MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **62** (2009) 261-274

ISSN 0340 - 6253

Enhanced Evolutionary and Heuristic Algorithms for Haplotype Reconstruction Problem Using Minimum Error Correction Model

Mehdi Kargar [a, c], Hadi Poormohammadi [b], Leila Pirhaji [c, e],

Mehdi Sadeghi [d,*], Hamid Pezeshk [f], Changiz Eslahchi [b]

[a] *Computer Engineering Department, Sharif University of Technology, Tehran, Iran.*

[b] *Faculty of Mathematics, Shahid-Beheshti University, Tehran, Iran.*

[c] *Bioinformatics Group, School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran.*

[d] *National Institute of Genetic Engineering and Biotechnology, Tehran-Karaj Highway, Tehran, Iran.*

[e] *Department of Biotechnology, College of Science, University of Tehran, Tehran, Iran.*

[f] *Center of Excellence in Biomathematics, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran.*

(Received October 10, 2008)

Abstract

Construction of two haplotypes from a set of Single Nucleotide Polymorphism (SNP) fragments is referred to as haplotype reconstruction problem. One of the most important computational models for this problem is Minimum Error Correction (MEC). Since MEC is an NP-hard problem, here we propose a heuristic algorithm for haplotype reconstruction problem. The algorithm is Particle Swarm Optimization (PSO) which is an evolutionary algorithm (EA). Evolutionary algorithms are stochastic search algorithms that imitate the natural biological evolution or the social behavior of species. In contrast to MEC model, our algorithm produces results in feasible time and it could be applied to large datasets. Our results suggest that the algorithm has less reconstruction error rate compared to other algorithms. This error is also very close to zero when the algorithm is applied to actual biological data. A comprehensive comparison between PSO and four famous algorithms in the literature is presented. A discussion on input parameters influencing reconstruction error rate is also presented.

* Corresponding author. E-mail address: sadeghi@nrcgeb.ac.ir

1. Introduction

Considering the rapid development of human genome sequencing process, in recent years, the identification of genetic differences has become one of the most interesting areas of research in bioinformatics. The human genome is a sequence formed by nucleotide alphabet {A, C, G, T}. In 99.9% of the positions of human genome, there are no differences across the gene sequence [1, 2]. Single Nucleotide Polymorphism (SNP) is the most frequent form of human genetic variation. In human genomes there are two copies of each chromosome. These copies are called paternal and maternal copies. A sequence of SNPs from each of the two chromosomes is called haplotype.

A nucleotide observed in an SNP position is called allele. In human, SNPs are almost always biallelic. Thus, two alleles can be denoted by 0 and 1, and a certain haplotype can be illustrated as a string of {0, 1} [3]. An SNP position, in which the two alleles have the same nucleotide, is called homozygous, while it is called heterozygous if the two alleles differ. Haplotypes have important roles in many branches of biology, for example in designing new drugs for complex diseases [4]. Determining haplotypes is a costly, difficult and time consuming process [5].

Using two enzymes with different concentration, it is possible to break two copies of chromosomes into small fragments. Thus a fragment is a substring of consecutive SNPs on a chromosome [3, 6]. It is possible to reconstruct the original haplotypes from these fragments. However, error might happen because haplotypes are determined using experimental methods. In addition, some SNP positions in a fragment might not be identified. In the last case, a '-' symbol, called gap, is used to represent missing data in the corresponding SNP position. An SNP matrix is used to determine the set of fragments. In this matrix, rows are fragments and columns are SNP positions.

The computational problem is as follows. Given an SNP matrix obtained by sequencing a DNA from two copies of a chromosome, how two haplotypes may be reconstructed using the SNP matrix. This process is called haplotype reconstruction problem [6].

Considering different types of errors, some models for haplotype reconstruction problem is introduced by Lancia, et al. [6]. Removal models are Minimum Fragment Removal (MFR) and Minimum SNP Removal (MSR). Time complexity analysis of MFR and MSR has been proposed by Bafna [7]. Practical algorithms for removal models have been proposed by Rizzi [8]. Another model is Longest Haplotype Reconstruction (LHR) which is discussed by Rudi,

et al. [9]. A statistical version of haplotype reconstruction problem, based on SNP fragments is proposed by Li, et al. [10]. One of the most important computational models for this problem is Minimum Error Correction (MEC), which was first introduced by Lippert, et al. [3], is an NP-hard problem. MEC is a popular model and is used widely in haplotype reconstruction process [11]. Recently a new model based on MEC called Minimum Conflict Individual Haplotyping (MCIH) has been proposed by Zhang, et al. [12]. This is also an NP-hard problem.

Since MEC is an NP-hard problem, some heuristic algorithms have been proposed for solving it. Also there is an exact algorithm based on Branch-and-Bound method for solving MEC problem. The exact algorithm has exponential time complexity and is not applicable to large datasets. In this paper we propose a new evolutionary algorithm based on particle swarm optimization method for solving MEC problem. Also we compare the algorithms in the literature and consider different types of parameters that influence the results.

The rest of this paper is organized as follows: section 2 includes a formal statement of the problem and related works. In section 3 the proposed algorithm is presented in detail. The pseudocode of the algorithm is also presented in section 3. The process of preparing datasets is introduced in section 4. In section 5, we present our results and compare them with the results obtained by other methods. Concluding remarks are presented in section 6.

2. Formal Statement of the Problem and Related Works

Suppose that there are a set $F = \{f_1, f_2, ..., f_m\}$ of fragments and a set $S = \{s_1, s_2, ..., s_n\}$ of SNPs positions obtained by sequencing two copies of chromosome. Define an $m \times n$ SNP matrix M = (m_{ij}), whose rows are fragments and columns are SNPs. The entry m_{ij} has the value from set {0, 1, -}. The symbol ‘‘-’’ called gap is used for characterizing missing data in the shotgun sequencing process or the entries that a fragment does not cover (so the size of each fragment is assumed to be n, the size of the haplotype). Let $x,y \in \{0,1,-\}$ and define

$$d(x,y) = \begin{cases} 1, & if\ x, y \in \{0,1\}\ and\ x \neq y \\ 0, & otherwise, \end{cases}$$

where *d(x,y)* is the distance between two chromosomes. Then the distance between two SNP fragments $f_i = \{f_{i1}, ..., f_{in}\}$, $f_j = \{f_{j1}, ..., f_{jn}\}$ is defined as $HD(f_i, f_j) = \sum_{j=1}^{n} d(f_{ij}, f_{kj})$.

We say that two fragments f_i , f_j are in conflict if $HD(f_i, f_j) > 0$,otherwise we say they are compatible. The distance between two haplotypes is defined in a similar way. For $HD(f_i, f_j) > 0$, it means that f_i and f_j are not from the same copy of a chromosome or there are errors in some SNP positions of f_i or f_j . If there are no errors in SNP matrix *M*, the rows of *M* can be partitioned into two sets F_1 and F_2 of pair wise compatible fragments and from each F_i *(i=1,2)* we can reconstruct a haplotype by fragment overlap. At this time we say *M* is feasible, otherwise infeasible. One of the most popular models for haplotype reconstruction problem is Minimum Error Correction (MEC). MEC is defined as what follows.

MEC Problem: Given an SNP matrix *M*, find and correct minimum number of errors (convert 0 to 1 or vice versa) so that the resulting matrix is feasible.

There are several algorithms concerning MEC model in the literature. Branch-and-bound algorithm presented by Wang in [13], is an algorithm producing an exact optimal solution for haplotype reconstruction problem. In fact, branch-and-bound is an exact and non-polynomial algorithm to avoid exhaustive search [14]. Using this algorithm, a classification of fragments into two disjoint sets S_1 and S_2 is obtained. Two haplotypes h_1 and h_2 are then reconstructed from S_1 and S_2 . Branch-and-bound is an exponential algorithm. In some cases the algorithm must check for all pair of sets to find optimal solution. This might result in infeasible time for operation even if the numbers of fragments are small.

Fast Hare is a heuristic algorithm for solving haplotype reconstruction problem [15]. Fast Hare is also based on MEC model. In the first step, Fast Hare sorts the SNP matrix based on the most left non gap SNP position of each fragment. The second step corresponds to reconstruct two haplotypes from this sorted SNP matrix. Fast Hare is a polynomial time algorithm and produces results in feasible time. Fast Hare runs in $O(n \log n)$.

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects which are similar to each other within the same cluster. A heuristic algorithm based on clustering analysis in data mining for haplotype reconstruction problem is introduced by Wang [16]. Based on hamming distance and similarity between two fragments, the iterative clustering algorithm produces two clusters

of fragments; then, in each iteration, the algorithm assigns a fragment to one of the clusters. In the last step, the algorithm produces two haplotypes from the set of two SNP fragments.

Genetic algorithms are derivative-free, stochastic optimization methods based on the concepts of natural selection and evolutionary processes [17]. Genetic Algorithms work with a random population of solutions (chromosomes). The fitness of each chromosome is determined by evaluating it against an objective function. To simulate the natural survival of the fittest process, best chromosomes exchange information (through crossover or mutation) to produce offspring chromosomes [18]. Genetic Algorithm (GA) is a well known evolutionary algorithm. A Genetic Algorithm is proposed by Wang [13] for solving MEC problem. They define a fitness function and mutation and cross over operations considering the haplotype reconstruction problem. In the next section, our algorithm is presented.

3. Particle Swarm Optimization Algorithm

This section describes the particle swarm optimization (PSO) algorithm for solving haplotype reconstruction problem. Let *M* be an SNP matrix with *m* rows and *n* columns. *m* is the number of fragments and *n* is the length of the haplotype. Thus each row represents a fragment and each column represents an SNP position. It should be noted that the goal of the algorithm is to produce two disjoint sets of fragments; say S_1 and S_2; then it produces two haplotypes from these sets. We need to infer two haplotypes from two sets of fragments by fragment overlap. Let $N_j^0(S_i)$ and $N_j^1(S_i)$ denote the number of 0s and 1s, respectively, in SNP position j of set S_i. The haplotypes h_1 and h_2 are produced from sets S_1 and S_2 by

$$h_{ij} = \begin{cases} 1 & if\ N_j^1(S_i) > N_j^0(S_i); \\ 0 & otherwise, \end{cases}$$

where $i = 1, 2$ and $j = 1, 2, \ldots, n$ and h_{ij} is *jth* SNP position of *ith* haplotype produced by *i*th set. We assume that the distance between two input haplotypes is in its maximum value. Therefore two input haplotypes are complements of each other. In the preprocess step, we remove those SNPs which contain a high percentage of 1s (0s), ignoring gaps, e.g. those SNPs with 90% of 1s (0s) will be removed from SNP matrix. For this case, we insert the most observed allele (1s or 0s) into two constructed haplotypes at that position and remove the corresponding column from the SNP matrix.

Evolutionary algorithms (EAs) are stochastic search methods that mimic the behavior of natural biological evolution and the social behavior of species. Considering the emergence and evolution of biological and social order has been a fundamental goal of evolutionary algorithms. Particle Swarm Optimization (PSO) is one of the modern evolutionary algorithms developed by Kennedy and Eberhart [23]. The PSO simulates the behavior of a group (swarm) of migrating birds, trying to reach an unknown destination. Like other evolutionary algorithms, PSO directs search using a population of particles (birds), corresponding to individuals. Each particle represents a candidate solution which could be an optimal solution. This mimics a swarm of birds that communicate together when they fly. Each bird flies in a specific direction, and identifies the bird that is in the best location by communicating with other birds. Each bird moves using a velocity which might change through the time when the algorithm is run. The process repeats until the swarm reaches a desired destination.

We use a binary string corresponding to an individual. This binary string divides the SNP fragments into two disjoint sets. Thus, the length of the binary string is the number of SNP fragments. Hence, if the *ith* position of an individual is set to 0, then the *ith* SNP fragment is assigned to set S_1, otherwise to set S_2. For every individual in a population, we need to assign an evaluation function called fitness. Considering MEC model, the goodness or badness of an individual depends upon the number of error corrections needed for the corresponding clustering. Thus, we use the following fitness function for evaluating the fitness of individual, say, *ind*:

$$fitness(ind) = m \times n - \sum_{i=1}^{2} \sum_{f \in S_i}^{2} HD(f, h_i)$$

in which h_i is the corresponding haplotype of set S_i, m is number of fragments and n is the number of SNP positions. It should be noted that an SNP matrix is feasible if and only if there exists an individual, *ind*, such that $fitness(ind) = m \times n$. The Algorithm is initialized with a group of random individuals (particles). The number of individuals, during the execution of the algorithm, is constant. Let us assume that, this is *popSize*. During the running of the algorithm, each individual keep three values. These values are: the best position reached in previous phases, P_i; its flying velocity V_i and its current position X_i. In each iteration of the algorithm, the position of the best individual, P_i, is calculated as the best fitness of all individuals. In each iteration, all individuals update their velocities as follows [19]:

$$V_{i+1} \rightarrow \gamma \times V_i + c_1 \times rand1(.) \times (P_i - X_i) + c_2 \times rand2(.) \times (P_g - X_i),$$

where c_1 and c_2 are two positive constants called learning factors (in this work we set c_1 and c_2 equal to 2). *rand1(.)* and *rand2(.)* are two functions generating pseudo random numbers in [0,1]. γ is an inertia weight value proposed by Shi and Eberhart [19] to control the influence of previous velocity values on the current velocity. γ starts with a large weight and then decreases as time goes by to support local search over global search. V_i should satisfy inequality $V_{\max} \geq V_i \geq -V_{\max}$; which $V_{\max}$ is an upper bound on the maximum value of the velocity of the particle (individual). The new position X_{i+1} is updated as:

$$X_{i+1} = X_i + V_{i+1}$$

After calculating the new position, X_{i+1}, the particle flies towards it. The pseudocode of PSO is presented in Table 1.

Table 1. Pseudocode of PSO Algorithm

Give proper parameter settings, population size *popSize*, number of generation *gnumber*, maximum change of a particle velocity $V_{\max}$
Generate random population of size *popSize* individuals (particles)
for each individual *ind*
    calculate *fitness(ind)*
initialize the value of γ
counter $\leftarrow 0$
while *counter* < *gnumber*
    for each individual *ind*
        calculate *fitness(ind)*
        set *pBest(ind)* as the best position of individual *ind* until this phase
    set *gBest* as the best fitness of all individuals
    for each individual *ind*
        calculate particle velocity using $V_{i+1} \rightarrow \gamma \times V_i + c_1 \times rand1() \times (P_i - X_i) + c_2 \times rand2() \times (P_g - X_i)$
        calculate particle position using $X_{i+1} = X_i + V_{i+1}$
    update the value of γ
    counter ++

4. Dataset

In this section the procedure of producing three input datasets are discussed. The first two datasets are real and the last one is simulated.

4.1. Angiotensin converting enzyme (ACE) Dataset

Angiotensin converting enzyme (ACE) is encoded by the gene DCP1. The dataset of ACE from 11 individuals is available [20]. There are 52 biallelic SNP's out of 78 available SNP positions in ACE. As we have mentioned in our method, we consider biallelic SNP's. Among these 11 genotypes corresponding to 11 individuals, there are two identical genotypes. We obtain a data set of 8 pairs of haplotypes, after removing one of the repeated genotypes (a pair of haplotypes) and those genotypes for which there exist at most one heterozygous SNP position [20].

4.1. Chromosome 5q31 Dataset

The dataset presented by Daly [21] contains 258 haplotype pairs and 103 SNP positions. Those haplotype pairs with less than 20% missing alleles are considered. From these pairs, 18 genotypes have at most one heterozygous position. We obtain 129 haplotype pairs from the dataset, after removing these genotypes [21].

4.1. Simulated Dataset

We generate data under some assumptions based on reality. We use CelSim, a popular simulator based on shotgun assembly [22]. In this part we introduce CelSim and it's parameters concerning our purpose to produce an SNP matrix. First CelSim makes two haplotypes of length L which are complements of each other. CelSim includes a distance parameter, D, which is the distance between two haplotypes. We set D equal to 100% in this work concerning that our haplotypes are complements of each other. Then it produces equal number of copies, *CopyNum*, from these haplotypes, cuts them and then it produces F fragments. The number of haplotypes, F, for a given haplotype of length l, was assumed to be $F=L/5$ (e.g. for a haplotype of length 80, number of fragments, F, is 16). The length of each fragment is assumed to be somewhere between *LMin* and *LMax*. We set *LMin* equal to 3 and *LMax* equal to 8 as it is the case for real biological data. Thus we have an SNP matrix which is error free. We include some errors, *ReadingError* and *GapError* in SNP matrix. *ReadingError* is concerned with changing 0 to 1 and vice versa and *GapError* is concerned to

changing 0 or 1 to '-' in an SNP matrix. The algorithms are run 500 times with different inputs. Input dataset for each run is the same for all algorithms.

5. Results and Discussion

In this section the results of running Fast Hare, clustering algorithm, GA and the proposed algorithm, PSO on datasets are presented. Some discussions on the performance of the algorithms are also available. The algorithms were implemented on a 2.8 GHz Pentium 4 PC using JAVA language. The simulations show that Clustering and PSO algorithms produce better results, in terms of certain error, in comparison with Fast Hare and GA. We also present results of branch-and-bound algorithm. It should be noted here that we are not able to run branch and bound algorithm on large input datasets. The branch-and-bound algorithm can quickly process instances within 30 SNP fragments and 50 SNP sites under various error rates. Beyond this range, the time taken by the algorithm increases rapidly, especially when the error rate of fragments is high [13]. However, in the process of producing datasets, the fragments of each of the haplotypes are determined. Thus, we can assign fragments of original haplotype h_1 (h_2) to set S_1 (S_2). Then two reconstructed haplotypes are produced by clusters. Finally, we have results of branch-and-bound algorithm without running it. It should be mentioned that branch-and-bound algorithm produces exact optimal solution for haplotype reconstruction problem [13]. It means that the branch-and-bound algorithm produces results with the minimum value of error rate which is possible. For calculating error rate, we use the degree of similarity between the reconstructed haplotypes and the original haplotypes. Assume that h_1 and h_2 are original haplotypes, and h'_1 and h'_2 are reconstructed haplotypes. Reconstruction error rate is defined by:

$$reconstruction\ error\ rate = \frac{\min\{r_{11}+r_{22}, r_{12}+r_{21}\}}{2\times haplotypeLength},$$

where r_{ij} is the hamming distance between an actual haplotype and a constructed haplotype i,j=0,1.

We want to study the relationship between multiple values of reading error rate and gap error rate with reconstruction error rate. As we might expect, when reading error rate or gap error rate increases, reconstruction error rate increases too. Figures 1, 2 and 3 show the results of running algorithms with different reading error rate on ACE, Daly and Celsim dataset. The

gap error rate is set to 5 percent in the figures. It should be noted that the results of running algorithms on ACE and CelSim are very close to each other. It is because the numbers of SNP positions are approximately equal in CelSim and ACE datasets.

These results suggest that when reading error rate is smaller than 3%, PSO and clustering algorithm produce more accurate results. The reconstruction error rates of these algorithms are very close to Branch-and-Bound results. This is very close to zero. When error rate increases form 3%, the results of PSO is better than the other algorithms. Also this is close to the results of branch-and-bound algorithm. When reading error rate is larger than 10%, GA produces more accurate results than Fast Hare and clustering algorithm.

When haplotype length increases, reconstruction error rate increases. In Figure 4 we present results of running PSO algorithm on CelSim dataset for haplotypes of different lengths, from 20 to 60. The gap error is set to 5 percent. The number of copies is set to 5.

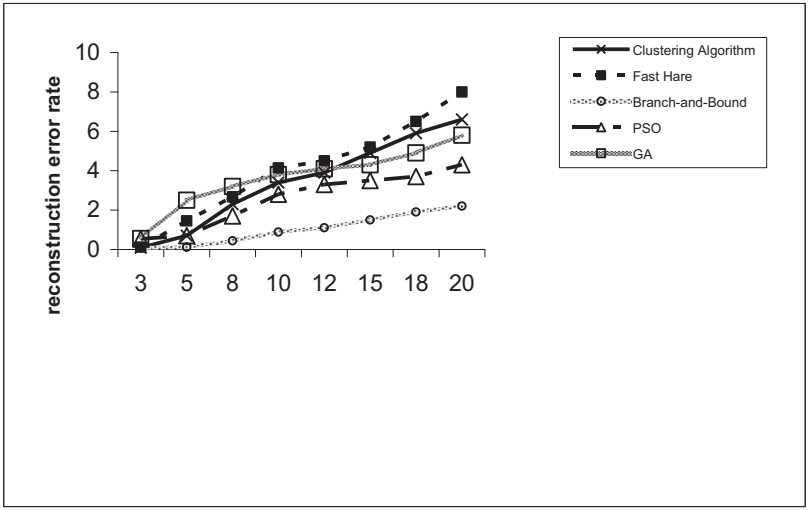

Figure 1. The reconstruction error rate of the MEC model by different algorithms on ACE dataset. Haplotype length is 52, number of copy is 5, gap error is 10 percent.

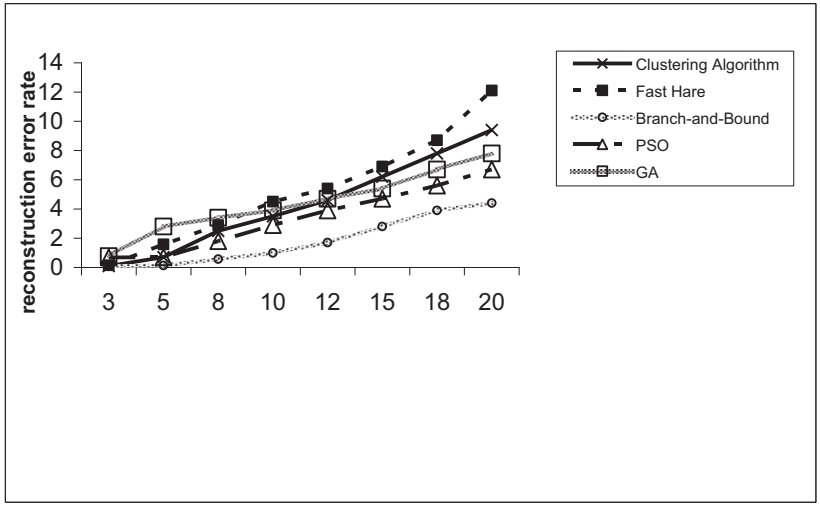

Figure 2. The reconstruction error rate of the MEC model by different algorithms on Daly dataset. Haplotype length is 112, number of copy is 5, gap error is 10 percent.

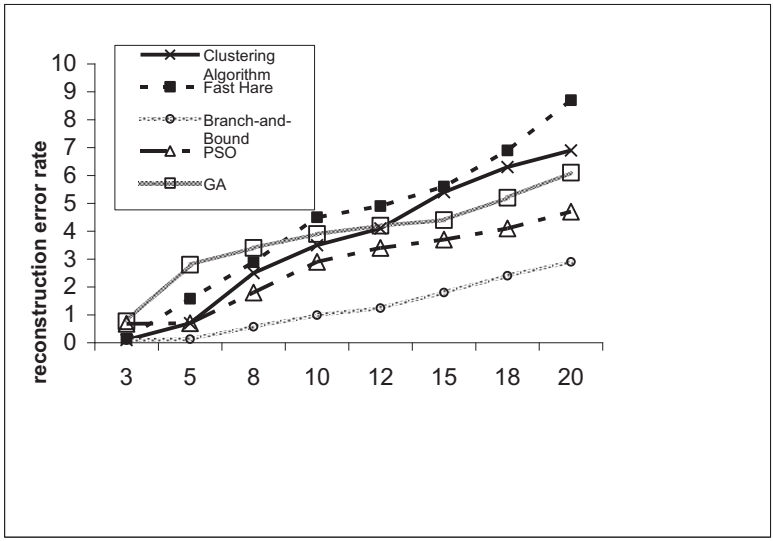

Figure 3. The reconstruction error rate of the MEC model by different algorithms on CelSim dataset. Haplotype length is 50, number of copy is 5, gap error is 10 percent.

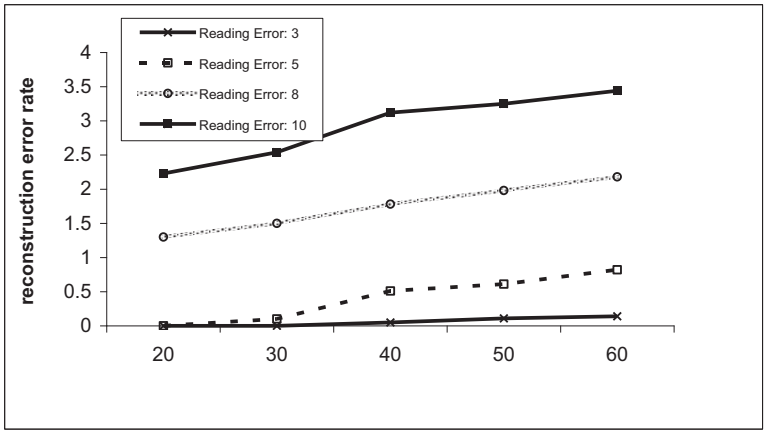

Figure 4. The reconstruction error rate of the MEC model by PSO algorithms on CelSim dataset. Haplotype length is varied, number of copy is 5, gap is set to 5 percent.

The running times of PSO, GA, clustering algorithm and Fast Hare are presented in Table 2. It should be noted that we are not able to run branch-and-bound algorithm on large input datasets. Thus, the running time of branch-and-bound is not available. This table suggests that Fast Hare produce results faster than other algorithms. It is because Fast Hare has $O(n \log n)$ time complexity, where n is the number of fragments. In addition, the running times of the clustering algorithm and Fast Hare depend on the number of fragments. But the running times of PSO and GA do not considerably change when the number of fragments varies.

Table 2. Running times (in seconds) of the algorithms. Haplotype length is 50. We are not able to run branch-and-bound on datasets with more than 30 fragments because it has exponential time complexity.

Number of fragments	Clustering	PSO	Fast Hare	GA
40	0.33	2.98	0.11	3.18
60	0.86	3.13	0.18	3.35
80	1.97	3.35	0.26	3.52
100	4.15	3.75	0.46	3.81

5. Conclusion

We have studied evolutionary, heuristic and exact algorithms for haplotype reconstruction problem. Our algorithm, PSO, is an evolutionary algorithm. The second algorithm is based on clustering analysis and the techniques used in data mining. The third and fourth algorithms are GA and a heuristic algorithm called Fast Hare. The last one is branch-and-bound algorithm that produces exact optimal solution for haplotype reconstruction problem. Branch-and-bound is an exponential algorithm and we are not able to run it on large input datasets.

To summarize, our simulation results suggest that when reading error rate is smaller than 3% PSO and clustering algorithms can produce better results in terms of reconstruction error rate, compared to other algorithms. Also in comparison with minimum possible reconstruction error rate which is achievable by branch-and-bound algorithm, the results achievable by PSO are very close to it. Especially when reading error rate is in the range of actual biological data. Evolutionary and heuristic algorithms produce results in feasible time compared to the time complexity of branch-and-bound, which is exponential. Thus, we believe that evolutionary algorithms could be applied on haplotype reconstruction problem and these produce more accurate results.

Acknowledgements

Hamid Pezeshk would like to thank the Department of Research Affairs of University of Tehran. This research was in part supported by a grant from IPM (No. CS1385-1-02).

References

[1] M. R. Hoehe, K. Kopke, B. Wendel, K. Rohde, C. Flachmeier, K. K. Kidd, W. H. Berrettini, and G. M. Church, Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence, *Hum. Mol. Gene.* **9** (2000) 2895–2908.

[2] J. D. Terwilliger and K. M. Weiss, Linkage disequilibrium mapping of complex disease: fantasy and reality?, *Curr. Opin. Biotechnol.* **9** (1998) 579–594.

[3] R. A. Lippert, R. Schwartz, G. Lancia and S. Istrail, Algorithmic strategies for the SNPs haplotype assembly problem, *Brief. Bioinformatics* **3** (2002) 23–31.

[4] S. Service, D. Temple Lang, N. Freimer and L. Sandkuijl, Linkage-disequilibrium mapping of disease genes by reconstruction of ancestral haplotypes in founder populations, *Am. J. Hum. Genet*. **64** (1999) 1728–1738.

[5] J. C. Stephens, J. A. Schneider, D. A. Tanguay, J. Choi, T. Acharya and S.E. Stanley, Haplotype variation and linkage disequilibrium in 313 human genes, *Science* **293** (2001) 489–493.

[6] G. Lancia, V. Bafna, S. Istrail, R. Lippert and R. Schwartz, SNPs problems, complexity and algorithms, *LNCS* **61** (2001) 182–193.

[7] V. Bafna, S. Istrail, G. Lancia, and R. Rizzi, Polynomial and APX-hard cases of the individual haplotyping problem, *Theor. Comput. Sci.* **335** (2005) 109–125.

[8] R. Rizzi, V. Bafna, S. Istrail and G. Lancia, Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem, *LNCS* **24** (2002) 29–43.

[9] C. Rudi, L. Iersel, S. Kelk and J. Tromp, On the complexity of the single individual SNP haplotyping problem, *Proc. WABI2005* (2005) 127–132.

[10] L. M. Li, J. H. Kim and M. S. Waterman, Haplotype reconstruction from SNP alignment, *J. Comp. Biol.* **11** (2004) 507–518.

[11] X. S. Zhang, R. S. Wang, L. Y. Wu and L. N. Chen, Models and algorithms for haplotyping problem, *Curr. Bioinf.* **11** (2006) 105–114.

[12] X. S. Zhang, R. S. Wang, L. Y. Wu and W. Zhang, Minimum conflict individual haplotyping from SNP fragments and related genotype, *Evol. Bioinformatics Online* **2** (2006) 271–280.

[13] R. S. Wang, L. Y. Wu, Z. P. Li and X. S. Zhang, Haplotype reconstruction from SNP fragments by minimum error correction, *Bioinformatics* **21** (2005) 2456–2462.

[14] W. I. Koontz, P. M. Narendra and K. Fukunaga, A branch and bound clustering algorithm, *IEEE Trans. Comp.* **24** (1975) 908–915.

[15] A. Panconesi, M. Sozio, Fast hare: A fast heuristic for single individual SNP haplotype reconstruction, *Proc. WABI2004* (2004) 266–277.

[16] Y. Wang, E. Feng and R. Wang, A clustering algorithm based on two distance functions for MEC model, *Comput. Biol. Chem.* **31** (2007) 148–150.

[17] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, MA: Addison-Wesley Publishing Co, 1989.

[18] Z. Michalewicz C. Janikow, Genocop: a genetic algorithm for numerical optimization problems with linear constraints, *ACM* **39** (1996) 23–29.

[19] Y. Shi, R. Eberhart, A modified particle swarm optimizer, *Proc. IEEE international conference on evolutionary computation*. (1998) 69–73.

[20] M. J. Rieder, S. L. Taylor, A. G. Clark and D. A. Nickerson , Sequence variation in the human angiotensin converting enzyme , *Nat. Genet.* **62** (1999) 22–59.

[21] M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson and E. S. Lander, High-resolution haplotype structure in the human genome, *Nat. Genet.* **29** (2001) 229–232.

[22] G. Myers, A dataset generator for whole genome shotgun sequencing, *Proc. Int. Conf. Intell. Syst. Mol. Niol.* (1999) 202–210.

[23] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proc. IEEE international conference on neural networks (Perth, Australia)* (1995) 1942–1948.