*INFORMS Journal on Computing*
Vol. 22, No. 2, Spring 2010, pp. 195–209
ISSN 1091-9856 | EISSN 1526-5528 | 10 | 2202 | 0195

DOI 10.1287/ijoc.1090.0333




# A Class Representative Model for Pure Parsimony Haplotyping

Daniele Catanzaro
Graphs and Mathematical Optimization, Computer Science Department, Université Libre de Bruxelles, B-1050 Brussels, Belgium, dacatanz@ulb.ac.be

Alessandra Godi
Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, 00185 Roma Italy, alessandragodi@inwind.it

Martine Labbé
Graphs and Mathematical Optimization, Computer Science Department, Université Libre de Bruxelles, B-1050 Brussels, Belgium, mlabbe@ulb.ac.be

Haplotyping estimation from aligned single nucleotide polymorphism fragments has attracted increasing attention in recent years because of its importance in the analysis of fine-scale genetic data. Its application fields range from mapping of complex disease genes to inferring population histories, passing through designing drugs, functional genomics, and pharmacogenetics. The literature proposes several criteria for haplotyping populations, each of them characterized by biological motivations. One of the most important haplotyping criteria is parsimony, which consists of finding the minimum number of haplotypes necessary to explain a given set of genotypes. Parsimonious haplotype estimation is an $\mathcal{NP}$-hard problem for which the literature has proposed several integer programming (IP) models. Here, we describe a new polynomial-sized IP model based on the concept of class representatives, already used for the coloring problem. We propose valid inequalities to strengthen our model and show, through computational experiments, that our model outperforms the best IP models currently known in literature.

*Key words*: haplotype inference; computational biology; integer programming
*History*: Accepted by Harvey Greenberg, former Area Editor for Computational Biology and Medical Applications; received October 2008; revised January 2009, February 2009, March 2009; accepted March 2009. Published online in *Articles in Advance* July 29, 2009.

## 1. Introduction

Diploid organisms, such as humans, are characterized by having the DNA organized in pairs of chromosomes, of which one copy is inherited from the father and the other from the mother. The recent completion of the sequencing phase of the Human Genome Project (Venter et al. 2001) showed that such copies are extremely similar and that the genomes of two different individuals are identical in more than 99% of the overall number of nucleotides. Nevertheless, differences at the genomic level (also known as *polymorphisms*) occur, on average, every 1,000 bases (Chakravarti 1998) and are (excluding the recombination process) the predominant form of human variation as well as of genetic diseases (Hoehe et al. 2000, Terwilliger and Weiss 1998).

When a site (i.e., the position of a specific nucleotide) of the genome shows a statistically significant variability within a population (i.e., a set of individuals) it is then called a *single nucleotide polymorphism* (SNP). Specifically, a site is considered a SNP if for a minority of the population a certain nucleotide is observed (called the least frequent allele) while for the rest of the population a different nucleotide is observed (the most frequent allele). For a given SNP, an individual can be either homozygous (i.e., possess the same allele on both chromosomes) or heterozygous (i.e., possess two different alleles). The values of a set of SNPs on a particular chromosome region define a *haplotype*. Haplotyping an individual therefore consists of determining, for each copy of a given chromosome region, a pair of haplotypes.

Haplotyping populations of individuals has attracted an increasing amount of attention in recent years (Helmuth 2001, Marshall 1999) because of its importance in the analysis of fine-scale genetic data (Clark et al. 1998, Schwartz et al. 2002). For example, haplotypes are necessary in evolutionary studies to extract the information needed to detect diseases and to reduce the number of tests to be carried out. In

functional genomics, haplotypes are used to discover a functional gene or to study an altered response of an organism to a particular therapy. In human pharmacogenetics, haplotypes explain why people react differently to different types or amounts of drugs. In fact, because SNPs affect the structure and function of proteins and enzymes, they may influence the way in which a drug is absorbed and metabolized.

Direct sequencing of haplotypes via experimental methods is both time-consuming and expensive, and therefore current molecular sequencing methods generally provide more general genotype information. Specifically, genotype data provide information about the multiplicity of each SNP allele of a given individual, i.e., knowledge about its homozygous or heterozygous nature. Unfortunately, a drawback of using genotype data is that information regarding which heterozygous site SNP variants came from the same chromosome copy remains unknown (Wang and Xu 2003). Hence, in silico haplotyping methods become attractive alternatives and in some cases the only viable way for haplotyping populations (Lancia et al. 2004).

The simplest way for haplotyping a population is described in Bonizzoni et al. (2003), Gusfield (2003), and Lancia et al. (2004) and can be resumed as follows: first, to experimentally obtain genotype data, and subsequently, for each individual, to retrieve the haplotypes computationally—i.e., to find a set of haplotypes such that, if they are assumed to be the corresponding set of chromosome copies, then computing the multiplicity of each SNP allele one can obtain exactly the genotypes given. However, this approach requires the presence of some haplotyping criterion.

Several criteria have been proposed for haplotyping populations, each of them based on biological motivations (see, for example, Bafna et al. 2003, Clark et al. 1998, Eskin et al. 2003, Excoffier and Slatkin 1995, Fallin and Schork 2000, Lancia and Rizzi 2006, Niu et al. 2002, Qin et al. 2002, Stephens and Donnelly 2003, Stephens et al. 2001). In this article we consider the *parsimony* criterion (Lancia et al. 2004). The idea at the core of the parsimony criterion is that under many plausible explanations of an observed phenomenon, the one requiring the fewest assumptions should be preferred (Semple and Steel 2003). Hence, because the number of distinct haplotypes observed in a population is much smaller than the number of possible haplotypes, the parsimony-based approaches aim to determine the minimum number of different haplotypes that, combined in pairs over time, have given rise to a set of observed genotypes (Lancia et al. 2004).

The problem of haplotyping populations under parsimony (hereafter denoted as the pure parsimony haplotyping (PPH) problem) is known to be $\mathcal{APX}$-hard (Lancia et al. 2004). This result justifies the development of several enumerative optimization algorithms that aim to solve exactly instances of PPH. Specifically, Gusfield (2003) first proposed an integer-programming model to tackle instances of PPH. The author described a model, exponential in size, characterized by two kinds of variables—one for haplotypes and the other for haplotype pairs—and by the exhaustive generation of the set of all haplotypes compatible with some genotype in the input. Similar integer programming models were also used by Brown and Harrower (2004) and Bertolazzi et al. (2008). To minimize the number of distinct haplotypes, Brown and Harrower (2004) proposed constructing haplotype vectors by associating a variable to each site; they subsequently used constraints to establish the exact haplotype structures. On the other hand, Bertolazzi et al. (2008) first formulated PPH as a minimization problem characterized by a polynomial number of variables and constraints. Then the authors turned the problem into a maximization problem and strengthened the model by using clique inequalities, symmetry breaking, inequalities, and dominance relations. Whereas Gusfield (2003) and Bertolazzi et al. (2008) used commercial mixed-integer programing solvers (CPLEX and Xpress-MP, respectively) to get solutions to their models, Brown and Harrower (2004) used a branch-and-cut algorithm to solve their polynomial model. A comparison of their results shows that the Brown and Harrower (2004) polynomial model is well suited for big dimension samples, whereas the Gusfield (2003) and Bertolazzi et al. (2008) models are more efficient for medium dimensions, and specifically, when the recombination level (i.e., the parameter that affects the structure of the haplotypes) increases. So far, to our best knowledge, data sets containing 68 genotypes and 75 SNPs represent the limit size instances of PPH that can be exactly analyzed (Brown and Harrower 2006).

In this article we present a new polynomial-sized integer programming (IP) model for PPH. (See the Online Supplement, available at http://joc.pubs.informs.org/ecompanion.html/, for codes and data.) The idea at the core of the model is based on class representatives already proposed by Campelo et al. (2008) for the coloring problem (Garey and Johnson 2003). Computational experiments show that our model outperforms the existing polynomial IP models for PPH (Bonizzoni et al. 2003, Brown and Harrower 2006, Gusfield 2003, Lancia et al. 2004) both on real and simulated data, and allows haplotyping of populations containing hundreds of genotypes and SNPs. The model is compact, easy to implement, solvable with standard solvers, and usable in those cases for which the parsimony criterion is well suited for haplotyping populations.

## 2. Notation and Problem Formulation

Assume that a population of diploid individuals is given and that for each individual the knowledge of

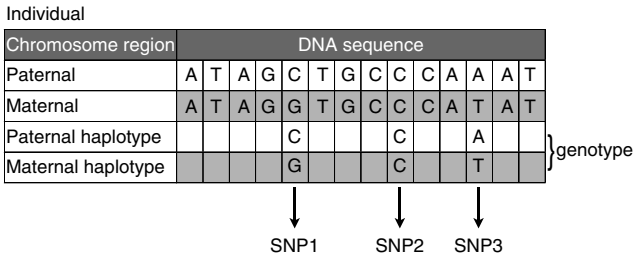


**Figure 1 An Example of Haplotypes and SNPs**

a specific (paternal and maternal) chromosome region is available. The sites of the chromosome regions from the individuals at which polymorphism arises are called SNP. Given a SNP, the most frequent allele in the population is encoded with the value "0" and the least frequent allele with the value "1."

For an individual in the population, a SNP is homozygous if the corresponding paternal and maternal nucleotides are equal, and heterozygous otherwise. The set of SNPs of a paternal (or maternal) chromosome region of an individual defines a *haplotype*.

Haplotypes can be encoded as strings of fixed length over an alphabet $\Sigma = \{0, 1\}$. For example, consider the paternal and maternal chromosome regions from an individual as shown in Figure 1. Encode the first SNPs (i.e., "C" and "G") as 0 and 1, respectively; the second SNPs (i.e., "C" and "C") as 1; and the third SNPs (i.e., "A" and "T") as 0 and 1, respectively. Then the paternal haplotype is the string (010) and the maternal haplotype is the string (111).

Because at each SNP only three possibilities can arise (homozygous of type 0 or 1, or heterozygous), a genotype can be encoded as strings of fixed length over an alphabet $\Sigma = \{0, 1, 2\}$, where entries equal to 0 (or 1) denote homozygous sites of type 0 (or 1), and entries equal to 2 denote heterozygous sites.

Given a pair of haplotypes $h_i$ and $h_j$, define the operator sum $\oplus$ among $h_i$ and $h_j$ as the genotype $g_k$ whose $p$th entry is $h_{ip}$ if $h_{ip} = h_{jp}$, and 2 otherwise. For example, the genotype obtained by summing the paternal and maternal haplotypes in Figure 1 is $g_k = (212)$. We say that a genotype $g_k$ is resolved from the pair of haplotypes $h_i$ and $h_j$ if $g_k = h_i \oplus h_j$.

