Desalination and Water Treatment
www.deswater.com
doi: 10.5004/dwt.2020.25453

185 (2020) 168–174
May

A bio-inspired algorithm for a classical water resources allocation problem based on Adleman–Lipton model

Anjun Deng[a], Zhaocai Wang[b,a,*], Haifeng Liu[b], Tunhua Wu[c,*]

[a]*State Key Laboratory of Simulation and Regulation of River Basin Water Cycle, China Institute of Water Resources and Hydropower Research, Beijing 100048, China, Tel./Fax: +86 10 68786631; email: anjdeng@163.com (A. Deng)*
[b]*College of Information, Shanghai Ocean University, Shanghai 201306, China, Tel./Fax: +86 21 61900624; email: zcwang1028@163.com (Z. Wang), hfliu@shou.edu.cn (H. Liu)*
[c]*First Affiliated Hospital, Wenzhou Medical University, Wenzhou 325035, China, Tel./Fax: +86 577 86689913; email: appll188@163.com (T. Wu)*

Received 22 July 2019; Accepted 8 January 2020

ABSTRACT

The shortage of water resources has always been one of the difficult problems that plagued world economic development and social stability. The imbalance and uneven distribution of water resources between regions exacerbate this contradiction. Therefore, the optimal economical and reasonable water resource design method has become the hope for solving the problem of water resource shortage. As a classic resource allocation and scheduling problem, the rural postman problem has been widely used in practice. The problem includes finding the minimum cost tour after traversing all edges in the required edge subset at least once in an undirected graph, but so far, there is no effective algorithm to solve this problem. Based on bio-heuristic computing model and deoxyribonucleic acid (DNA) molecular operations, a parallel bio-computing algorithm for rural postman problem is proposed in the paper. We use a DNA algorithm to solve the rural postman problem with n vertices in $O(n^2)$ time complexity. Then, we prove the feasibility of the algorithm in theory and verify the authenticity of the algorithm in the simulation experiment. Compared with previous algorithms, DNA computing algorithm not only has higher computational efficiency and lower error rate but also has huge storage capacity and parallel computing ability, which makes the algorithm has better applicability in dealing with large-scale problems.

Keywords: Water resources dispatching; Rural postman problem; Adleman–Lipton model; Bio-inspired algorithm

1. Introduction

As one of the indispensable key resources for human survival and economic development, water plays an inestimable role in promoting social progress. However, not only the spatial distribution of global water resources is uneven, but also the annual precipitation also varies widely. In addition, the lack of water resources protection measures and the impact of human social activities directly lead to a shortage of water resources and uneven distribution in some areas. With the increase of urban population and economic development, the demand for water resources is increasing year by year, but the available water resources are decreasing, and the contradiction between supply and demand of urban water resources is becoming more and more serious. At present, as a densely populated area, water shortages in large cities are an important problem restricting urban development. Actively seeking available water resources and making full use of any form of water resources is the only way to reduce the pressure on cities to use water.

---

* Corresponding authors.

How to more economically and rationally allocate and distribute water resources in limited areas in the urban agglomeration network has become one of the urgent issues for urban development.

Now, about 1.4 billion people in the world lack safe and clean drinking water. It is estimated that about 2.3 billion people in the world will face water shortage by 2025. Therefore, water will become one of the bottlenecks restricting social and global economic development. Due to the shortage of water resources and the rapid deterioration of the ecological environment, the efficient allocation algorithm of water resources has great application value. Some scholars have proposed new algorithms to solve the problem of resource allocation. Davijani et al. [1] used the particle swarm optimization (PSO) algorithm to solve the allocation of water resources based on the maximization of employment in the agriculture and industry sectors, and achieved distinct effects. Yu and Lu [2] presented an innovative comprehensive PPMGWO model for optimal allocation of cross-border river basin water resources. Their results show that the amount of water that can be distributed among all control measures shows an overall trend, and that water resources will be used reasonably and rationally in the future. How to alleviate the crisis and realize the sustainable utilization of water resources by improving the efficiency of water resources utilization is an urgent issue in front of the academic interface. Therefore, it is necessary to quantitatively model actual problems and solve them through efficient calculation algorithms to achieve optimal allocation and sustainable use of regional water resources.

In this paper, we will use an efficient computing algorithm to solve a classical problem of optimal allocation of water resources - rural postal problem (RPP). As an arc routing problem, the rural postman problem is to find a minimum cost tour of a specified arc set in a graph $G = (V,E)$, with the particularity that only a subset E_R ($E_R \subseteq E$) of arcs is required to be passed at least once. Because of its wide application in logistics, transportation, wireless network, and parallel computing, this problem is particularly important. It can be applied to the selection of water sources, the distribution of pipeline paths, the design of water flow, and so on.

