Applied Mathematics and Computation 219 (2012) 2267–2270

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Solving traveling salesman problem in the Adleman–Lipton model

Zhaocai Wang [a,*], Yiming Zhang [b], Weihua Zhou [c], Haifeng Liu [a]

[a] College of Information, Shanghai Ocean University, Shanghai 201306, PR China
[b] School of Economics and Management, Yancheng Institute of Technology, Yancheng 224051, PR China
[c] The Central Laboratory, Shandong Vocational College of Science and Technology, Weifang 261053, PR China

ARTICLE INFO

Keywords:
DNA computing
The traveling salesman problem
Adleman–Lipton model
NP-complete problem

ABSTRACT

The traveling salesman problem is to find a minimum cost (weight) path for a given set of cities (vertices) and roads (edges). The path must start at a specified city and end there after going through all the other given cites only once. It is a classical NP-complete problem in graph theory. In this paper, we consider a DNA procedure for solving the traveling salesman problem in the Adleman–Lipton model. The procedure works in $O(n)$ steps for the traveling salesman of an edge-weighted graph with n vertices.

1. Introduction

DNA computing is a newly emerging interdisciplinary science that uses DNA molecular biotechnologies to solve problems in computer science or mathematics. DNA computing can execute billions of operations simultaneously. DNA also provides a huge storage capacity since they encode information on the molecular scale. As the first work for DNA computing, Adleman [1] presented an idea of solving the Hamiltonian path problem of size n in $O(n)$ steps using DNA molecules. Lipton [2] demonstrated that Adleman's experiment could be used to determine the NP-complete satisfiability problem. In recent years, lots of papers have occurred for designing DNA procedures and algorithms to solve various NP-complete problems [3–9].

In this paper, a DNA procedure is introduced for figuring out solutions of the traveling salesman problem: for an edge-weighted graph $G = (V, E)$ find a minimum cost (weight) path. The path must begin at a specified city (vertice) and end there after going through all the other given cites (vertices) only once. For instance, the edge-weighted graph G in Fig. 1 defines such a problem. We assume that the starting and ending vertice is v_1. It is not difficult to find that the path $v_1 \to v_7 \to v_6 \to v_5 \to v_4 \to v_3 \to v_2 \to v_1$ with total weight 21 is a solution to the traveling salesman problem for graph G in Fig. 1.

The rest of this paper is organized as follows. In Section 2, the Adleman–Lipton model is introduced in detail. Section 3 introduces a DNA algorithm for solving the traveling salesman problem and the complexity of the proposed algorithm is described. We give conclusions in Section 4.

2. The Adleman–Lipton model

The DNA operations proposed by Aldeman [1] and Lipton [2] are described below. These operations will be used for figuring out solutions of the traveling salesman problem in this paper. The Adleman–Lipton model: A (test) tube is a set of molecules of DNA (i.e., a multi-set of finite strings over the alphabet {A,C,G,T}). Given a tube, one can perform the following operations [9]: Merge, Copy, Detect, Separation, Selection, Cleavage, Annealing, Denaturation, Discard and Read. Since these

* Corresponding author.
E-mail address: zcwang@shou.edu.cn (Z. Wang).

0096-3003/$ - see front matter Crown Copyright © 2012 Published by Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2012.08.073

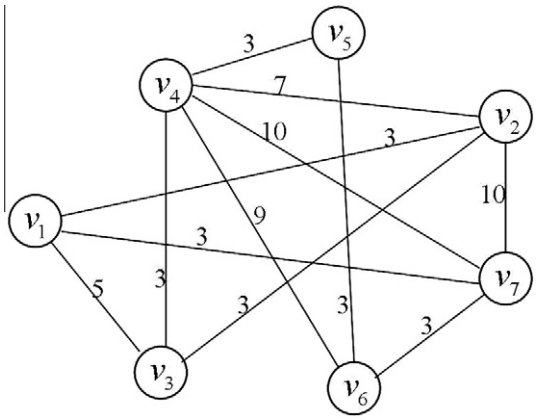

Fig. 1. An edge-weighted graph G with 7 vertices.

