# Efficient and Tight Upper Bounds for Haplotype Inference by Pure Parsimony using Delayed Haplotype Selection

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

[1] School of Electronics and Computer Science, University of Southampton, UK
jpms@ecs.soton.ac.uk
[2] IST/INESC-ID, Technical University of Lisbon, Portugal
{ines,assg}@sat.inesc-id.pt, aml@inesc-id.pt

**Abstract.** Haplotype inference from genotype data is a key step towards a better understanding of the role played by genetic variations on inherited diseases. One of the most promising approaches uses the pure parsimony criterion. This approach is called Haplotype Inference by Pure Parsimony (HIPP) and is NP-hard as it aims at minimising the number of haplotypes required to explain a given set of genotypes. The HIPP problem is often solved using constraint satisfaction techniques, for which the upper bound on the number of required haplotypes is a key issue. Another very well-known approach is Clark's method, which resolves genotypes by greedily selecting an explaining pair of haplotypes. In this work, we combine the basic idea of Clark's method with a more sophisticated method for the selection of explaining haplotypes, in order to explicitly introduce a bias towards parsimonious explanations. This new algorithm can be used either to obtain an approximated solution to the HIPP problem or to obtain an upper bound on the size of the pure parsimony solution. This upper bound can then used to efficiently encode the problem as a constraint satisfaction problem. The experimental evaluation, conducted using a large set of real and artificially generated examples, shows that the new method is much more effective than Clark's method at obtaining parsimonious solutions, while keeping the advantages of simplicity and speed of Clark's method.

## 1 Introduction

Over the last few years, an emphasis in human genomics has been on identifying genetic variations among different people. A comprehensive search for genetic influences on disease involves examining all genetic differences in a large number of affected individuals. This allows the systematic test of common genetic variants for their role in disease. These variants explain much of the genetic diversity in our species, a consequence of the historically small size and shared ancestry of the human population. One significant effort in this direction is represented by the HapMap Project[23], a project that aims at developing a haplotype map of the human genome and represents the best known effort to develop a public resource that will help finding genetic variants associated with specific human diseases.

For a number of reasons, these studies have focused on the tracking of the inheritance of Single Nucleotide Polymorphisms (SNPs), point mutations found with only two common values in the population. This process is made more difficult because of technological limitations. Current methods can directly sequence only short lengths of DNA at a time. Since the sequences of the chromosomes inherited from the parents are very similar over long stretches of DNA, it is not possible to reconstruct accurately the sequence of each chromosome. Therefore, at a genomic site for which an individual inherited two different values, it is currently difficult to identify from which parent each value was inherited. Instead, currently available sequencing methods can only determine that the individual is ambiguous at that site.

Most diseases are due to very complex processes, where the values of many SNPs affect, directly and indirectly, the risk. Due to a phenomenon known as linkage disequilibrium, the values of SNPs in the same chromosome are correlated with each other. This leads to the conservation, through generations, of large haplotype blocks. These blocks have a fundamental role in the risk of any particular individual for a given disease. If we could identify maternal and paternal inheritance precisely, it would be possible to trace the structure of the human population more accurately and improve our ability to map disease genes. This process of going from genotypes (which may be ambiguous at specific sites) to haplotypes (where we know from which parent each SNP is inherited) is called haplotype inference.

This paper introduces a greedy algorithm for the haplotype inference problem called Delayed Haplotype Selection (DS) that extends and improves the well-known Clark's method[5]. We should note that recent work on Clark's method studied a number of variations and improvements, none similar to DS, and all performing similarly to Clark's method. This new algorithm takes advantage of new ideas that have appeared recently, such as those of pure parsimony[10]. A solution to the haplotype inference by pure parsimony (HIPP) problem provides the smallest number of haplotypes required to explain a set of genotypes. This algorithm can then be used in two different ways: (1) as a standalone procedure for giving an approximate solution to the HIPP problem or (2) as an upper bound to the HIPP solution to be subsequently used by pure parsimony algorithms which use upper bounds on their formulation. Experimental results, obtained on a comprehensive set of examples, show that, for the vast majority of the examples, the new approach provides a very accurate approximation to the pure parsimony solution.