The RPP was originally proposed by Orloff [3]. Due to its wide application, RPP has recently attracted considerable attention from some researchers [4–8]. In the past half-century, more and more heuristic methods were proposed to solve the RPP. It includes the following algorithms: exact methods [5–7], heuristic methods [8,9] and metaheuristics [10–12]. Córdoba et al. [6] proposed heuristic algorithms to solve the RPP based on Monte Carlo methods. The basis of the idea is to simulate a vehicle traveling randomly over a graph. Ghiani and Laporte [7] studied the properties of polyhedron and proposed a branch cutting algorithm to solve the problem. The algorithm can solve the optimization problem in medium solution time, involving up to 350 randomly generated vertices. Later, he used a constructive heuristic algorithm for the problem and achieved better results compared with the classical Frederickson procedure [8]. Archetti et al. [11] used a re-optimization algorithm to remove edges from the optimal RPP patrol to ensure that the tightness ratio is equal to 3/2. Computational tests are carried out on these algorithms and Fredericks algorithm, and the performance of these algorithms and Fredericks algorithm running from scratch is compared. However, in the parallel computing system, the rural postman problem is a typical non-deterministic polynomial (NP) problem [12], which is considered to be one of the most challenging problems. With the enlargement of the scale of the problem and the explosion of data volume, it is even impossible if limited computing time is needed.

The RPP can be simply defined as follows: given an undirected graph $G = (V,E,E_R)$ with V representing the set of nodes, E representing the set of edges, c_{ij} is the cost of traversing edge e_{ij} ($e_{ij} \in E$), and $E_R \subseteq E$ representing the set of edges that must be traversed. Then, the RPP is to find a postman tour with the lowest total cost, which starts with a depot, traverses each edge of the E_R at least once, and then returns to the original depot. The solution to the problem has the following limitations:

(1) Every edge in the E_R is traversed at least once;
(2) The route is continuous and closed;
(3) The route starts and ends at the same appointed vertex (depot);
(4) Sum of edge weights (cost) in a route is the minimum.

For example, a graph $G = (V,E,E_R)$, $|V| = 6$, $|E| = 12$ in Fig. 1 is such a problem, and we set vertex (depot) v_1 as the departure and termination one, $E_R = \{e_{12},e_{25},e_{34},e_{56}\}$. After logical calculations, the optimal route of transportation is shown in Table 1.

This paper presents a parallel deoxyribonucleic acid (DNA) algorithm based on Adelman–Lipton model to solve the rural postman problem with $O(n^2)$ time complexity.

The organizational structure of this paper is as follows. Section 2 introduces the basic knowledge of the Adelman–Lipton model. In section 3, DNA parallel algorithms are proposed to solve the rural postman problem. Section 4 analyses the feasibility and computational complexity of the proposed DNA algorithms, and section 5 gives the simulation results of DNA computing. We conclude in section 6.

2. DNA computing and Adleman–Lipton model

Feynman published a visionary report describing the possibility of building microcomputers [13]. After half a

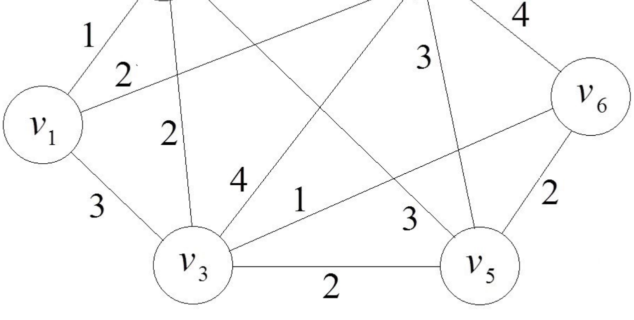

Fig. 1. Edge-weighted graph G with six vertices and 12 edges.

century, computer miniaturization has made remarkable progress, but his goal is still far. Due to the limitations of computer hardware development, traditional algorithms based on silicon computing are difficult to solve some complex NP-hard problems. In view of this situation, some efficient parallel intelligent algorithms have been proposed. Adleman [14] firstly stated and proved that DNA computing could be used to solve the directed Hamiltonian path problem, which is one of the well-known NP problems. Lipton [15] showed that Adelman's experiment can solve the satisfactoriness problem, which is another important NP problem. Biocomputing has been recognized by researchers for its high parallel efficiency, large storage capacity, and low energy consumption. Researchers have established some typical biocomputing models, such as Adleman–Lipton model [14,15], self-assembly model [16], restriction endonuclease model [17], sticker model [18], surface model [19] and hairpin model [20]. Based on these models, DNA programs and algorithms are designed to solve other NP problems [21–35].

Adleman–Lipton model [14,15] uses basic biological experiment operations to achieve the established objectives and uses them as the operational logic of the DNA algorithms. The tools for biological experiments are a series of test tubes including some strings on the alphabet $\{A,C,G,T\}$ in the Adleman–Lipton model. Following is a list of biological experiment operations on test tubes:

1. *Merge* ($T_1,T_2,\dots,T_n$): the strands of test tubes $T_1,T_2,\dots,T_n$ are mixed in tube T_1, and the others are empty.
2. *Annealing (T)*: it can generate all possible double strands of tube T, and remain them in it through some biological operations, such as ligation and polymerase chain reaction (PCR).
3. *Denaturation (T)*: it can separate all the double strands in the test tube T into corresponding single strands:

$$\begin{bmatrix}\alpha\beta\\ \overline{\alpha\beta}\end{bmatrix}\Rightarrow\left[\alpha\beta\right],\left[\overline{\alpha\beta}\right].$$

4. *Separation* (T_1,X,T_2): for a set of strings X and test tube T_1, it moves biological strands containing X from test tube T_1 to test tube T_2 and retains the remaining biological strands in test T_1.
5. *Discard (T)*: it removes the biological strands from the original test tube T and makes it empty.
6. *Copy* ($T_1,T_2,\dots,T_n$): given a test tube T_1, it can replicate the biological strands in test tube T_1 to test tube $T_2,\dots,T_n$.
7. *Sort* (T_1,T_2,T_3): given a test tube T_1, it moves the shortest biological strands in test tube T_1 to test tube T_2, the longest biological strands to test tube T_3, and the rest are still in test tube T_1.
8. *Read (T)*: given a test tube T, it recognizes the composition of biological strands in tube T.

Although DNA computing is different from computer computing, its principles and ideas are the same. Since all the above operations can be performed by a constant biological step in the laboratory, it is reasonable to assume that the complexity of each operation is $O(1)$ [22].

3. DNA computing and Adleman–Lipton model

3.1. Preliminary conception

The basic framework to solve the rural postman problem is as follows: generate various biological chains to represent continuous paths starting and ending from specified vertices, make sure that the routes pass through all established sides at least once, select the shortest routing represented by the biological strands as the optimal solution of the problem. Specifically, the algorithm is implemented in four steps to achieve corresponding purposes.

Step 1: the route is continuous from end to end;
Step 2: route begins and ends at a specified vertex;
Step 3: route through all designated edges at least once;
Step 4: in all feasible routes, the sum of weights of passing edges is the minimum.

3.2. Elemental information representation

In the paper, we use symbol A_i,B_i($i \in \{1,2,\dots,n\}$) represents the information of vertex v_i, $B_iw_{ij}A_j$ ($i,j \in \{1,2,\dots,n\}$) for edge e_{ij}, and symbols B_i, A_j as the starting and ending signs of the edge e_{ij}. Meanwhile, to generate double chains of continuous paths, we also design complementary chains of vertices $\overline{A_iB_i}$ ($i \in \{1,2,\dots,n\}$). Furthermore, to distinguish different path chains, we set the start and end symbols # of the path chain. For searching the optimal path chain, we reasonably set the chain length of each symbol as follows: $||B_iw_{ij}A_j|| = c_{ij} \times t$ mer, c_{ij} is the weight of the edge e_{ij}; $||A_i|| = ||B_i|| = ||\#|| = l$ mer, t and l are positive integers and their values are determined by the scale of the problem [33]; then $||w_{ij}|| = c_{ij} \times t - 2l$ mer, we set w_{ij} consists of a series of certain nucleotides X, the number of which is equal to $c_{ij} \times t - 2l$.

3.3. Algorithms for a general model

For a general rural postman problem having n vertices ($v_1,v_2,\dots,v_n$), m edges ($e_{ij} \in E$) with weight w_{ij}, edge subset E_R, respectively, $E_R \subseteq E$. We use adjacency matrix to denote the connection relationship between different vertices, which $a_{ij} = 1$ means vertex v_i links with v_j and $a_{ij} = 0$ denotes connectionless:

$$M(G)=\left[a_{ij}\right],a_{ij}=\begin{cases}1 & \text{vertex } i \text{ is connected to vertex } j\\ 0 & \text{otherwise}\end{cases}$$

Taking Fig. 1, for example, the adjacency matrix for edge set E and E_R are shown as follows:

$$M(G_1)=\begin{bmatrix}0&1&1&1&0&0\\1&0&1&1&1&0\\1&1&0&1&1&1\\1&1&1&0&1&1\\0&1&1&1&0&1\\0&0&1&1&1&0\end{bmatrix}M(G_2)=\begin{bmatrix}0&1&0&0&0&0\\1&0&0&0&1&0\\0&0&0&1&0&0\\0&0&1&0&0&0\\0&1&0&0&0&1\\0&0&0&0&1&0\end{bmatrix}$$

