International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:5, 2014

An Improved Ant Colony Algorithm for Genome Rearrangements

Essam Al Daoud

***Abstract*—**Genome rearrangement is an important area in computational biology and bioinformatics. The basic problem in genome rearrangements is to compute the edit distance, i.e., the minimum number of operations needed to transform one genome into another. Unfortunately, unsigned genome rearrangement problem is NP-hard. In this study an improved ant colony optimization algorithm to approximate the edit distance is proposed. The main idea is to convert the unsigned permutation to signed permutation and evaluate the ants by using Kaplan algorithm. Two new operations are added to the standard ant colony algorithm: Replacing the worst ants by re-sampling the ants from a new probability distribution and applying the crossover operations on the best ants. The proposed algorithm is tested and compared with the improved breakpoint reversal sort algorithm by using three datasets. The results indicate that the proposed algorithm achieves better accuracy ratio than the previous methods.

***Keywords*—**Ant colony algorithm, Edit distance, Genome breakpoint, Genome rearrangement, Reversal sort.

I. Introduction

BESIDES the local mutations, i.e., deleting, inserting, and substituting a single nucleotide, another kind of mutation also occurs in the chromosomes, changing the DNA sequence at a higher level. These mutations relocate parts of a chromosome and put it back into the chromosome at another position or in reverse orientation [1]. This means that the genes themselves are not altered by these mutations, but only their order changes. This kind of mutations is called genome rearrangements. To build a phylogenetic tree of closely related species, a comparison of the sequences of single genes does not help much, since the differences are too small [2]. On the other hand, often the order of the genes varies a lot. Thus, it looks promising to compare the entire genomes with each other and to use the number of genome rearrangements needed to transform one genome into another as a measure of evolutionary distance. Two types of intrachromosomal transformations, the first is the reversal where a part of the gene sequence is rearranged in reverse orientation. The second type is the transposition where a part of the gene sequence is rearranged in the same orientation at another position [3]. Thus, the genome rearrangement problem can be formalized as the problem of transforming a given permutation into another given permutation with the minimum number of reversals [4]. Reversal genome rearrangement problem can be divided into two models. The first model is the signed permutation model which can be solved exactly in polynomial time; the second model is the unsigned permutation model which belongs to the NP-hard combinatorial problems. Therefore, several approximation algorithms have been suggested to solve unsigned permutation model [5], [6].

E. Al-Daoud is with faculty of Science and Information Technology, Computer Science Department, Zarka University, Jordan (Tel +962-796680005, e-mail: essamdz@zpu.edu.jo).

II. Genome Arrangement Problem

The pairwise genome rearrangement problem is to find an optimal scenario transforming one genome to another. This type of arrangement is called reversal (inversion) which is a contiguous interval of genes is located into the reverse order. Let $\pi = (\pi_1, \pi_2, ..., \pi_n)$ be a permutation of order n. For $1 \le i < j \le n$, an (i, j)-reversal is a permutation $\rho(i, j)$, such that [7], [8]:

$$\pi.\rho(i, j) = (\pi_1, ..., \pi_{i-1}, \pi_{j-1}, ..., \pi_{i+1}, \pi_i, \pi_{j+1}, ..., \pi_n)$$

Given two permutations $\pi = (\pi_1, \pi_2, ..., \pi_n)$ and $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ the reversal distance $d(\pi, \alpha)$ is the smallest number of reversals that will take π to α. To simplify the calculation consider one of the two permutations is the identity (1, 2, …n). Subsequnce the reversal distance can be written as $d(\pi)$. For Example the Table I shows that the reversal distance of π=(4, 3, 2, 1, 7, 6 5) is 5. However, the optimal reversal is given in Table II. Thus the optimal distance $d(\pi)$=2