An instance of PPH is specified by indicating a set $\mathcal{G}$ of $m$ genotypes. A feasible solution to PPH is constituted by (i) a set $\mathcal{H}$ of (at most $2m$ (Lancia et al. 2004)) haplotypes and (ii) for each genotype $g_k \in \mathcal{G}$, a pair of haplotypes, say, $\{h_i, h_j\}$, resolving $g_k$ such that $g_k = h_i \oplus h_j$. An optimal (or minimum) solution to PPH is a feasible solution in which the set $\mathcal{H}$ has the minimum cardinality (Blain et al. 2009). As an example, an instance of PPH and two alternative (minimum) solutions are shown in Figure 2. It is worth nothing that, although having the same set of haplotypes, Solutions 1 and 2 of Figure 2 are different.

Bipartite graphs can be used to represent feasible solutions to PPH (see Blain et al. 2009). Specifically, a feasible solution can be represented by such a graph $(\mathcal{H}, \mathcal{G}, \mathcal{E})$ in which each vertex $g_k \in \mathcal{G}$ is of degree 2 and the two other vertices, say, $h_i$ and $h_j$, adjacent to $g_k$ satisfy $g_k = h_i \oplus h_j$ (i.e., $h_i$ and $h_j$ are *mates* for $g_k$; see Blain et al. 2009). The bipartite graphs corresponding to Solutions 1 and 2 of Figure 2 are depicted in Figures 3(a) and 3(b), respectively. In the bipartite graph representation of a solution, each haplotype is incident to a subset of genotypes. The family of the subsets of genotypes induced by the haplotypes of a feasible solution satisfies the following three properties: (1) each subset of genotypes shares one haplotype, (2) each genotype belongs to exactly two subsets, and (3) every pair of subsets intersects in at most one genotype. As an example, the haplotypes of Solution 1 in Figure 2 induce the family of subsets of Figure 4(a) satisfying properties (1)–(3). Specifically, the subsets are induced by the following four haplotypes: $h_1 = \{1100\}$ inducing the subset $S_1 = \{2222, 1202\}$, $h_2 = \{0011\}$ inducing the subset $S_2 = \{2222, 0212\}$, $h_3 = \{1001\}$ inducing the subset $S_3 = \{1202\}$, and finally, $h_4 = \{0110\}$ inducing the subset $S_4 = \{0212\}$. Similarly, the haplotypes of Solution 2 in Figure 2 induce the family of subsets of Figure 4(b) satisfying properties (1)–(3). Specifically, the subsets are induced by the following four haplotypes: $h_1 = \{1100\}$ inducing the subset $S_1 = \{1202\}$, $h_2 = \{0011\}$ inducing the subset $S_2 = \{0212\}$, $h_3 = \{1001\}$ inducing the subset $S_3 = \{2222, 1202\}$, and finally, $h_4 = \{0110\}$ inducing the subset $S_4 = \{2222, 0212\}$.

## 3. Integer Programming Models

In this section we describe an IP model for PPH. First, we describe a basic model that requires a polynomial number of variables and constraints. We subsequently show how to reduce the model size by exploiting the particular structure and properties of the variables involved. Finally, we provide valid inequalities to further strengthen the model.

### 3.1. Basic Model

The integer programming model for PPH that we propose is based on the class representatives with the smallest index, similar to the one proposed by Campelo et al. (2008) to tackle the coloring problem (Garey and Johnson 2003). Following the properties of a feasible solution presented in the previous section, we associate an index to each subset $S$ of genotypes induced by a haplotype $h$. Specifically, if $i$ is the smallest index of a genotype belonging to $S$, then $i$ is the index associated to $S$ and the subset will be denoted as $S_i$. Because each genotype $g_k$ belongs to exactly two subsets, it may happen that $g_k$ is itself the genotype with the smallest index in

Instance of PPH

| Genotypes | SNP | | | |
|---|---|---|---|---|
| Genotype 1 | 2 | 2 | 2 | 2 |
| Genotype 2 | 1 | 2 | 0 | 2 |
| Genotype 3 | 0 | 2 | 1 | 2 |

Minimum solution 1

| Haplotypes | SNP | | | |
|---|---|---|---|---|
| Haplotype 1 | 1 | 1 | 0 | 0 |
| Haplotype 2 | 0 | 0 | 1 | 1 |
| Haplotype 3 | 1 | 0 | 0 | 1 |
| Haplotype 4 | 0 | 1 | 1 | 0 |

| Conflation | | | | |
|---|---|---|---|---|
| Genotype 1 | = | Haplotype 1 | ⊕ | Haplotype 2 |
| Genotype 2 | = | Haplotype 1 | ⊕ | Haplotype 3 |
| Genotype 3 | = | Haplotype 2 | ⊕ | Haplotype 4 |

Minimum solution 2

| Haplotypes | SNP | | | |
|---|---|---|---|---|
| Haplotype 1 | 1 | 1 | 0 | 0 |
| Haplotype 2 | 0 | 0 | 1 | 1 |
| Haplotype 3 | 1 | 0 | 0 | 1 |
| Haplotype 4 | 0 | 1 | 1 | 0 |

| Conflation | | | | |
|---|---|---|---|---|
| Genotype 1 | = | Haplotype 3 | ⊕ | Haplotype 4 |
| Genotype 2 | = | Haplotype 1 | ⊕ | Haplotype 3 |
| Genotype 3 | = | Haplotype 2 | ⊕ | Haplotype 4 |

**Figure 2 Graphical Representation of an Instance of PPH and Two Alternative Solutions**

both subsets. In this case a dummy genotype $g_{k'}$ is added, and the subset $S_{k'}$ is created. As an example, one can imagine that the haplotype $h_1$ induces the subset $S_i = \{g_i, g_j, g_k, \ldots\}$, $h_2$ induces the subset $S_{i'} = \{g_i, g_l, g_r, g_s, \ldots\}$, $h_3$ induces the subset $S_k = \{g_k, g_l, g_s, g_t, \ldots\}$, and so on. We remark that the index $k'$ can be considered only if $k$ was previously used, i.e., if the subset $S_k$ already exists. For the sake of notation, in the rest of the paper we will refer to the genotype $g_k$ as "the genotype $k$."

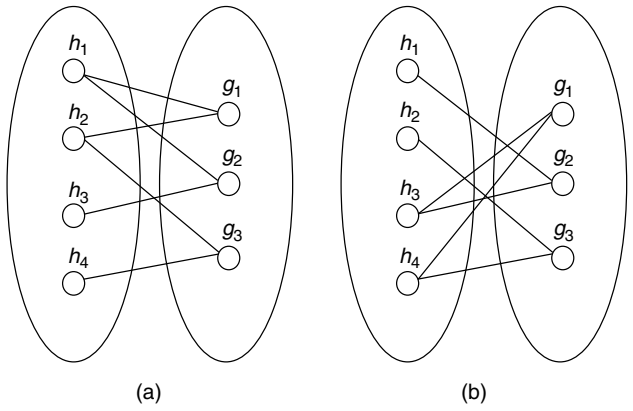


**Figure 3 Bipartite Graph Representation of Solutions 1 and 2 of Figure 2**

Because at most $2m$ haplotypes are necessary to resolve $m$ genotypes (Lancia et al. 2004), the indices $i$ of the subsets $S_i$ can vary inside $K \cup K'$, where $K = \{1, \ldots, m\}$ and $K' = \{1', \ldots, m'\}$. Assume that an order is defined on $K \cup K'$ in such a way that $1 < 1' < 2 < 2' < \cdots < m < m'$. Define $x_i$, $\forall\, i \in K \cup K'$, as a decision variable equal to 1 if, in the solution, there exists a haplotype inducing a subset $S_i$ of genotypes whose smallest index genotype is $i$, and 0 otherwise. Denote $y_{ij}^k$, $\forall\, k \in K$, $\forall\, i, j \in K \cup K'$, $i < j$, as a decision variable equal to 1 if the genotype $k$ belongs to the subsets $S_i$ and $S_j$, and 0 otherwise. Finally, denote $\mathcal{P}$ as the set of the $p$ SNPs characterizing each genotype $g_k$ in $\mathcal{G}$, and let $z_{ip}$, $\forall\, i \in K \cup K'$, $p \in \mathcal{P}$, be a decision variable equal to 1 if the haplotype inducing the subset $S_i$ of genotypes has such a value at the $p$th site, and 0 otherwise. Variables $z_{ip}$ describe explicitly the haplotypes of the solution. Then, an integer programming model for PPH follows:

**Formulation 1.** *Basic Model* (*BM*)

$$\min \sum_{i \in K \cup K'} x_i \tag{1}$$

$$\text{s.t.} \quad x_{i'} \le x_i \quad \forall\, i \in K, \tag{2}$$

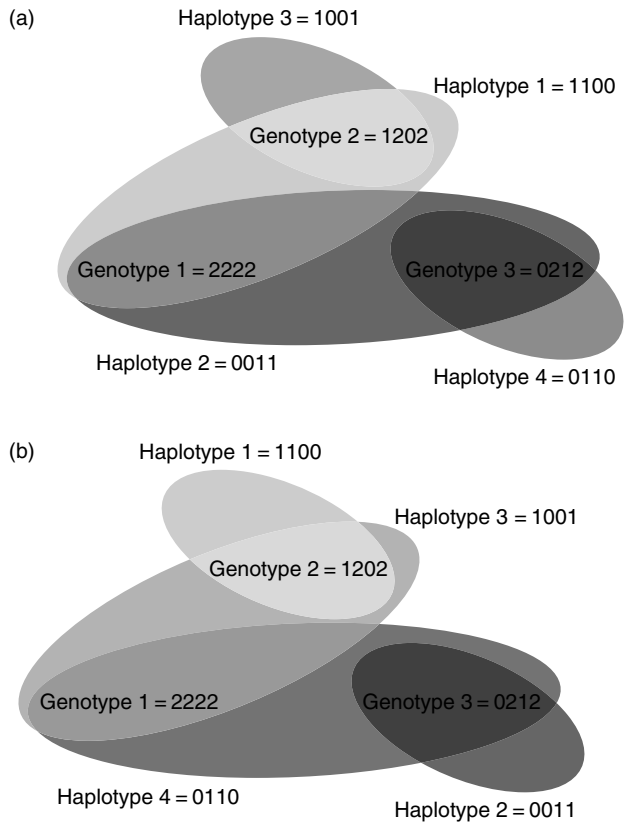


**Figure 4 Examples of Subsets of Genotypes Induced by the Two Alternative Solutions of Figure 2**

$$\sum_{i,\, j \in K \cup K'} y_{ij}^k \geq 1 \quad \forall k \in K, \tag{3}$$

$$\sum_{j \in K \cup K' : j \geq i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \leq x_i \quad \forall k \in K, \forall i \in K \cup K', \tag{4}$$