This paper is organised as follows. The next section introduces key concepts, describes the problem from a computational point of view, and points to related work, including Clark's method and pure parsimony approaches. Based on Clark's method, section 3 describes a new algorithm called *Delayed Haplotype Selection*. Afterwards section 4 gives the experimental results obtained with the new algorithm, which are compared with other methods and evaluated from the point of view of a parsimonious solution. Finally, section 5 presents the conclusions and points directions for future research work, including the integration of the greedy algorithm in pure parsimonious algorithms.

## 2 Problem Formulation and Related Work

### 2.1 Haplotype Inference

A *haplotype* represents the genetic constitution of an individual chromosome. The underlying data that forms a haplotype is generally viewed as the set of SNPs in a given region of a chromosome. Normal cells of diploid organisms contain two haplotypes, one inherited from each parent. The *genotype* represents the conflated data of the two haplotypes. The value of a particular SNP is usually represented by X, Y or X/Y, depending on whether the organism is homozygous with value X, homozygous with value Y or heterozygous. The particular base that the symbols X and Y represent varies with the SNP in question. For instance, the most common value in a particular location may be the guanine (G) and the less common variation cytosine (C). In this case, X will mean that both parents have guanine in this particular site, Y that both parents have cytosine at this particular site, and X/Y that the parents have different bases at this particular site. Since mutations are relatively rare, the assumption that at a particular site only two bases are possible does not represent a strong restriction. This assumption is supported by the so called *infinite sites model*[14], that states that only one mutation has occurred in each site, for the population of a given species.

Starting from a set of genotypes, the haplotype inference task consists in finding the set of haplotypes that gave rise to that set of genotypes. The variable $n$ denotes the number of individuals in the sample, and $m$ denotes the number of SNP sites. Without loss of generality, we may assume that the two values of each SNP are either 0 or 1. Value 0 represents the wild type and value 1 represents the mutant. A haplotype is then a string over the alphabet $\{0,1\}$. Genotypes may be represented by extending the alphabet used for representing haplotypes to $\{0,1,2\}$. A specific genotype is denoted by $g_i$, with $1 \leq i \leq n$. Furthermore, $g_{ij}$ denotes a specific site $j$ in genotype $g_i$, with $1 \leq j \leq m$. We say that a genotype $g_i$ can be explained by haplotypes $h_k$ and $h_l$ iff for each site $g_{ij}$:

$$g_{ij} = \begin{cases} h_{kj} & \text{if } h_{kj} = h_{lj} \\ 2 & \text{if } h_{kj} \neq h_{lj} \end{cases}$$

In general, if a genotype $g_i$ has $r \geq 1$ heterozygous sites, then there are $2^{r-1}$ pairs that can explain $g_i$. The objective is to find the set $\mathcal{H}$ of haplotypes that is most likely to have originated the set of genotypes in $\mathcal{G}$.

**Definition 1.** *(Haplotype Inference) Given a set $\mathcal{G}$ of $n$ genotypes, each of length $m$, the haplotype inference problem consists in finding a set $\mathcal{H}$ of $2 \cdot n$ haplotypes, not necessarily different, such that for each genotype $g_i \in \mathcal{G}$ there is at least one pair of haplotypes $(h_k, h_l)$, with $h_k$ and $h_l \in \mathcal{H}$ such that the pair $(h_k, h_l)$ explains $g_i$.*

*Example 1.* (Haplotype Inference) Consider genotype 02122 having 5 SNPs, of which 1 SNP is homozygous with value 0, 1 SNP is homozygous with value 1, and the 3 remaining SNPs are heterozygous (thus having value 2). Genotype 02122 may then be explained by four different pairs of haplotypes: (00100, 01111), (01100, 00111), (00110, 01101) and (01110, 00101).

We may distinguish between a number of approaches that are usually used for solving the haplotype inference problem: the statistical, the heuristic and the combinatorial approaches. The statistical approaches[19, 22] use specific assumptions about the underlying haplotype distribution to approximate different genetic models, and may obtain highly accurate results. The heuristic approaches include, among others, Clark's method[5]. Finally, most combinatorial approaches are based on the pure parsimony criterion[10]. The later has shown to be one of the most promising alternative approaches to statistical models[3, 17].

