Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090241

A Genetic Algorithm to Solve the Subset Sum Problem based on Parallel Computing

Lei Li[1], *Kai Zhao*[2,*] *and Zuwen Ji*[3]

[1] Department of civil engineering, Xi'an University of Architecture & Technology, Xi'an 710055, PR China
[2] Academic Affair Office, Pingdingshan University, Pingdingshan 467000, PR China
[3] State Key Laboratory of Simulation and Regulation of River Basin Water Cycle, China Institute of Water Resources and Hydropower Research, Beijing 100048, P. R. China

Received: 12 Jun. 2014, Revised: 10 Sep. 2014, Accepted: 12 Sep. 2014
Published online: 1 Mar. 2015

Abstract: The subset sum problem is to find subsets in a given number set, meanwhile number sum of the subset is equal to appointed value. It is a classical NP-complete problem in graph theory. It can be solved by the electronic computer in exponential time. In this paper, we consider a DNA procedure for solving the subset sum problem in the Adleman-Lipton model. The procedure works in $O(n)$ steps for the subset sum problem of an undirected graph with n vertices. The innovation of the procedure is the ingenious choice of the vertices strands' length, which can get the solution of the problem in proper length range and simultaneity simplify the complexity of the computation.

Keywords: genetic algorithm; The subset sum problem; parallel computing

1 Introduction

DNA computating is calculated by molecular biology method to solve complex mathematical problems, which is a combined product of biology, mathematics, computer and information science. According to certain rules in the original problem of data information onto the DNA molecular chain, DNA computating uses the double helix structure and DNA molecule complementary principle of information coding, get the solution space data pool of the problem. After a series of biochemical reaction controlled parallel manipulation of DNA molecules, it excludes non feasible solution chain. Finally, it get the ultimate DNA chain using molecular biological extraction technique, which is the solution of the problem. As the pathbreaking work of DNA computation, Leonard Adleman [1] succeeded in solving an instance of the Directed Hamiltonian Path Problem solely by manipulating DNA strings, and also illustrated the potential parallel ability of DNA computating in 1994. Lipton [2] testified that the same method could be used to solve another NP-complete problem (satisfiability problem) in 1995. Since then, DNA computing, as a interdisciplinary science using DNA molecular biotechnologies to solve conundrum problems of computer science and computational mathematics, has a wide application prospect in solving difficult problems. Huge storage capacity, massive parallelism and low energy consumption are primary advantages of DNA computing. The advantages imply that we can utilize DNA molecule to solve harder, larger problems such as NP-complete problems in linearly increasing time complexity, in contrast to the exponentially increasing time complexity required by electronic computers. Recently, DNA computing has attracted more and more attention and interest from research scholars. In recent years, lots of papers have occurred for designing DNA procedures and algorithms to solve various NP-complete problems [3,4,5,6,7,8,9,10]. However, most of the previous works in DNA computing do not require the consideration of the representation of numerical data in DNA strands. In fact, many practical applications in the real world involve edge-weighted or vertice-weighted graph problems such as shortest path problem, subset sum problem, etc. Therefore, representation of numerical data in DNA strands is an important issue toward expanding the capability of DNA computing to solve numerical optimization problems. There have been some previous works to represent the numerical data with DNA.

* Corresponding author e-mail: jerrys76@163.com

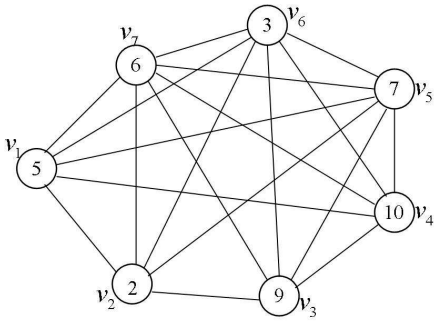

Fig. 1. An vertice-weighted graph G with 7 vertices

Narayanan et al. [11] presented a conceptual encoding method that represents costs with the lengths of DNA strands. Shin et al. [12] proposed a method for representing the real numbers in fixed-length DNA strands by varying the number of hydrogen bonds. Yamamura et al. [13] proposed a concentration control method which encoded the numerical data by means of the concentrations of DNA strands.

In this paper, a DNA procedure is introduced for figuring out solutions of the subset sum problem: for an vertice-weighted graph $G = (V,E)$ find vertice subsets. The subsets contain some vertices of V, which sum of vertices weight is equal to specified L. For instance, the vertices-weighted graph G in Fig. 1 defines such a problem. We assume that the specified L is 23. It is not difficult to find that the vertice subset $\{v_1, v_2, v_5, v_6, v_7\}$ with corresponding weight $\{5,2,7,3,6\}$ is one solution to the subset sum problem for graph G in Fig. 1. We encode the numerical data by means of the lengths of DNA strands, the same way as that in [11]. A DNA procedure is formally presented by means of the DNA operations proposed by Adleman [1] and Lipton [2].

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 subset sum problem and the complexity of the proposed algorithm is described. We give conclusions in Section 4.