$$y_{kk'}^k \leq x_{k'} \quad \forall k \in K, \tag{5}$$

$$z_{kp} = z_{k'p} = 0 \quad \forall k \in K, \forall p \in \mathcal{P}\colon g_{kp} = 0, \tag{6}$$

$$z_{kp} = z_{k'p} = 1 \quad \forall k \in K, \forall p \in \mathcal{P}\colon g_{kp} = 1, \tag{7}$$

$$z_{kp} + z_{k'p} = 1 \quad \forall k \in K, \forall p \in \mathcal{P}\colon g_{kp} = 2, \tag{8}$$

$$z_{ip} \leq 1 - \sum_{j \in K \cup K' : j \geq i} y_{ij}^k - \sum_{j \in K \cup K' : j < i} y_{ji}^k \quad \forall p \in \mathcal{P}, \forall k \in K\colon g_{kp} = 0, \forall i \in K \cup K', i \neq k, k', \tag{9}$$

$$z_{ip} \geq \sum_{j \in K \cup K' : j \geq i} y_{ij}^k + \sum_{j \in K \cup K' : j < i} y_{ji}^k \quad \forall p \in \mathcal{P}, \forall k \in K\colon g_{kp} = 1, \forall i \in K \cup K', i \neq k, k'', \tag{10}$$

$$z_{ip} + z_{jp} \geq y_{ij}^k \quad \forall p \in \mathcal{P}, \forall k \in K\colon g_{kp} = 2, \forall i, j \in K \cup K', \tag{11}$$

$$z_{ip} + z_{jp} \leq 2 - y_{ij}^k \quad \forall p \in \mathcal{P}, \forall k \in K\colon g_{kp} = 2, \forall i, j \in K \cup K', \tag{12}$$

$$x_i, z_{ip}, y_{ij}^k \in \{0, 1\}. \tag{13}$$

The objective function (1) represents the number of distinct haplotypes or equivalently the cardinality of $\mathcal{H}$. Because the index $i'$ is considered only if $i$ is already used, constraints (2) imply that if the haplotype $h_i$ is not used, then $h_{i'}$ should not be used. Constraints (3) impose that each genotype $g_k$ must belong to exactly two subsets $S_i$, $S_j$, and constraints (4) force $x_i$ to be 1, i.e., to take haplotype $h_i$ into account, if some genotype $g_k$ is resolved by $h_i$. Constraints (5) are a consequence of the definition of the dummy genotype $k'$; actually, they constitute a special version of constraints (4) when genotype $k$ is resolved by haplotype $k'$. Constraints (6)–(8) translate the sum operation among haplotypes. Specifically, constraints (6) impose that the $p$th SNP of the haplotype $h_i$ ($h_{i'}$), which induces the subset $S_i$ ($S_{i'}$), must be set to 0 when genotype $g_i$ has its $p$th SNP equal to 0. By analogy, constraints (7) impose that the $p$th SNP of the haplotype $h_i$ ($h_{i'}$) must be set to 1 when genotype $g_i$ has its $p$th SNP equal to 1. Constraints (8) impose that exactly one among the $p$th SNPs of the haplotypes $h_i$ and $h_{i'}$ can be set to 1 when genotype $g_i$ has its $p$th SNP is equal to 2. Constraints (9) establish the relations between variables $z_{ip}$ and $y_{ij}^k$. Specifically, they force the $p$th SNP of the haplotype $h_i$ to be equal to 0 when at least one genotype $g_k$, whose $p$th entry is equal to 0, belongs to the induced subset $S_i$. By analogy, constraints (10) force the $p$th SNP of the haplotype $h_i$ to be equal to 1 when at least one genotype $g_k$, whose $p$th entry is equal to 1, belongs to the induced subset $S_i$. Finally, constraints (11) and (12) impose that the sum operation among haplotypes is respected for any pair of haplotypes $h_i$ and $h_j$ in the solution.

### 3.2. Reducing Model Size

It is worth noting that the particular nature of the set of indices $K \cup K'$ can be exploited to reduce the size of BM. In fact, given that $y_{ij}^k = 1$ if and only if $k$ belongs to two subsets having $g_i$ and $g_j$ for the smallest index genotype, we need to define variables $y_{ij}^k$ only when $i < j \leq k$ or $i = k$ and $j = k'$. For example, variable $y_{1,1'}^2$ does not need to be defined as well as all variables $y_{ii'}^k$ for all $i, k \in K$, $k \neq i$. Similarly, variables $y_{ik'}^k$ or $y_{k'i}^k$ (depending on whether $k$ is smaller or larger than $i$) do not need to be defined for $i \in K \cup K'$ with $i \neq k$ and $i \neq k'$. In fact, if $y_{ik'}^k = 1$, then $k$ belongs to two subsets, one represented by $i$ and the other by $k'$, which contradicts the assumption that the dummy genotype $k'$ can be considered only if $k$ is already used. By extending this analysis to all the possible cases in which variables $y_{ij}^k$ are redundant and assuming that variable $y_{11'}^1 = 1$, it is easy to see that variables $y_{ij}^k$ do not need to be defined whenever they belong to one of the following sets:

$$R_1 = \{y_{ij}^k\colon k \in K, i, j \in K \cup K', j < i < k\}, \tag{14}$$

$$R_2 = \{y^k_{ik'} \colon k \in K, i \in K \cup K', i \le (k-1)'\}, \quad (15)$$

$$R_3 = \{y^k_{ii'} \colon k \in K, i \in K \cup K', 2 \le i \le k-1\}. \quad (16)$$

The sets of redundant variables can be further expanded by observing that for each triplet of genotypes $\{g_i, g_j, g_k\}$ such that the respective $p$th SNP is $g_{ip} = 0$, $g_{jp} = 0$, and $g_{kp} = 2$, variable $y^k_{ij}$ is necessarily equal to 0 since the containment of the genotype $g_k$ to the subsets $S_i$ and $S_j$ would violate the sum operator among haplotypes at least at the $p$th SNP. Extending this argument to all the possible combinations of triplets of genotypes that violate the haplotype sum operator, we see that the following proposition, whose proof is omitted, holds:

Proposition 1. *The set of variables*

$$R_4 = \{y^k_{ij} \colon i, j, k \in K, p \in \mathcal{P}, g_{kp} = 2, g_{ip} = g_{jp} \ne 2\} \quad (17)$$

*is redundant.*

Similar arguments can be used to prove the following proposition:

Proposition 2. *The set of variables*

$$R_5 = \{y^k_{ij} \colon i, j, k \in K, p \in \mathcal{P}, g_{kp} + g_{ip} = 1 \text{ or } g_{kp} + g_{jp} = 1\} \quad (18)$$

*is redundant.*

Proof. By contradiction, if the set $R_5$ is not redundant, then for some $p \in \mathcal{P}$, genotype $k$ may belong to two subsets $S_i$ and $S_j$. Without loss of generality, assume $g_{kp} = 0$ and $g_{ip} = 1$. Because haplotype $h_i$ is associated with the subset $S_i$ and explains genotype $i$, it must have the $p$th SNP equal to 1; otherwise, the sum operator would be violated. In turn, this implies that genotype $k$ cannot be resolved by $h_i$ because a necessary condition for its resolution is that the $p$th SNP of $h_i$ be equal to 0. Hence, independent of $j$, $y^k_{ij} = 0$ in any feasible solution of PPH. □

Note that removing the redundant variables $y^k_{ij}$ of Propositions 1 and 2 can be performed in $O(m^3 n)$. Finally, a similar process of reduction can be applied to the variables $z_{ip}$ both by removing those whose value is fixed by constraints (6) and (7) and by the set of redundant variables (17) and (18). In this way, only variables $z_{ip}$ involved in constraints (8) need to be defined. In conclusion, denoting $Y$ as the set of variables $y^k_{ij}$, $R = \bigcup_q R_q$, and $\widehat{Y} = Y \backslash R$, we can be reduce BM to the following:

**Formulation 2.** *Reduced Model* (*RM*)

$$\min \sum_{i \in K \cup K'} x_i \quad (19)$$

$$\text{s.t. } x_{i'} \le x_i \quad \forall i \in K, \quad (20)$$

$$\sum_{i, j \colon y^k_{ij} \in Y \backslash R} y^k_{ij} \ge 1 \quad \forall k \in K, \quad (21)$$

$$\sum_{j \colon y^k_{ij} \in Y \backslash R, j \ge i} y^k_{ij} + \sum_{j \colon y^k_{ij} \in \widehat{Y}, j < i} y^k_{ji} \le x_i \quad \forall k \in K, \forall i \colon y^k_{ij} \in \widehat{Y}, \quad (22)$$

$$y^k_{kk'} \le x_{k'} \quad \forall k \in K, \quad (23)$$

$$z_{kp} + z_{k'p} = 1 \quad \forall k \in K, \forall p \in \mathcal{P} \colon g_{kp} = 2, \quad (24)$$

$$z_{ip} \le 1 - \sum_{j \colon y^k_{ij} \in \widehat{Y}, j \ge i} y^k_{ij} - \sum_{j \colon y^k_{ij} \in \widehat{Y}, j < i} y^k_{ji} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i \colon y^k_{ij} \in \widehat{Y}, g_{kp} = 0, g_{ip} = 2, \quad (25)$$

$$z_{ip} \ge \sum_{j \colon y^k_{ij} \in \widehat{Y}, j \ge i} y^k_{ij} + \sum_{j \colon y^k_{ij} \in \widehat{Y}, j < i} y^k_{ji} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i \colon y^k_{ij} \in \widehat{Y}, g_{kp} = 1, g_{ip} = 2, \quad (26)$$

$$z_{ip} \ge y^k_{ij} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i, j \colon y^k_{ij} \in \widehat{Y}, g_{ip} = 2, g_{jp} = 0, g_{kp} = 2, \quad (27)$$

$$z_{jp} \ge y^k_{ij} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i, j \colon y^k_{ij} \in \widehat{Y}, g_{ip} = 0, g_{jp} = 2, g_{kp} = 2, \quad (28)$$

$$z_{ip} \le 1 - y^k_{ij} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i, j \colon y^k_{ij} \in \widehat{Y}, g_{ip} = 2, g_{jp} = 1, g_{kp} = 2, \quad (29)$$

$$z_{jp} \le 1 - y^k_{ij} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i, j \colon y^k_{ij} \in \widehat{Y}, g_{ip} = 1, g_{jp} = 2, g_{kp} = 2, \quad (30)$$

$$z_{ip} + z_{jp} \ge y^k_{ij} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i, j \colon y^k_{ij} \in \widehat{Y}, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2, \quad (31)$$

$$z_{ip} + z_{jp} \le 2 - y^k_{ij} \quad \forall p \in \mathcal{P}, \forall k \in K, \forall i, j \colon y^k_{ij} \in \widehat{Y}, g_{ip} = 2, g_{jp} = 2, g_{kp} = 2, \quad (32)$$

$$x_i, z_{ip}, y^k_{ij} \in \{0, 1\}. \quad (33)$$

Note that the integrality of variables $y^k_{ij}$ suffices to guarantee the integrality of variables $x_i$. Hence, we denote by RMM the formulation in which the integrality condition on variables $x_i$ is relaxed.

