(2^{n-1}-1)$ sets of intervals and hence $(2^{n-1}-1)$ recombination values. There is a relationship between these $(2^{n-1}-1)$ recombination values and the $(2^{n-1}-1)$ haplotype frequencies as specified by a linear system

$$\Theta = \Gamma A_n$$

where A_n is the m-by-m matrix with $m = 2^{n-1}$ being equal to the number of haplotype classes, and Θ and Γ are m-vectors containing the recombination values and haplotype frequencies respectively, see Section 2 for details about the derivation of $\Theta = \Gamma A_n$.

When the number n of loci increases, the size of A_n increases exponentially and therefore the cost of solving $\Theta = \Gamma A_n$ is very expensive. Here we will first establish the structure of A_n and a recursive formula relating A_{n+1} and A_n. We then present a recursive solver based on the recursiev formula to solve $\Theta = \Gamma A_n$ efficiently.

The rest of this paper is organized as follows. In Section 2, we give some background and basic properties on the matrix A_n. In Section 3, we show that A_n is nonsingular and give the explicit form for its inverse. According to the explicit form of A_n^{-1}, we obtain the haplotype frequencies efficiently by using a recursive scheme. We also give a cost analysis for the proposed algorithm. Numerical examples are given to illustrate the effectiveness of the proposed method. Finally, concluding remarks are given in Section 4.

2. The Recombination Matrix A_n

In this section, we give some background of the recombination matrix A_n. In the multilocus situation ($n \geq 3$), we denote a haplotype of n loci by a $(n-1)$ string of 0s and 1s with respect the ith digit representing the recombination status of the $(i+1)$th allele with respect to the first allele. This string of $(n-1)$ digits specifies the recombination status between all $n(n-1)/2$ pairs of loci. Here pairs of loci with different digits are recombinants while the others are non-recombinants. Such strings refer to different rows of the matrix A_n. To apply the concept of recombination values of a set of non-contiguous intervals, we let the inclusion or the exclusion of the intervals be denoted by a vector of 0s and 1s, where 0 represents exclusion and 1 represents inclusion. Such intervals refer to different columns of the matrix A_n. For each haplotype class and each set of intervals, we set the entry of A to 1 exactly when there is an odd number of crossovers for the intervals in the set.

For examples, in the case of four loci, W, X, Y and Z, there are eight possible haplotype classes, 000, 001, 010, 011, 100, 101, 110 and 111. Each represents a unique combination of recombination status between the six possible pairs of loci (WX, WY, WZ, XY, XZ and YZ). There are seven possible sets of intervals (001, 010, 011, 100, 101, 110, 111), excluding the set with no intervals. In this case, the relationship between the haplotype classes and the recombination values can be described as follows:

Haplotype classes	Interval sets						
WXYZ	001	010	011	100	101	110	111
000	0	0	0	0	0	0	0
001	1	0	1	0	1	0	1
010	1	1	0	0	1	1	0
011	0	1	1	0	0	1	1
100	0	1	1	1	1	0	0
101	1	1	0	1	0	0	1
110	1	0	1	1	0	1	0
111	0	0	0	1	1	1	1

- The gamete of the haplotype class “001” is the recombinant with respect to the loci W and Z, and is the non-recombinant with respect to the loci W, X and Y. Correspondingly, the crossover only occurs in the interval YZ, and therefore, we assign one in the sets of intervals (001, 011, 101 and 111) as these intervals including YZ contribute the frequencies to the haplotype class “001”. By using the same arguments, the haplotype class “100” can be considered similarly.
- For the haplotype class “011”, the gamete is the recombinant with respect to the loci W and Y and the loci W and Z, and is the non-recombinant with respect to the loci W and X. In this case, the crossover only occurs in the interval XY. The sets of intervals including XY contributing to the frequencies of the haplotype class “011” are 010, 011, 110 and 111. The haplotype class “110” can be considered similarly.
- The gamete of the haplotype class “010” is the recombinant with respect to the loci W and Y, and is the non-recombinant with respect to the loci W, X and Z. It also implies that such haplotype is also the the recombinant with respect to the loci X and Y, and also the loci Y and Z. Correspondingly, the crossover only occurs in the interval XY or YZ, and therefore, we assign one in the sets of intervals (001, 010, 101 and 111) as these intervals including XY or YZ contribute the frequencies to the haplotype class “010”. We note that the sets of intervals (011 and 111) include both XY and YZ and therefore the value 0 is assigned to them since an odd number of crossovers occurring in the intervals is counted. By using the same arguments, the haplotype class “101” can be considered similarly.
- For the haplotype class ”111”, the gamete is the recombinant with respect to the loci W and X, the loci W and Y, and the loci W and Z. In this case, the crossover only occurs in the interval WX. The sets of intervals contributing to the frequen-

cies of the haplotype class "111" are 100, 101, 110 and 111.

Finally, we note that the sum of all haplotype frequencies should be equal to one. With the above table and the additional constraint, the matrix A_4 is given as follows:

$$\begin{array}{c} \\ \text{Haplotype classes} \end{array} \begin{array}{c} \begin{array}{cccccccc} & \multicolumn{7}{c}{\text{Interval sets}} \\ & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \begin{array}{l} 000 \rightarrow \\ 001 \rightarrow \\ 010 \rightarrow \\ 011 \rightarrow \\ 100 \rightarrow \\ 101 \rightarrow \\ 110 \rightarrow \\ 111 \rightarrow \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}.$$

In the following discussion, the binary strings of haplotype classes and interval sets are represented in ascending order, and the properties of the recombination matrix A_n can be summarized as follows:

(1) All the entries in the first column of A_n are equal to 1.
(2) The first row of A_n is a unit row vector with the first entry being equal to 1.
(3) For the (i,j)th entry of A_n, we express the integers i and j in a binary system:

$$i = 1 + \sum_{k=0}^{n-2} a_k^{(i)} 2^k \quad \text{and} \quad j = 1 + \sum_{k=0}^{n-2} b_k^{(j)} 2^k.$$

The haplotype class is represented by $a_0^{(i)} a_1^{(i)} \cdots a_{n-2}^{(i)}$ and the set of intervals is represented by $b_1^{(i)} b_2^{(i)} \cdots b_{n-2}^{(i)}$. The value of the (i,j)th entry of A_n is determined by the following formula:

$$[A_n]_{i,j} = \left(a_0^{(i)} b_0^{(i)} + \sum_{k=1}^{n-2} (a_k^{(i)} - a_{k-1}^{(i)}) b_k^{(j)} \right) \pmod 2.$$

We note that when $a_k^{(i)}$ and $a_{k-1}^{(i)}$ are different, it refers to the case that the gamete is recombinant with respect to themselves, and hence such interval should be included in the interval set if $b_k^{(j)}$ is equal to 1. The $a_0^{(i)}$ already indicates whether the gamete is recombinant with respect to the first two loci. Finally, the value one is assigned to $[A_n]_{i,j}$ under the modulo arithmetic if the number of intervals included is an odd number.

According to the above properties of A_n, we can construct the recombination matrix and then solve the linear system $\Theta = \Gamma A_n$ to obtain the haplotype frequencies. Since the size of A_n increases exponentially with respect to the number of loci n, fast solvers are required in order to compute haplotype frequencies efficiently in linkage analysis of

multiple loci. Next we present a recursive formula for A_{n+1} and A_n based on the nice structure of the matrix A_{n+1}.

Theorem 2.1. *For $n \geq 1$, the recombination matrix A_{n+1} is given recursively as follows:*

$$A_{n+1} = \left(\begin{array}{c|c} A_n & A_n - R \\ \hline PA_n & N - PA_n + R \end{array}\right)$$

where

$$P = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}, \qquad R = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix} \quad \text{and} \quad N = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & \dots & 1 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix}.$$

Proof. First of all, we partition A_{n+1} into four blocks, i.e.,

$$A_{n+1} = \left(\begin{array}{c|c} M_1 & M_2 \\ \hline M_3 & M_4 \end{array}\right)$$

where M_i are 2^{n-1}-by-2^{n-1} matrices. We note that the binary strings of haplotype classes and interval sets are represented in ascending order. Therefore, for the matrix M_1 corresponding to the first 2^{n-1} rows and the first 2^{n-1} columns, the first digit of their corresponding haplotype classes and interval sets is equal to 0. It implies that M_1 is just the recombination matrix A_n for the n loci problem.

For the submatrix M_2, we note that the first digit of the interval sets corresponding the columns of A_{n+1} between $(2^{n-1}+1)$th to 2^nth, is equal to 1. Since the first digit of the corresponding haplotype classes is equal to 0, there is no contribution of such haplotype classes to the interval set "$100\cdots000$". We assign the zero entries for the first column of M_2, and the other entries are the same as the matrix M_1. Therefore the resulting matrix M_2 is equal to $(A_n - R)$.

For the submatrix M_3, the corresponding haplotype class "$i_1 i_2 \cdots i_n$" can be viewed as the same as the haplotype class "$(1-i_1)(1-i_2)\cdots(1-i_n)$". The contributions of the haplotype class "$i_1 i_2 \cdots i_n$" and the haplotype class "$(1-i_1)(1-i_2)\cdots(1-i_n)$" to the interval sets are the same. It means that the kth row of the matrix M_3 is equal to the $(2^{n-1}-k+1)$th row of the matrix M_1. Such permutation can be implemented as $M_3 = PM_1$.

For the submatrix M_4, by using the similar argument for the submatrix M_3, the kth row of the matrix M_4 is equal to the $(2^{n-1}-k+1)$th row of the matrix M_2. Since the first digit of all the haplotype classes and all the interval sets corresponding to the matrix M_4 is equal 1, all the entries of M_4 should increase by 1. We note that an odd number of crossovers occurring in the set intervals is counted in the recombination matrix. Therefore, the matrix M_4 is given by $N - P(A_n - R)$. Hence the result follows. □

In the next section, we demonstrate that an efficient solver based on the recursive formula for A_{n+1} can be developed to solve the linear system $\Theta = \Gamma A_n$.

3. Recursive Solvers

Since the number n of loci increases, the size of A_n increases exponentially. Fast solvers are required in order to compute haplotype frequencies efficiently in linkage analysis of multiple loci. In this section, we show that A_n is nonsingular for $n \geq 2$, and study the structure of A_n^{-1}. We then present our recursive solvers.

Theorem 3.1. *For $n \geq 1$, A_{n+1} is nonsingular, and we have the following properties of A_{n+1}^{-1}:*

(a) The matrix A_{n+1}^{-1} is given by $\frac{1}{2}\begin{pmatrix} A_n^{-1}+G & (A_n^{-1}-G)P \\ A_n^{-1}-G-H & (H+G-A_n^{-1})P \end{pmatrix}$ where

$$G = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \quad \text{and} \quad H = \frac{1}{2^{n-2}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

[Here we assume that $A_1 = 1$.]

(b) The first row of A_{n+1}^{-1} is a unit row vector with the first entry being equal to 1.

(c) The row sum of A_{n+1}^{-1} is equal to zero except for the first row of A_{n+1}^{-1}.

Proof. Here we use mathematical induction. Let $S(k)$ be a statement that A_k is invertible and A_k^{-1} satisfies above properties. To begin with, we notice that A_2 and its inverse are:

$$A_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad A_2^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}.$$

For $k = 3$, A_3 and its inverse are given by

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad A_3^{-1} = \frac{1}{2}\begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{pmatrix}.$$

It is clear that the last two properties are satisfied. We note that

$$\begin{aligned} &\frac{1}{2}\begin{pmatrix} A_2^{-1}+G & (A_2^{-1}-G)P \\ A_2^{-1}-G-H & (H+G-A_2^{-1})P \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \left[\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right]\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \frac{1}{2^0}\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} & \left[\frac{1}{2^0}\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}\right]\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{pmatrix} = A_3^{-1}. \end{aligned}$$

The statement is true for $k = 2$ and $k = 3$.

Now we assume $S(k)$ is true. We are going to prove that $S(k+1)$ is true. By using Theorem 2.1, we have

$$A_{k+1} = \begin{pmatrix} A_k & A_k - R \\ PA_k & N - PA_k + R \end{pmatrix}$$

Let us consider the following matrix-matrix multiplication:

$$\frac{1}{2}\begin{pmatrix} A_k^{-1} + G & (A_k^{-1} - G)P \\ A_k^{-1} - G - H & (H + G - A_k^{-1})P \end{pmatrix}\begin{pmatrix} A_k & A_k - R \\ PA_k & N - PA_k + R \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

where $P^2 = I$. Our task here is to show that the above right-hand-side matrix is the identity matrix. We expand the product of the two matrices and we have

$$\begin{aligned} C_{11} &= \frac{1}{2}[(A_k^{-1} + G)A_k + (A_k^{-1} - G)PPA_k] = \frac{1}{2}[(A_k^{-1} + G)A_k + (A_k^{-1} - G)A_k] \\ &= \frac{1}{2}[I + GA_k + I - GA_k] = I \end{aligned}$$

and

$$\begin{aligned} C_{12} &= \frac{1}{2}[(A_k^{-1} + G)(A_k - R) + (A_k^{-1} - G)P(N - PA_k + R)] \\ &= \frac{1}{2}[(A_k^{-1} + G)(A_k - R) + (A_k^{-1} - G)PP(N - A_k + R)] \\ &= \frac{1}{2}[(A_k^{-1} + G)(A_k - R) + (A_k^{-1} - G)(N - A_k + R)] \\ &= \frac{1}{2}[I - A_k^{-1}R + GA_k - GR + A_k^{-1}N - I + A_k^{-1}R - GN + GA_k - GR] \\ &= \frac{1}{2}[A_k^{-1}N - GN] = 0. \end{aligned}$$

According to Theorem 2.1, the first row of A_k is a unit row vector with first entry being equal to 1, we obtain $GA_k = GR = (1, 0, \ldots, 0)$. By proposition 3.1, we obtain $A_k^{-1}N = GN = 2^{n-2}H$. Thus we have

$$\begin{aligned} C_{21} &= \frac{1}{2}[(A_k^{-1} - G - H)A_k + (H + G - A_k^{-1})PPA_k] \\ &= \frac{1}{2}[(A_k^{-1} - G - H)A_k + (H + G - A_k^{-1})A_k] \\ &= \frac{1}{2}[I - GA_k - HA_k + HA_k + GA_k - I] = 0 \end{aligned}$$

and

$$\begin{aligned}
C_{22} &= \frac{1}{2}[(A_k^{-1} - G - H)(A_k - R) + (H + G - A_k^{-1})P(N - PA_k + R)] \\
&= \frac{1}{2}[(A_k^{-1} - G - H)(A_k - R) + (H + G - A_k^{-1})PP(N - A_k + R)] \\
&= \frac{1}{2}[(A_k^{-1} - G - H)(A_k - R) + (H + G - A_k^{-1})(N - A_k + R)] \\
&= \frac{1}{2}[I - GA_k - HA_k - A_k^{-1}R + GR + HR + HN - HA_k + HR + GN - GA_k \\
&\quad + GR - A_k^{-1}N + I - A_k^{-1}R] \\
&= I + \frac{1}{2}[2GR + 2HR - 2GA_k - 2HA_k - 2A_k^{-1}R + HN + GN - A_k^{-1}N] = I.
\end{aligned}$$

Hence (a) is proved.

By using the induction assumption, it is easy to show that each row sum of $(A_k^{-1} + G) + (A_k^{-1} - G)P$ is equal to zero except the first row. Also it is clear that the first row sum of $(A_k^{-1} + G) + (A_k^{-1} - G)P$ is equal to two. Moreover, we have

$$\begin{aligned}
A_k^{-1} - G - H + (H + G - A_k^{-1})P &= A_k^{-1} - G - H + H + (G - A_k^{-1})P \\
&= (A_k^{-1} - G) + (G - A_k^{-1})P.
\end{aligned}$$

Therefore we can show that each row sum of $(A_k^{-1} - G - H) + (H + G - A_k^{-1})P$ is equal to zero. Thus (b) and (c) are proved. □

By using Theorem 3.1, a recursive method can be developed to solve the linear system $\Theta = \Gamma A_n$. The next theorem states how to solve the linear system $\Theta = \Gamma A_n$ without storing A_n^{-1}.

Theorem 3.2. *The complexity for solving* Γ *in* $\Theta = \Gamma A_n$ *with* n *loci is of* $O(n2^n)$.

Proof. To begin with, let us consider the complexity for calculating $2^{n-1}\Theta A_{n+1}^{-1}$ given that the computational complexity of the inverse of $2^{n-2}XA_n^{-1}$ is $\psi(n)$, where X is a 1-by-2^{n-1} vector. By Theorem 3.1, we have

$$\Theta A_{n+1}^{-1} = \frac{1}{2}\begin{pmatrix}\Theta_1 & \Theta_2\end{pmatrix}\begin{pmatrix} A_n^{-1} + G & (A_n^{-1} - G)P \\ A_n^{-1} - G - H & (H + G - A_n^{-1})P \end{pmatrix}$$

where $\Theta = (\Theta_1, \Theta_2)$. It implies that

$$2^{n-1}\Theta A_{n+1}^{-1} = \begin{pmatrix} 2^{n-2}(\Theta_1 + \Theta_2)A_n^{-1} + 2^{n-2}(\Theta_1 - \Theta_2)G + -2^{n-2}\Theta_2 H \\ 2^{n-2}(\Theta_1 - \Theta_2)A_n^{-1}P - 2^{n-2}(\Theta_1 - \Theta_2)GP + 2^{n-2}\Theta_2 HP \end{pmatrix}^T.$$

Firstly, we observe that the cost for $2^{n-2}G$ requires one operation and there is no computational cost for $2^{n-2}H$ as they are given by

$$2^{n-2}G = \begin{pmatrix} 2^{n-2} & 0 \cdots & 0 \\ 0 & 0 \cdots & 0 \\ \vdots & \vdots \ddots & \vdots \\ 0 & 0 \cdots & 0 \end{pmatrix} \quad \text{and} \quad 2^{n-2}H = \begin{pmatrix} 1 & 1 \cdots & 1 \\ 0 & 0 \cdots & 0 \\ \vdots & \vdots \ddots & \vdots \\ 0 & 0 \cdots & 0 \end{pmatrix}.$$

The computational cost for obtaining either $(\Theta_1 + \Theta_2)$ or $(\Theta_1 - \Theta_2)$ requires 2^{n-1} operations. The cost for $2^{n-2}(\Theta_1+\Theta_2)A_n^{-1}$ and $2^{n-2}(\Theta_1-\Theta_2)A_n^{-1}$ requires $2\psi(n)$ operations. The cost for $2^{n-2}(\Theta_1-\Theta_2)G$ requires one operation as $2^{n-2}G$ contains only one non-zero element in $2^{n-2}G$. Similarly, there is no cost involved for the computation of $2^{n-2}\Theta_2 H$. This is also true for the matrix multiplication of P as it is just a permutation. Thus, the total computational cost $\psi(n+1)$ of $2^{n-1}\Theta A_{n+1}^{-1}$ is equal to $2\psi(n) + 5 \cdot 2^{n-1} + 4$. It is easy to deduce that

$$\psi(n+1) = 3 \cdot 2^{n-1} + 5(n-1)2^{n-1} + 4 \cdot (2^{n-1} - 1) = 5n \cdot 2^{n-1} + (2^n - 4) = O(n2^n).$$

Hence the result follows. □

Theorem 3.3. *The storage cost for solving* Γ *in* $\Theta = \Gamma A_n$ *with* n *loci is* $3 \cdot 2^n - 5$.

Proof. To begin with, let us denote the storage cost for computing $2^{n-2}\Theta A_n^{-1}$ by $\phi(n)$. According to Theorem 3.2, we need to store such components

$$2^{n-2}G, \quad 2^{n-2}H, \quad \Theta_1 \quad \text{and} \quad \Theta_2.$$

Their corresponding storage cost are are 1, 2^{n-1}, 2^{n-1} and 2^{n-1} respectively. The computational procedure of solving Γ in $\Theta = \Gamma A_n$ is summarized as follows:

Procedure	Current Storage requirement
Start with $2^{n-2}A_n^{-1}$	$\phi(n)$
Load Θ_1, Θ_2	$\phi(n) + 2^n$
Compute $\Theta_1 + \Theta_2$, $\Theta_1 - \Theta_2$	$\phi(n) + 2^n + 2^n$
Remove Θ_1	$\phi(n) + 2^n + 2^{n-1}$
Compute $X_1 = 2^{n-2}(\Theta_1 + \Theta_2)A_n^{-1}$	$\phi(n) + 2^n + 2^{n-1} + 2^{n-1}$
Remove $\Theta_1 + \Theta_2$	$\phi(n) + 2^n + 2^{n-1}$
Compute $X_2 = 2^{n-2}(\Theta_1 - \Theta_2)A_n^{-1}$	$\phi(n) + 2^n + 2^{n-1} + 2^{n-1}$
Remove $2^{n-2}A_n^{-1}$	2^{n+1}
Compute $X_2 = X_2 P$	2^{n+1}
Create $2^{n-2}G$	$2^{n+1} + 1$
Compute $Y = 2^{n-2}(\Theta_1 - \Theta_2)G$	$2^{n+1} + 1 + 1$
Remove $\Theta_1 - \Theta_2$, $2^{n-2}G$	$2^n + 2^{n-1} + 1$
Compute $2^{n-2}\Theta_2 H$	$2^n + 2^{n-1} + 1 + 2^{n-1}$
Remove Θ_2	$2^n + 2^{n-1} + 1$
Compute $X_1 + Y - 2^{n-2}\Theta_2 H$	$2^n + 2^{n-1} + 1 + 2^{n-1}$
Remove X_1	$2^n + 2^{n-1} + 1$
Compute $Y = YP$	$2^n + 2^{n-1} + 1$
Compute $X_2 - Y + 2^{n-2}\Theta_2 H$	$2^n + 2^{n-1} + 1 + 2^{n-1}$
Remove $X_2, Y, 2^{n-2}\Theta_2 H$	2^n

Table 1: The Storage of the Algorithm.

From the above procedure, the maximum storage requirement is either $\phi(n) + 2^{n+1}$ or $2^{n+1} + 1$. Since $\phi(n+1) = \phi(n) + 2^{n+1} = \cdots = 2^{n+2} - 5 + 2^{n+1}$, the total storage requirement is $3 \cdot 2^{n+1} - 5$. □

3.1. *Computational Results*

In this subsection, we demonstrate the effectiveness of the proposed recursive solver for solving $\Theta = \Gamma A_n$. Here we perform our test in a MATLAB platform with CPU=AMP 1800+ and memory=512Mb. Table 2 shows the times (in seconds) required for computing ΘA_n^{-1} and the ratio between the computational times of ΘA_n^{-1} $\Theta' A_{n-1}^{-1}$. We remark that the complexity of the proposed recursive algorithm for the n loci problem is of $O((n-1)2^n)$. From Table 1, we find that the computational times only increase linearly with respect to n for our tested cases. It clearly shows that the proposed recursive method is highly efficient.

n	10	11	12	13	14	15	16	17
time (seconds)	0.05	0.11	0.22	0.33	0.77	1.43	2.86	5.65
ratio	–	2.20	2.00	1.50	2.33	1.86	2	1.98
n	18	19	20	21	22	23	24	25
time (seconds)	11.37	22.91	46.08	92.83	187.68	379.04	765.94	1812.82
ratio	2.01	2.01	2.01	2.01	2.02	2.02	2.02	2.37

Table 1: The Computational Times for different n.

4. Concluding Remarks

In this paper, we give a systematic formulation for the linkage analysis problem and an efficient recursive solver is also proposed for solving the haplotype frequencies in multiple loci linkage analysis. The complexity of our method is shown to be $O((n-1)2^n)$ for n loci problem. It is much more efficient when compared to $O(2^{3n})$ operations required by the classical Gaussian elimination method. Previous applications of the linkage analysis only consider small values of n, see for instance (Sham 1998 and Zhao 1990). With our formulation of the problem and also the fast recursive solver, practitioners can now consider larger n and we expect the method can be more popular.

References

1. D. Curtis, Another Procedure for the Preliminary Ordering of Loci on Two-Point Lod Scores, *Annals of Human Genetics*, 58, 65-75, 1994.
2. G. Golub and C. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore 1989.
3. L. Kruglyak and E. S. Lander, High-Resolution Genetic Mapping of Complex Traits, *American Journal of Human Genetics*, 56, 1212-23, 1995.
4. U. Liberman and S. Karlin, Theoretical Models of Genetic MapFunctions, *Theoretical Population Biology*, 25, 331-46, 1984.
5. P. Sham, *Statistics in Human Genetics*, Edward Arnold, 1998.
6. D. E. Weeks, G. M. Lathrop and J. Ott, Multipoint Mapping under Genetic Interference, *Human Heredity*, 43, 86-97, 1993.
7. L. P. Zhao, E. Thompson and R. Prentice, Joint Estimation of Recombination Fractions and Interference Coefficients in Multilocus Linkage Analysis, *American Journal of Human Genetics*, 47, 255-265, 1990.