### 2.2 Clark's Method

Clark's method is a well-known algorithm that has been proposed to solve the haplotype inference problem[5]. Clark's algorithm has been widely used and is still useful today. This method considers both haplotypes and genotypes as vectors. The method starts by identifying genotype vectors with zero or one ambiguous sites. These vectors can be resolved in only one way, and they define the initially resolved haplotypes. Then, the method attempts to resolve the remaining genotypes by starting with the resolved haplotypes. The following step infers a new resolved vector *NR* from an ambiguous vector A and an already resolved genotype vector R.

Suppose A is an ambiguous genotype vector with $r$ ambiguous sites and R is a resolved vector that is a haplotype in one of the $2^{r-1}$ potential resolutions of vector A. Then the method infers that A is the conflation of the resolved vector R and another unique vector NR. All of the ambiguous positions in A are set in NR to the opposite value of the position in R. Once inferred, this vector is added to the set of known resolved vectors, and vector A is removed from the set of unresolved vectors.

The key point to note is that there are many ways to apply the resolution rule, since for an ambiguous vector A there may be many choices for vector R. A wrong choice may lead to different solutions, or even leave orphan vectors, in the future, i.e., vectors that cannot be resolved with any already resolved vector R.

The Maximum Resolution (MR) problem[9] aims at finding the solution of the Clark's algorithm with the fewest orphans, i.e. with the maximum number of genotypes resolved. This problem is NP-hard as shown by Gusfield[9], who also proposed an integer linear programming approach to the MR problem.

### 2.3 Pure Parsimony

Chromosomes in the child genome are formed by combination of the corresponding chromosomes from the parents. Long stretches of DNA are copied from each parent, spliced together at recombination points. Since recombination is relatively infrequent, large segments of DNA are passed intact from parent to child. This leads to the well known fact that the actual number of haplotypes in a given population is much smaller than the number of observed different genotypes. The haplotype inference by pure parsimony approach was proposed by Hubbel but only described by Gusfield[9].

**Definition 2.** *(Haplotype Inference by Pure Parsimony) The haplotype inference by pure parsimony (HIPP) approach aims at finding a solution for the haplotype infer-*

*ence problem that minimises the total number of distinct haplotypes used. The problem of finding such a parsimonious solution is APX-hard (and, therefore, NP-hard)[16].*

*Example 2.* (Haplotype Inference by Pure Parsimony) Consider the following example, taken from a recent survey on the topic[11], where the set of genotypes is: 02120, 22110, and 20120. There are solutions that use five different haplotypes[1], but the solution (00100, 01110), (01110, 10110), (00100, 10110) uses only three different haplotypes.

It is known that the most accurate solutions based on Clark's method are those that infer a small number of distinct haplotypes[10, 20]. Although Clark's method has sometimes been described as using the pure parsimony criterion[19, 1, 22], this criterion is not explicitly used and an arbitrary choice of the resolving haplotype does not lead to a pure parsimony solution. The present paper proposes a method that, while still based on Clark's method, explicitly uses the pure parsimony criterion, leading to more precise results.

Several approaches, have been proposed to solve the HIPP problem. The first algorithms are based on integer linear programming[10, 2, 24], whereas the most recent and competitive encode the HIPP problem as a constraint satisfaction problem (either using propositional satisfiability[17, 18] or pseudo-Boolean optimization[4]).

One should note that the implementation of exact algorithms for the HIPP problem often requires computing either lower or upper bounds on the value of the HIPP solution[24, 18]. Clearly, Clark's method can be used for providing upper bounds on the solution of the HIPP problem. Besides Clark's method, which is efficient but in general not accurate, existing approaches for computing upper bounds to the HIPP problem require worst-case exponential space, due to the enumeration of candidate pairs of haplotypes[12, 24]. Albeit impractical for large examples, one of these approaches is used in Hapar[24], a fairly competitive HIPP solver when the number of possible haplotype pairs is manageable.

