Efficient and Accurate Haplotype Inference by Combining Parsimony and Pedigree Information

Ana Graça[1], Inês Lynce[1], João Marques-Silva[2], and Arlindo L. Oliveira[1]

[1] INESC-ID/IST, Technical University of Lisbon
{assg,ines}@sat.inesc-id.pt, aml@inesc-id.pt
[2] CSI/CASL, University College Dublin
jpms@ucd.ie

Abstract. Existing genotyping technologies have enabled researchers to genotype hundreds of thousands of SNPs efficiently and inexpensively. Methods for the imputation of non-genotyped SNPs and the inference of haplotype information from genotypes, however, remain important, since they have the potential to increase the power of statistical association tests. In many cases, studies are conducted in sets of individuals where the pedigree information is relevant, and can be used to increase the power of tests and to decrease the impact of population structure on the obtained results. This paper proposes a new Boolean optimization model for haplotype inference combining two combinatorial approaches: the Minimum Recombinant Haplotyping Configuration (MRHC), which minimizes the number of recombinant events within a pedigree, and the Haplotype Inference by Pure Parsimony (HIPP), that aims at finding a solution with a minimum number of distinct haplotypes within a population. The paper also describes the use of well-known techniques, which yield significant performance gains. Concrete examples include symmetry breaking, identification of lower bounds, and the use of an appropriate constraint solver. Experimental results show that the new PedRPoly model is competitive both in terms of accuracy and efficiency.

Keywords: haplotype inference, Boolean optimization

1 Introduction

The majority of complex diseases are influenced by both environmental and genetic factors. Existing technologies have enabled researchers to genotype hundreds of thousands of single nucleotide polymorphisms (SNPs) in a single run. Data obtained with genotyping or sequencing technologies for thousands of individuals will be available in the near future. This will enable researchers to conduct whole genome association studies in an unprecedented scale to detect increasingly subtle and more complex associations between genomes and diseases [33].

Despite these technological advances, existing technologies still generate genotypes obtained from the conflation of two haplotypes on homologous chromosomes. Hence, haplotypes must be inferred computationally using the experimentally identified genotypes. Inference of the haplotypes is made possible by the fact that, in many cases, there exists a strong correlation between the allele present in a particular SNP and other nearby sites. A given combination of alleles in one chromosome is termed a haplotype, and the deviation from independence that exists between alleles is known as linkage disequilibrium.

Haplotype inference from genotype data remains an important and challenging task. The identification of haplotypes allows to develop haplotype-based association studies [5]. In addition, most imputation methods require the haplotype data [30].

Many haplotype inference methods apply to unrelated individuals. Nonetheless, pedigree information is available in many studies and can be used to improve the results of inference methods. Combinatorial methods for haplotype inference have been shown to be practical and relevant, either for phasing families [20] or unrelated individuals [10]. Recently, a study comparing the haplotype inference methods using pedigrees and unrelated individuals [21] concluded that taking into consideration both pedigree and population information leads to improvements on the precision of haplotype inference methods.

This paper proposes the combination of two well-known haplotype inference methods: pure parsimony, which aims at finding a solution that uses the minimum possible number of distinct haplotypes, and the minimum recombinant approach used to phase individuals organized in pedigrees by minimizing the number of recombination events within each pedigree. The resulting haplotype inference model, PedRPoly, is shown to be quite competitive and more accurate than the existing methods for haplotype inference from pedigrees, in particular, using the minimum recombinant approach [20]. Note that a simpler and fairly inefficient version of this model was first outlined in [9].

The models developed in this paper represent challenging combinatorial optimization problems, which can be viewed as a special case of multi-objective optimization. The solution methods for these problems combine techniques often used in the fields of operations research and artificial intelligence. Furthermore, the proposed models yield accuracy results that conclusively outperform the minimum recombinant approach. This paper describes the first PedRPoly algorithm which can actually be used in practice, since it is both efficient and accurate.

This paper is organized in two main parts. The first part describes the PedRPoly model, which combines the pure parsimony and the minimum recombinant approaches. This model is enhanced with several constraint modeling techniques. These techniques aim at improving the efficiency of the method and include the identification of lower bounds, symmetry breaking and a heuristic sorting technique. The second part conducts a comprehensive experimental evaluation of the accuracy and efficiency of PedRPoly on a large set of instances. The experimental evaluation was also used to select the best performing constraint solver, among a representative number of solvers.

2 Haplotype Inference

Variations in the DNA sequence are the basis for evolution. *Single Nucleotide Polymorphisms* (SNPs) are the most common variations between human beings, and occur when a nucleotide base (A, T, C, G) is changed to other nucleotide base at a single DNA position. Moreover, the mutant type nucleotide must be represented in a significant percentage of the population (normally 1%). An example of a SNP is the mutation of the DNA sequence AC**T**TGAC to AC**A**TGAC, where the third nucleotide is changed from T to A. DNA is organized in structures called chromosomes. Chromosomes contain the genetic information coded in different type of substructures, of which the best known are the genes. A gene is a sequence of DNA bases which encodes a specific protein.

A given combination of SNPs in a single chromosome is called a *haplotype*. Moreover, SNPs within a haplotype tend to be inherited together. The deviation from independence that exists between SNPs is known as *linkage disequilibrium* (LD).

Diploid organisms, such as human beings, have pairs of homologous chromosomes, with each chromosome in a pair inherited from a single parent. In practice, experimental technology is only able to obtain genotypes, which correspond to the conflated data of two haplotypes on homologous chromosomes. The haplotype inference problem consists in obtaining the set of haplotype pairs which originated a given set of genotypes.

Considering that the assumptions underlying the infinite-site model [14] are valid, we may assume that each SNP can only have two values (called alleles). Each haplotype can therefore be represented by a binary string, with size $m \in \mathbb{N}$, where 0 represents the wild type allele and 1 represents the mutant type allele. Each site of the haplotype h_i is represented by $h_{i\,j}$ $(1 \leq j \leq m)$. In addition, each genotype is represented by a string, with size m, over the alphabet $\{0, 1, 2\}$, and each site of the genotype g_i is represented by $g_{i\,j}$. Each genotype is explained by two haplotypes. A genotype g_i is explained by a pair of haplotypes $(h_i^a,\, h_i^b)$, which is represented by $g_i = h_i^a \oplus h_i^b$, if

$$g_{i\,j} = \begin{cases} h_{i\,j}^a & \text{if } \; h_{i\,j}^a = h_{i\,j}^b \\ 2 & \text{if } \; h_{i\,j}^a \neq h_{i\,j}^b \end{cases} .$$

A genotype site $g_{i\,j}$ with either value 0 or 1 is a homozygous site (the same allele is inherited from both parents), whereas a site with value 2 is a heterozygous site (different alleles are inherited from each parent).

Definition 1. *(Haplotype Inference) Given a set $\mathcal{G}$ of n genotypes, each with size m, the haplotype inference problem consists in finding a set of haplotypes $\mathcal{H}$, such that each genotype $g_i \in \mathcal{G}$ is explained by two haplotypes $h_i^a, h_i^b \in \mathcal{H}$.*

Observe that for each genotype g with k heterozygous sites, there are 2^{k-1} non-ordered pairs of haplotypes that can explain g. For example, genotype $g = 022$ can be explained either by haplotypes $(000, 011)$ or by haplotypes $(001, 010)$.

Most often genotyping procedures leave a percentage of missing data, i.e. genotype positions with unknown values. To represent missing sites, the alphabet of the genotypes is extended to $\{0, 1, 2, ?\}$.

Pedigrees Pedigree data adds new relevant information to the haplotype inference problem. A pedigree refers to the genealogical tree which allows studying the inheritance of genes within a family. The *Mendelian laws of inheritance* are well established assumptions and, in particular, state that all sites in a single haplotype are inherited from a single parent, assuming there are no mutations within a pedigree. We assume that haplotype h^a is inherited from the father and h^b is inherited from the mother. Nonetheless, a recombination may happen. A recombination occurs when two haplotypes of a parent are mixed together and the recombinant is inherited by the child. For example, suppose a father has the haplotype pair $(000, 111)$ and the haplotype that he passed on to his child is 100. Here, one recombination event must have occurred: haplotypes 000 and 111 got shuffled and originated a new haplotype $h = 100$. Therefore, the child has inherited the first allele from the paternal grandmother, while second and third alleles were inherited from the paternal grandfather. In a pedigree, an individual is a *founder* if he does not have parents on the pedigree (and a *non-founder* if he has both parents on the pedigree).

2.1 Minimum Recombinant Haplotype Configuration

Most rule-based haplotype inference methods for pedigrees assume no recombination within each pedigree [35, 37, 22]. The assumption of no recombination is valid in many cases because recombination events are rare in DNA regions with high linkage disequilibrium. Nonetheless, this assumption can be violated even for some dense markers [19]. Therefore, a more realistic approach consists in minimizing the number of recombinations within pedigrees [13, 31, 20].

Definition 2. *The minimum recombinant haplotype configuration (MRHC) problem aims at finding a haplotype inference solution for a pedigree which minimizes the number of required recombination events.*

For example, suppose the father has the genotype $g_1 = 202$, the mother has the genotype $g_2 = 212$ and the child has the genotype $g_3 = 222$. One possible solution to the haplotype inference problem is $g_1 = 001 \oplus 100$, $g_2 = 010 \oplus 111$ and $g_3 = 101 \oplus 010$. However, this solution implies that one recombination has occurred, because the child has not inherited an integral haplotype from his father, but a mixture of his paternal grandparents haplotypes. A different solution to this example admits no recombination and, therefore, is a MRHC solution: $g_1 = 000 \oplus 101$, $g_2 = 010 \oplus 111$ and $g_3 = 000 \oplus 111$.

In general, there can be a significant number of MRHC solutions to the same problem. For instance, $g_1 = 000 \oplus 101$, $g_2 = 010 \oplus 111$, $g_3 = 101 \oplus 010$ is another 0-recombinant solution for the previous example, i.e. another MRHC solution.

The MRHC problem has been shown to be a NP-hard [19, 25] problem. The PedPhase tool [20] implements an integer linear programming (ILP) model for MRHC with missing alleles.

2.2 Haplotype Inference by Pure Parsimony

The haplotype inference by pure parsimony problem consists in finding a solution to the haplotype inference problem which minimizes the number of distinct haplotypes [12].

Natural phenomena tend to be explained parsimoniously, using the minimum number of required entities. The haplotype inference by pure parsimony approach is also biologically motivated by the fact that individuals from the same population have the same ancestors and mutations do not occur often. Moreover, it is also well-known that the number of haplotypes in a population is much smaller than the number of genotypes [34].

Definition 3. *The haplotype inference by pure parsimony (HIPP) approach aims at finding a minimum-cardinality set of haplotypes $\mathcal{H}$ that can explain a given set of genotypes $\mathcal{G}$.*

For example, consider the set of genotypes $\mathcal{G} = \{g_1,\ g_2,\ g_3\} = \{202, 212, 222\}$. There are solutions using 6 different haplotypes: $\mathcal{H}_1 = \{101, 000, 111, 010, 001, 110\}$, such that $g_1 = 101 \oplus 000$, $g_2 = 111 \oplus 010$ and $g_3 = 001 \oplus 110$. However the HIPP solution only requires 4 distinct haplotypes: $\mathcal{H}_2 = \{101, 000, 111, 010\}$ such that $g_1 = 101 \oplus 000$, $g_2 = 111 \oplus 010$ and $g_3 = 000 \oplus 111$.

The HIPP problem is NP-hard [16]. RPoly [10] is a state-of-the-art solver implementing a 0-1 ILP model for solving the HIPP problem.

3 The PedRPoly Model

This section describes the *minimum recombinant maximum parsimony* model, denoted as PedRPoly model. In practice, the PedRPoly model is a combination of the MRHC PedPhase model [20] and the HIPP RPoly model [10].

Definition 4. *Given sets of pedigrees from the same population, the minimum recombinant maximum parsimony (MRMP) model aims at finding a haplotype inference solution which first minimizes the number of recombination events within pedigrees and then minimizes the number of distinct haplotypes used.*

Figure 1 illustrates two trios (mother ◯, father □ and child ◊) from two families A and B with the corresponding genotypes. The figure includes three haplotype inference solutions. Solution 1 is a 0-recombinant solution with 7 distinct haplotypes {100, 101, 000, 111, 011, 001, 110}. Solution 2 is a 1-recombinant solution (there is one recombination event in family B) using 5 distinct haplotypes {100, 101, 000, 111, 011}. Solution 3 is a 0-recombinant solution using 5 distinct haplotypes {100, 101, 000, 111, 011}.

According to the PedRPoly model, solution 3 is preferred to the other solutions. Solution 3 is both a MRHC and a HIPP solution. Consequently, solution 3 is a MRMP solution. If there exists no solution that minimizes both criteria, then preference is given to the MRHC criterion. Hence, the MRHC solution which uses the smallest number of distinct haplotypes would be chosen.

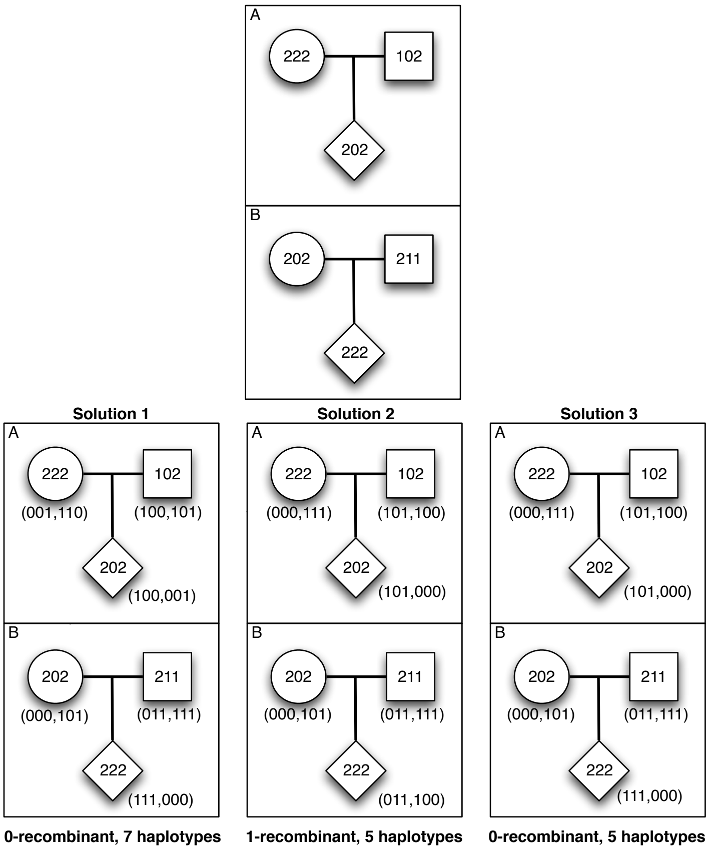

Fig. 1. Solutions for haplotype inference with two trios

The minimum recombinant maximum parsimony model combines the PedPhase and the RPoly models, in a new 0-1 integer linear programming model, so called PedRPoly. A 0-1 integer linear programming problem aims at finding a Boolean assignment to the variables which optimizes the value of a given cost function, subject to a set of linear constraints. The cost function and the general constraints of the PedRPoly model are detailed in Table 1, which is described in the following paragraphs.

Following the RPoly model, PedRPoly associates two haplotypes, h_i^a and h_i^b, with each genotype g_i, and these haplotypes are required to explain g_i. Moreover, PedRPoly associates a variable $t_{i\,j}$ with each heterozygous site $g_{i\,j}$, such that $t_{i\,j} = 1$ indicates that the mutant value was inherited from the father ($h^a_{i\,j} = 1$) and the wild value was inherited from the mother ($h^b_{i\,j} = 0$) whereas $t_{i\,j} = 0$ indicates that the wild value was inherited from the father ($h^a_{i\,j} = 0$) and the

Table 1. The PedRPoly Model

minimize:	$((2n+1)\times\sum_{\text{non-founder } i}\ \sum_{j=1}^{m-1}(r^1_{i\,j}+r^2_{i\,j}))+\sum_{i=1}^{n}(u^a_i+u^b_i)$	
subject to:		
Equation	Constraint	Indexes
	Mendelian laws of inheritance rules (Table 2)	
(1)	$-r^l_{i\,j}+g^l_{i\,j}-g^l_{i\,j+1}\leq 0$ $-r^l_{i\,j}-g^l_{i\,j}+g^l_{i\,j+1}\leq 0$	$l\in\{1,2\}$ $1\leq i\leq n$, i non-founder $1\leq j\leq m-1$
(2)	$\neg(R\Leftrightarrow S)\Rightarrow x^{p\,q}_{i\,k}$ (Table 3)	$p,q\in\{a,b\}$ $1\leq k<i\leq n$
(3)	$\sum_{k<i\,;\,q\in\{a,b\}} x^{p\,q}_{i\,k}-u^p_i\leq 2i-3$	$1<i\leq n$ $p\in\{a,b\}$

mutant value was inherited from the mother ($h^b_{i\,j}=1$). In addition, PedRPoly associates two variables with each missing site. Variable $t^a_{i\,j}$ is associated with the paternal haplotype site $h^a_{i\,j}$, whereas variable $t^b_{i\,j}$ is associated with the maternal haplotype site $h^b_{i\,j}$. The values of h^a_i and h^b_i at homozygous sites are implicitly assumed.

The grandparental origin of each site of the haplotypes must be considered when analyzing recombination events within pedigrees. Following the MRHC PedPhase model, for each non-founder individual i and site j, two variables are defined: $g^1_{i\,j}$ and $g^2_{i\,j}$. The assignment $g^1_{i\,j}=0$ means that the paternal allele of individual i at site j (i.e. $h^a_{i\,j}$) comes from the paternal grandfather, and $g^1_{i\,j}=1$ means that $h^a_{i\,j}$ comes from the paternal grandmother, i.e.

$$g^1_{i\,j}=\begin{cases}0 \text{ if } & h^a_{i\,j}=h^a_{f(i)\,j}\\ 1 \text{ if } & h^a_{i\,j}=h^b_{f(i)\,j}\end{cases},$$

where $f(i)$ corresponds to the father of individual i. In a similar way, $g^2_{i\,j}=0$ ($g^2_{i\,j}=1$) means that the maternal allele of individual i at site j comes from the maternal grandfather (grandmother), i.e.

$$g^2_{i\,j}=\begin{cases}0 \text{ if } & h^b_{i\,j}=h^a_{m(i)\,j}\\ 1 \text{ if } & h^b_{i\,j}=h^b_{m(i)\,j}\end{cases},$$

where $m(i)$ corresponds to the mother of individual i.

Constraints to ensure that the Mendelian laws of inheritance are satisfied are defined in Table 2. Note that PedRPoly only associates variables with heterozygous and missing sites (inspired by RPoly), while PedPhase also associates

Table 2. Mendelian laws of inheritance rules regarding variables $g^1_{i\,j}$. (The constraints involving variables $g^2_{i\,j}$ are defined similarly. $f(i)$ corresponds to the father of i. $1 \leq i \leq n$, i non-founder, $1 \leq j \leq m$.)

Condition	Constraint
$g_{i\,j} = 0 \wedge g_{f(i)\,j} = 2$	$t_{f(i)\,j} \Leftrightarrow g^1_{i\,j}$
$g_{i\,j} = 0 \wedge g_{f(i)\,j} =?$	$(g^1_{i\,j} \vee \neg t^a_{f(i)\,j}) \wedge (\neg g^1_{i\,j} \vee \neg t^b_{f(i)\,j})$
$g_{i\,j} = 1 \wedge g_{f(i)\,j} = 2$	$t_{f(i)\,j} \Leftrightarrow \neg g^1_{i\,j}$
$g_{i\,j} = 1 \wedge g_{f(i)\,j} =?$	$(g^1_{i\,j} \vee t^a_{f(i)\,j}) \wedge (\neg g^1_{i\,j} \vee t^b_{f(i)\,j})$
$g_{i\,j} = 2 \wedge g_{f(i)\,j} = 0$	$\neg t_{i\,j}$
$g_{i\,j} = 2 \wedge g_{f(i)\,j} = 1$	$t_{i\,j}$
$g_{i\,j} = 2 \wedge g_{f(i)\,j} = 2$	$(g^1_{i\,j} \vee t_{i\,j} \vee \neg t_{f(i)\,j}) \wedge (g^1_{i\,j} \vee \neg t_{i\,j} \vee t_{f(i)\,j}) \wedge$ $(\neg g^1_{i\,j} \vee t_{i\,j} \vee t_{f(i)\,j}) \wedge (\neg g^1_{i\,j} \vee \neg t_{i\,j} \vee \neg t_{f(i)\,j})$
$g_{i\,j} = 2 \wedge g_{f(i)\,j} =?$	$(g^1_{i\,j} \vee t_{i\,j} \vee \neg t^a_{f(i)\,j}) \wedge (g^1_{i\,j} \vee \neg t_{i\,j} \vee t^a_{f(i)\,j}) \wedge$ $(\neg g^1_{i\,j} \vee t_{i\,j} \vee \neg t^b_{f(i)\,j}) \wedge (\neg g^1_{i\,j} \vee \neg t_{i\,j} \vee t^b_{f(i)\,j})$
$g_{i\,j} =? \wedge g_{f(i)\,j} = 0$	$\neg t^a_{i\,j}$
$g_{i\,j} =? \wedge g_{f(i)\,j} = 1$	$t^a_{i\,j}$
$g_{i\,j} =? \wedge g_{f(i)\,j} = 2$	$(g^1_{i\,j} \vee t^a_{i\,j} \vee \neg t_{f(i)\,j}) \wedge (g^1_{i\,j} \vee \neg t^a_{i\,j} \vee t_{f(i)\,j}) \wedge$ $(\neg g^1_{i\,j} \vee t^a_{i\,j} \vee t_{f(i)\,j}) \wedge (\neg g^1_{i\,j} \vee \neg t^a_{i\,j} \vee \neg t_{f(i)\,j})$
$g_{i\,j} =? \wedge g_{f(i)\,j} =?$	$(g^1_{i\,j} \vee t^a_{i\,j} \vee \neg t^a_{f(i)\,j}) \wedge (g^1_{i\,j} \vee \neg t^a_{i\,j} \vee t^a_{f(i)\,j}) \wedge$ $(\neg g^1_{i\,j} \vee t^a_{i\,j} \vee \neg t^b_{f(i)\,j}) \wedge (\neg g^1_{i\,j} \vee \neg t^a_{i\,j} \vee \neg t^b_{f(i)\,j})$

variables with homozygous sites. The new definition of variables associated with sites requires the redefinition of the constraints related with Mendelian laws. For instance, consider the first constraint of Table 2, $t_{f(i)\,j} \Leftrightarrow g^1_{i\,j}$, for the case $g_{i\,j} = 0$ and $g_{f(i)\,j} = 2$. Clearly, if $t_{f(i)\,j} = 1$ (representing that individual $f(i)$ has inherited value 1 from his father and value 0 from his mother) then $g^1_{i\,j} = 1$ (representing that individual i must have inherited the value 0 from his paternal grandmother) and conversely.

In addition, in order to allow counting the number of recombinations, the model defines new variables r. For each non-founder individual i, variable $r^1_{i\,j}$ ($r^2_{i\,j}$) is assigned value 1 if a recombination took place at site j, to create the paternal (maternal) haplotype of individual i. Thus, $r^l_{i\,j} = 1$ if $g^l_{i\,j} \neq g^l_{i\,j+1}$, for $l \in \{1, 2\}$ and $1 \leq j \leq m - 1$, which is ensured by constraints (1) in Table 1. Here, another simplification to the original MRHC is considered. Actually, in the PedPhase model, $r^l_{i\,j} = 1$ *if and only if* $g^l_{i\,j} \neq g^l_{i\,j+1}$. Observe that an implication, instead of an equivalence, is sufficient for correctness and reduces in half the number of these constraints.

Moreover, the model defines variables to count the number of distinct haplotypes used. Let $x^{p\,q}_{i\,k}$, with $p, q \in \{a, b\}$ and $1 \leq k < i \leq n$, be 1 if haplotype p of genotype g_i (h^p_i) and haplotype q of genotype g_k (h^q_k) are different. The conditions on the $x^{p\,q}_{i\,k}$ variables are based on the values of variables $t_{i\,j}$ and $t_{k\,j}$ for heterozygous sites and of variables $t^a_{i\,j}$, $t^b_{i\,j}$, $t^a_{k\,j}$ and $t^b_{k\,j}$ for missing sites, and are described by equations (2) in Table 1.

Table 3. Definition of predicates R and S, accordingly to index values.

Condition	Constraint
$g_{i\,j} \neq 2 \wedge g_{k\,j} = 2$	$R = (g_{i\,j} \Leftrightarrow (q \Leftrightarrow a))$ and $S = t_{k\,j}$
$g_{k\,j} \neq 2 \wedge g_{i\,j} = 2$	$R = (g_{k\,j} \Leftrightarrow (p \Leftrightarrow a))$ and $S = t_{i\,j}$
$g_{i\,j} = 2 \wedge g_{k\,j} = 2$	$R = (p \Leftrightarrow q)$ and $S = (t_{i\,j} \Leftrightarrow t_{k\,j})$
$g_{i\,j} =? \wedge g_{k\,j} \notin \{2, ?\}$	$R = t^p_{i\,j}$ and $S = g_{k\,j}$
$g_{k\,j} =? \wedge g_{i\,j} \notin \{2, ?\}$	$R = t^q_{k\,j}$ and $S = g_{i\,j}$
$g_{i\,j} =? \wedge g_{k\,j} = 2$	$R = (q \Leftrightarrow a)$ and $S = (t^p_{i\,j} \Leftrightarrow t_{k\,j})$
$g_{k\,j} =? \wedge g_{i\,j} = 2$	$R = (p \Leftrightarrow a)$ and $S = (t^q_{k\,j} \Leftrightarrow t_{i\,j})$
$g_{i\,j} =? \wedge g_{k\,j} =?$	$R = t^p_{i\,j}$ and $S = t^q_{k\,j}$

Furthermore, the model needs variables u to denote when one of the haplotypes, associated with a given genotype, is different from all previous haplotypes. Hence, u^p_i, with $p \in \{a, b\}$ and $1 \leq i \leq n$, is 1 if haplotype p of genotype g_i is different from all previous haplotypes. Then, the conditions on the u^p_i variables are based on the conditions for the $x^{p\,q}_{i\,k}$ variables, with $1 \leq k < i$ and $q \in \{a, b\}$. These conditions are described by equations (3) in Table 1.

Finally, the cost function consists in minimizing the number of recombination events and the number of distinct haplotypes, which are, respectively, given by the sum of variables r and u,

$$\textit{minimize} \quad ((2n+1) \times \sum_{(\text{non-founder } i)} \sum_{j=1}^{m-1} (r^1_{i\,j} + r^2_{i\,j})) + \sum_{i=1}^{n} (u^a_i + u^b_i). \tag{1}$$

Given that higher importance is given to the minimum recombinant criterion, a larger weight is given to the number of recombinations. Note that $2n$ is a trivial upper bound on the number of haplotypes in the solution, and therefore giving weight $2n + 1$ to the number of recombinations implies that a MRHC solution is always preferred. The idea of giving more weight to the number of recombinations is biological motivated by the fact that recombination events within haplotypes in a pedigree are rare. Moreover, note that a larger number of recombinants suggests a larger number of haplotypes. In general, a recombination event generates a new haplotype, whereas without recombination, the haplotypes of the child are exact copies of the parents' haplotypes. Nonetheless, different weights w, $1 < w < 2n + 1$, were also tried but did not lead to improvements neither on accuracy or efficiency.

Finally, we would like to point out that a two-step approach which obtains all MRHC solutions first and then picks the solution with the smallest number of haplotypes would not be practical. The number of MRHC solutions is, in general, significantly large, specially with higher missing rates, and usually it is not feasible to compute all solutions. In addition, note that the number of all MRHC solutions is the product of the number of MRHC solutions for each pedigree. On the other hand, the minimum recombinant maximum parsimony criterion reduces the search space and results confirm that it produces more accurate results.

4 Improving Efficiency

This section describes three improvements on the original PedRPoly model, which contribute for an efficient haplotype inference model. The practical contribution of each technique is detailed in Section 5.1.

4.1 Lower Bounds

The integration of lower bounds is a modeling technique implemented previously in other approaches [26, 11]. The algorithms for computing lower bounds rely on information regarding (in)compatible genotypes. Two genotypes are declared *compatible* if does *not* exist a site for which one genotype has value 0 and the other genotype has value 1. Otherwise, the genotypes are *incompatible.* Clearly, two incompatible genotypes cannot be explained by the same haplotypes. Given the incompatibility relation we can create an *incompatibility* graph I, where each vertex is a genotype, and two vertexes are linked with an edge if they are incompatible. Suppose I has a clique of size k. Hence, the number of required haplotypes is at least $2 \cdot k - \sigma$, where σ is the number of genotypes in the clique which do not have heterozygous sites.

In addition, an analysis of the structure of the genotypes allows the lower bound to be further increased. The objective of the new procedure is to identify heterozygous sites which require at least one additional haplotype given a set of previously chosen genotypes. For each genotype g not in the clique, if the genotype has a heterozygous site and all compatible genotypes have the same value at that site (either 0 or 1), then g is guaranteed to require one additional haplotype to be explained. Hence the lower bound can be increased by 1.

Therefore, the lower bound procedure provides a list of genotypes with an indication of the contribution of each genotype to the lower bound. Each genotype either contributes with $+2$, indicating that 2 new haplotypes will be required for explaining this genotype, or with $+1$, indicating that 1 new haplotype will be required.

This technique has been included in the PedRPoly model. In practice, the implementation of lower bounds allows the variables u associated with haplotypes affected by the lower bound to be fixed and, consequently, the clauses used for constraining the value *need not* to be generated. Indeed, if g_i is a genotype contributing with $+2$ to the lower bound, then $u_i^a = 1$ and $u_i^b = 1$. Moreover, if g_i is a genotype contributing with $+1$ to the lower bound, then either u_i^a or u_i^b can be assigned 1. The new model with integration of lower bounds will be named *PedRPoly-LB*.

4.2 Sorting Genotypes

The order in which the genotypes are organized, before the model is generated, can have an important impact on the efficiency of the solver. In particular, note that variables u designate whether a haplotype associated with a genotype is different from all previous haplotypes. We used as an heuristic the lexicographic order on the genotypes, defined by a total order on the genotype sites where $0 < 1 < 2 <\ ?$, i.e.

$$g_{i\,j} < g_{l\,j} \wedge (\forall_{\{k:\ k<j\}}\ g_{i\,k} = g_{l\,k}) \Rightarrow i < l. \qquad (2)$$

The new model, which integrates lower bounds and where the genotypes are sorted according to the lexicographic order is named *PedRPoly-LB-Ord*.

4.3 Symmetries

Symmetry breaking is a well-known technique for pruning the search space and, therefore, contributing to the efficiency of a model. Note that, in general, in the haplotype inference problem, if a genotype g is explained by haplotype pair (h^a, h^b), then g is also explained by haplotype pair (h^b, h^a). Within pedigrees, this symmetry on pairs of haplotypes does not exist for every individual. For non-founders, symmetry is already broken by imposing that the first haplotype comes from the father and the second haplotype comes from the mother. Nonetheless, the symmetry can be broken on founders. This symmetry is broken by introducing a new constraint for each heterozygous founder, imposing that the first heterozygous site $g_{i\,j}$ is explained with $h^a_{i\,j} = 1$ and $h^b_{i\,j} = 0$, i.e.

$$g_{i\,j} = 2 \wedge (\forall_{\{k:\ k<j\}}\ g_{i\,k} \neq 2) \Rightarrow t_{i\,j} = 1. \qquad (3)$$

The new model which includes breaking symmetry on founders is named *PedRPoly-LB-Ord-Sym*.

5 Experimental Evaluation

This section has a threefold purpose. First, it illustrates the contribution of each technique described in Section 4 to the efficiency of PedRPoly. Second, it presents the results obtained using a number of constraint optimization solvers to solve the PedRPoly model, enabling the user not only to choose the best constraint solver for PedRPoly, but also indicating which solvers are more appropriate for solving Boolean constraint problems with multiple cost functions. Finally, it tests the accuracy of the PedRPoly model against the accuracy of the PedPhase approach.

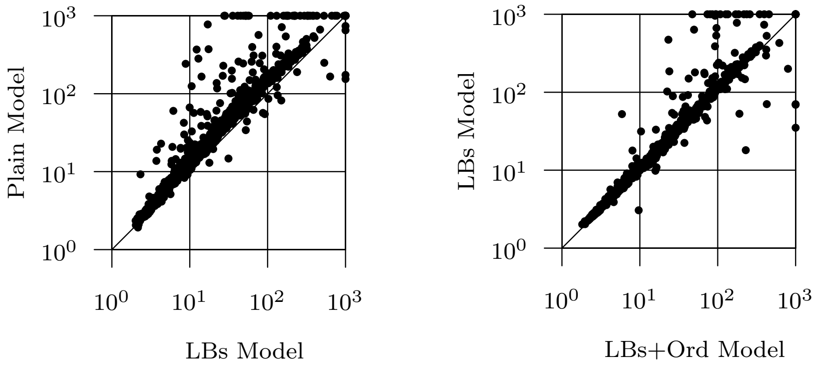

Fig. 2. CPU time comparison between models: plain PedRPoly model vs PedRPoly-LB model and PedRPoly-LB model vs PedRPoly-LB-Ord model

Experimental Data The experimental data was simulated using the SimPed software [17]. SimPed generates haplotypes for families, given the pedigree structure, as well as the haplotypes and their frequencies for founders. The haplotypes for founders and their frequencies were obtained from 7 real data sets of experimentally identified haplotypes [2, 29], and correspond to the A-G data sets already used in other haplotyping studies [6]. The number of SNPs range from 5 to 47. Note that haplotyping regions with tens of SNPs are still relevant in several association studies. Moreover, larger regions can always be partitioned into small blocks [36].

In addition, the same three pedigree structures used by PedPhase [20] were considered: pedigree 1 with 15 individuals, pedigree 2 with 29 individuals and pedigree 3 with 17 individuals. Pedigree 3 contains a mating loop, which means that two mating individuals have a common ancestor in the pedigree. Each simulated instance consists of 10 replicates of the given pedigree, simulating 10 different families from the same population. Hence, the number of genotypes per instance may be 150, 290 or 170. Recombination events are uniformly distributed between alleles with probabilities 0.1%, 0.5% and 1%. Three variations on missing rates were considered: 1%, 10% and 20%. For each combination of parameters, 5 independent replicates were selected, resulting in a total of 945 $(= 7 \times 3^3 \times 5)$ input trials.

Genotyping errors have not been simulated. Nonetheless, genotype errors do not represent a significant limitation because they can be minimized by previously applying an appropriate error detection software [32].

Experimental Setup All results were obtained on a Intel Xeon 5160 server (3.0GHz, 1333Mhz, 4GB) running Red Hat Enterprise Linux WS4. PedPhase ILP was run

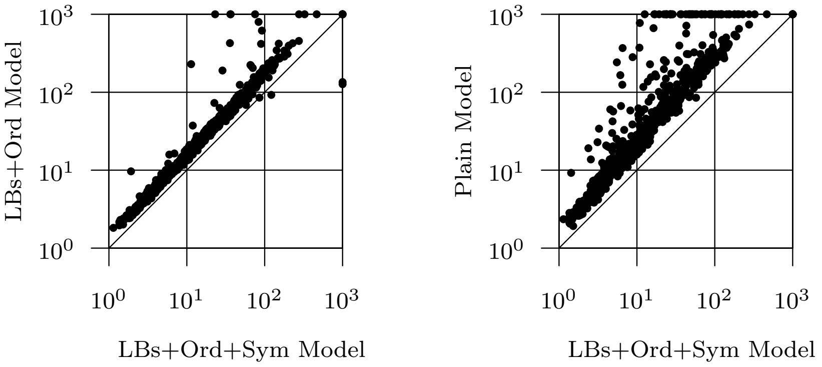

Fig. 3. CPU time comparison between models: PedRPoly-LB-Ord model vs PedRPoly-LB-Ord-Sym model and plain PedRPoly model vs PedRPoly-LB-Ord-Sym model

on Windows because this software is not available for Linux. Results are presented for a timeout of 1000 seconds and a memory limit of 3.5 GB.

5.1 Efficiency

This section studies the contribution of each modeling technique to improving the efficiency of PedRPoly. Furthermore, a significant number of constraint optimization solvers is tested. The use of an appropriate optimization solver with the model contributes for an efficient haplotype inference solver. In what follows, we used PedRPoly with the Boolean multilevel optimization (BMO) Max-SAT solver provided by the authors [4].

Lower Bounds Figure 2 (left) provides a scatter plot which compares the performance of the plain PedRPoly with PedRPoly implementing the identification of lower bounds, within a timeout of 1000 seconds. Each point in the plot corresponds to a problem instance, where the x-axis corresponds to the CPU time required by PedRPoly-LB and the y-axis corresponds to the CPU time required by the plain PedRPoly. Points in the 10^3 lines represent instances which cannot be solved within 1000 seconds.

PedRPoly-LB reduces in half the number of instances aborted by the plain PedRPoly. The plain model aborts 59 instances while PedRPoly-LB is not able to solve 26 instances. PedRPoly-LB solves 37 instances which the plain PedRPoly aborts, although being able to solve 4 instances which PedRPoly-LB aborts. Moreover, PedRPoly-LB is faster than plain PedRPoly for more than 94% of the problem instances.

Table 4. The PedRPoly model: comparison between models (timeout 1000 sec; memory limit 3.5 GB)

Solver	# Solved inst.	% Solved inst.	Avg run time (sec)
PedRPoly	886/945	93.76%	62.50
PedRPoly-LB	919/945	97.25%	47.30
PedRPoly-LB-Ord	933/945	98.73%	41.64
PedRPoly-LB-Ord-Sym	938/945	99.26%	24.08

Sorting Genotypes Figure 2 (right) compares the performance of PedRPoly-LB with PedRPoly-LB-Ord. PedRPoly-LB-Ord does not solve 12 instances but is able to solve 17 instances which PedRPoly-LB aborts. However, there are 3 instances which PedRPoly-LB solves and PedRPoly-LB-Ord is not able to solve. Moreover, PedRPoly-LB-Ord is faster than PedRPoly-LB for 86% of the instances.

Symmetries Figure 3 (left) compares the performance of PedRPoly-LB-Ord with PedRPoly-LB-Ord-Sym. The final model is able to solve 938 out of 945 instances. PedRPoly-LB-Ord-Sym solves 7 instances which PedRPoly-LB-Ord aborts and aborts 2 instances which PedRPoly-LB-Ord solves. Moreover, PedRPoly-LB-Ord-Sym is faster than PedRPoly-LB-Ord for 99% of the instances.

Moreover, figure 3 (right) compares the performance of plain PedRPoly and PedRPoly-LB-Ord-Sym. The later is faster than the former for *all* instances, and solves 52 instances which the plain model aborts. These facts illustrate the importance of the improved model in the efficiency of PedRPoly.

Table 4 summarizes the improvement achieved by combining modeling techniques. Overall, PedRPoly-LB-Ord-Sym outperforms all other models, being capable of solving 99.26% of the instances within 1000 seconds, and using an average run time of 24 seconds. In the remainder of the paper, PedRPoly-LB-Ord-Sym will be denoted simply by PedRPoly.

Solvers A key issue for the efficiency of the haplotype inference solver is to select an adequate underlying optimization solver. In this subsection, 8 different optimization solvers were tested for solving the final version of the PedRPoly model. Integer linear programming, pseudo-Boolean optimization and also weighted Max-SAT solvers were considered. Scip [1] (version 1.2.0) combines constraint programming and mixed integer programming methodologies. Cplex (version 12.1) is an IBM/ILOG commercial linear programming optimization tool. Weighted Max-SAT solvers were also tested: MaxSat_bmo [4], WPM1 [3], WMaxSatz [18] (version 2.5), and IncWMaxSatz [23]. MiniSat+ [7] and Bsolo [27] (version 3.5) are pseudo-Boolean optimization solvers, also known as 0-1 ILP solvers.