### 3.3. Strengthening Inequalities

In this section, we provide valid inequalities to strengthen RM.

Proposition 3. *The inequality*

$$\sum_{k \in K \colon y^k_{ij} \in \widehat{Y}} y^k_{ij} \le x_i \quad \forall i, j \colon \exists y^k_{ij} \in \widehat{Y} \quad (34)$$

*is valid for RM.*

Proof. Two subsets $S_i$ and $S_j$ of a feasible solution can share at most one genotype. So it follows that $\sum_{k\in K} y_{ij}^k \in \{0,1\}$. If the solution is such that $\sum_{k\in K} y_{ij}^k = 0$, then the inequality reduces to $x_i \geq 0$, which is trivially valid. If the solution is such that $\sum_{k\in K} y_{ij}^k = 1$, then genotype $k$ belongs to the subset represented by genotype $i$, which in turn implies $x_i = 1$, and the inequality is again valid. □

Similarly, it is easy to see that the following proposition holds:

Proposition 4. *The inequality*

$$\sum_{k\in K:\, y_{ij}^k \in \hat{Y}} y_{ij}^k \leq x_j \quad \forall\, i,\ j\colon \exists y_{ij}^k \in \hat{Y} \tag{35}$$

*is valid for RM.*

Note that $\sum_{k\in K} y_{ij}^k \in \{0,1\}$ also implies $\sum_{k\in S} y_{ij}^k \in \{0,1\}$, for any $S \subseteq K$. Let $\mathcal{G}_k^1 = \{k \in K\colon g_{kp} = 1\}$. The following proposition holds:

Proposition 5. *The inequality*

$$z_{ip} \geq \sum_{k\in \mathcal{G}_k^1:\, y_{ij}^k \in \hat{Y}} y_{ij}^k \quad \forall\, p \in \mathcal{P},\ \forall\, i,\ j\colon y_{ij}^k \in \hat{Y} \tag{36}$$

*is valid for RM.*

Proof. If the solution is such that $\sum_{k\in \mathcal{G}_k^1} y_{ij}^k = 0$, then the inequality reduces to $z_{ip} \geq 0$, which is trivially valid. If the solution is such that $\sum_{k\in \mathcal{G}_k^1} y_{ij}^k = 1$, then the inequality reduces to $z_{ip} = 1$, which is again valid. □

Note that inequality (36) only applies if $g_{ip} = 2$. In fact, if $g_{ip} = 1$, (36) is redundant, and if $g_{ip} = 0$, the corresponding variables $y_{ij}^k$ are not defined.

By analogy, fixing $i$, $j$, and $p$, consider $\mathcal{G}_k^0 = \{k \in K\colon g_{kp} = 0\}$. Then, the following proposition holds:

Proposition 6. *The inequality*

$$z_{ip} \leq x_i - \sum_{k\in \mathcal{G}_k^0:\, y_{ij}^k \in \hat{Y}} y_{ij}^k \quad \forall\, p \in \mathcal{P},\ \forall\, i,\ j\colon \exists y_{ij}^k \in \hat{Y} \tag{37}$$

*is valid for RM.*

Proof. If the solution is such that $\sum_{k\in \mathcal{G}_k^0} y_{ij}^k = 0$, then the inequality reduces to $z_{ip} \leq x_i$, which is trivially valid. If the solution is such that $\sum_{k\in \mathcal{G}_k^0} y_{ij}^k = 1$, then genotype $k$ belongs to the subset represented by genotype $i$, which implies $x_i = 1$ and thus $z_{ip} = 0$. □

Also in this case, the inequality (37) only applies if $g_{ip} = 2$. In fact, if $g_{ip} = 0$ and if $g_{ip} = 1$, the corresponding variables $y_{ij}^k$ are not defined.

Defining $\mathcal{G}_k^2 = \{k \in K\colon g_{kp} = 2\}$, we have the following proposition:

Proposition 7. *The inequality*

$$z_{ip} + z_{jp} \geq \sum_{k\in \mathcal{G}_k^2:\, y_{ij}^k \in \hat{Y}} y_{ij}^k + 2 \sum_{k\in \mathcal{G}_k^1:\, y_{ij}^k \in \hat{Y}} y_{ij}^k \quad \forall\, p \in \mathcal{P}, \forall\, i, j\colon \exists y_{ij}^k \in \hat{Y} \tag{38}$$

*is valid for RM.*

Proof. The right-hand side of the inequality $\sum_{k\in \mathcal{G}_k^2} y_{ij}^k + 2\sum_{k\in \mathcal{G}_k^1} y_{ij}^k \in \{0,1,2\}$. If the solution is such that $\sum_{k\in \mathcal{G}_k^2} y_{ij}^k + 2\sum_{k\in \mathcal{G}_k^1} y_{ij}^k = 0$, then the inequality reduces to $z_{ip} + z_{jp} \geq 0$, which is trivially valid. If the solution is such that $\sum_{k\in \mathcal{G}_p^2} y_{ij}^k = 1$, then the inequality reduces to $z_{ip} + z_{jp} \geq 1$, which is still valid. Finally, if the solution is such that $\sum_{k\in \mathcal{G}_k^1} y_{ij}^k = 1$, then the inequality reduces to $z_{ip} + z_{jp} = 2$, which is again valid. In fact, if haplotypes $h_i$ and $h_j$ resolve genotype $k$, then they must be characterized by having the $p$th SNP equal to 1. □

Similar arguments can be used to prove the following propositions:

Proposition 8. *The inequality*

$$z_{ip} + z_{jp} \leq x_i + x_j - \sum_{k\in \mathcal{G}_k^2:\, y_{ij}^k \in \hat{Y}} y_{ij}^k - 2 \sum_{k\in \mathcal{G}_k^0:\, y_{ij}^k \in \hat{Y}} y_{ij}^k \quad \forall\, p \in \mathcal{P},\ \forall\, i, j\colon \exists y_{ij}^k \in \hat{Y} \tag{39}$$

*is valid for RM.*

A distinct class of inequalities are obtained by introducing the concept of *conflict* among genotypes (Bertolazzi et al. 2008) at the $p$th SNP. Specifically, consider a pair of two genotypes $(g_i, g_j)$ and assume that there exists a conflict; i.e., one of the two genotypes (e.g., $g_i$) is such that $g_{ip} = 1$, whereas the corresponding entry of the other is such that $g_{jp} = 0$ (or vice versa). Then this condition is sufficient to exclude common haplotypes explaining $g_i, g_j$ because their existence would violate the haplotype sum operator. The rationale can be extended to all the subset of genotypes having conflicts and can be formalized as follows.

Define $CG$ as a graph whose set of vertices is the set of genotypes $\mathcal{G}$ and whose set of edges contains all the pairs of genotypes $(g_i, g_j)$ that are in conflict. Then, the following proposition holds:

Proposition 9. *Let $C$ be a clique in $CG$, then the inequality*

$$\sum_{k\in C} \sum_{j:\, y_{ij}^k \in \hat{Y},\, j\neq i} y_{ij}^k \leq x_i \quad \forall\, i\colon y_{ij}^k \in \hat{Y},\ \forall\, C \subseteq CG \tag{40}$$

*is valid for RM.*

Proof. Only one genotype in $C$ can be shared by two subsets $S_i$ and $S_j$ of a feasible solution. This implies that $\sum_{k\in C} \sum_{j:\, y_{ij}^k \in \hat{Y},\, j\neq i} y_{ij}^k \in \{0,1\}$. If the solution is such that $\sum_{k\in C} \sum_{j:\, y_{ij}^k \in \hat{Y},\, j\neq i} y_{ij}^k = 0$, then the inequality reduces to $x_i \geq 0$, which is valid. If the solution is such that $\sum_{k\in C} \sum_{j:\, y_{ij}^k \in \hat{Y},\, j\neq i} y_{ij}^k = 1$, then the genotype $k$ belongs to the subset represented by genotype $i$, which in turn implies $x_i = 1$; hence, the inequality is again valid. □

Proposition 9 can be further extended to variables $z_{ip}$. Specifically, define the set $E_p^1 = \{(g_i, g_j) \in \mathcal{G}_k^1 : \exists q \in \mathcal{P}, q \neq p, g_{iq} + g_{jq} = 1\}$ for all $p \in \mathcal{P}$, and consider the graph $CG_p^1 = (\mathcal{G}_k^1, E_p^1) \subset CG$. Then the following proposition holds:

Proposition 10. *The inequality*

$$z_{ip} \geq \sum_{k \in C^1} \sum_{j : y_{ij}^k \in \widehat{Y},\, j \geq i} y_{ij}^k + \sum_{k \in C^1} \sum_{j : y_{ij}^k \in \widehat{Y},\, j < i} y_{ji}^k$$

$$\forall p \in \mathcal{P},\ \forall i : y_{ij}^k \in \widehat{Y},\ \forall \textit{ clique } C^1 \subseteq CG_p^1 \quad (41)$$

*is valid for RM.*

By analogy, defining the set $E_p^0 = \{(g_i, g_j) \in \mathcal{G}_k^0 : \exists q \in \mathcal{P}, q \neq p, g_{iq} + g_{jq} = 1\}$, for all $p \in \mathcal{P}$, and considering the graph $CG_p^0 = (\mathcal{G}_k^0, E_p^0) \subset CG$, we have the following proposition:

Proposition 11. *The inequality*

$$z_{ip} \leq x_i - \sum_{k \in C^0} \sum_{j : y_{ij}^k \in \widehat{Y},\, j \geq i} y_{ij}^k - \sum_{k \in C^0} \sum_{j : y_{ij}^k \in \widehat{Y},\, j < i} y_{ji}^k$$

$$\forall p \in \mathcal{P},\ \forall i : y_{ij}^k \in \widehat{Y},\ \forall \textit{ clique } C^0 \subseteq CG_p^0 \quad (42)$$

*is valid for RM.*

The proofs of Propositions 10 and 11 are quite similar to the ones for Propositions 7 and 8 and are omitted. Finally, observe that Propositions 10 and 11 imply Proposition 9 and only apply if $g_{ip} = 2$.

## 4. Experiments

In this section we analyze the performances of our model to solve the pure parsimony haplotyping problem. Our experiments were motivated by a number of goals—namely, to evaluate, with respect to BM, the benefits obtained by removing the redundant variables and including the valid inequalities previously described; to compare the performances of our model with the ones obtained by Brown and Harrower's HybridIP model (Brown and Harrower 2006), currently the best model for PPH; and finally, to allow the analysis of larger data sets with respect to the ones currently analyzed.

Similar to Brown and Harrower (2006), we emphasize that our experiments aim simply to evaluate the runtime performance of our model for solving PPH. We neither attempt to study the efficiency of PPH for haplotype inference nor compare the accuracy of our algorithm to haplotype inference solvers that do not use the parsimony criterion; this analysis has been performed by both Gusfield (2003) and Wang and Xu (2003), and we refer the interested reader to these articles.

### 4.1. Implementation

We implemented BM and RM by means of Mosel 2.0 of Xpress-MP, Optimizer version 18, running on a Pentium 4, 3.2 GHz, equipped with 2 GB of RAM, and operating system Gentoo release 7 (kernel linux 2.6.17). We have combined RM with different constraints obtaining two other models—specifically, RM including the valid inequalities (hereafter indicated with SM) and RMM including the valid inequalities and such that variables $x_i$ are declared continuous (hereafter indicated with SMM). To evaluate the benefits of the valid inequalities, we have deactivated the Xpress Optimizer automatic cuts and the pre-solving strategy. Finally, we have used the Xpress-MP primal heuristic to generate the first upper bound.

### 4.2. Separation Oracle for the Valid Inequalities

When using models SM and SMM, the valid inequalities (34) and (35) are loaded together with RM. On the contrary, the valid inequalities (36)–(39) are dynamically generated by means of a separation oracle. Specifically, let $(\bar{x}, \bar{y}, \bar{z})$ be a current solution at a given node of the search tree. For each $p \in \mathcal{P}$, we build the set $\mathcal{G}_k^1$ and for all $i, j$ such that $\exists y_{ij}^k \in \widehat{Y}$, and we check if $\bar{z}_{ip} < \sum_{k \in \mathcal{G}_k^1 : y_{ij}^k \in \widehat{Y}} \bar{y}_{ij}^k$. If so, the inequality

$$z_{ip} \geq \sum_{k \in \mathcal{G}_k^1 : y_{ij}^k \in \widehat{Y}} y_{ij}^k$$

is added. To reduce the computational overhead, no more than 10 cuts are added at each node of the search tree. A similar procedure is used to separate inequalities (37)–(39). If no inequality of types (36)–(39) is added to the formulation, then the separation oracle checks for violated clique inequalities (41)–(42). Before running the exact search, we precompute the graph $CG$ for the instance analyzed. At a given node of the search tree, we subsequently pick at random an index $p \in \mathcal{P}$ and an index $i$: $y_{ij}^k \in \widehat{Y}$, and we dynamically construct the graph $CG_p^1$ and set $\overline{w}_k := \sum_{j : y_{ij}^k \in \widehat{Y},\, j \geq i} \bar{y}_{ij}^k + \sum_{j : y_{ij}^k \in \widehat{Y},\, j < i} \bar{y}_{ji}^k$, $k \in CG_p^1$. If $\bar{z}_{ip} \geq \sum_{k \in \mathcal{G}_p^1} \overline{w}_k$, then there does not exist any violated inequality in $CG_p^1$, and we select at random a different index $i$; otherwise, we proceed by finding a maximum clique in $CG_p^1$. After testing alternative (heuristic and exact) solution strategies for finding a maximum clique in a graph, a well-known $\mathcal{NP}$-hard problem (Pardalos and Xue 1994), we opted for Nemhauser and Trotter's exact algorithm (Pardalos and Xue 1994), which proved to be, for the data sets analyzed, a good trade-off between speed (a few milliseconds) and efficiency. Specifically, Nemhauser and Trotter's algorithm consists in solving the model

**Formulation 3.**

$$\max \quad z_o = \sum_{k=1}^{|\mathcal{G}_k^1|} \overline{w}_k t_k \quad (43)$$

$$\text{s.t.} \quad t_i + t_j \leq 1 \quad \forall (i, j) \in \bar{E}_p^1, \quad (44)$$

$$t_i \in \{0, 1\}, \quad (45)$$

where $t_i$ denotes a decision variable equal to 1 if vertex $t_i \in \mathcal{G}_k^1$ belongs to the clique and 0 otherwise, and $\bar{E}_p^1$ is the complement of the edge set $E_p^1$. If the optimal value of the objective function is greater than $\bar{z}_{ip}$, then the inequality

$$z_{ip} \geq \sum_{k \in C^1} \sum_{j: y_{ij}^k \in \hat{Y},\, j \geq i} y_{ij}^k + \sum_{k \in C^1} \sum_{j: y_{ij}^k \in \hat{Y},\, j < i} y_{ji}^k$$

is added to the formulation. A similar procedure is used for separating the clique inequalities (42). To speed up the cut generation, branching priorities on the integer variables were introduced. Specifically, first the branches are performed on variables $z_{ip}$, then on variables $x_i$, and finally on variables $y_{ij}^k$. Moreover, the search is performed previously on variables $z_{ip}$ whose relaxed values are in the interval $(0, 0.5]$ and subsequently on those whose relaxed values are in the interval $(0.5, 1)$.

### 4.3. Basic and Reduced Models on Simulated Data Sets

Analogously to Brown and Harrower (2006), we used Hudson's ms program (Hudson 1990) to generate benchmark simulated data sets to compare the performances of BM and RM. Under the neutral evolution hypothesis (Kimura 1983), ms is able to generate many independent replicate samples (haplotypes) through a variety of assumptions about migration, recombination rate, and population size. A coalescent approach (Hudson 1990) is at the core of the simulator: a random genealogy of the sample is generated, and then mutations are randomly placed on the genealogy. A parameter called recombination level allows one to tune the heterogeneity degree of the haplotypes in a population: as $r$ increases, haplotypes become increasingly different from each other, and hence the number of heterozygous sites in genotype samples tends to become large.

Using a strategy similar to the one already proposed by Brown and Harrower (2006) we generated three collections: 10 data sets containing 80 samples of 50 SNPs each; 10 data sets containing 140 samples of 70 SNPs each; and 10 data sets containing 200 samples of 100 SNPs each, respectively. For each data set, the recombination level was randomly chosen in $\{0, 4, 16\}$. The reason for choosing nonzero recombination levels was due to the fact that positive recombination levels may lead to more difficult PPH instances (Brown and Harrower 2006). To obtain data sets of genotypes, sample data sets were paired randomly so that all of them were used. This process led to the generation of three data sets of 10 instances, each containing 40 genotypes, 70 genotypes, and 100 genotypes, respectively. Duplicate genotypes were removed though a preprocessing step. It is worth noting that this reduction does not interfere with the value of the optimal solution or reduce the gap between the optimal solution of the IP problem and its relaxation. Furthermore, Brown and Harrower's preprocessing step consisting in the elimination of duplicate columns was also implemented (Brown and Harrower 2006).

**Table 1 Overview of the Data Sets Analyzed to Compare the Basic and the Reduced Models**

| Genotypes | SNPs | Recombination level | Mean no. of 2s sites | Max no. of 2s per genotype (mean) | No. of data sets |
|---|---|---|---|---|---|
| 40 | 50 | Random in {0, 4, 16} | 204.60 | 13.80 | 10 |
| 70 | 70 | Random in {0, 4, 16} | 579.90 | 17.31 | 10 |
| 100 | 100 | Random in {0, 4, 16} | 1,047.20 | 22.30 | 10 |

Table 1 summarizes the details about the genotype data sets used, whereas Table 2 shows the results obtained by BM and RM. The second and third columns of Table 2 indicate the mean and the standard deviation of the time (RAM) required by BM and RM to solve a given data set. The fourth column indicates the maximum number of branch-and-bound branches required by BM and RM to solve a given data set. Finally, the fifth column shows the overall number of instances solved by BM and RM on each data set.

In general, numerical experiments show that RM performs better than BM both in terms of runtime and memory. The reduction of variables $y_{ij}^k$ and $z_{ip}$ (and the corresponding constraints) performed by RM approaches 99% of the overall number of variables and constraints required by BM. As a consequence, this strategy leads to a model requiring only a few megabytes of RAM, whereas several hundred of megabytes of RAM are required by the BM model (the main reason the BM model is unable to tackle instances of PPH larger than 40 genotypes).

Because of its better performance, we prefer RM to BM for the remainder of our numerical analysis.

### 4.4. Performance Analysis on Brown and Harrower's Data Sets

We used Brown and Harrower's data sets (Brown and Harrower 2006) for testing the performances of our models versus Brown and Harrower's HybridIP state-of-the-art model for PPH. Specifically, using Hudson's (1990) ms program, the authors created two families of data sets (called the *uniform* and *nonuniform* data sets) by randomly pairing the resulting haplotypes. The distinction in the two simulated methods comes in how the random pairing is performed. In the uniform data sets the haplotypes are randomly paired by sampling uniformly from the set of distinct haplotypes. In the nonuniform data sets the haplotypes are sampled uniformly from the collection of haplotypes generated by the coalescent process. In this collection,

**Table 2 Results Obtained by the Basic and the Reduced Models on the Data Sets Analyzed**

| | Time (s) | | RAM required (MB) | | Max no. of branches | | Fraction solved | |
|---|---|---|---|---|---|---|---|---|
| Data sets | BM | RM | BM | RM | BM | RM | BM | RM |
| $40 \times 50$s | $20.572 \pm 0.77$ | $1.551 \pm 0.078$ | $325 \pm 20$ | $5 \pm 1$ | 1 | 1 | 10/10 | 10/10 |
| $70 \times 70$s | — | $9.995 \pm 0.94$ | $>1{,}058$ | $9 \pm 3$ | — | 1 | 0/10 | 10/10 |
| $100 \times 100$s | — | $39.249 \pm 2.33$ | $>1{,}933$ | $14 \pm 6$ | — | 1 | 0/10 | 10/10 |

haplotypes may not be unique, so some haplotypes are sampled with higher frequency than others. As shown in Table 3, both the uniform and nonuniform data sets consist of collections of 30 or 50 genotypes having 10, 30, 50, 75, or 100 SNPs each. Each data set contains a number of instances varying between 15 and 50.

Brown and Harrower (2006) also considered biological inputs from chromosomes 10 and 21 over all four HapMap (International HapMap Consortium 2004) populations. For each input length, the authors selected sequences of lengths 30, 50, and 75, giving a total of eight data sets consisting of three instances each. Table 4 summarizes the main characteristics of Brown and Harrower's biological data sets.

We show in Tables 5–7 the performances of our models on the uniform, nonuniform, and biological data sets, respectively. Specifically, the columns of each table evidence the mean, the maximum, and the minimum of the solution time, the gap (i.e., the difference between the optimal value found and the value of linear relaxation at the root node of the search tree, divided by the optimal value), the number of branches performed, and the number of cuts added to solve each group of instances belonging to a given data set.

**4.4.1. Uniform Data Sets.** Brown and Harrower (2006) showed that the HybridIP model is able to solve instances of the data sets having a genotype length of 30 or less. Specifically, the authors observed that HybridIP needs a run time ranging from 7 seconds to 2 minutes to solve instances of a data set having genotype length of 10, and from 30 seconds to an hour to solve instances of the data set having genotype length of 30. However, HybridIP is only partially able to solve instances of the data sets having a genotype length of 30 or greater. In fact, HybridIP solved 70% of the instances of the data set having a genotype length of 50, 60% of instances of the data set having a genotype length of 75, and only 30% of the instances of the data set having a genotype length of 100. The run times range from five seconds to two hours.

On the contrary, our numerical experiments (see Table 5) show that RM on average is able to solve instances of the data sets having genotype length of 10 in about 8 seconds, although five instances of the data set $50 \times 10$r16 (specifically, instances 02, 04, 06, 11, and 13) took a bit longer (22.44, 11.152, 30.623, 16.535, and

**Table 3 Characteristics of Brown and Harrower's (2006) Artificial Data Sets**

| Genotypes | SNPs | Max no. of 2s in a genotype | Mean no. of 2s in a genotype | Recombination level |
|---|---|---|---|---|
| Uniform | | | | |
| 50 | 10 | 8 | 3.1 | 0 |
| 50 | 10 | 9 | 3.3 | 4 |
| 50 | 10 | 9 | 2.9 | 16 |
| 50 | 30 | 23 | 8.2 | 0 |
| 30 | 50 | 42 | 14.1 | 0 |
| 30 | 75 | 52 | 19.8 | 0 |
| 30 | 100 | 68 | 25.1 | 0 |
| Nonuniform | | | | |
| 50 | 10 | 7 | 1.7 | n.a. |
| 50 | 30 | 18 | 5.0 | n.a. |
| 50 | 50 | 33 | 8.4 | n.a. |
| 30 | 75 | 41 | 15.8 | n.a. |
| 30 | 100 | 66 | 19.1 | n.a. |

**Table 4 Characteristics of Brown and Harrower's (2006) Biological Data Sets**

| Data set | Genotypes | SNPs | Max no. of 2s in a genotype | Mean no. of 2s in a genotype |
|---|---|---|---|---|
| Biological | | | | |
| Test-chr10-CEU-30 | 35 | 30 | 19 | 0.402778 |
| Test-chr10-CEU-50 | 35 | 50 | 24 | 0.235 |
| Test-chr10-CEU-75 | 35 | 75 | 39 | 0.166667 |
| Test-chr10-HCB-30 | 20 | 30 | 17 | 0.271667 |
| Test-chr10-HCB-50 | 20 | 50 | 25 | 0.11 |
| Test-chr10-HCB-75 | 20 | 75 | 41 | 0.238667 |
| Test-chr10-JPT-30 | 11 | 30 | 13 | 0.251515 |
| Test-chr10-JPT-50 | 11 | 50 | 27 | 0.356364 |
| Test-chr10-JPT-75 | 11 | 75 | 40 | 0.437576 |
| Test-chr10-YRI-30 | 33 | 30 | 20 | 0.243434 |
| Test-chr10-YRI-50 | 33 | 50 | 29 | 0.365455 |
| Test-chr10-YRI-75 | 33 | 75 | 47 | 0.452121 |
| Test-chr21-CEU-30 | 32 | 30 | 19 | 0.427083 |
| Test-chr21-CEU-50 | 32 | 50 | 24 | 0.264375 |
| Test-chr21-CEU-75 | 32 | 75 | 39 | 0.1875 |
| Test-chr21-HCB-30 | 7 | 30 | 17 | 0.447619 |
| Test-chr21-HCB-50 | 7 | 50 | 25 | 0.297143 |
| Test-chr21-HCB-75 | 7 | 75 | 41 | 0.502857 |
| Test-chr21-JPT-30 | 15 | 30 | 13 | 0.184444 |
| Test-chr21-JPT-50 | 15 | 50 | 27 | 0.261333 |
| Test-chr21-JPT-75 | 15 | 75 | 40 | 0.355556 |
| Test-chr21-YRI-30 | 68 | 30 | 20 | 0.118137 |
| Test-chr21-YRI-50 | 68 | 50 | 29 | 0.177353 |
| Test-chr21-YRI-75 | 68 | 75 | 47 | 0.260392 |

**Table 5 Performances of RM**

| | Time (sec.) | | | Gap (%) | | | Nodes | | |
|---|---|---|---|---|---|---|---|---|---|
| Data set | Average | Max | Min | Average | Max | Min | Average | Max | Min |
| Uniform | | | | | | | | | |
| $50 \times 10$ | 1.143 | 2.404 | 0.102 | 0.000 | 0 | 0 | 1.000 | 1 | 1 |
| $50 \times 10$r4 | 1.730 | 6.104 | 0.043 | 1.179 | 10 | 0 | 1.000 | 1 | 1 |
| $50 \times 10$r16 | 8.092 | 30.623 | 2.011 | 1.644 | 10.7692 | 0 | 1.533 | 9 | 1 |
| $50 \times 30$ | 11.772 | 53.42 | 2.732 | 2.440 | 7.14286 | 0 | 2.000 | 15 | 1 |
| $30 \times 50$ | 8.922 | 47.467 | 0.73 | 1.694 | 7.69231 | 0 | 10.260 | 75 | 1 |
| $30 \times 75$ | 15.624 | 35.693 | 1.358 | 1.649 | 6.66667 | 0 | 24.300 | 92 | 1 |
| $30 \times 100$ | 10.142 | 31.994 | 2.593 | 1.402 | 7.35294 | 0 | 8.500 | 25 | 1 |
| Nonuniform | | | | | | | | | |
| $50 \times 10$ | 0.634 | 1.726 | 0.127 | 0.513 | 7.69231 | 0 | 2.400 | 11 | 1 |
| $50 \times 30$ | 11.882 | 30.411 | 1.59 | 1.164 | 6.25 | 0 | 11.867 | 35 | 1 |
| $30 \times 50$ | 10.764 | 24.108 | 0.815 | 0.890 | 4.09091 | 0 | 20.533 | 61 | 1 |
| $30 \times 75$ | 22.389 | 61.869 | 3.537 | 1.038 | 5.55556 | 0 | 62.286 | 387 | 1 |
| $30 \times 100$ | 74.925 | 462.791 | 12.953 | 1.521 | 4.7619 | 0 | 216.071 | 1,679 | 8 |
| Biological | | | | | | | | | |
| CHR10-CEU | 102.792 | 305.103 | 0.774 | 0.000 | 0 | 0 | 270.333 | 807 | 1 |
| CHR10-HCB | 38.058 | 96.324 | 8.746 | 2.593 | 7.77778 | 0 | 67.000 | 151 | 1 |
| CHR10-JPT | 0.895 | 1.583 | 0.368 | 1.515 | 4.54545 | 0 | 7.000 | 11 | 1 |
| CHR10-YRI | 73.723 | 116.127 | 31.353 | 1.111 | 3.33333 | 0 | 89.667 | 123 | 63 |
| CHR21-CEU | 18.868 | 54.562 | 0.428 | 1.515 | 4.54545 | 0 | 49.667 | 145 | 1 |
| CHR21-HCB | 0.182 | 0.456 | 0.017 | 0.000 | 0 | 0 | 8.000 | 19 | 1 |
| CHR21-JPT | 1.781 | 2.87 | 0.967 | 0.833 | 2.5 | 0 | 15.667 | 29 | 1 |
| CHR21-YRI | 2,349.331 | 6,819.2 | 50.012 | 0.000 | 0 | 0 | 3,815.667 | 11,199 | 123 |

11.822 seconds, respectively). Instances of the data set having a genotype length of 30 on average have been solved in about 11.772 seconds, with two instances (02 and 08) taking more than the average time (53.42 and 38.642 seconds, respectively). The remaining ones took less than nine seconds. Instances of the data sets having genotype length greater than 30 on average were solved in at most 15.624 seconds, with a maximum solution of 47.467—hence, much less than HybridIP. Moreover, the time taken by the Xpress-MP primal heuristic to obtain the first feasible solution was always smaller than one second even in the most difficult instance. However, it is worth noting that the hardware used in our experiments is twice as fast than that of Brown and Harrower (2006); hence, a direct comparison of the runtimes is unfair. The comparison of the gap and the branches instead provides a more independent insight about the effectiveness

**Table 6 Performances of SM**

| | Time (sec.) | | | Gap (%) | | | Nodes | | | Cuts | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data set | Average | Max | Min | Average | Max | Min | Average | Max | Min | Average | Max | Min |
| Uniform | | | | | | | | | | | | |
| $50 \times 10$ | 0.835 | 2.339 | 0.085 | 0.000 | 0 | 0 | 1.000 | 1 | 1 | 0.667 | 10 | 0 |
| $50 \times 10$r4 | 1.350 | 3.269 | 0.023 | 0.667 | 10 | 0 | 1.000 | 1 | 1 | 2.000 | 10 | 0 |
| $50 \times 10$r16 | 5.848 | 21.293 | 1.611 | 1.201 | 5.55556 | 0 | 4.733 | 49 | 1 | 4.600 | 20 | 0 |
| $50 \times 30$ | 8.305 | 32.291 | 1.627 | 1.609 | 7.14286 | 0 | 5.267 | 34 | 1 | 8.667 | 32 | 0 |
| $30 \times 50$ | 6.227 | 82.677 | 0.366 | 0.934 | 9.09091 | 0 | 48.240 | 1,109 | 1 | 37.080 | 741 | 0 |
| $30 \times 75$ | 10.836 | 20.595 | 1.519 | 0.979 | 6.66667 | 0 | 49.900 | 175 | 1 | 36.900 | 116 | 10 |
| $30 \times 100$ | 9.515 | 27.67 | 2.413 | 1.305 | 6.66667 | 0 | 25.800 | 103 | 1 | 23.400 | 69 | 10 |
| Nonuniform | | | | | | | | | | | | |
| $50 \times 10$ | 0.492 | 0.956 | 0.086 | 0.513 | 7.69231 | 0 | 2.800 | 9 | 1 | 6.933 | 17 | 0 |
| $50 \times 30$ | 7.780 | 24.36 | 0.88 | 0.556 | 5.55556 | 0 | 24.867 | 77 | 1 | 21.667 | 58 | 0 |
| $30 \times 50$ | 10.553 | 38.569 | 0.754 | 0.540 | 5.15873 | 0 | 110.667 | 559 | 1 | 62.867 | 203 | 0 |
| $30 \times 75$ | 27.939 | 92.472 | 2.888 | 0.722 | 5.55556 | 0 | 282.786 | 1,477 | 1 | 138.571 | 569 | 10 |
| $30 \times 100$ | 126.111 | 7,227.52 | 5.798 | 0.000 | 3.99743 | 0 | 2,434.000 | 94,846 | 34 | 1,748.857 | 31,915 | 36 |
| Biological | | | | | | | | | | | | |
| CHR10-CEU | 46.759 | 138.315 | 0.282 | 0.000 | 0 | 0 | 137.000 | 401 | 1 | 50.667 | 142 | 0 |
| CHR10-HCB | 1,026.015 | 3,067.22 | 5.362 | 1.754 | 5.26316 | 0 | 11,041.667 | 32,981 | 29 | 9,948.333 | 29,760 | 20 |
| CHR10-JPT | 0.977 | 2.453 | 0.124 | 0.000 | 0 | 0 | 61.667 | 173 | 6 | 46.000 | 104 | 17 |
| CHR10-YRI | 73.674 | 135.232 | 38.146 | 0.617 | 1.85185 | 0 | 199.000 | 253 | 141 | 97.000 | 125 | 73 |
| CHR21-CEU | 449.307 | 1,346.36 | 0.227 | 1.515 | 4.54545 | 0 | 4,289.000 | 12,857 | 1 | 1,810.333 | 5,419 | 0 |
| CHR21-HCB | 0.096 | 0.2 | 0.008 | 0.000 | 0 | 0 | 2.667 | 4 | 1 | 9.333 | 16 | 0 |
| CHR21-JPT | 2.297 | 4.42 | 0.085 | 0.282 | 3.84615 | 0 | 85.667 | 209 | 1 | 65.333 | 152 | 0 |
| CHR21-YRI | 95.334 | 160.563 | 30.105 | 0.000 | 0 | 0 | 205.000 | 269 | 141 | 112.000 | 131 | 93 |

**Table 7 Performances of SMM**

| | Time (sec.) | | | Gap (%) | | | Nodes | | | Cuts | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data set | Average | Max | Min | Average | Max | Min | Average | Max | Min | Average | Max | Min |
| Uniform | | | | | | | | | | | | |
| $50 \times 10$ | 1.139 | 3.383 | 0.088 | 0.000 | 0 | 0 | 1.667 | 4 | 1 | 3.200 | 13 | 0 |
| $50 \times 10$r4 | 3.306 | 17.695 | 0.021 | 0.000 | 0 | 0 | 2.933 | 16 | 1 | 7.333 | 17 | 0 |
| $50 \times 10$r16 | 22.720 | 102.463 | 3.864 | 0.000 | 0 | 0 | 17.800 | 89 | 1 | 16.133 | 31 | 0 |
| $50 \times 30$ | 35.562 | 152.909 | 1.285 | 0.000 | 0 | 0 | 18.867 | 61 | 1 | 17.600 | 44 | 0 |
| $30 \times 50$ | 23.860 | 415.269 | 0.787 | 0.121 | 3.125 | 0 | 93.900 | 2,399 | 1 | 67.260 | 1,736 | 0 |
| $30 \times 75$ | 24.163 | 85.899 | 1.935 | 0.000 | 0 | 0 | 95.300 | 369 | 1 | 70.100 | 259 | 13 |
| $30 \times 100$ | 15.769 | 47.697 | 4.1 | 0.000 | 0 | 0 | 42.100 | 223 | 1 | 29.900 | 129 | 11 |
| Nonuniform | | | | | | | | | | | | |
| $50 \times 10$ | 0.976 | 1.611 | 0.251 | 0.000 | 0 | 0 | 5.733 | 19 | 1 | 11.400 | 20 | 0 |
| $50 \times 30$ | 22.448 | 78.588 | 2.02 | 0.196 | 2.94118 | 0 | 62.133 | 203 | 7 | 37.267 | 93 | 13 |
| $30 \times 50$ | 34.690 | 266.58 | 2.919 | 0.144 | 1.85185 | 0 | 311.933 | 3,367 | 9 | 138.000 | 1,227 | 14 |
| $30 \times 75$ | 81.194 | 445.243 | 7.713 | 0.054 | 0.757576 | 0 | 685.643 | 4,063 | 21 | 413.429 | 1,910 | 21 |
| $30 \times 100$ | 658.337 | 7,324.47 | 6.603 | 0.000 | 1.333 | 0 | 11,124.214 | 116,887 | 14 | 9,114.571 | 101,623 | 22 |
| Biological | | | | | | | | | | | | |
| CHR10-CEU | 104.346 | 303.915 | 0.777 | 0.000 | 0 | 0 | 224.667 | 649 | 8 | 86.333 | 226 | 15 |
| CHR10-HCB | 1,089.757 | 3,245.86 | 7.693 | 0.292 | 0.877193 | 0 | 9,190.333 | 27,391 | 77 | 8,411.333 | 25,112 | 36 |
| CHR10-JPT | 1.910 | 4.07 | 0.319 | 0.000 | 0 | 0 | 54.000 | 135 | 11 | 44.000 | 87 | 20 |
| CHR10-YRI | 191.857 | 464.219 | 39.36 | 0.000 | 0 | 0 | 287.333 | 401 | 179 | 176.667 | 299 | 71 |
| CHR21-CEU | 892.729 | 2,670.99 | 0.662 | 0.000 | 0 | 0 | 4,376.000 | 13,103 | 8 | 1,771.333 | 5,277 | 18 |
| CHR21-HCB | 0.187 | 0.411 | 0.008 | 0.000 | 0 | 0 | 6.333 | 13 | 1 | 15.333 | 23 | 10 |
| CHR21-JPT | 6.327 | 13.354 | 0.865 | 0.000 | 0 | 0 | 183.333 | 463 | 8 | 126.000 | 285 | 18 |
| CHR21-YRI | 206.543 | 7,242.43 | 84.659 | 0.000 | 0 | 0 | 346.667 | 6,204 | 401 | 221.333 | 2,814 | 299 |

of our model. To this end, we observe that, unlike HybridIP, the gap is 0 in at least the 50% of the instances analyzed and that several instances are typically solved without branching.

SM shows better performances than RM (see Table 6) both in terms of solution times and gaps. In fact, on average, SM was able to solve instances of the data sets having genotype length of 10 in about 5.8 seconds, and maximum and minimum solution times were always smaller than RM. The first two data sets were solved at the root node, whereas an instance of the third data set needed at most 49 branches to reach the optimum. The efficiency of the valid inequalities is confirmed by both the smaller gaps than RM and the average number of cuts added during the exact search. This trend is also confirmed for the data sets in which genotype length was greater than or equal to 30, with the only exception being instance 19 of the data set $30 \times 50$, which took 82.677 seconds versus 25.791 seconds of RM. This instance had a gap of 2.63158 (versus 0.0 of RM), a number of branches equal to 1,109 (versus 75 of RM), and the number of cuts added equal to 741. The anomaly of this instance seems to be due to the valid inequalities (34) and (35), which negatively interfere with the primal heuristic of the Xpress Optimizer.

Finally, SMM shows worse running time performances than RM and SM (see Table 7) in all uniform data sets. This is in contrast to the average, maximum, and minimum gaps, which are always smaller than RM and SM. Once again, the anomalous behavior is due to the valid inequalities (34) and (35), which interfere both with the primal heuristic of the Xpress Optimizer and with the automatic branching strategy.

**4.4.2. Nonuniform Data Sets.** Brown and Harrower (2006) showed that on the nonuniform data sets the HybridIP solves the instances having lengths 10 and 30. Specifically, the instances having genotypes of length 10 are solved in 1 second, and ten out of the fifteen length 30 instances are solved in less than 15 seconds. However, the HybridIP was only able to solve six instances having genotype length of 50, although those instances were solved in 39 seconds or less. Again, the Hybrid was able to solve 5 of the 15 instances with a genotype length of 75, two of which were solved in less than one minute. Finally, the HybridIP is able to solve three instances having genotype length 100, two of which solved in under two minutes.

Numerical experiments show that our models outperform HybridIP on the nonuniform data sets. In fact, as shown in Table 5, RM on average is able to solve instances of the data sets having genotype length of 10 in about 0.634 seconds, with a maximum runtime of 1.726 seconds; instances having genotype length of 30 in about 11.882 seconds, with a maximum runtime of 30.411 seconds; instances having genotype length of 50 in about 10.764 seconds, with a maximum runtime of 24.108 seconds; instances having genotype length of 75 in about 22.389 seconds, with a maximum

**Table 8 Performances of the Models as Function of the Number of Genotypes and SNP Length**

| | | SNP length | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Genotypes | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| RM | 50 | 21.539 | 47.546 | 70.814 | 95.420 | 119.308 | 141.682 | 160.54 | 184.586 |
| RM | 75 | 76.530 | 145.127 | 225.173 | 314.501 | 396.974 | 457.972 | 560.465 | 615.152 |
| RM | 100 | 189.736 | 353.716 | 544.839 | 765.423 | 962.542 | 1,119.63 | 927.976 | 731.711 |
| RM | 200 | 845.646 | 1,648.66 | 5,237.8 | 3,539.14 | 5,355.9 | 5,443.56 | 6,244.8 | — |
| RM | 300 | 3,944.94 | — | — | — | — | — | — | — |
| SM | 50 | 14.671 | 19.974 | 26.511 | 40.017 | 62.375 | 77.621 | 98.120 | 112.353 |
| SM | 75 | 41.658 | 94.860 | 113.958 | 171.644 | 281.06 | 217.39 | 391.29 | 490.83 |
| SM | 100 | 104.031 | 296.637 | 259.112 | 399.288 | 610.626 | 532.14 | 771.02 | 541.03 |
| SM | 200 | 625.018 | 1,118.448 | 3,665.77 | 1,608.02 | 4,019.28 | — | — | — |
| SM | 300 | — | — | — | — | — | — | — | — |
| SMM | 50 | 32.552 | 65.412 | 115.869 | 156.771 | 131.420 | 176.008 | 212.697 | 244.881 |
| SMM | 75 | 108.810 | 271.341 | 191.23 | 398.29 | 503.754 | 531.244 | 571.221 | 799.84 |
| SMM | 100 | 265.200 | 361.299 | 600.554 | 846.443 | 1,207.97 | 1,757.992 | 1,727.02 | 1,047.39 |
| SMM | 200 | 1,315.935 | 1,706.01 | 6,923.9 | 4,760.1 | 7,144.45 | — | — | — |
| SMM | 300 | — | — | — | — | — | — | — | — |

runtime of 61.869 seconds; and finally, instances having genotype length of 100 in about 74.925 seconds, with a maximum runtime of 462.791 seconds. SM performs better than RM when solving instances having length of 50 or smaller but on average is slower for larger instances. Specifically, SM is always faster than RM in almost all instances of the data set $30 \times 75$, with the only exception being the instances of 02 and 14, whose solution times were 85.551 and 92.472 seconds, respectively. Similarly, SM is always faster than RM in almost all instances of the data set $30 \times 100$, with the only exception being the instances of 01, 09, and 14 whose solution times were 267.156, 663.219, and 327.615 seconds, respectively. Moreover, SM was unable to solve within the limit time the instance 00 (its solution time was 7,227.52).

Finally, SMM shows worse running time performances than RM and SM (see Table 7) in all nonuniform data sets. It is worth noting that SMM is characterized by the same difficulties as SM for the instances of 02 and 14 of the data set $30 \times 75$, and for 00, 01, 09, and 14 of the data set $30 \times 100$ (although the gaps are always better than the corresponding ones of RM and SM).

**4.4.3. Biological Data Sets.** To complete the performance analysis on Brown and Harrower's data sets (Brown and Harrower 2006), we tested our model on the biological data sets. Brown and Harrower showed that HybridIP was able to solve 16 of the 24 biological instances. Unfortunately, the run times for this data set were not provided by the authors.

In general, the numerical experiments show that on the biological data sets SM generally has better performance for the data sets CHR10-HCB, CHR21-CEU, and CHR21-JPT, whose instances chr10-HCB-75, chr10-CEU-30, and chr10-JPT-30 took 3,067.22, 1,346.36, and 4.42 seconds, respectively. Possibly, the implementation of a primal heuristic could be helpful for decreasing the solution time of those instances in which the valid inequalities (34) and (35) negatively interfere with the Xpress Optimizer primal heuristic. However, the benefits of the valid inequalities are evident. Specifically, in CHR10-CEU the solution time is halved, and in CHR21-YRI the solution time for the most difficult instance decreases from almost two hours to almost three minutes.

### 4.5. Performance Analysis on Larger Data Sets

In some circumstances it may be necessary to analyze data sets of multiple genes belonging to a common chromosome (e.g., when dealing with the human leukocyte antigen system belonging to chromosome 6). In such cases, data sets containing large numbers of SNPs could easily arise. For this reason, we have generated new classes of data sets with the aim of studying the performances of our models on larger data sets. Specifically, under the same sample generation described in §4.3, we generated five classes of data sets characterized by 50, 75, 100, 200, and 300 genotypes each. Each class contains eight instances characterized by a SNP length ranging between 100 and 800. We have fixed the runtime limit to two hours and considered as a measure of performance the time (in seconds) necessary to solve each instance. The results, listed in Table 8, show that SM is characterized by better runtime performances, although it is unable (together with SMM) to analyze data sets containing more than 200 genotypes and

600 SNPs. Only RM is able to analyze data sets containing more than 200 genotypes and 600 SNPs. This fact is due to the overhead of RAM introduced by the separation algorithm. Possibly, a combined use of the Xpress automatic cuts and pre-solving with fine-tuning on the number of valid inequalities added by the separation oracle at each node of the search tree could allow SM to analyze larger data sets.

## 5. Conclusion

In this paper, we described a new polynomial formulation for the pure parsimony haplotyping problem. The idea is that a solution for PPH induces a family of genotype subsets such that (i) each subset of genotypes shares one haplotype, (ii) each genotype belongs to exactly two subsets, and (iii) every pair of subsets intersects in at most one genotype. This observation led us to develop a new integer linear model for PPH based on the class representatives with smallest index, already proposed for the coloring problem (Campelo et al. 2008). We have suggested techniques to reduce the overall number of variables and constraints required by our models and provided valid inequalities to strengthen them. Computational experience showed that, under the same sample generation conditions used by Gusfield (2003), Brown and Harrower (2006), and Bertolazzi et al. (2008), our models outperform all existing IP models for PPH, whose limit is represented by data sets characterized by 68 genotypes of 75 SNPs each (Brown and Harrower 2006). Our model can be applied to larger real biological genotype data sets than the ones proposed by Brown and Harrower (2006). The model is compact, polynomial sized, easy to implement, solvable with standard solvers, and usable in those cases for which the parsimony principle is well suited for haplotyping inference.

### Acknowledgments

The first author acknowledges support from the Belgian National Fund for Scientific Research (F.N.R.S.), of which he is a Research Fellow. The first and third authors also acknowledge support from Communauté Française de Belgique–Actions de Recherche Concertées (ARC). Finally, the authors thank Daniel G. Brown and Ian M. Harrower (who kindly provided the benchmark data sets), the area editor, the associate editor, and the anonymous reviewers for helpful comments on the previous version of the manuscript.

### References

Bafna, V., D. Gusfield, G. Lancia, S. Yooseph. 2003. Haplotyping as perfect phylogeny: A direct approach. *J. Comput. Biology* **10**(3–4) 323–340.

Bertolazzi, P., A. Godi, M. Labbé, L. Tininini. 2008. Solving haplotyping inference parsimony problem using a new basic polynomial formulation. *Comput. Math. Appl.* **55**(5) 900–911.

Blain, P., C. Davis, J. Silva, C. Vinzant. 2009. Diversity graphs. S. Bulenko, W. A. Chaovalitwongse, P. M. Pardalos, eds. *Clustering Challenges in Biological Networks*. World Scientific, Hackensack, NJ, 129–150.

Bonizzoni, P., G. Della Vedova, R. Dondi, L. Jing. 2003. The haplotyping problem: A view of computational models and solutions. *Internat. J. Comput. Sci. Tech.* **18**(6) 675–688.

Brown, D., I. M. Harrower. 2004. A new integer programming formulation for the pure parsimony problem in haplotype analysis. I. Jonassen, J. Kim, eds. *Algorithms in Bioformatics: Proc. Fourth Annual Workshop. Lecture Notes in Computer Science*, Vol. 3240. Springer-Verlag, Berlin, 254–265.

Brown, D., I. M. Harrower. 2006. Integer programming approaches to haplotype inference by pure parsimony. *IEEE Trans. Comput. Biol. Bioinformatics* **3**(2) 141–154.

Campelo, M., V. Campos, R. Correa. 2008. On the asymmetric representatives formulation for the vertex coloring problem. *Discrete Appl. Math.* **156**(7) 1097–1111.

Chakravarti, A. 1998. It's raining SNPs, hallelujah? *Nature Genetics* **19**(3) 216–217.

Clark, A. G., K. Weiss, D. Nickerson, S. Taylor, A. Buchanan, J. Stengard, V. Salomaa et al. 1998. Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase. *Amer. J. Human Genetics* **63**(2) 595–612.

Eskin, E., E. Halperin, R. M. Karp. 2003. Efficient reconstruction of haplotype structure via perfect phylogeny. *J. Bioinformatics Comput. Biol.* **1**(1) 1–20.

Excoffier, L., M. Slatkin. 1995. Maximum likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biol. Evolution* **12**(5) 921–927.

Fallin, D., N. J. Schork. 2000. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *Amer. J. Human Genetics* **67**(4) 947–959.

Garey, M. R., D. S. Johnson. 2003. *Computers and Intractability*. Freeman, New York.

Gusfield, D. 2003. Haplotype inference by pure parsimony. R. Baeza-Yates, E. Chávez, M. Crochemore, eds. *Combinatorial Pattern Matching 14th Annual Sympos. Lecture Notes in Computer Science*, Vol. 2676. Springer-Verlag, Berlin, 144–155.

Helmuth, L. 2001. Genome research: Map of the human genome 3.0. *Science* **293**(5530) 583–585.

Hoehe, M. R., K. Kopke, B. Wendel, K. Rohde, C. Flachmeier, K. K. Kidd, W. H. Berrettini, G. M. Church. 2000. Sequence variability and candidate gene analysis in complex disease: Association of $\mu$ opioid receptor gene variation with substance dependence. *Human Molecular Genetics* **9**(19) 2895–2908.

Hudson, R. R. 1990. Gene genealogies and the coalescent process. D. Futuyma, J. Antonovics, eds. *Oxford Survey of Evolutionary Biology*, Vol. 7. Oxford University Press, Oxford, UK, 1–44.

International HapMap Consortium. 2004. Integrating ethics and science in the International HapMap Project. *Nature Rev. Genetics* **5**(6) 467–475.

Kimura, M. 1983. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge, UK.

Lancia, G., R. Rizzi. 2006. A polynomial case of the parsimony haplotyping problem. *Oper. Res. Lett.* **34**(3) 289–295.

Lancia, G., M. C. Pinotti, R. Rizzi. 2004. Haplotyping populations by pure parsimony: Complexity of exact and approximate algorithms. *INFORMS J. Comput.* **16**(4) 348–359.

Marshall, E. 1999. Drug firms to create public database of genetic mutations. *Science* **284**(5413) 406–407.

Niu, T., Z. S. Qin, X. Xu, J. S. Liu. 2002. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Amer. J. Human Genetics* **70**(1) 157–169.

Pardalos, P. M., J. Xue. 1994. The maximum clique problem. *J. Global Optim.* **4**(3) 301–328.

Qin, Z. S., T. Niu, J. S. Liu. 2002. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *Amer. J. Human Genetics* **71**(5) 1242–1247.

Schwartz, R., A. G. Clark, S. Istrail. 2002. Methods for inferring block-wise ancestral history from haploid sequences. R. Guigo, D. Gusfield, eds. *Algorithms in Bioinformatics: Second Internat. Workshop* (*WABI'02*). *Lecture Notes in Computer Science*, Vol. 2452. Springer-Verlag, Berlin, 44–59.

Semple, C., M. Steel. 2003. *Phylogenetics*. Oxford University Press, New York.

Stephens, M., P. Donnelly. 2003. A comparison of Bayesian methods for haplotype reconstruction from population genotype data. *Amer. J. Human Genetics* **73**(5) 1162–1169.

Stephens, M., N. J. Smith, P. Donnelly. 2001. A new statistical method for haplotype reconstruction from population data. *Amer. J. Human Genetics* **68**(4) 978–989.

Terwilliger, J. D., K. M. Weiss. 1998. Linkage disequilibrium mapping of complex disease: Fantasy and reality? *Current Opinions Biotechnology* **9**(6) 579–594.

Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith et al. 2001. The sequence of the human genome. *Science* **291**(5507) 1304–1351.

Wang, L., Y. Xu. 2003. Haplotype inference by maximum parsimony. *Bioinformatics* **19**(14) 1773–1780.