TABLE I
Reversal Sort by Using 5 Permutation

Permutation	Sequences						
$\rho(1,2)$	**4**	**3**	**2**	**1**	**7**	**6**	**5**
$\rho(2,3)$	**3**	**4**	**2**	**1**	**7**	**6**	**5**
$\rho(3,4)$	**3**	**2**	**4**	**1**	**7**	**6**	**5**
$\rho(1,3)$	**3**	**2**	**1**	**4**	**7**	**6**	**5**
$\rho(5,7)$	**1**	**2**	**3**	**4**	**7**	**6**	**5**
	1	**2**	**3**	**4**	**5**	**6**	**7**

TABLE II
Optimal Reversal Sort

Permutation	Sequences						
$\rho(1,4)$	**4**	**3**	**2**	**1**	**7**	**6**	**5**
$\rho(5,7)$	**1**	**2**	**3**	**4**	**7**	**6**	**5**
	1	**2**	**3**	**4**	**5**	**6**	**7**

The reversal distance can be reduced by using breakpoint reversal sort. Let $\pi = (\pi_1, \pi_2, ..., \pi_n)$ be a permutation of order n. A breakpoint of π is a pair:

$$(i, i+1) \in \{0, ..., n\} \times \{1, ..., n+1\}$$

of positions such that $|\pi_i - \pi_{i+1}| \neq 1$ holds, and b(π) is the number of breakpoints of π. For example the number of breakpoints of π =(8 2 7 6 5 1 4 3 9 10) is b(π)=5 which are: (8| 2 |7 6 5| 1| 4 3| 9 10). Any permutation can be partitioned into increasing strips (overlined) and decreasing strips (underlined). For example the increasing and the decreasing strips of π = (0 2 1 3 4 5 8 7 6 9) is π = ($\overline{0}$ $\underline{2\ 1}$ $\overline{3\ 4\ 5}$ $\underline{8\ 7\ 6}$ $\overline{9}$). Algorithm 1 uses breakpoint reduction to approximate the reversal sort problem. Algorithm 2 improves the breakpoint reversal sort. The lower bound approximation of the second algorithm is 2-approximation [9].

Algorithm 1. **Breakpoint reversal sort**:
while b(π) > 0 **do**
  if π has a decreasing strip **then**
    perform a reversal that decreases b(π)
  else
    perform a reversal that turns an increasing strip into a decreasing strip;

Algorithm 2. **Improved breakpoint reversal sort**:
while b(π) > 0 **do**
  choose a reversal reducing b(π) by the largest amount, resolving ties among those that reduce b(π) by one in favor of reversals that leave a decreasing strip.

Table III illustrates the reversal sort by using breakpoint, the increasing and the decreasing strips.

TABLE III
REVERSAL SORT BY USING ALGORITHM 2

Sequences	Breakpoint
$\overline{0}$ $\underline{2}$ $\overline{8\ 7\ 6\ 5}$ $\underline{1}$ $\overline{4\ 3\ 9}$	b(π) = 5
$\overline{0\ 2\ 3\ 4\ 1}$ $\overline{5\ 6\ 7\ 8\ 9}$	b(π) = 3
$\overline{0\ 1}$ $\underline{4\ 3\ 2}$ $\overline{5\ 6\ 7\ 8\ 9}$	b(π) = 2
$\overline{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9}$	b(π) = 0

In another hand, the signed permutation model can be solved exactly in polynomial time by using Kaplan or Badar method [10], [11].

III. ANT COLONY METHODS

Ant Colony Optimization (ACO) is a metaheuristic inspired by the behavior of ant colonies. In the last few years, ACO has received increased attention by the scientific community as can be seen by the growing number of publications and the different fields of application [12]. It has been used for solving the combinatorial optimization problems, such as Traveling Salesman Problem (TSP), vehicle routing, telecommunications networks, knapsack problem, assignment problem, maximum clique and scheduling problem. Real ants are able to find the shortest path between their nest and food sources because of the chemical substance (pheromone) that they deposit on their way [13]. The pheromone evaporates over time so the shortest paths will contain more pheromone and will subsequently attract a greater number of ants. Therefore ACO algorithms use two factors for guiding the search process namely the pheromone and the heuristic information. Three different versions of Ant system (AS) were proposed [14]. These were called, antquantity, ant-cycle and ant-density. Nowadays, when referring to AS, one actually refers to ant-cycle since the two other variants were abandoned because of their inferior performance. Let m be the number of the artificial ants. The initial pheromone is $\tau_{ij} = \tau_0 = m/C$ where i is the current node, j the next node and C is the length of a tour generated by the nearest-neighbor heuristic. The probability with which ant k, currently at node i and the next node is j:

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \in N_i^k} \tau_{il}^\alpha \eta_{il}^\beta} \quad (1)$$

where $\eta_{ij} = 1/d_{ij}$ is a heuristic value that is available a priori, α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information, and N_i^k is the feasible neighborhood of ant k when being at node i. Pheromone evaporation is implemented by

$$\tau_{ij} = (1-\rho)\tau_{ij} \quad (2)$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate which is used to forget the bad decisions. The pheromone is updated as following

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k \quad (3)$$

where

$$\Delta\tau_{ij}^k = \begin{cases} 1/C & \text{if the arc belongs to tour k} \\ 0 & \text{otherwise} \end{cases}$$

Rank-Based Ant (RBA) System is another improvement over AS is proposed by Bullnheimer et al. [15]. The AS_{rank} pheromone update rule is

$$\tau_{ij} = \tau_{ij} + \sum_{r=1}^{w-1} (w-r)\Delta\tau_{ij}^r + w\Delta\tau_{ij}^{bs} \quad (4)$$

where bs is the best-so-far tour, r indicates to the best r^{th} ant and w is the best ranked ants. Ant Colony System ACS suggested by Dorigo and Gambardella [16]. The next node is calculated as following:

$$j = \begin{cases} \max_{l \in N_i^k} \tau_{ij}\eta_{ij}^{\beta} & \text{if rand} \le q_0 \\ P_{ij}^k & \text{Otherwise} \end{cases}$$

where q_0 is a parameter. The global pheromone update rule is:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad (5)$$

In addition to the global formula, in ACS the ants use a local pheromone update rule that is used immediately after crossed an edge. The local pheromone update rule is:

$$\tau_{ij} = (1-\xi)\tau_{ij} + \xi\tau_0 \quad (6)$$

where ξ and τ_0 are two parameters

IV. An Improved Ant Colony Algorithm

The reversal distance problem of unsigned permutation is NP-hard problem. Therefore many approximation algorithms have been suggested to solve this problem. In this section a new approximation algorithm based on modified ant optimization is proposed. The basic idea is to convert the unsigned permutation to signed permutation and evaluate the ant tour by using Kaplan algorithm. The modified ant optimization barrows the crossover operation from the genetic algorithm. Moreover, for each iteration, new ants are sampled, the best ants are kept and the worst ants are replaced. Algorithm 3 shows the modified ant optimization for genome rearrangement problem.

Algorithm 3: An improved ant optimization
1- Divide the string into *k* subsets, each one of length *t* (in this study *t=5*)
2- For each ant
  2.1-Select v subsets randomly
  2.2-Let all the elements in the selected subsets be negative (in this study *v=k/5*)
  2.3-Let all the elements in the unselected subsets be positive
  2.4-Each ant consists from all the negative subsets and all the positive subsets
3- Use Kaplan algorithm to evaluate each ant
4- Select the best *r=m/3* ants (*m* is the number of ants)
5- Update pheromone using only the best ants, the pheromone of character *i* becomes:

$$\tau_i = \tau_i + \frac{1}{nr}\sum_{j=1}^{r}\delta^j \quad (7)$$

where

$$\delta_i = \begin{cases} 1 & \text{if character } i \text{ is negative} \\ 0 & \text{Otherwise} \end{cases}$$

6- Construct the new population
  6.1- *m/3* ants are the best ants selected from the previous generation
  6.2- *m/3* ants are constructed by using two points crossover that are applied to the best ants
  6.3- *m/3* ants are constructed by using a new probability distribution, and a random value *rand* in *(0, 1)* such that:

$$\text{character } i \text{ is } \begin{cases} \text{Negative} & \text{if } rand < p_i \\ \text{Positive} & \text{Otherwise} \end{cases}$$

where

$$p_i = \frac{\tau_i^{\alpha}}{\sum_{l \in ch}\tau_l^{\alpha}} \quad (8)$$

7- Apply pheromone evaporation

$$\tau_i = (1-\rho)\tau_i \quad (9)$$

8- Check the stopping condition, if met then terminate, else go to step 3.

V. Experimental Results

To test the suggested method, unsigned datasets with various sizes, 50, 100 and 200, were randomly generated, the permutation are generated by performing numerous swaps on the identity sequences. Therefore the exact solution is known in advance in this study. The experiments have three main goals. The first is to observe how close reversal distance values to optimal values, the second is to compare the suggested method with the improved breakpoint reversal sort (IBRS) and the third is to find the ratio between the approximate methods and the exact solution. The ratio is calculated as following:

$$Ratio = \frac{\text{Approximate Method}}{\text{Exact Solution}} \quad (10)$$

The results of the suggested method are the average of 10 runs. The best parameters in this study are α = 0.6, pheromone evaporation rate ρ =0.75 and the number of the ants is 30. Tables IV-VI show the ratio and compare the reversal distances of IBRS and the new ant optimization algorithm, the average ratio of each algorithm indicates to that the new approach outperforms the previous approaches for all sequences with various breakpoints and length. For example the average ratio between the new algorithm and the exact solution are 1.18, 1.22 and 1.19 while the average ratio between the IBRS algorithm and the exact solution are 1.76, 1.6 and 1.57 for the datasets of the sizes 50, 100 and 200 respectively.

TABLE IV
NUMERICAL COMPARISON USING DATASET OF LENGTH 50

No	b(π)	Exact d(π)	IBRS		New Ant Algorithm	
			d(π)	Ratio	d(π)	Ratio
1	30	24	38	1.58	26	1.08
2	25	20	32	1.60	23	1.15
3	23	16	27	1.69	19	1.19
4	18	15	24	1.60	19	1.27
5	20	16	26	1.63	18	1.13
6	25	19	30	1.58	23	1.21
7	35	26	42	1.62	30	1.15
8	14	8	16	2.00	9	1.13
9	22	17	31	1.82	21	1.24
10	17	10	25	2.50	13	1.30
Avg.	22.9	17.1	29.1	1.76	20.1	1.18

TABLE V
NUMERICAL COMPARISON USING DATASET OF LENGTH 100

No	b(π)	Exact d(π)	IBRS		New Ant Algorithm	
			d(π)	Ratio	d(π)	Ratio
1	52	32	50	1.56	37	1.16
2	70	45	74	1.64	55	1.22
3	48	33	56	1.70	40	1.21
4	45	35	55	1.57	41	1.17
5	55	34	55	1.62	43	1.26
6	43	30	52	1.73	35	1.17
7	48	28	42	1.50	33	1.18
8	50	30	47	1.57	39	1.30
9	52	34	52	1.53	41	1.21
10	41	28	45	1.61	37	1.32
Avg.	50.4	32.9	52.8	1.60	40.1	1.22

TABLE VI
NUMERICAL COMPARISON USING DATASET OF LENGTH 200

No	b(π)	Exact d(π)	IBRS		New Ant Algorithm	
			d(π)	Ratio	d(π)	Ratio
1	113	81	140	1.73	91	1.12
2	122	84	133	1.58	102	1.21
3	105	74	112	1.52	105	1.43
4	90	66	101	1.53	77	1.17
5	107	71	108	1.52	79	1.11
6	98	72	116	1.61	85	1.18
7	118	78	123	1.58	93	1.19
8	114	81	130	1.60	88	1.09
9	94	65	102	1.57	85	1.31
10	102	72	108	1.50	79	1.10
Avg.	106.3	74.35	117.30	1.57	88.40	1.19

VI. CONCLUSION

With the increased availability of whole genome sequences, genome rearrangement algorithms have been used to estimate the evolutionary distance between present-day genomes, and to reconstruct the gene order of ancestral genomes. In this study an improved ant colony algorithm to find the nearest minimal reversal distance between genomes is presented. The suggested procedure uses Kaplan algorithm to evaluate each ant for each iteration, replaces the worst ants by a new ants sampled from a new probability distribution and generates a set of ants by using the crossover operation. The average ratio of each algorithm indicates to that the new approach outperforms the previous approaches for all datasets with various breakpoints and length.

REFERENCES

[1] G. Fertin, A. Labarre, I. Rusu, E. Tannier, S. Vialette, *Combinatorics of Genome Rearrangements*. MIT Press, 2009.
[2] A. Bergeron, J. Mixtacki, J. Stoye, "A unifying view of genome rearrangements," Proc 6th Workshop Algs in Bioinf (WABI'06), Volume 4175 of Lecture Notes in Comp Sci Springer Verlag, Berlin, 163-173, 2006.
[3] G. Tesler, "Efficient algorithms for multichromosomal genome rearrangements," J Comput Syst Sci, 65(3):587-609, 2002.
[4] P. A. Pevzner, M. S. Waterman, "Open combinatorial problems in computational molecular biology," In Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems, 158-173, 1995.
[5] V. Bafna, P. A. Pevzner, "Genome rearragements and sorting by reversals," SIAM Journal on Computing, 25(2):272–289, 1996.
[6] N. El-Mabrouk, "Sorting signed permutations by reversals and insertions/deletions of contiguous segments," Journal of Discrete Algorithms, 1:105-122, 2001.
[7] A. Caprara, R. Rizzi, "Improved approximation for breakpoint graph decomposition and sorting by reversals," J. of Combin Optimization, 6(2):157-182, 2002.
[8] H. Kaplan, E. Verbin,"Effficient data structures and a new randomized approach for sorting signed permutations by reversals," In Proc. 14th Annual Symposium on Combinaotrial Pattern Matching (CPM '03), 170–185, 2003.
[9] Q. P. Gu, S. Peng, H. Sudborough, "A 2-approximation algorithm for genome rearrangements by reversals and transpositions," Theoretical Computer Science, 210(2): 327–339, 1999.
[10] H. Kaplan, R. Shamir, R. E. Tarjan, "Faster and simpler algorithm for sorting signed permutations by reversals," SIAM Journal of Computing, 29(3):880–892, 2000.
[11] D. A. Bader, B. M. E. Moret, M. Yan "A linear-time algorithm for computing inversion distance between signed permutations with an experimental study," Journal of Computational Biology, 8(5): 483–491, 2001.
[12] V. Ganapathy, T. Tang, S. Parasuraman," Improved Ant Colony Optimization for Robot Navigation," Proceeding of the 7th International Symposium on Mechatronics and its Applications (ISMA10), Sharjah, UAE, April 20-22, 2010.
[13] Z. Zhang, Z. Feng, Z. Ren, "Novel Ant Colony Optimization Algorithm Based on Order Optimization," Journal of Xi'An Jiaotong University, 44(2): 15-19, 2010.
[14] M. Dorigo, L. M. Gambardella, M. Middendorf, T. Stutzle, "Ant Colony Optimization," IEEE Transactions on Evolutionary Computation, 6(4): 317–365,2002.
[15] B. Bullnheimer, R. F. Hartl, C. Strauss, "A new rank-based version of the Ant System: A computational study," Central European Journal for Operations Research and Economics, 7(1): 25–38. 1999.
[16] M. Dorigo, L. M. Gambardella, "Ant colonies for the traveling salesman problem," BioSystems, 43(2), 73–81, 1997.

Essam Al Daoud received his BSc from Mu'tah university, MSc from Al Al-Bayt university, and his PhD in computer science from university Putra Malaysia in 2002. Currently, he is an associate professor in the computer science department at Zarqa university, Jordan. His research interests include machine learning, optimization, quantum computation and cryptography.