Table 5. The PedRPoly model: comparison using different solvers (timeout 1000 sec; memory limit 3.5 GB)

Solver	# Solved inst.	% Solved inst.	Avg run time (sec)
MaxSat_bmo	938/945	99.26%	24.08
WPM1	911/945	96.40%	14.75
Cplex	553/945	58.52%	84.73
Scip	455/945	48.15%	139.07
MiniSat+	260/945	27.51%	238.45
IncWMaxSatz	221/945	23.39%	71.04
Bsolo	160/945	16.93%	170.26
WMaxSatz	16/945	1.69%	113.16

Table 5 summarizes the performance of the different solvers. Clearly, the solver which is able to solve a larger number of instances is MaxSat_bmo, which solves 99.26% of the instances.

The second best performing solver is WPM1 which solves 96.40% of the instances. The third and fourth best performing solvers are the integer programming solvers. Cplex solves 58.52% and Scip solves 48.15% of the instances, followed by MiniSat+ which solves 27.51% and IncWMaxSatz which solves 23.39% of the problem instances. Bsolo solves 16.93% and WMaxSatz solves 1.69% of the instances. Most of the instances aborted by IncWMaxSatz and WMaxSatz were due to limitations in the internal data structures used by these solvers.

5.2 Accuracy

This section analyzes the gains in accuracy of PedRPoly, that integrates both HIPP with MRHC, with PedPhase [20], which uses only the MRHC approach. Two different commonly used error rates were considered. The *switch error rate* measures the percentage of possible switches in haplotype orientation, used to recover the correct phase in an individual [24]. Missing alleles are not considered for computing the switch error. The *missing error rate* (or *genotype inference error rate*) is the percentage of incorrectly inferred missing data [28].

Note that instances for which at least one of the solvers is unable to give a solution have been removed from the comparison. PedPhase is able to solve 99.8% of the instances, whereas and PedRPoly is able to solve 99.3%. As a result, 9 out of 945 instances have been left out.

Figure 4 presents a bar graph comparing the switch error rate of PedRPoly with the switch error rate of PedPhase. Results have been organized by parameter value: missing rate, recombination rate and pedigree. Each value is the average of the error rate for the instances generated with the corresponding parameter value. PedRPoly is more accurate than PedPhase for 67.09% of the instances. The two solvers have equal error rates for 19.55% of the instances. For 13.35% of the instances, PedRPoly is less accurate than PedPhase.

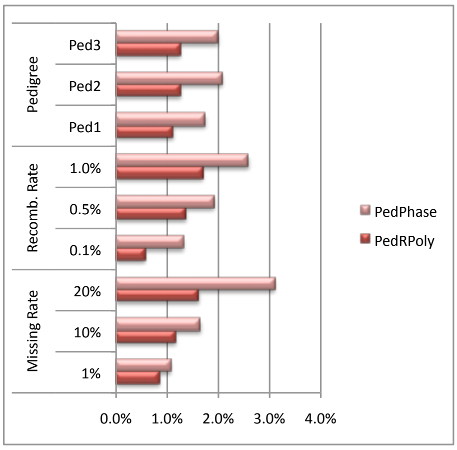

Fig. 4. Switch error: comparing PedRPoly and PedPhase

Figure 5 presents a bar graph for evaluating the missing error rate of the two tools. PedRPoly is more accurate than PedPhase for 73.93% of the instances. The two solvers have equal error rate for 12.39% of the instances. For 13.68% of the instances, PedRPoly is less accurate than PedPhase. Indeed, the population information included by the PedRPoly model is shown to be particularly important for inferring missing genotypes. Overall, we can conclude that PedRPoly consistently outperforms PedPhase in terms of accuracy.

Although the goals of the paper are to show that MRHC combined with HIPP has better accuracy than MRHC alone, and the development of optimizations to the plain model, two distinct statistical methods for haplotype inference within pedigrees were also evaluated. The methods evaluated are Superlink [8] and PhyloPed [15], and both exhibit error rates higher than PedRPoly.

Finally, the number of distinct haplotypes in the PedRPoly solution and in the PedPhase solution was compared with the number of haplotypes used in the real solution. The number of haplotypes in the PedRPoly solution is exactly the same as in the real solution for 60% of the instances, and for more than 99.6% the number of haplotypes in the PedRPoly solution differs from the number of haplotypes in the real solution by less than 5 haplotypes. PedPhase solutions, however, are less similar to the real solutions with respect to the number of haplotypes. For the same set of instances, PedPhase has the same number of haplotypes as the real solution for 12.3% of the instances, and differs from the real solution by less than 5 haplotypes for 45.8% of the instances.

6 Conclusions

This paper addresses the problem of haplotype inference from pedigrees, and proposes a new Boolean optimization model for haplotype inference which com-

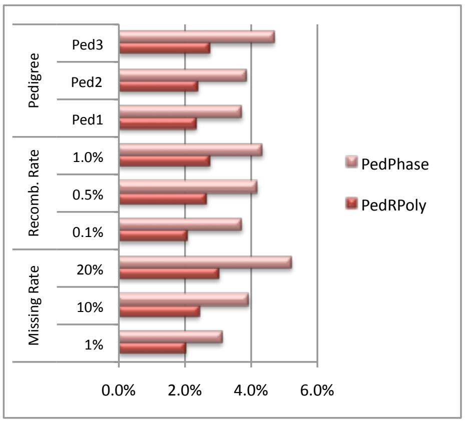

Fig. 5. Missing error: comparing PedRPoly and PedPhase

bines the pure parsimony approach with the minimum recombinant approach. Moroever, the paper details the integration of well-known modeling techniques exploited in order to improve the performance of the method. These techniques include integration of lower bounds, ordering heuristics and symmetry breaking, as well as the selection of an appropriate constraint solver. The new PedRPoly approach was tested on a set of instances of considerable dimension. Experimental results show that the new approach is both accurate and efficient when compared to other methods.

The problem instances generated by the combined model represent challenging combinatorial optimization problems, related with multi-objective optimization. A number of techniques was suggested to improve the performance for this class of problem instances. Future research will address further optimizations to the model, aiming at improving both accuracy and efficiency. The topic of improved efficiency will also involve the development of solvers capable of solving Boolean-based multi-objective optimization problems.

Acknowledgments. This work is partially supported by Fundação para a Ciência e Tecnologia under research project SHIPs (PTDC/EIA/64164/2006) and PhD grant (SFRH/BD/28599/2006), and by Microsoft under contract 2007-017 of the Microsoft Research PhD Scholarship Programme.

References

[1] T. Achterberg, T. Berthold, T. Koch, and K. Wolter. Constraint Integer Programming: a new approach to integrate CP and MIP. In *International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Problems (CPAIOR'08)*, pages 6–20, 2008.