3.3.1. Generate data pool

We firstly set

$$T_1 = \{\#B_1 w_{1k} A_k,\ A_i w_{ij} B_j,\ B_l w_{l1} A_1\# \mid e_{1k}, e_{ij}, e_{l1} \in E\}$$

$$T_2 = \left\{\overline{A_1\#}, \overline{\#B_1}, \overline{A_k B_k} \mid k = 1,2,\ldots,n\right\}$$

$$T_3 = \left\{\overline{w_{ij}}, \mid e_{ij} \in E\right\}$$

Algorithm 3.1: Generate various routings combination strands.

```
(1-1) Merge($T_1$,$T_2$);
(1-2) Merge($T_1$,$T_3$);
(1-3) Annealing($T_1$);
(1-4) Denaturation($T_1$).
```

3.3.2. Starting and ending vertex

Algorithm 3.2: Select the routings that start and end at the specified vertex.

```
(2-1) Separation($T_1$,{$\#A_1$},$T_4$);
(2-2) Discard($T_1$);
(2-3) Separation($T_4$,{$B_1\#$},$T_5$);
(2-4) Discard($T_4$).
```

3.3.3. Traversing specified edges

The feasible solutions of RPP require that all designated edges must be passed through. We need to check whether the various combinatorial path chains satisfy the condition.

Algorithm 3.3: Determine whether all specified edges have passed.

```
For $i = 1$ to $i = n$
    For $j = 1$ to $j = n$
(3-1) If($b_{ij} = 1$)
(3-2-1) Copy($T_5$,$T_6$);
(3-2-2) Separation($T_5$,$A_i w_{ij} B_j$,$T_7$);
(3-2-3) Separation($T_6$,$A_j w_{ji} B_i$,$T_8$);
(3-2-4) Merge($T_7$,$T_8$);
(3-2-5) Discard($T_5$);
(3-2-6) Discard($T_6$);
    Else
(3-3) Continue.
    End for
End for
```

3.3.4. Finding the optimal routing

The shortest feasible solution path chains mean having the minimum edge weight sum. So, we have to judge and pick them out.

Algorithm 3.4: search solutions strands.

```
(4-1) Sort($T_7$,$T_9$,$T_{10}$);
(4-2) Read($T_9$).
```

Finally, we can get the exact solutions strands for the rural postman problem.

4. Feasibility and complexity of the DNA algorithms

The following theorems prove that our proposed algorithms can solve the rural postman problem in a certain time complexity and space strands length range.

Theorem 1. The rural postman problem could be solved by using our designed DNA computing operations.

Proof. We use DNA chains to represent different vertices and edges information and generate various routing combinations in the data reaction pool. Next, in the rural postman problem, every routing should start from the designated city and terminate in the same vertex. We set start and terminate the vertex v_1, and select the corresponding DNA strands by selecting strands with symbols $\#B_1$ and $A_1\#$ in Algorithm 3.2. Since the postman is required to pass through all the designated edges at least once in the RPP, the path strands should contain all the symbols of the designated edges. In Algorithm 3.3, we select feasible strands passing through all specified edges.

The strands S_1 after Algorithm 3.3 can be described: $\#B_1 w_{1i_1} A_{i_1} B_{i_1} w_{i_1 i_2} A_{i_2} \ldots B_{i_p} w_{i_p i_{p+1}} A_{i_{p+1}} \ldots B_{i_k} w_{i_k 1} A_1\# \qquad i_p \in \{1,2,\ldots,n\}$.

Besides, we initially set the chain length as $||A_i w_{ij} B_j|| = c_{ij} \times t$, $||\#|| = t$ mer, and $||A_i|| = ||B_i|| = l$ mer ($1 \le i \le n$), so the length of DNA strands S_1 is:

$$\|S_1\| = \|\#\| + \underbrace{\|B_1\| + \|w_{1i_1}\| + \|A_{i_1}\|}_{c_{1i_1}{}^*t} + \underbrace{\|B_{i_1}\| + \|w_{i_1 i_2}\| + \|A_{i_2}\|}_{c_{i_1 i_2}{}^*t} + \ldots +$$

$$\underbrace{\|B_{i_p}\| + \|w_{i_p i_{p+1}}\| + \|A_{i_p}\|}_{c_{i_p i_{p+1}}{}^*t} + \ldots \underbrace{\|B_{i_k}\| + \|w_{i_k 1}\| + \|A_1\|}_{c_{i_k 1}{}^*t} + \|\#\|$$

$$= t + c_{1i_1} \times c_{i_1 i_2} \times t + \ldots + c_{i_p i_{p+1}} \times t + \ldots + c_{i_k 1} \times t + t$$

$$= \left(c_{1i_1} + c_{i_1 i_2} + \ldots + c_{i_p i_{p+1}} + \ldots + c_{i_k 1} + 2\right)t$$

$$\propto \left(c_{1i_1} + c_{i_1 i_2} + \ldots + c_{i_p i_{p+1}} + \ldots + c_{i_k 1}\right)$$

Therefore, the length of the strand in S_1 is proportional to the sum of the edge weights of the routing which it represents. At the same time, in all possible routes, the solution of the rural postman problem has the minimum sum of edge weights. In Algorithm 3.4, we can compare the length of the strands to obtain the optimal solution of the problem.