10 manipulations are implemented with a constant number of biological steps for DNA strands [3], we assume that the complexity of each manipulation is $O(1)$ steps.

3. DNA algorithm for the traveling salesman problem

Let $G = (V, E)$ be a edge-weighted graph with the set of vertices $V = \{v_k | k = 1, 2, \ldots, n\}$ and the set of edges $E = \{e_{i,j} | 1 \leqslant i, j \leqslant n, i \neq j\}$. Note that both $e_{i,j}$ and $e_{j,i}$ are in E if the vertices v_i and v_j are connected by an edge. Without loss of generality, we assume that v_1 is the starting and ending vertex. Let $|E| = s$, Then, $s \leqslant \frac{1}{2}n(n+1)$.

In the following, we use the symbols $\#, A_k, B_k (k = 1, 2 \ldots, n)$ and $w_{i,j}$ to denote distinct DNA singled strands for which $||\#|| = ||A_k|| = ||B_k||$, where $||\cdot||$ denotes the length of the DNA singled strand. The symbols $A_k B_k (k = 2, 3, \ldots, n)$ denote the vertex v_k. Meanwhile the symbols $\#A_1B_1, A_1B_1\#$ denote the starting and ending vertex v_1 which the symbol $\#$ is the signal of starting and ending. Suppose that all weights in the given graph are commensurable. The DNA singled strands $w_{i,j}$ are used to denote the weights $k_{i,j}$ on the edges $e_{i,j} \in E$ with $||w_{i,j}|| = k_{i,j} w$ where w is a constant, e.g., take $w = 5$ mer in the following discussion, Then, the $||w_{i,j}|| = 5k_{i,j}$. Let $m = \max_{e_{i,j} \in E} 5k_{i,j}$ and $||\#|| = ||A_k|| = ||B_k|| = n * m = t$. For example, for the graph in Fig. 1. We can let $m = \max\{5*3, 5*5, 5*7, 5*9, 5*10\} = 50$ mer, Then $||\#|| = ||A_k|| = ||B_k|| = n * m = t = 7 * 50 = 350$ mer. Let

$$P = \{w_{i,j}, \#A_1B_1, A_1B_1\#, A_kB_k | e_{i,j} \in E, k = 2, 3, \ldots, n\};$$
$$Q = \{\overline{\#}, \overline{A_1}, \overline{B_1}, \overline{B_i w_{i,j} A_j} | e_{i,j} \in E, 1 \leqslant i, j \leqslant n, i \neq j\}.$$

We design the following algorithm to solve the traveling salesman problem and give the corresponding DNA operations as follows:

(1) We choose all possible paths, which start at v_1 and end at v_1.
  (1-1) *Merge*(P, Q);
  (1-2) *Annealing*(P);
  (1-3) *Denaturation*(P);
  (1-4) *Separation*$(P, \{\#A_1B_1\}, T_{tmp})$;
  (1-5) *Discard*(P);
  (1-6) *Separation*$(T_{tmp}, \{A_1B_1\#\}, P)$;
  After the above six steps of manipulations, the singled strands in tube P will encode all paths which start and end at v_1. For example, for the graph in Fig. 1, we have singled strands: $\#A_1B_1w_{1,2}A_2B_2w_{2,7}A_7B_7w_{7,6}\ A_6B_6w_{6,4}A_4B_4w_{4,3}A_3B_3w_{3,1}\ A_1B_1\# \in P$ which correspond to the path $v_1 \to v_2 \to v_7 \to v_6 \to v_4 \to v_3 \to v_1$ respectively. This operation can be finished in $O(1)$ steps since each manipulation above works in $O(1)$ steps.
(2) We choose all possible paths which pass all the other vertices (cities) at least once.
  For $k = 2$ to $k = n$.
  (2-1) *Separation*$(P, \{A_kB_k\}, T)$;
  (2-2) *Discard*(P);
  (2-3) *Merge*(T, P);
  End for
  In the above operations, we get the strands that denote starting and ending specified vertex v_1, synchronously going through all the other vertices at least once. For example, for the graph in Fig. 1, we have singled strands: