Mathematical Biosciences 231 (2011) 120–125

Contents lists available at ScienceDirect

Mathematical Biosciences

journal homepage: www.elsevier.com/locate/mbs

Mathematical properties and bounds on haplotyping populations by pure parsimony

I-Lin Wang *, Chia-Yuan Chang

Department of Industrial and Information Management, National Cheng Kung University, No.1 University Rd., Tainan 701, Taiwan

ARTICLE INFO

Article history:
Received 8 September 2009
Received in revised form 20 August 2010
Accepted 18 February 2011
Available online 24 February 2011

Keywords:
Haplotype inference
Pure parsimony
Bioinformatics
Integer programming
Compatible graph

ABSTRACT

Although the haplotype data can be used to analyze the function of DNA, due to the significant efforts required in collecting the haplotype data, usually the genotype data is collected and then the population haplotype inference (PHI) problem is solved to infer haplotype data from genotype data for a population. This paper investigates the PHI problem based on the pure parsimony criterion (HIPP), which seeks the minimum number of distinct haplotypes to infer a given genotype data. We analyze the mathematical structure and properties for the HIPP problem, propose techniques to reduce the given genotype data into an equivalent one of much smaller size, and analyze the relations of genotype data using a compatible graph. Based on the mathematical properties in the compatible graph, we propose a maximal clique heuristic to obtain an upper bound, and a new polynomial-sized integer linear programming formulation to obtain a lower bound for the HIPP problem.

1. Introduction

The Post-Genomic Era focuses on functional genome analysis, including studying the knowledge about the genetic constitution of an individual chromosome called the haplotypes. Information from the haplotype data can be applied in various domains, such as linkage disequilibrium, inference of population evolutionary history, disease diagnosis, and customization of treatment for each individual [1]. However, since the time, labor, and expense involved in directly collecting the haplotype data require too much resources and efforts, the researchers usually collect the descriptions of one conflated pair of haplotypes called the genotype data, rather than the haplotype data [2] for further analysis. We are interested in solving the population haplotype inference (PHI) problem which infers the haplotype data for a population of diploid species from their genotype data. Since each genotype has to be resolved by a pair of haplotypes from a large number of possible haplotype pair candidates, the PHI problem is a difficult combinatorial problem.

Although there are many possible haplotype pairs for resolving a given genotype matrix, the real-world haplotype pairs are constituted by a very few amount of distinct haplotypes. For example, Drysdale et al. [3] identify 13 SNPs in the human β_2AR gene, which can be composed by many (e.g. with number up to $2^{13} = 8192$) possible haplotype combinations. However, among all the possible haplotype combinations, only 10 haplotypes are related to asthmatic cohort. Thus Gusfield [4] suggests a combinatorial optimization problem called the haplotype inference based on pure parsimony (HIPP) which seeks the minimum amount of distinct haplotypes to resolve a given genotype matrix.

Suppose we have m genotypes and each genotype contains n sites. These data can be expressed by an $m \times n$ genotype matrix G, where each row in the genotype matrix corresponds to a genotype data for one individual, each column stands for one SNP, and each element in G has value 0, 1, or 2. A site is called *homozygous wild type* if it has value 0, *homozygous mutant type* if it has value 1, and *heterozygous* if it has value 2. A site in a genotype is resolved if it has value 0 or 1, and ambiguous if it has value 2. A genotype is called resolved if there are two haplotypes such that for every site with value 0 or 1 in that genotype, the value of that site in the two haplotypes are either both with value 0 or both with value 1; for every site with value 2 in that genotype, one of the haplotype must have value 0 and the other haplotype must have value 1 in that site. The objective of an HIPP problem is to find a $2m \times n$ haplotype matrix, in which the ith row in the genotype matrix is resolved by the $(2i-1)$th and the $2i$th rows in the haplotype matrix, and the number of distinct haplotypes is minimized.

Take Fig. 1 for example. Given a genotype matrix $G = \{202,021,212\}$, there are 2, 1, and 2 possible haplotype pairs to resolve genotype 1, 2, and 3, respectively. Furthermore, there are 6, 5, 5, or 4 distinct haplotypes if we select (p_1, p_3, p_4), (p_1,p_3,p_5), (p_2,p_3,p_4) or (p_2,p_3,p_5) to resolve the genotype matrix, respectively. Using the pure parsimony criterion, (p_2,p_3,p_5) will be selected to resolve all the genotypes, since this combination induces the minimum number of distinct haplotypes.

* Corresponding author. Tel.: +886 6 2757575 53123; fax: +886 6 2362162.
E-mail address: ilinwang@mail.ncku.edu.tw (I-L. Wang).

0025-5564/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.mbs.2011.02.008

$$g_1 = 202 \Rightarrow p_1 = (000,101);\ p_2 = (001,100)$$

$$g_2 = 021 \Rightarrow p_3 = (001,011)$$

$$g_3 = 212 \Rightarrow p_4 = (010,111),\ p_5 = (011,110)$$

select $p_1, p_3, p_4 \Rightarrow 6$ distinct haplotypes

select $p_1, p_3, p_5 \Rightarrow 5$ distinct haplotypes

select $p_2, p_3, p_4 \Rightarrow 5$ distinct haplotypes

select $p_2, p_3, p_5 \Rightarrow 4$ distinct haplotypes

Fig. 1. A PHI example by pure parsimony criterion.

The HIPP problem has been shown to be an APX-hard problem by Lancia et al. [5]. The solution methods in the literature for solving the HIPP problem are either based on mathematical programming techniques such as integer linear programming (see [4,6]) and quadratic integer programming (see [7,8]), or heuristic algorithm (see [9]). In particular, Gusfield [4] gives the first ILP formulation called RTIP to model the HIPP problem. Although the RTIP is potentially exponential-sized, it is practically faster than the PolyIP, the polynomial-sized ILP formulation proposed by Brown and Harrower [6]. Both RTIP and PolyIP calculate the exact optimal solution for the HIPP problem, but they usually consume a lot of computational resources and time, and are not suitable for solving large-scale HIPP problems. Wang and Xu [10] give a branch and bound algorithm called HAPAR. HAPAR also takes a lot of computational time since it tries out all the combinations to solve the HIPP problem. Based on different Integer Quadratic Programming formulations, Huang et al. [7] give an approximation algorithm called SDPHapInfer and Kalpakis and Namjoshi [8] propose a heuristic algorithm to solve the HIPP problem. SDPHapInfer performs well for smaller cases but its error rates increase dramatically for larger cases (see [11], for details).The heuristic algorithm by Kalpakis and Namjoshi [8] require further investigation. On the other hand, the parsimonious tree growing heuristic algorithm called PTG by Li et al. [9] is very fast, but its effectiveness in terms of the optimality gap (i.e. the difference in the number of distinct candidate haplotypes used, compared with the optimal solution) remains to be evaluated.

Recently, Boolean Satisfiability (SAT) has been proposed for solving the HIPP problem with success. In particular, the SAT-based method by Lynce and Marques-Silva [12,13] can calculate exact optimal solutions for large-scale HIPP problems. The Pseudo-Boolean Optimization (PBO) model proposed by Graca et al. [14,15] also use similar techniques based on the PolyIP model, and are efficient for solving larger HIPP cases. Although these methods also discuss compatible relations between genotypes, they do not use these properties to construct integer programming model as proposed in this paper.

To design a better algorithm for solving the HIPP problem, this paper investigates the mathematical properties of the HIPP problem. We identify several mathematical properties which are useful in reducing the complexity of the HIPP problem. By introducing compatible relations between genotypes, we first give a heuristic to estimate the upper bound for the HIPP problem, and then estimate its lower bound by a new polynomial-sized ILP formulation.

The rest of this paper is organized as follows. Section 2 introduces the notations and compatible graph used for our analysis. Section 3 discusses the mathematical properties of the HIPP problem. Our upper bound heuristic and new ILP formulation for calculating the lower bound of the HIPP problem are illustrated in Section 4. Section 5 concludes the paper.

2. Preliminaries

Let $G = \{g_1, g_2, \ldots, g_n\} = [g_{i,j}]$ be an $m \times n$ genotype matrix, where each row $g_i = \{g_{i,1}, g_{i,2}, \ldots, g_{i,n}\}$. If $g_i = h_\alpha \otimes h_\beta$, we say genotype g_i can be resolved by a candidate haplotype pair h_α and h_β, where for each $j = 1, \ldots, n$, $(h_{\alpha,j}, h_{\beta,j}) = (0,0)$ or $(1,1)$ when $g_{i,j} = 0$ or 1, or $(h_{a,j}, h_{b,j}) = (0,1)$ or $(1,0)$ when $g_{i,j} = 2$, respectively. In such a case, we also say that h_α is the *conjugate* (or *complementary*) haplotype for h_β in resolving g_i. Note that any two of the three elements in the set $\{g_i, h_\alpha, h_\beta\}$ can uniquely determine the remaining $1 \times n$ vector by the relation $g_i = h_\alpha \otimes h_\beta$. For our convenience, we also define $G = G_0 \cup G_1 \cup G_2$, where G_0, G_1, and G_2 represent the set of genotypes that contains zero, one, and at least two heterozygous sites, respectively.

A genotype may be resolved by many possible haplotype pairs. Denote $CHP(i)$ the set of candidate haplotypes pairs that can resolve genotype g_i. Let ς be a $1 \times n$ row vector with element 0, 1, or 2. We say ς_a is *compatible* with ς_b, denoted by $\varsigma_a \sim \varsigma_b$, if $(\varsigma_{a,j}, \varsigma_{b,j}) \notin \{(0,1),(1,0)\}$ for each $j = 1, \ldots, n$; otherwise, we say ς_a is *incompatible* with ς_b, denoted by $\varsigma_a \nsim \varsigma_b$. By the relations of compatibility between different genotypes in G, we can construct an undirected compatible graph of m nodes and $\tilde{a}$ arcs, denoted by $CG(G)$, where each node α represents a genotype g_α in G and node α connects node β by an arc when $g_\alpha \sim g_\beta$. Fig. 2 gives an example of a compatible graph.

When $g_{\alpha,j} = g_{\beta,j}$ for each $j = 1, \ldots, n$ and $\alpha \neq \beta$, we say g_α is a *duplicated* genotype of g_β, denoted by $g_\alpha \equiv g_\beta$. When two haplotype pairs $(h_{\alpha_1}, h_{\alpha_2})$ and $(h_{\beta_1}, h_{\beta_2})$ satisfy that $h_{\alpha_1} = h_{\beta_1}$ (thus $h_{\alpha_2} = h_{\beta_2}$) or $h_{\alpha_1} = h_{\beta_2}$ (thus $h_{\alpha_2} = h_{\beta_1}$), we say they are equivalent, denoted by $(h_{\alpha_1}, h_{\alpha_2}) \equiv (h_{\beta_1}, h_{\beta_2})$. Let $Z(G)$ represent the optimal objective value for an HIPP problem with an input genotype matrix G.

In the next section, we will give several mathematical properties for solving the HIPP problem. The summarized mathematical properties are useful in reducing the complexity of the HIPP problem, which help to design a better algorithm or formulation for solving the HIPP problem.

3. Mathematical properties for the HIPP problem and compatible graph

Here we propose several mathematical properties useful for developing efficient HIPP algorithms.

Property 1. *Different genotypes g_α and g_β cannot be resolved by the same haplotype pair. In other words, if $g_\alpha \neq g_\beta$ for some $\alpha \neq \beta$, then $CHP(\alpha) \cap CHP(\beta) = \varnothing$.*

Proof. Suppose $CHP(\alpha) \cap CHP(\beta) \neq \varnothing$ and $(h_a, h_b) \in CHP(\alpha) \cap CHP(\beta)$. Then $g_\alpha = h_a \otimes h_b = g_b$, which contradicts the assumption that $g_\alpha \neq g_\beta$. □

Property 2. *Suppose $g_\alpha \equiv g_\beta$ for some $\alpha \neq \beta$, $g_\alpha, g_\beta \in G$. Let $G' = G \backslash \{g_\beta\}$, then $Z(G') = Z(G)$.*

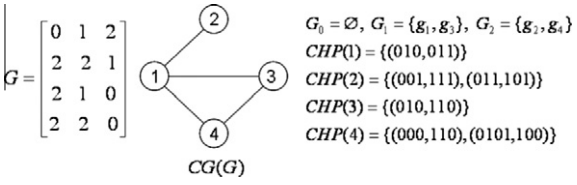

Fig. 2. A compatible graph example.

Proof. Let $(h_{\alpha_1}, h_{\alpha_2})$ and $(h_{\beta_1}, h_{\beta_2})$ be the optimal solution of the HIPP problem that resolves g_α and g_β in G, respectively. Whether $(h_{\alpha_1}, h_{\alpha_2}) \equiv (h_{\beta_1}, h_{\beta_2})$ or not, resolving g_β by $(h_{\alpha_1}, h_{\alpha_2})$ will not increase $Z(G)$ since both h_{α_1} and h_{α_2} have been counted in $Z(G)$. Since resolving g_β by $(h_{\alpha_1}, h_{\alpha_2})$ has the same effect as removing g_β from G, thus $Z(G') \leqslant Z(G)$.

Suppose $Z(G') < Z(G)$, then we may construct another genotype G'' by adding a g_β to G'. Since the optimal solution for resolving G' and $(h_{\alpha_1}, h_{\alpha_2})$ for g_β can be used as a feasible solution for resolving G'' with the objective value equal to $Z(G')$, we know $Z(G'') \leqslant Z(G')$. However, $G'' \equiv G$ and thus $Z(G) \leqslant Z(G')$ contradicting the assumption of $Z(G') < Z(G)$. Therefore, $Z(G') = Z(G)$. □

Property 3. *Suppose that $g_{i,\alpha} \in \{0, 1\}$ for each $i = 1, \ldots, m$, then we may divide the original HIPP problem into at most two HIPP subproblems where all the elements in the αth column of one subproblem have value 0 and the elements of the same column in the other subproblem have value 1.*

Proof. If all the elements in the αth column of G have the same value 0 (or 1), all the candidate haplotypes must have the same value 0 (or 1) in their αth column. Thus removing the entire αth column will not affect the original optimal solution.

Similarly, if some but not all of the genotypes have their αth element equal to 0 (or 1), their candidate haplotypes must also have their αth element equal to 0 (or 1), which also means that these candidate haplotypes cannot resolve those genotypes whose αth element equal to 1 (or 0). Therefore, we can divide G into two submatrices by the values in their αth column. □

Property 4. *Duplicated or complementary columns can be removed.*

Proof. See Section 7.1.3 in Brown and Harrower [6] and Section 5 in Li et al. [9] for details. □

These four properties are helpful in designing more efficient algorithms or providing good preprocessing operations to reduce the complexity of the HIPP problem. For example, Properties 2 and 4 imply a preprocessing procedure that removes duplicated rows, duplicated columns, or complementary columns will not affect the optimal solution to the original HIPP problem. In practice, these techniques do help to reduce the size of the formulations (see [4,6,9]), especially for those formulations with size exponential to the number of columns.

Property 3 also suggests a divide-and-conquer technique, where several subproblems with fewer rows and columns can be identified according to the value of 0 or 1. The same property also implies that each subproblem will share common solutions, so that the merge of the solutions can be easily implemented.

When we reduce the size of the original genotype matrix by these properties, note that we should not remove any column or row that contains all 2's, since different position of the "0" and "1" that resolves each "2" will affect the objective value of the HIPP problem. For convenience, the genotype matrix G to be analyzed afterwards in this paper is assumed to contain no duplicated rows, duplicated columns, and complementary columns.

4. Upper and lower bounds for the HIPP problem based on the compatible graph

4.1. An upper bound based on maximal cliques in the compatible graph

Using the compatible graph $CG(G)$, we may derive a theoretical lower bound and upper bound for the optimal objective value of the HIPP problem.

Lemma 1

(a) *For any clique C of size $|C|$ in $CG(G)$, we can always resolve those genotypes in C by at most $|C| + 1$ haplotypes. Furthermore, if there exists a genotype in C and it belongs to G_0 (i.e. it contains no value of 2), we can reduce the upper bound of the inferred haplotypes to be $|C|$.*

(b) *Suppose we can cover all the nodes in $CG(G)$ by w cliques, $C_1, C_2, \ldots, C_w$, then we can always resolve G by at most $m + w$ haplotypes. If there are totally σ genotypes that belong to G_0, we can reduce the upper bound of the inferred haplotypes to be $m + w - \sigma$.*

Proof

(a) Any clique in a compatible graph represents a set of compatible genotypes where any two of them can be resolved by a common haplotype. By definition of the compatible relations, there must exist a common haplotype that can resolve all the genotypes in a clique. In particular, that common haplotype can be easily determined column by column: whenever a column among those genotypes in the same clique contains 1 or 0 (other than 2), we let the common haplotype to have 1 (or 0), respectively in that site; otherwise, we are free to select either 1 or 0 to be on that site. Based on the common haplotype, one can easily derive the other $|C|$ haplotypes to resolve those $|C|$ genotypes in C.
If there exists a genotype in C and it belongs to G_0, such a genotype can be uniquely resolved by a pair of the same haplotypes. Since we assume that G contains no duplicated genotypes, there exists at most one such genotype in each clique. Thus the total number of distinct inferred haplotypes in this case is at most $|C| + 1 - 1 = |C|$.

(b) By (a), we can resolve all the genotypes in C_i by at most $|C_i| + 1$ haplotypes. Thus totally we can resolve G by at most $\sum_{i=1}^{w}(|C_i| + 1) = m + w$ haplotypes.
If there are totally σ genotypes whose elements are either 0 or 1, each of such genotypes must be located in different clique, or otherwise there exist duplicated genotypes. By (a), each clique C_i that contains such a genotype can be resolved by at most $|C_i|$ haplotypes, which makes the upper bound of the inferred haplotypes become $m + w - \sigma$. □

Lemma 1 suggests a heuristic to solve the HIPP problem, although not to the optimality. In particular, one may iteratively solve the maximum clique problem over the compatible graph $CG(G)$ to find the maximum clique $\widehat{C}$, then remove $\widehat{C}$ which constructs a reduced compatible graph $CG(\widehat{G}) = CG(G) \setminus \widehat{C}$, where $\widehat{G}$ contains the remaining genotypes. These steps can be repeated until the reduced compatible graph becomes empty. Since solving a maximum clique problem is NP-complete, one may instead solve a maximal clique problem which can be done in polynomial time.

Here we give an example to show there may exist better solution than the solution obtained on the maximal clique heuristic. Suppose $G = \{g_1, g_2, g_3\} = \{210, 012, 212\}$, one can easily identify a common haplotype 010 and draw the compatible graph that forms a clique of size 3. Using the maximal clique heuristic, we obtain the solution as $g_1 = 010 \otimes 110$, $g_2 = 010 \otimes 011$, and $g_3 = 010 \otimes 111$, where totally 4 distinct haplotypes are used. However, the optimal solution is to resolve g_3 by $110 \otimes 011$, where totally 3 distinct haplotypes are sufficient.

4.2. A lower bound based on a polynomial-sized integer linear programming formulation

The compatible graph also provides a means to estimate the lower bound of the HIPP problem. Given an $m \times n$ genotype matrix

G, where there exist no duplicated rows, duplicated columns, and complementary columns, we may seek the best possible combination of the inferred haplotypes and genotypes based on their compatible relations defined in $CG(G)$. To simplify the original HIPP problem and concentrate on the compatible relations between inferred haplotypes and genotypes, we neglect the detailed site values for each candidate haplotype and represent it by an abstract "object" called *haplotype object*. At first glance, we may think that each genotype is associated with at most two haplotype objects, and the problem becomes to select as few haplotype objects as possible without violating the compatible relations.

Note that not all the genotypes require to be associated with two haplotype objects. For example, any $g_i \in G_0$ suffices to have a single haplotype object h_{2i-1} since its inferred haplotype pair contains identical haplotypes. It is also possible to further reduce the number of haplotype objects for each $g_i \in G_1$, depending on its compatible relations with other genotypes. Let $CGT(i) = \{g_\gamma : g_\gamma \sim g_i\}$ represent the set of all the genotypes g_γ compatible with g_i (i.e. the set of the nodes adjacent to i in the compatible graph). Note that each $g_i \in G_1$ cannot have more than two compatible genotypes belonging to G_0, thus $CGT(i)$ can only fall into one of the three cases: (1) only two genotypes $g_{\gamma_1}, g_{\gamma_2} \in CGT(i)$ belong to G_0, but all the other genotypes in $CGT(i) \setminus \{g_{\gamma_1}, g_{\gamma_2}\}$ belong to $G_1 \cup G_2$; (2) only one $g_\gamma \in CGT(i)$ belongs to G_0 but all the other genotypes in $CGT(i)\setminus\{g_\gamma\}$ belong to $G_1 \cup G_2$; and (3) each $g_\gamma \in CGT(i)$ belongs to $G_1 \cup G_2$. For the first case, we know immediately that $g_i = g_{\gamma_1} \otimes g_{\gamma_2}$. Since both g_{γ_1} and g_{γ_1} are in G_0, which means each of them is respectively associated with a single haplotype object, it is no longer necessary to associate any haplotype object to g_i. For the second case, since g_γ must be one of the inferred haplotype to g_i, it suffices to associate g_i with only a single haplotype object h_{2i-1}. For the third case, since there is no clear clue to cut off any haplotype objects, we associate two haplotype objects h_{2i-1} and h_{2i} to g_i. Similarly to the third case of $g_i \in G_1$, for each $g_i \in G_2$ we also associate it with two haplotype objects h_{2i-1} and h_{2i}. Note that in our setting, if h_{2i} is associated with h_{2i}, then implicitly h_{2i-1} will also exist.

Take the genotypes in Fig. 3 for example, there are $m = 7$ genotypes. Since $g_2 \in G_0$, we associate it with h_3 only. Among those 3 genotypes in G_1, both g_4 and g_7 fall into case 2, while g_5 corresponds to the case 3. Thus we assign h_7 to g_4, h_{13} to g_7, and h_9 and h_{10} to g_5. Similar tog$_5$, we respectively assign two haplotype objects to each genotype in $G_2 = \{g_1, g_3, g_6\}$.

By introducing the haplotype objects, we try to identify the haplotype objects that can be shared as more genotypes as possible and satisfy the given compatible relations between genotypes. Since we look for the best haplotype objects instead of the exact haplotype vectors, the solution we obtain can be used as a lower bound for the original HIPP problem.

For our convenience, we define $CHO(i) = \{k : g_{\lceil k/2 \rceil} \sim g_i\}$ to represent the index set for those candidate haplotype objects compatible to g_i, and $IG(k) = \{i : g_i \sim g_{\lceil k/2 \rceil}\}$ to represent the index set for those genotypes compatible to h_k. In our formulation, for each $g_i \in G$, we assign a binary variable $x_{i,k}$ to represent whether the genotype g_i selects the haplotype object h_k (i.e. $x_{i,k} = 1$) or not (i.e. $x_{i,k} = 0$) among all possible $k \in CHO(i)$. For each haplotype object h_k, we assign a binary variable y_k to represent whether h_k has been selected to resolve some genotype in the optimal solution (i.e. $y_k = 1$) or not (i.e. $y_k = 0$). Thus the objective function becomes to minimize $\sum_k y_k$. We give the following ILP formulation (HIPP$_{LB}$) to calculate a good lower bound for the HIPP problem.

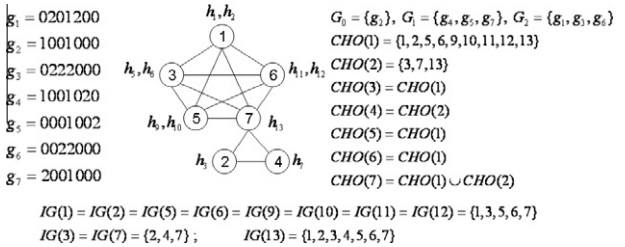

Fig. 3. An illustrative example for explaining the haplotype objects.

$$\min \sum_k y_k \qquad (\text{HIPP}_{\text{LB}})$$

$$\text{s.t.} \sum_{i \in IG(k)} x_{i,k} \leqslant M y_k \; \forall k \qquad (1)$$

$$x_{i,2i-1} \geqslant 1 \quad \forall i \in G_0 \cup G_1 \text{ and } h_{2i} \text{ does not exist} \qquad (2)$$

$$x_{i,2i-1} + x_{i,2i} \geqslant 1 \quad \forall i \in G_1 \text{ and } h_{2i} \text{ exists} \qquad (3)$$

$$1 \leqslant \sum_{k \in CHO(i)} x_{i,k} \leqslant 2 \quad \forall i \in G_1 \cup G_2 \text{ and } h_{2i} \text{ exists} \qquad (4)$$

$$\sum_{k \in CHO(\alpha) \cap CHO(\beta)} u_{\alpha,\beta,k} \leqslant 1 \quad \forall g_\alpha, g_\beta \in G_2 \text{ such that } g_\alpha \sim g_\beta \qquad (5)$$

$$x_{\alpha,k} + x_{\beta,k} - u_{\alpha,\beta,k} \leqslant 1 \quad \forall k \in CHO(\alpha) \cap CHO(\beta), \forall g_\alpha, g_\beta \in G_2 \text{ such that } g_\alpha \sim g_\beta \qquad (6)$$

$$x_{\alpha,2\beta-1} + x_{\alpha,2\beta} \leqslant 1 \quad \forall g_\alpha \in G_2, g_\beta \in G_1 \text{ such that } g_\alpha \sim g_\beta \text{ and } h_{2\beta} \text{ exists} \qquad (7)$$

$$x_{\alpha,k} + x_{\beta,k} \leqslant 1 \quad \forall g_\alpha, g_\beta \in G_2, g_\alpha \nsim g_\beta, \quad \forall k \in CHO(\alpha) \cap CHO(\beta) \qquad (8)$$

$$y_k, x_{i,k}, u_{\alpha,\beta,k} \in \{0,1\} \quad \forall k, i, \alpha, \beta \qquad (9)$$

For each haplotype object h_k, constraints (1) force the optimal solution to select it (i.e. $y_k = 1$), as long as it is used to infer any of its compatible genotypes g_i (i.e. $x_{i,k} = 1$). Note that M is a number no less than $|IG(k)|$. Besides the binary variable constraints (9), constraints (2)–(8) can be categorized into four types:

(a) For each genotype $g_i \in G_0 \cup G_1$, at least one of its inferred haplotype objects (either h_{2i-1}, or h_{2i} if it exists) can be uniquely determined, and therefore it has to be selected to resolve g_i (i.e. $x_{i,2i-1} + x_{i,2i} \geqslant 1$, if h_{2i} exists, which also implies the existence of h_{2i-1}; otherwise $x_{i,2i-1} \geqslant 1$). This is formulated in constraints (2) and (3).

(b) For each genotype $g_i \in G_1 \cup G_2$ that is associated with two haplotype objects (i.e. h_{2i} exists), either one or two haplotype objects from any of its compatible genotypes (including itself) have to be selected. In other words, we have to select at least one and at most two among those haplotypes objects compatible to g_i in the solution. This is formulated in constraint (4).

(c) Since there exist no duplicated rows, by Property 1 in Section 3, any two compatible genotypes will not select more than one common haplotype object. In particular, for each pair of adjacent nodes α and β in the compatible graph (i.e. $g_\alpha \sim g_\beta$), we discuss the following cases:

Case (c.1) Both g_α and g_β are in G_2: we assign a binary variable $u_{\alpha,\beta,k}$ to represent whether both g_α and g_β select h_k (i.e. $u_{\alpha,\beta,k} = 1$) or not (i.e. $u_{\alpha,\beta,k} = 0$). In this case, g_α and g_β will not select more than one common haplotype object by Property 1 in Section 3, thus $\sum_{k \in CHO(\alpha) \cap CHO(\beta)} u_{\alpha,\beta,k} \leqslant 1$, and $x_{\alpha,k} + x_{\beta,k} - u_{\alpha,\beta,k} \leqslant 1$ for each $k \in CHO(\alpha) \cap CHO(\beta)$, as formulated in constraints (5), (6).

Case (c.2) $g_\alpha \in G_2$, $g_\beta \in G_1$ and g_β is associated with two haplotype objects: In this case, $x_{\alpha,2\beta-1} + x_{\alpha,2\beta} \leqslant 1$ ensures that g_α will not select those two haplotype objects associated with g_β at the same time. This is done in constraints (7).

(d) Any two incompatible genotypes will not select any common haplotype object. To formulate this, for each pair of nonadjacent nodes α and β in the compatible graph (i.e. $g_\alpha \nsim g_\beta$), if both g_α and g_β are in G_2, we set $x_{\alpha,k} + x_{\beta,k} \leqslant 1$ to ensure they will not select any common haplotype object h_k, for each $k \in CHO(\alpha) \cap CHO(\beta)$, as formulated in constraints (8).

We now use the compatible graph in Fig. 3 to illustrate our formulation as follows: The objective function is to minimize $y_1 + y_2 + y_3 + y_5 + y_6 + y_7 + y_9 + y_{10} + y_{11} + y_{12} + y_{13}$. The case of $k = 1$ and $k = 3$ in constraints (1) are $x_{1,1} + x_{3,1} + x_{5,1} + x_{6,1} + x_{7,1} \leqslant 5y_1$ and $x_{2,3} + x_{4,3} + x_{7,3} \leqslant 5y_3$. Constraints (2) are composed by $x_{2,3} = 1, x_{4,7} = 1$, and $x_{7,13} = 1$. Constraint (3) gives $x_{5,9} + x_{5,10} \geqslant 1$. The case of $i = 1$ in constraints (4) is formulated by $x_{1,1} + x_{1,2} + x_{1,5} + x_{1,6} + x_{1,9} + x_{1,10} + x_{1,11} + x_{1,12} + x_{1,13} = 2$. The case of $\alpha = 1$ and $\beta = 3$in constraints (5) corresponds to $u_{1,3,1} + u_{1,3,2} + u_{1,3,5} + u_{1,3,6} + u_{1,3,9} + u_{1,3,10} + u_{1,3,11} + u_{1,3,12} + u_{1,3,13} \leqslant 1$. The case of $\alpha = 1$, $\beta = 3$, and $k = 2$ in constraint (6) gives $x_{1,2} + x_{3,2} + u_{1,3,2} \leqslant 1$. Constraint (7) are composed by $x_{1,9} + x_{1,10} \leqslant 1$, $x_{3,9} + x_{3,10} \leqslant 1$, and $x_{6,9} + x_{6,10} \leqslant 1$. There is no constraint (8) for this example since all the genotypes in G_2 are compatible to each other.

For the example illustrated in Fig. 3, its optimal HIPP objective value is 7. Our upper bound in Section 4.1 gives $m + w - \sigma = 7 + 2 - 1 = 8$, since the compatible graph can be covered by two cliques: 1–3–5–6–7 and 2–4. The optimal HIPP_{LB} objective value for this example is 5, which selects h_1, h_3, h_7, h_9, and h_{13} to satisfy the compatible relations. On the other hand, Lynce and Marques-Silva [13,16] suggested $2\kappa - \sigma$ as a lower bound, where κ denotes the size of a maximum clique in the complement of $CG(G)$ and σ is the number of genotypes in that clique which do not have heterozygous sites. For this specific example, $\sigma = 0$ and $\kappa = 2$ since the maximum clique in the complement of $CG(G)$ corresponds to an arc and we can find some arc $(i,4)$ where $g_i \notin G_0$, thus $2\kappa - \sigma = 4$, which is worse than our lower bound. More computational experiments will be conducted in next section, and the results show our lower bounds are consistently better than theirs.

HIPP_{LB} contains $O(m^3)$ binary variables and constraints, which is polynomial-sized. It may be possible to derive those inferred haplotype vectors for the optimal haplotype objects based on the solution of HIPP_{LB}. If such a feasible inferred haplotype vectors can be calculated, one would obtain an optimal solution to the HIPP problem. Otherwise, the optimal solution of HIPP_{LB} can be used as a new cut (i.e. constraint) to further improve the lower bound.

4.3. Preliminary computational experiments on simulated data

We use the program by Hudson [17] to simulate a $2m \times n$ haplotype matrix, and then randomly pair two haplotypes from these $2m$ haplotypes to produce an $m \times n$ genotype matrix in a way that none of the $2m$ haplotypes is repeatedly paired. Similar simulation settings can also be found in [4,6,7,10].

All the computational experiments are conducted on a Personal Computer with Intel Core2 1.83 GHz CPU, 2 GB RAM and Windows XP operating system. Three heuristics are implemented and evaluated: (1) our upper bound heuristic proposed in Section 4.1 that iteratively calculates for maximal cliques in the compatible graph, denoted as "UB_{MQ}", is implemented in C++, compiled by Visual C++; (2) our lower bound heuristic proposed in Section 4.2 that solves for a polynomial-sized ILP HIPP_{LB}, denoted by "LB_{ILP}", is implemented in C++, compiled by Visual C++, and linked with CPLEX 9.0 callable library; and (3) the lower bound heuristic suggested in [13,16], denoted by "LB_{LM}". We implement LB_{LM} using the maximum clique algorithm by Konc and Janečič [18]. Moreover, we compare UB_{MQ}, LB_{ILP}, and LB_{LM} with "RPoly", the pseudo-Boolean optimization model by Graca et al. [14,15] that calculates the optimal HIPP solutions.

Table 1–4 lists the results of UB_{MQ}, LB_{ILP}, LB_{LM} and RPoly on solving four problem sets of different genotype matrix sizes (10×10, 20×10, 20×20, and 30×10), where 5 random test cases for each problem set have been generated.

The results of our computational experiments indicate our proposed upper bounds and lower bounds do serve their purposes, although some of them seem not so tight in our tests. Nevertheless, LB_{ILP} performs consistently better than LB_{LM} in all the test cases. The upper bound may be further improved by better techniques to solve the maximum clique problem. The lower bound may also be improvable by introducing new constraints such as relations of genotypes to haplotype objects among more than two genotypes, or relations between the haplotype objects of compatible genotypes depending on individual site. On the other hand, although not listed here, all of our proposed algorithms can calculate the results very efficiently, compared with other time-consuming ILP techniques such as the RTIP by Gusfield [4] and PolyIP by Brown and Harrower [6].

Table 1
Computational results of UB_{MQ}, LB_{ILP}, LB_{LM}, and RPoly on 5 random 10×10 cases.

Problem set	10×10			
Algorithms	UB_{MQ}	LB_{ILP}	LB_{LM}	RPoly
Case 1	6	4	1	6
Case 2	4	2	1	4
Case 3	7	4	2	6
Case 4	8	4	3	7
Case 5	10	5	4	8

Table 2
Computational results of UB_{MQ}, LB_{ILP}, LB_{LM}, and RPoly on 5 random 20×10 cases.

Problem set	20×10			
Algorithms	UB_{MQ}	LB_{ILP}	LB_{LM}	RPoly
Case 1	5	3	1	4
Case 2	8	4	1	6
Case 3	6	2	1	6
Case 4	7	3	1	6
Case 5	12	5	2	9

Table 3
Computational results of UB_{MQ}, LB_{ILP}, LB_{LM}, and RPoly on 5 random 20×20 cases.

Problem set	20×20			
Algorithms	UB_{MQ}	LB_{ILP}	LB_{LM}	RPoly
Case 1	11	3	1	10
Case 2	9	4	2	6
Case 3	16	5	3	12
Case 4	15	5	3	10
Case 5	13	3	2	9

Table 4
Computational results of UB_{MQ}, LB_{ILP}, LB_{LM}, and RPoly on 5 random 30×10 cases.

Problem set	30×10			
Algorithms	UB_{MQ}	LB_{ILP}	LB_{LM}	RPoly
Case 1	8	3	3	5
Case 2	8	4	3	4
Case 3	12	4	4	7
Case 4	10	3	2	6
Case 5	11	4	1	6

5. Conclusions

This paper analyzes the mathematical properties for the HIPP problem. In particular, we show several properties that can be used to reduce the size of the original HIPP problem. Some properties also suggest efficient solution methods to divide the original problem into several disjoint subproblems of smaller size. Based on the compatible relations between genotypes, we suggest a heuristic

that iteratively solves the maximal clique problems for computing a good feasible solution that provides a good upper bound. A polynomial-sized ILP has also been proposed to seek the fewest haplotype objects that resolve all the genotypes, based on the compatible relations. Our computational experiments indicate our proposed lower bound technique performs consistently better than the one by Lynce and Marques-Silva [13] and Hudson [17], and thus can be used to speed up some ILP-based HIPP solution methods such as RTIP or PolyIP.

For future research directions, we suggest to investigate tighter upper and lower bounds, whether based on the ILP or Boolean variables.

Acknowledgements

I-Lin Wang was partly supported by the National Science Council of Taiwan under Grant NSC97-2221-E-006-173.

References

[1] L. Helmuth, Map of the human genome 3.0., Science 293 (2001) 583.
[2] Y. Futatsugawa, T. Kubota, A. Ishiguro, H. Suzuki, H. Ishikawa, T. Iga, PCR-based haplotype determination to distinguish CYP2B6∗1/∗7 and ∗5/∗6, Clin. Chem. 50 (8) (2004) 1472.
[3] C. Drysdale, D. McGraw, C. Stack, J. Stephens, R. Judson, K. Nandabalan, K. Arnold, G. Ruano, S. Liggett, Complex promoter and coding region β_2-adrenergic receptor haplotypes alter receptor expression and predict in vivo responsiveness, Proc. Natl. Acad. Sci. USA. 97 (2000) 10483.
[4] D. Gusfield, Haplotype inference by pure parsimony, in: Combinatorial Pattern Matching: 14th Annual Symposium, 2003, pp. 144–155.
[5] G. Lancia, C.M. Pinotti, R. Rizzi, Haplotyping populations by pure parsimony: complexity, exact and approximation algorithms, INFORMS J. Comput. 16 (2004) 348.
[6] D.G. Brown, I.M. Harrower, Integer programming approaches to haplotype Inference by pure parsimony, IEEE ACM Trans. Comput. Biol. 3 (2006) 141.
[7] Y.T. Huang, K.M. Chao, T. Chen, An approximation algorithm for haplotype inference by maximum parsimony, J. Comput. Biol. 12 (2005) 1261.
[8] K. Kalpakis, P. Namjoshi, Haplotype phasing using semidefinite programming, in: Proceedings of the Fifth IEEE Symposium on Bioinformatics and Bioengineering, 2005, pp. 145–152.
[9] Z. Li, W. Zhou, X. Zhang, L. Chen, A parsimonious tree-grow method for haplotype inference, Bioinformatics 21 (2005) 3475.
[10] L. Wang, Y. Xu, Haplotype inference by maximum parsimony, Bioinformatics 19 (2003) 1773.
[11] H. Yang, On solving population haplotype inference problems, Master thesis, Institute of Information Management, National Cheng Kung University, Tainan, Taiwan, 2003.
[12] I. Lynce, J. Marques-Silva, SAT in Bioinformatics: making the case with haplotype inference, in: International Conference on Theory and Applications of Satisfiability Testing, Seattle, USA, August 2006.
[13] I. Lynce, J. Marques-Silva, Haplotype inference with Boolean satisfiability, Int. J. Artif. Intell. Tools 17 (2) (2008) 355.
[14] A. Graca, J. Marques-Silva, I. Lynce, A. Oliveira, Efficient haplotype inference with pseudo-Boolean optimization, in: Algebraic Biology, Hagenberg, Austria, July 2007.
[15] A. Graca, J. Marques-Silva, I. Lynce, A. Oliveira, Efficient haplotype inference with combined CP and OR techniques, in: International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Paris, France, May 2008.
[16] I. Lynce, J. Marques-Silva, Efficient haplotype inference with Boolean satisfiability, in: National Conference on Artificial Intelligence (AAAI), Boston, USA, July 2006.
[17] R.R. Hudson, Generating samples under a Wright–Fisher neutral model of genetic variation, Bioinformatics 18 (2002) 337.
[18] J. Konc, D. Janečič, An improved branch and bound algorithm for the maximum clique problem, MATCH Commun. Math. Comput. Chem. 58 (2007) 569.