Theorem 2. The rural postman problem can be solved in $O(n^2)$ time complexity by using DNA computing.

Proof. Since the complexity of each biological operation is in O(1) time complexity [22,36–43], the operations of the total algorithm are completed within a limited time complexity. The total time complexity T of the DNA algorithm is as follows:

T(Algorithm 3.1) = $O(1)$;

T(Algorithm 3.2) = $O(1)$;

T(Algorithm 3.3) = $O(n^2)$;

T(Algorithm 3.4) = $O(1)$;

T = T(Algorithm 3.1) + T(Algorithm 3.2) + T(Algorithm 3.3) + T(Algorithm 3.4)

= $O(1)$+ $O(1)$+ $O(n^2)$+ $O(1)$

= $O(n^2)$

In short, we can solve the rural postman problem in $O(n^2)$ time scale complexity.

5. DNA computing simulation experiment

The calculation of DNA depends on the operational accuracy of biochemical molecules, which will lead to errors in the application of biochemical reactions, which will accumulate and expand in experiments. Therefore, the validity of sequence design is an important guarantee for the reliability of DNA computing. To obtain better computational performance in hybrid reactions, we use the sequence design method in reference [44–46].

In this paper, a molecular computing tool, Biopython, is used as a development platform to generate DNA sequences suitable for laboratory algorithms. Braich's program runs on Windows 10 using Intel CPU, 8GB main memory and Visual C++ compiler. The encoding program generates DNA sequences for solving RPP. When adding a new DNA sequence, it is necessary to determine whether the chain satisfies the restriction conditions. If the generated DNA sequence cannot pass any restrictions, the program will regenerate new other sequences. If these restrictions are met, DNA sequences can be accepted.

So taking the rural postman problem in Fig. 1 as a test example, the programs generate random sequences to form A_i, B_i, #, X represented in Table 2. We set $t = 7$ and $||A_i|| = ||B_i|| = 3$, due to $||A_i w_{ij} B_j|| = c_{ij} \times t$, so $||w_{ij}|| = c_{ij} \times 7 - ||A_i|| - ||B_i|| = 7c_{ij} - 6$ and $||\#|| = t = 7$. Table 3 shows the DNA edge sequences generated by the Braich's program.

Table 1
Optimal postman tours

	Optimal route of transportation	Weight sum
Tour 1	$v_1 \to v_4 \to v_3 \to v_6 \to v_5 \to v_2 \to v_1$	13
Tour 2	$v_1 \to v_2 \to v_5 \to v_6 \to v_3 \to v_4 \to v_1$	13

Table 2
Sequences chosen to represent A_i, B_i, # and X ($i \in \{1,2,\ldots,6\}$) for the rural postman problem

Bit	3′–5′ DNA sequence	Bit	3′–5′ DNA sequence
A_1	TTC	B_1	CTG
A_2	ACA	B_2	ATA
A_3	TAT	B_3	GTC
A_4	TAA	B_4	CGA
A_5	ATC	B_5	TGC
A_6	GCA	B_6	AGC
#	TACCTTA	X	G

Table 3
Sequences chosen to represent the elements $B_i w_{ij} B_j$ ($e_{ij} \in E$) for the rural postman problem

Edge	3′–5′ DNA sequence	Edge	3′–5′ DNA sequence
$B_1w_{12}A_2$	CTGGACA	$B_1w_{13}A_3$	CTGGGGTAT
$B_1w_{14}A_4$	CTGGGTAA	$B_2w_{21}A_1$	ATAGTTC
$B_2w_{23}A_3$	ATAGGTAT	$B_2w_{24}A_4$	ATAGGGTAA
$B_2w_{25}A_5$	ATAGGGATC	$B_3w_{31}A_1$	GTCGGGTTC
$B_3w_{32}A_2$	GTCGGACA	$B_3w_{34}A_4$	GTCGGGGTAA
$B_3w_{35}A_5$	GTCGGATC	$B_3w_{36}A_6$	GTCGGCA
$B_4w_{41}A_1$	CGAGGTTC	$B_4w_{42}A_2$	CGAGGGACA
$B_4w_{43}A_3$	CGAGGGGTAT	$B_4w_{45}A_5$	CGAGGGATC
$B_4w_{46}A_6$	CGAGGGGGCA	$B_5w_{52}A_2$	TGCGGGACA
$B_5w_{53}A_3$	TGCGGTAT	$B_5w_{54}A_4$	TGCGGGTAA
$B_5w_{56}A_6$	TGCGGGCA	$B_6w_{63}A_3$	AGCGTAT
$B_6w_{64}A_4$	AGCGGGGTAA	$B_6w_{65}A_5$	AGCGGATC