The lack of approaches both accurate and efficient for computing upper bounds, prevented their utilization in recent HIPP solvers, for instance, in SHIPs[18]. Algorithm 1 summarizes the top-level operation of SHIPs. This SAT-based algorithm iteratively determines whether there exists a set $\mathcal{H}$ of distinct haplotypes, with $r = |\mathcal{H}|$ such that each genotype $g \in \mathcal{G}$ is explained by a pair of haplotypes in $\mathcal{H}$. The algorithm considers increasing sizes for $\mathcal{H}$, from a lower bound $lb$ to an upper bound $ub$. Trivial lower and upper bounds are, respectively, $1$ and $2 \cdot n$. For each value of $r$ considered, a CNF formula $\varphi^r$ is created, and a SAT solver is invoked (identified by the function call SAT($\varphi^r$)). The algorithm terminates for a size of $\mathcal{H}$ for which there exist $r = |\mathcal{H}|$ haplotypes such that every genotype in $\mathcal{G}$ is explained by a pair of haplotypes in $\mathcal{H}$, i.e. when the constraint problem is satisfiable. (Observe that an alternative would be to use binary search.)

This paper develops an efficient and accurate approach for haplotype inference, inspired by pure parsimony, and which can be used to compute tight upper bounds

[1] In general, up to $2 \cdot n$ distinct haplotypes may be required to explain $n$ genotypes. However, in this particular case, there is no solution with six distinct haplotypes.

**Algorithm 1** Top-level SHIPs algorithm

```
SHIPs($\mathcal{G}, lb$)
r ← lb
while (true)
    do Generate $\varphi^r$ given $\mathcal{G}$ and $r$
       if SAT($\varphi^r$) = $true$
          then return $r$
          else $r \leftarrow r + 1$
```

to the HIPP problem. Hence, the proposed approach can be integrated in any HIPP approach, including Hapar[24] and SHIPs[18].

## 3 Delayed Haplotype Selection

A key drawback of haplotype inference algorithms based on Clark's method is that these algorithms are often too greedy, at each step seeking to explain *each* non-explained genotype with the most recently chosen haplotype. As a result, given a newly selected haplotype $h_a$, which can explain a genotype $g_t$, a new haplotype $h_b$ is generated that only serves to explain $g_t$. If the objective is to minimize the number of haplotypes, then the selection of $h_b$ may often be inadequate.

This section develops an alternative algorithm which addresses the main drawback of Clark's method. The main motivation is to avoid the excessive greediness of Clark's method in selecting new haplotypes. Therefore a *delayed* greedy algorithm for haplotype *selection* (DS) is used instead.

In contrast to Clark's method, where identified haplotypes are included in the set of chosen haplotypes, the DS algorithm maintains two sets of haplotypes. The first set, the *selected* haplotypes, represents haplotypes which have been chosen to be included in the target solution. A second set, the *candidate* haplotypes, represents haplotypes which can explain one or more genotypes not yet explained by a pair of selected haplotypes.

The initial set of selected haplotypes corresponds to all haplotypes which are required to explain the genotypes with no more than one heterozygous sites, i.e. genotypes which are explained with either one or exactly two haplotypes. Clearly, all these haplotypes must be included in the final solution.

At each step, the DS algorithm chooses the candidate haplotype $h_c$ which can explain the largest number of genotypes. The chosen haplotype $h_c$ is then used to identify additional candidate haplotypes. Moreover, $h_c$ is added to the set of selected haplotypes, and all genotypes which can be explained by a pair of selected haplotypes are removed from the set of unexplained genotypes. The algorithm terminates when all genotypes have been explained.

Each time the set of candidate haplotypes becomes empty, and there are still more genotypes to explain, a new candidate haplotype is generated. The new haplotype is selected greedily as the haplotype which can explain the largest number of genotypes

**Algorithm 2** Delayed Haplotype Selection

```
DelayedHaplotypeSelection($\mathcal{G}$)
 1  ▷ $\mathcal{H}_S$ is the set of selected haplotypes; $\mathcal{H}_C$ is the set of candidate haplotypes
 2  $\mathcal{H}_S \leftarrow$ CalcInitialHaplotypes($\mathcal{G}$)
 3  $\mathcal{G} \leftarrow$ RemoveExplainedGenotypes($\mathcal{G}, \mathcal{H}_S$)
 4  for each $h \in \mathcal{H}_S$
 5        do
 6            for each $g \in \mathcal{G}$
 7                do if CanExplain($h, g$)
 8                      then $h_c \leftarrow$ CalcExplainPair($h, g$)
 9                            $\mathcal{H}_C \leftarrow \mathcal{H}_C \cup \{h_c\}$
10                            Associate $h_c$ with $g$
11  while ($\mathcal{G} \neq \emptyset$)
12        do if ($\mathcal{H}_C = \emptyset$)
13              then
14                    $h_c \leftarrow$ PickCandHaplotype($\mathcal{G}$)
15                    $\mathcal{H}_C \leftarrow \{h_c\}$
16            $h \leftarrow h_c \in \mathcal{H}_C$ associated with largest number of genotypes
17            $\mathcal{H}_C \leftarrow \mathcal{H}_C - \{h\}$
18            $\mathcal{H}_S \leftarrow \mathcal{H}_S \cup \{h\}$
19            $\mathcal{G} \leftarrow$ RemoveExplainedGenotypes($\mathcal{G}, \mathcal{H}_S$)
20            for each $g \in \mathcal{G}$
21                do if CanExplain($h, g$)
22                      then $h_c \leftarrow$ CalcExplainPair($h, g$)
23                            $\mathcal{H}_C \leftarrow \mathcal{H}_C \cup \{h_c\}$
24                            Associate $h_c$ with $g$
25  $\mathcal{H}_S \leftarrow$ RemoveNonUsedHaplotypes($\mathcal{H}_S$)
26  return $\mathcal{H}_S$
```

not yet explained. Clearly, other alternatives could be considered, but the experimental differences, obtained on a large set of examples, were not significant.

Observe that the proposed organization allows selecting haplotypes which will not be used in the final solution. As a result, the last step of the algorithm is to remove from the set of selected haplotypes all haplotypes which are not used for explaining any genotypes.

The overall delayed haplotype selection algorithm is shown in Algorithm 2 and summarizes the ideas outlined above. Line 2 computes the set of haplotypes $\mathcal{H}_S$ associated with genotypes $\mathcal{G}$ with one or zero heterozygous sites, since these haplotypes must be included in the final solution. Line 3 removes from $\mathcal{G}$ all genotypes that can be explained by a pair of haplotypes in $\mathcal{H}_S$. The same holds true for line 19. Lines 6 to 10 and 20 to 24 correspond to the candidate haplotype generation phase, given newly selected haplotypes. The DS algorithm runs in polynomial time in the number of genotypes and sites, a straightforward analysis yielding a run time complexity in $\mathcal{O}(n^2\,m)$.

**Table 1.** Classes of problem instances evaluated

| Class | #Instances | *min*SNPs | *max*SNPs | *min*GENs | *max*GENs |
|---|---|---|---|---|---|
| uniform | 245 | 10 | 100 | 30 | 100 |
| nonuniform | 135 | 10 | 100 | 30 | 100 |
| hapmap | 24 | 30 | 75 | 7 | 68 |
| biological | 450 | 13 | 103 | 5 | 50 |
| Total | 854 | 10 | 103 | 5 | 100 |

In practice, the delayed haplotype selection algorithm is executed multiple times, as in other recent implementations of Clark's method[20]. At each step, ties in picking the next candidate haplotype (see line 16) are randomly broken. The run producing the smallest number of haplotypes is selected.

Results in the next section suggest that delayed haplotype selection is a very effective approach. Nonetheless, it is straightforward to conclude that there are instances for which delayed haplotype selection will yield the same solution as Clark's method. In fact, it is possible for DS to yield solutions with more haplotypes than Clark's method. The results in the next section show that this happens very rarely. Indeed, for most examples considered (out of a comprehensive set of examples) DS is extremely unlikely to compute a larger number of haplotypes than Clark's method, and most often computes solutions with a significantly smaller number of haplotypes.

## 4 Experimental Results

This section compares the delayed haplotype selection (DS) algorithm described in the previous section with a recent implementation of Clark's method (CM)[8]. In addition, the section also compares the HIPP solutions, computed with a recent tool[18], with the results of DS and CM. As motivated earlier, the objectives of the DS algorithm are twofold: first to replace Clark's method as an approximation of the HIPP solution, and second to provide tight upper bounds to HIPP algorithms.

Recent HIPP algorithms are iterative[18], at each step solving a Boolean Satisfiability problem instance. The objective of using tight upper bounds is to reduce the number of iterations of these algorithms. As a result, the main focus of this section is to analyze the absolute difference, in the number of haplotypes, between the computed upper bound and the HIPP solution.

### 4.1 Experimental setup

The instances used for evaluating the two algorithms have been obtained from a number of sources[18], and can be organized into four classes shown in Table 1. For each class, Table 1 gives the number of instances, and the minimum and maximum number of SNPs and genotypes, respectively [2]. The uniform and nonuniform classes of instances

[2] Table 1 shows data for the original non-simplified instances. However, all instances were simplified using well-known techniques[3] before running any of the haplotype inference algorithms.

are the ones used by other authors[3], but extended with additional, more complex, problem instances. The hapmap class of instances is also used by the same other authors[3]. Finally, the instances for the biological class are generated from data publicly available[13, 21, 7, 6, 15]. To the best of our knowledge, this is the most comprehensive set of examples used for evaluating haplotype inference solutions.

All results shown were obtained on a 1.9 GHz AMD Athlon XP with 1GB of RAM running RedHat Linux. The run times of both algorithms (CM and DS) were always a few seconds at most, and no significant differences in run times were observed between CM and DS. As a result, no run time information is included below.

### 4.2 Experimental evaluation

The experimental evaluation of the delayed haplotype selection (DS) algorithm is organized in two parts. The first part compares DS with a publicly available recent implementation of Clark's method (CM)[8], whereas the second part compares DS with an exact solution to the Haplotype Inference by Pure Parsimony (HIPP) problem[18]. In all cases, for both CM and DS, we select the best solution out of 10 runs. Other implementations of Clark's method could have been considered[20]. However, no significant differences were observed among these implementations when the objective is to minimize the number of computed haplotypes.

The results for the first part are shown in Figure 1. The scatter plot shows the difference of CM and of DS with respect to the exact HIPP solution for the examples considered. The results are conclusive. DS is often quite close to the HIPP solution, whereas the difference of CM with respect to the HIPP solution can be significant. While the distance of DS to the HIPP solution never exceeds 16 haplotypes, the distance of CM can exceed 50 haplotypes. Moreover, for a large number of examples, the distance of DS to the HIPP solution is 0, and for the vast majority of the examples the distance does not exceed 5 haplotypes. In contrast, the distance of CM to the HIPP solution often exceeds 10 haplotypes.

The second plot in Figure 1 provides the distribution of the difference between the number of haplotypes computed with DS and with CM. A bar associated with a value $k$ represents the number of examples for which CM exceeds DS by $k$ haplotypes. With one exception, DS always computes a number of haplotypes no larger than the number of haplotypes computed with CM. For the single exception, DS exceeds CM in 1 haplotype (hence -1 is shown in the plot). Observe that for 85% of the examples, DS outperforms CM. Moreover, observe that for a reasonable number of examples (40.1%, or 347 out of 854) the number of haplotypes computed with CM exceeds DS in more than 5 haplotypes. Finally, for a few examples (3 out of 854), CM can exceed DS by more than 40 haplotypes, the largest value being 46 haplotypes.

It should also be noted that, if the objective is to use either DS or CM as an upper bound for an exact HIPP algorithm, then a larger number of computed haplotypes represents a less tight, and therefore less effective, upper bound. Hence, DS is clearly preferable as an upper bound solution.

The results for the second part, comparing DS to the HIPP solution, are shown in Figure 2. As can be observed for the majority of examples (78.7%, or 672 out of 854), DS computes the HIPP solution. This is particularly significant when DS is used

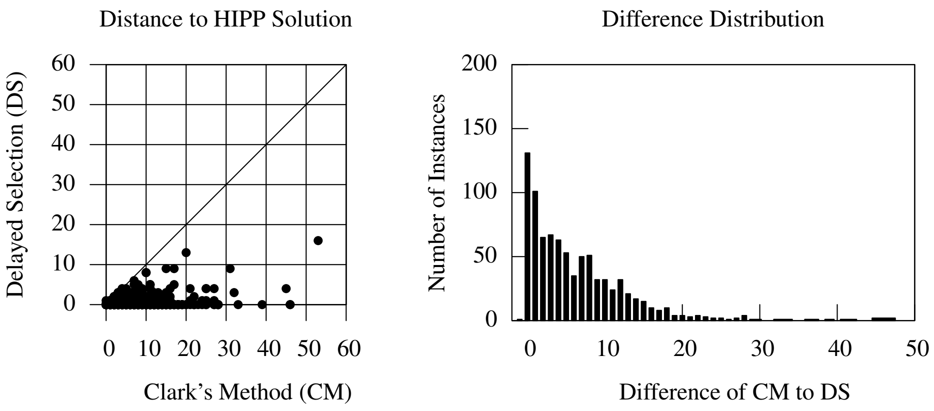

**Fig. 1.** Comparison of Clark's Method (CM) with Delayed Haplotype Selection (DS)

as an upper bound for recent HIPP algorithms[18]. For examples where DS computes the HIPP solution, exact HIPP algorithms are only required to prove the solution to be optimum. For a negligible number of examples (0.9%, or 8 out of 854) the difference of DS to the HIPP solution exceeds 5 haplotypes. Hence, for the vast majority of examples considered, DS provides a tight upper bound to the HIPP solution.

The results allow drawing the following conclusions. First, DS is a very effective alternative to CM when the objective is to minimize the total number of computed haplotypes. Second, DS is extremely effective as an upper bound for exact HIPP algorithms. For most examples (99.1%, or 846 out of 854) the number of haplotypes identified by DS is within 5 haplotypes of the target HIPP solution.

## 5 Conclusions and Future Work

This paper proposes a novel approach for haplotype selection, which addresses one of the main drawbacks of Clark's method[5]: its excessive greediness. This is achieved by delaying haplotype selection, one of the greedy steps of Clark's method. This approach leads to a tight upper bound that can be used when modelling this problem as a constraint satisfaction problem. The main context for the work is the development of efficient and accurate upper bounding procedures for exact algorithms for the Haplotype Inference by Pure Parsimony (HIPP) problem. Nevertheless, the proposed approach can also serve as a standalone haplotype inference algorithm. Experimental results, obtained on a comprehensive set of examples, are clear and conclusive. In practice, the new *delayed haplotype selection* (DS) algorithm provides quite tight upper bounds, of far superior quality than a recent implementation of Clark's method. For the vast majority of the examples considered, the results for DS are comparable to those for HIPP, and for a large percentage of the examples, DS computes the actual HIPP solution.

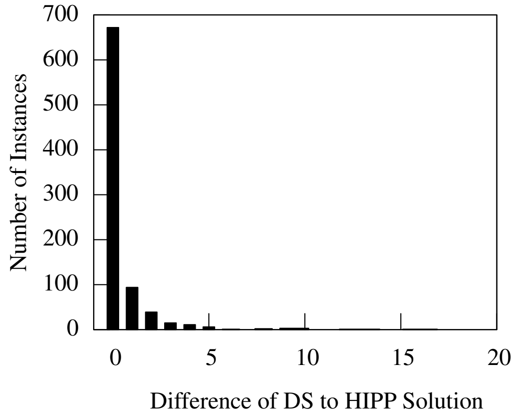

**Fig. 2.** Comparison of Delayed Haplotype Selection (DS) with HIPP solution

As mentioned earlier, recent approaches for the HIPP problem iterate through increasingly higher lower bounds[18]. This implies that solutions to the haplotype inference problem are only obtained when the actual solution to the HIPP problem is identified. Thus, these recent approaches to the HIPP problem[18] *cannot* be used for computing approximate HIPP solutions. The work described in this paper provides an efficient and remarkably tight approach for computing upper bounds. This allows recent HIPP based algorithms[18] to compute the exact solution by iterating through decreasing upper bounds. Hence, at each step a solution to the haplotype inference problem is identified, and, therefore, these methods can be used for approximating the exact HIPP solution. The integration of the DS algorithm in recent solutions to the HIPP problem is the next natural step of this work.

**Acknowledgments** This work is partially supported by Fundação para a Ciência e Tecnologia under research project POSC/EIA/61852/2004 and PhD grant SFRH/BD/28599/2006, by INESC-ID under research project SHIPs, and by Microsoft under contract 2007-017 of the Microsoft Research PhD Scholarship Programme.

## References

1. R.M. Adkins. Comparison of the accuracy of methods of computational haplotype inference using a large empirical dataset. *BMC Genet*, 5(1):22, 2004.
2. D. Brown and I. Harrower. A new integer programming formulation for the pure parsimony problem in haplotype analysis. In *Workshop on Algorithms in Bioinformatics*, 2004.
3. D. Brown and I. Harrower. Integer programming approaches to haplotype inference by pure parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(2):141–154, April-June 2006.
4. A. Graça, J. Marques-Silva, I. Lynce, and A. Oliveira. Efficient haplotype inference with pseudo-Boolean optimization. In *Algebraic Biology 2007*, pages 125–139, July 2007.

5. A. G. Clark. Inference of haplotypes from pcr-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–122, 1990.
6. M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29:229–232, 2001.
7. C. M. Drysdale, D. W. McGraw, C. B. Stack, J. C. Stephens, R. S. Judson, K. Nandabalan, K. Arnold, G. Ruano, and S. B. Liggett. Complex promoter and coding region $\beta_2$-adrenergic receptor haplotypes alter receptor expression and predict in vivo responsiveness. In *National Academy of Sciences*, volume 97, pages 10483–10488, September 2000.
8. G. Greenspan and D. Geiger. High density linkage disequilibrium mapping using models of haplotype block variation. *Bioinformatics*, 20, Supp. 1, 2004.
9. D. Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *Journal of Computational Biology*, 8(3):305–324, August 2001.
10. D. Gusfield. Haplotype inference by pure parsimony. In *14th Annual Symposium on Combinatorial Pattern Matching (CPM'03)*, pages 144–155, 2003.
11. D. Gusfield and S.H. Orzach. *Handbook on Computational Molecular Biology*, volume 9 of *Chapman and Hall/CRC Computer and Information Science Series*, chapter Haplotype Inference. CRC Press, December 2005.
12. Y-T. Huang, K-M. Chao, and T. Chen. An approximation algorithm for haplotype inference by maximum parsimony. *Journal of Computational Biology*, 12(10):1261–1274, 2005.
13. B. Kerem, J. Rommens, J. Buchanan, D. Markiewicz, T. Cox, A. Chakravarti, M. Buchwald, and L. C. Tsui. Identification of the cystic fibrosis gene: Genetic analysis. *Science*, 245:1073–1080, 1989.
14. M. Kimura and J.F. Crow. The number of alleles that can be maintained in a finite population. *Genetics*, 49(4):725–738, 1964.
15. D. L. Kroetz, C. Pauli-Magnus, L. M. Hodges, C. C. Huang, M. Kawamoto, S. J. Johns, D. Stryke, T. E. Ferrin, J. DeYoung, T. Taylor, E. J. Carlson, I. Herskowitz, K. M. Giacomini, and A. G. Clark. Sequence diversity and haplotype structure in the human abcd1 (mdr1, multidrug resistance transporter). *Pharmacogenetics*, 13:481–494, 2003.
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. I. Lynce and J. Marques-Silva. Efficient haplotype inference with Boolean satisfiability. In *National Conference on Artificial Intelligence (AAAI)*, July 2006.
18. I. Lynce and J. Marques-Silva. SAT in bioinformatics: Making the case with haplotype inference. In *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2006.
19. T. Niu, Z. Qin, X. Xu, and J. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70:157–169, 2002.
20. S. H. Orzack, D. Gusfield, J. Olson, S. Nesbitt, L. Subrahmanyan, and Jr. V. P. Stanton. Analysis and exploration of the use of rule-based algorithms and consensus methods for the inferral of haplotypes. *Genetics*, 165:915–928, October 2003.
21. M. J. Rieder, S. T. Taylor, A. G. Clark, and D. A. Nickerson. Sequence variation in the human angiotensin converting enzyme. *Nature Genetics*, 22:481–494, 2001.
22. M. Stephens, N. Smith, and P. Donelly. A new statistical method for haplotype reconstruction. *American Journal of Human Genetics*, 68:978–989, 2001.
23. The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437:1299–1320, 27 October 2005.
24. L. Wang and Y. Xu. Haplotype inference by maximum parsimony. *Bioinformatics*, 19(14):1773–1780, 2003.