[2] A. Andrés, A. Clark, L. Shimmin, E. Boerwinkle, C. Sing, and J. Hixson. Understanding the accuracy of statistical haplotype inference with sequence data of known phase. *Genetic Epidemiology*, 31(7):659–671, 2007.

[3] C. Ansótegui, M. L. Bonet, and J. Levy. Solving (weighted) partial MaxSAT through satisfiability testing. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'09)*, pages 427–440, 2009.

[4] J. Argelich, I. Lynce, and J. Marques-Silva. On solving Boolean multilevel optimization problems. In *International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 393–398, 2009.

[5] I. Cheng, K. L. Penney, D. O. Stram, L. Le Marchand, E. Giorgi, C. A. Haiman, L. N. Kolonel, M. Pike, J. Hirschhorn, B. E. Henderson, and M. L. Freedman. Haplotype-based association studies of IGFBP1 and IGFBP3 with prostate and breast cancer risk: the multiethnic cohort. *Cancer Epidemiol Biomarkers Prev*, 15(10):1993–1997, 2006.

[6] S. Climer, G. Jäger, A. R. Templeton, and W. Zhang. How frugal is mother nature with haplotypes? *Bioinformatics*, 25(1):68–74, 2009.

[7] N. Eén and N. Sörensson. Translating pseudo-Boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:1–26, 2006.

[8] M. Fishelson, N. Dovgolevsky, and D. Geiger. Maximum likelihood haplotyping for general pedigrees. *Human Heredity*, 59(1):41–60, 2005.

[9] A. Graça, I. Lynce, J. Marques-Silva, and A. Oliveira. Haplotype inference combining pedigrees and unrelated individuals. In *Workshop on Constraint Based Methods for Bioinformatics (WCB'09)*, pages 27–36, 2009.

[10] A. Graça, J. Marques-Silva, I. Lynce, and A. Oliveira. Efficient haplotype inference with pseudo-Boolean optimization. In *Algebraic Biology (AB'07)*, pages 125–139, 2007.

[11] A. Graça, J. Marques-Silva, I. Lynce, and 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 Problems (CPAIOR'08)*, pages 308–312, 2008.

[12] D. Gusfield. Haplotype inference by pure parsimony. In *Symposium on Combinatorial Pattern Matching (CPM'03)*, pages 144–155, 2003.

[13] J. L. Haines. Chromlook: an interactive program for error detection and mapping in reference linkage data. *Genomics*, 14(2):517–519, 1992.

[14] M. Kimura. The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. *Genetics*, 61(4), 1969.

[15] B. Kirkpatrick, J. Rosa, E. Halperin, and R. M. Karp. Haplotype inference in complex pedigrees. In *International Conference on Research in Computational Molecular Biology (RECOMB'09)*, pages 108–120, 2009.

[16] G. Lancia, C. M. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony: complexity of exact and approximation algorithms. *INFORMS Journal on Computing*, 16(4):348–359, 2004.

[17] S. M. Leal, K. Yan, and B. Müller-Myhsok. SimPed: A simulation program to generate haplotype and genotype data for pedigree structures. *Human Heredity*, 60(2):119–122, 2005.

[18] C. M. Li, F. Manyà, N. O. Mohamedou, and J. Planes. Exploiting cycle structures in Max-SAT. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'09)*, pages 467–480, 2009.

[19] J. Li and T. Jiang. Efficient inference of haplotypes from genotypes on a pedigree. *Journal of Bioinformatics and Computational Biology*, 1(1):41–69, 2003.

[20] J. Li and T. Jiang. Computing the minimum recombinant haplotype configuration from incomplete genotype data on a pedigree by integer linear programming. *Journal of Computational Biology*, 12(6):719–739, 2005.

[21] X. Li and J. Li. Comparison of haplotyping methods using families and unrelated individuals on simulated rheumatoid arthritis data. In *BMC Proceedings*, pages S1–S55, 2007.

[22] X. Li and J. Li. Efficient haplotype inference from pedigree with missing data using linear systems with disjoint-set data structures. In *International Conference on Computational Systems Bioinformatics (CSB'08)*, pages 297–307, 2008.

[23] H. Lin, K. Su, and C. M. Li. Within-problem learning for efficient lower bound computation in Max-SAT solving. In *National Conference on Artificial Intelligence (AAAI'08)*, pages 351–356, 2008.

[24] S. Lin, A. Chakravarti, and D. J. Cutler. Haplotype and missing data inference in nuclear families. *Genome Research*, 14(8):1624–1632, 2004.

[25] L. Liu, C. Xi, J. Xiao, and T. Jiang. Complexity and approximation of the minimum recombinant haplotype configuration problem. *Theoretical Computer Science*, 378(3):316–330, 2007.

[26] I. Lynce, J. Marques-Silva, and S. Prestwich. Boosting haplotype inference with local search. *Constraints*, 13(1):155–179, 2008.

[27] V. Manquinho and J. Marques-Silva. Effective lower bounding techniques for pseudo-Boolean optimization. In *Design, Automation and Test in Europe Conference and Exhibition (DATE'05)*, pages 660–665, 2005.

[28] J. Marchini, D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z.S. Qin, H.M Munro, G.R. Abecassis, P. Donnelly, and International HapMap Consortium. A comparison of phasing algorithms for trios and unrelated individuals. *American Journal of Human Genetics*, 78(3):437–450, 2006.

[29] S. H. Orzack, D. Gusfield, J. Olson, S. Nesbitt, L. Subrahmanyan, and V. P. Stanton. Analysis and exploration of the use of rule-based algorithms and consensus methods for the inferral of haplotypes. *Genetics*, 165(2):915–928, 2003.

[30] Y. Pei, L. Zhang J. Li, C. J. Papasian, and H.-W. Deng. Analyses and comparison of accuracy of different genotype imputation methods. *PLoS ONE*, 3(10), 2008.

[31] D. Qian and L. Beckmann. Minimum-recombinant haplotyping in pedigrees. *American Journal of Human Genetics*, 70(6):1434–1445, 2002.

[32] M. Sánchez, S. Givry, and T. Schiex. Mendelian error detection in complex pedigrees using weighted constraint satisfaction techniques. *Constraints*, 13(1-2):130–154, 2008.

[33] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449:851–861, 2007.

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

[35] E. M. Wijsman. A deductive method of haplotype analysis in pedigrees. *American Journal of Human Genetics*, 41(3):356–373, 1987.

[36] K. Zhang, Z. Qin, T. Chen, J. S. Liu, M. S. Waterman, and F. Sun. HapBlock: haplotype block partitioning and tag SNP selection software using a set of dynamic programming algorithms. *Bioinformatics*, 21(1):131–134, 2005.

[37] K. Zhang, F. Sun, and H. Zhao. HAPLORE: a program for haplotype reconstruction in general pedigrees without recombination. *Bioinformatics*, 21(1):90–103, 2005.