It can also be used to calculate the enthalpy, entropy and free energy of the binding of each probe to the corresponding region in the library chains, which is shown in Table 4.

The programs calculate the average and standard deviations of enthalpy, entropy and free energy in all interactions. The energy level is described in Tables 5 and 6 gives the solution strands of the rural postman problem.

6. Conclusions

Based on Adelman–Lipton model, we use DNA algorithms to solve the rural postman problem through biological operations. Moreover, we simulated DNA experiments to solve the rural postmen problem. Compared with the previous algorithms, the algorithm has the following advantages: firstly, because of the development of computer programs to generate good DNA sequences in the solution space of the rural postman problem, the hybrid error rate of the algorithm is actually lower. Secondly, in the proposed DNA algorithm, the computational complexity of element information and the length of DNA chain increase linearly with the scale of the actual problem for the rural postman problem, which can be completed in $O(n^2)$ time complexity, faster than other previous algorithms, compared to $O(n^4)$ (n is number of vertices) [7], $O(n^3)$ [8], $O(m+n\log(n))$

Table 4
Energies for of binding each probe to its corresponding region on a library strand

Edges	Enthalpy energy *H*	Entropy energy *S*	Free energy *G*
$B_1w_{12}A_2$	107.3	270	25.6
$B_1w_{13}A_3$	107.9	275.5	25.1
$B_1w_{14}A_4$	102	256.1	23.8
$B_2w_{21}A_1$	97.6	249.9	23.3
$B_2w_{23}A_3$	111.9	289.3	25.2
$B_2w_{24}A_4$	104.1	274.9	23.2
$B_2w_{25}A_5$	112.7	295	26.1
$B_3w_{31}A_1$	97	248.3	23.5
$B_3w_{32}A_2$	111.2	279.2	26.1
$B_3w_{34}A_4$	104.1	271.2	22.8
$B_3w_{35}A_5$	111	283.1	26.3
$B_3w_{36}A_6$	104.5	265.9	24.9
$B_4w_{41}A_1$	105.6	276.6	23.5
$B_4w_{42}A_2$	115.3	290.7	27.8
$B_4w_{43}A_3$	103.1	266.5	23.3
$B_4w_{45}A_5$	105.9	272.1	24.4
$B_4w_{46}A_6$	113.3	281.6	26.5
$B_5w_{52}A_2$	101.7	265.8	22.8
$B_5w_{53}A_3$	99.1	253.5	23.3
$B_5w_{54}A_4$	108.1	267.9	25.3
$B_5w_{56}A_6$	96.5	240.5	23.1
$B_6w_{63}A_3$	100.1	248.2	23.4
$B_6w_{64}A_4$	118.3	297.2	26.4
$B_6w_{65}A_5$	101.4	251.9	24.9

Table 5
Energies over all probe/library strand interactions

	Enthalpy energy *H*	Entropy energy *S*	Free energy *G*
Average	105.821	269.621	24.608
Standard deviation	6.0061	15.7720	1.4304

Table 6
DNA sequences chosen to represent solutions to the rural postman problem in Fig. 1

3′–TACCTTACTGGGTAACGAGGGGTATGTCGGCAAGCG
GATCTGCGGGACAATAGTTCTACCTTA–5′

3′–TACCTTACTGGACAATAGGGATCTGCGGGCAAGCG
TATGTCGGGGTAACGAGGTTCTACCTTA–5′

(*m* is number of edges) [11] and so on. In addition, this study has certain reference value and application meaning to comprehensive management and water resource allocation in some river basins. Simultaneously, the solution of the rural postman problem has practical application value to the scheduling and allocation of water resources in agricultural production and provides a strong calculation basis for the rational allocation of water resources in the future.

However, DNA computing does not have a perfect computing model at present, and there are still some shortcomings. There are mainly the following aspects: first, the current DNA computing model can only be used to solve specific problems, and there is no generality. Second, the operation of a DNA experiment is not easy to control, which will affect the accuracy of experimental results.

The proposed DNA algorithm can further solve the scalability problems, including windy rural postman problem, mixed rural postman problem, and mixed general routing problem. The results show that the DNA algorithm can solve large-scale instances with reasonable time complexity. DNA computing, as a simple and effective new meta-heuristic algorithm, provides a method for further exploration.

Acknowledgments

It was supported by the National Key Research and Development Project of China (Grant No. 2018YFC0407404) and the Open Research Fund of State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research (grant No. IWHR-SKL-KF201905). The project was also financially supported by the National natural science foundation of China (Grant No. 11701363).

References

[1] M.H. Davijani, M.E. Banihabib, A.N. Anvar, S.R. Hashemi, Optimization model for the allocation of water resources based on the maximization of employment in the agriculture and industry sectors, J. Hydrol., 533 (2016) 430–438.
[2] S. Yu, H. Lu, An integrated model of water resources optimization allocation based on projection pursuit model-Grey wolf optimization method in a transboundary river basin, J. Hydrol., 559 (2018)156–165.
[3] C.S. Orloff, A fundamental problem in vehicle routing, Networks, 4 (1974) 35–64.
[4] N. Christofides, V. Campos, A. Corbern, An Algorithm for the Rural Postman Problem on a Directed Graph//Netflow at Pisa, Springer Berlin, Heidelberg, 1986, pp. 155–166.
[5] A. Corberán, J.M. Sanchis, A polyhedral approach to the rural postman problem, Eur. J. Oper. Res., 79 (1994) 95–114.
[6] P.F. de Córdoba, L.M.G Raffi, J.M. Sanchis, A heuristic algorithm based on Monte Carlo methods for the rural postman problem, Comput. Oper. Res., 25 (1998) 1097–1106.
[7] G. Ghiani, G. Laporte, A branch-and-cut algorithm for the undirected rural postman problem, Math. Program., 87 (2000) 467–481.
[8] G. Ghiani, L. Demetrio, R. Musmanno, A constructive heuristic for the undirected rural postman problem, Comput. Oper. Res., 33 (2006) 3450–3457.
[9] G.W. Groves, J.H. Van Vuuren, Efficient heuristics for the rural postman problem, Orion, 21 (2005) 33–51.
[10] M.L. Pérez-Delgado, A Solution to the Rural Postman Problem Based on Artificial Ant Colonies, Conference of the Spanish Association for Artificial Intelligence, Springer, Berlin, Heidelberg, 2007, pp. 220–228.
[11] C. Archetti, G. Guastaroba, M.G. Speranza, Reoptimizing the rural postman problem, Comput. Oper. Res., 40 (2013) 1306–1313.
[12] M.R Garey, D.S. Johnson, Computers and Intractability[M], Freeman, San Francisco, 1979.

[13] R.P. Feynman, In: D.H. Gilbert, ed., Minaturization, Reinhold, New York, 1961, pp. 282–296.

[14] L.M. Adleman, Molecular computation of solution to combinatorial problems, Science, 266 (1994) 1021–1024.

[15] R.J. Lipton, DNA solution of HARD computational problems, Science, 268 (1995) 542–545.

[16] E. Winfree, F. Liu, L.A. Wenzler, N.C. Seeman, Design and self-assembly of two dimensional DNA crystals, Nature, 394 (1998) 539–544.

[17] Q. Ouyang, P.D. Kaplan, S. Liu, A. Libchaber, DNA solution of the maximal clique problem, Science, 278 (1997) 446–449.

[18] S. Roweis, E. Winfree, R. Burgoyne, N.V. Chelyapov, M.F. Goodman, P.W.K. Rothemund, L.M. Adleman, A sticker based model for DNA computation, J. Comput. Biol., 5 (1998) 615–629.

[19] D.M. Xiao, W.X. Li, Z.Z. Zhang, L. He, Solving maximum cut problems in the Adleman-Lipton model, BioSystems, 82 (2005) 203–207.

[20] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, M. Hagiya, Molecular computation by DNA hairpin formation, Science, 288 (2000) 1223–1226.

[21] W.X. Li, D.M. Xiao, L. He, DNA ternary addition, Appl. Math. Comput., 182 (2006) 977–986.

[22] D.M. Xiao, W.X. Li, J. Yu, X.D. Zhang, Z.Z. Zhang, L. He, Procedures for a dynamical system on $\{0,1\}^n$ with DNA molecules, BioSystems, 84 (2006) 207–216.

[23] W.X. Li, E.M. Patrikeev, D.M. Xiao, A DNA Algorithm for the maximal matching problem, Autom. Remote Control, 76 (2015) 1797–1802.

[24] Z. Wang, D. Huang, H. Meng, C. Tang, A new fast algorithm for solving the minimum spanning tree problem based on DNA molecules computation, Biosystems, 114 (2013) 1–7.

[25] M.Y. Guo, W.L. Chang, M. Ho, J. Lu, J.N. Cao, Is optimal solution of every NP-complete or NP-hard problem determined from its characteristic for DNA-based computing, BioSystems, 80 (2005) 71–82.

[26] W.-L. Chang, K.W. Lin, J.-C. Chen, C.-C. Wang, L.C. Lu, M. Guo, M. Ho, Molecular Solutions of the RSA Public-key Cryptosystem on a DNA-based Computer, J. Supercomput., 61 (2012) 642–672.

[27] W.-L. Chang, T.-T. Ren, J. Luo, M. Feng, M. Guo, Quantum algorithms for bio-molecular solutions of the satisfiability problem on a quantum machine, IEEE Trans. Nanobiosci., 7 (2008) 215–222.

[28] Z.C. Wang, J. Tan, D.M. Huang, Y. Ren, Z. Ji, A biological algorithm to solve the assignment problem based on DNA molecules computation, Appl. Math. Comput., 244 (2014) 183–190.

[29] Z. Wang, Z. Ji, X. Wang, T. Wu, W. Huang, A new parallel DNA algorithm to solve the task scheduling problem based on inspired computational model, BioSystems, 162 (2017) 59–65.

[30] X.C. Liu, X.F. Yang, S.L. Li, Y. Ding, Solving the minimum bisection problem using a biologically inspired computational model, Theor. Comput. Sci., 411 (2010) 888–896.

[31] Z. Wang, J. Pu, L. Cao, J. Tan, A parallel biological optimization algorithm to solve the unbalanced assignment problem based on DNA molecular computing, Int. J. Mol. Sci., 16 (2015) 25338–25352.

[32] W.L. Chang, T.T. Ren, M. Feng, Quantum algorithms and mathematical formulations of biomolecular solutions of the vertex cover problem in the finite-dimensional hilbert space, IEEE Trans. Nanobiosci., 14 (2014) 121–128.

[33] H. Zhao, J. Zheng, J. Xu, W. Deng. Fault diagnosis method based on principal component analysis and broad learning system, IEEE Access, 7 (2019) 99263–99272.

[34] W. Deng, R. Yao, H. Zhao, X. Yang, G. Li, A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm, Soft Comput., 23 (2019) 2445–2462.

[35] W. Deng, H. Zhao, L. Zou, G. Li, X. Yang, D. Wu, A novel collaborative optimization algorithm in solving complex optimization problems, Soft Comput., 21 (2017) 4387–4398.

[36] Z. Ji, Z. Wang, X. Bao, X. Wang, T. Wu, Research on water resources optimal scheduling problem based on parallel biological computing, Desal. Water Treat., 111 (2018) 88–93.

[37] W. Deng, J. Xu, H. Zhao, An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem, IEEE Access, 7 (2019) 20281–20292.

[38] R.S. Braich, C. Johnson, P.W.K. Rothemund, N. Chelyapov, L.M. Adleman, Solution of a 20-variable 3-SAT problem on a DNA computer, Science, 296 (2002) 499–502.

[39] R.B.A. Bakar, J. Watada, W. Pedrycz, DNA approach to solve clustering problem based on a mutual order, Biosystems, 91 (2008) 1–12.

[40] K.H. Zimmermann, Z. Ignatova, I. Martínez-Pérez, DNA Computing Models[M], Springer-Verlag US, 2008, pp. 146–147.

[41] Z. Ji, Z. Wang, A. Deng, W. Huang, T. Wu, A new parallel algorithm to solve one classic water resources optimal allocation problem based on inspired computational model, Desal. Water Treat., 160 (2019) 214–218.

[42] Z. Wang, D. Huang, J. Tan, T. Liu, K. Zhao, L. Li, A parallel algorithm for solving the n-queens problem based on inspired computational model, Biosystems, 131 (2015) 22–29.

[43] H.Y. Zhang, X.Y. Liu, A CLIQUE algorithm using DNA computing techniques based on closed-circle DNA sequences, Biosystems, 105 (2011) 73–82.

[44] M. Yamamura, Y. Hiroto, T. Matoba, Solutions of shortest path problems by concentration control, Lect. Notes Comput. Sci., 2340 (2002) 231–240.

[45] R.S. Braich, C. Johnson, P.W.K. Rothemund, D. Hwang, N. Chelyapov, L.M. Adleman, Solution of a Satisfiability Problem on a Gel-based DNA Computer[C], International Workshop on DNA-Based Computers, Springer, Berlin, Heidelberg, 2000, pp. 27–42.

[46] Z. Wang, X. Ren, Z. Ji, W. Huang, T. Wu, A novel bio-heuristic computing algorithm to solve the capacitated vehicle routing problem based on Adleman–Lipton model, Biosystems, 184 (2019) 103997.