2 The genetic algorithm model

DNA computating is calculated by molecular biology method to solve complex mathematical problems, which is a combined product of biology, mathematics, computer and information science. According to certain rules in the original problem of data information onto the DNA molecular chain, DNA computating uses the double helix structure and DNA molecule complementary principle of information coding, get the solution space data pool of the problem. After a series of biochemical reaction controlled parallel manipulation of DNA molecules, it excludes non feasible solution chain. Finally, it get the ultimate DNA chain using molecular biological extraction technique, which is the solution of the problem. As the pathbreaking work of DNA computation, Leonard Adleman [1] succeeded in solving an instance of the Directed Hamiltonian Path Problem solely by manipulating DNA strings, and also illustrated the potential parallel ability of DNA computating in 1994. Lipton [2] testified that the same method could be used to solve another NP-complete problem (satisfiability problem) in 1995. Since then, DNA computing, as a interdisciplinary science using DNA molecular biotechnologies to solve conundrum problems of computer science and computational mathematics, has a wide application prospect in solving difficult problems. Huge storage capacity, massive parallelism and low energy consumption are primary advantages of DNA computing. The advantages imply that we can utilize DNA molecule to solve harder, larger problems such as NP-complete problems in linearly increasing time complexity, in contrast to the exponentially increasing time complexity required by electronic computers. Recently, DNA computing has attracted more and more attention and interest from research scholars.

A DNA(deoxyribonucleic acid) is a polymer, which is strung together from monomers called deoxyribonucleotides [15]. Distinct nucleotides are detected only with their bases. Those bases are, respectively, abbreviated as adenine (A) , guanine (G), cytosine (C), and thymine (T). Two strands of DNA can form (under appropriate conditions) a double strand, if the respective bases are the Watson-Crick complements of each other: A matches T and C matches G; also $3'$ end matches $5'$ end, e.g., the singled strands $5'CTGCAGTACACC3'$ and $3'GACGTCATGTGG5'$ can form a double strand. We also call the strand $3'GACGTCATGTGG5'$ as the complementary strand of $5'CTGCAGTACACC3'$ and simply denote $3'GACGTCATGTGG'$ by $\overline{CTGCAGTACACC}$. The length of a single stranded DNA is the number of nucleotides comprising the single strand. Thus, if a single stranded DNA includes 20 nucleotides, it is called a 20*mer*. The length of a double stranded DNA (where each nucleotide is base paired) is counted in the number of base pairs. Thus, if we make a double stranded DNA from a single stranded 20 mer, then the length of the double stranded DNA is 20 base pairs, also written as 20 bp.

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 genetic algorithm 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:

(1) *Merge* (T_1, T_2): for two given test tubes T_1, T_2, it stores the union $T_1 \bigcup T_2$ in T_1 and leaves T_2 empty;

(2) *Copy* (T_1,T_2): for a given test tube T_1, it produces a test tube T_2 with the same contents as T_1 ;

(3) *Detect (T)*: given a test tube T, it outputs "yes" if T contains at least one strand, otherwise, outputs "no";

(4) *Separation* (T_1,X,T_2): for a given test tube T_1 and a given set of strings X, it removes all single strands containing a string in X from T_1, and produces a test tube T_2 with the removed strands;

(5) *Selection* (T_1,L,T_2): for a given test tube T_1 and a given integer L, it removes all strands with length L from T_1, and produces a test tube T_2 with the removed strands;

(6) *Cleavage* $(T,\gamma_0\gamma_1)$: for a given test tube T and a string of two (specified) symbols $\gamma_0\gamma_1$, it cuts each double strand containing $\begin{bmatrix}\gamma_0\gamma_1\\ \overline{\gamma_0\gamma_1}\end{bmatrix}$ in T into two double strands as follows:

$$\begin{bmatrix}\alpha_0\gamma_0\gamma_1\beta_0\\ \alpha_1\overline{\gamma_0\gamma_1}\beta_1\end{bmatrix}\Longrightarrow\begin{bmatrix}\alpha_0\gamma_0\\ \alpha_0\overline{\gamma_0}\end{bmatrix},\begin{bmatrix}\gamma_1\beta_0\\ \overline{\gamma_1}\beta_0\end{bmatrix};$$

(7)*Annealing (T)*: for a given test tube T, it produces all feasible double strands in T. The produced double strands are still stored in T after annealing;

(8) *Denaturation (T)*: for a given test tube T, it dissociates each double strand in T into two single strands;

(9) *Discard (T)*: for a given test tube T, it discards the tube T;

(10) *Read(T)*: for a given tube T, the operation is used to describe a single molecule, which is contained in the tube T. Even if T contains many different molecules each encoding a different set of bases, the operation can give an explicit description of exactly one of them.

(11) *Append(T,Z)*: for a given test tube T and a given short DNA singled strand Z it appends Z onto the end of every strand in the tube T;

Since these ten manipulations are implemented with a constant number of biological steps for DNA strands [15], we assume that the complexity of each manipulation is $O(1)$ steps.

3 DNA algorithm for the Subset Sum Problem

Let $G=(V,E)$ be a vertice-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 the weight of vertice v_k is k_i. The vertices v_i and v_j can be divided up by the edges $e_{i,j}$ or $e_{j,i}$ in E. Without loss of generality, we assume that specified length is L.

In the following, we use the distinct DNA singled strands symbols $\#A_k\#$, $\#\overline{A_k}\#$ $(k=1,2,\ldots,n)$ to denote the vertice v_k for which $||\#||=||\overline{\#}||$, where $||\cdot||$ denotes the length of the DNA singled strand. Simultaneity the symbol $\#,\overline{\#}$ is the signal of division between different vertices. Suppose that all weights of vertices in the given graph are commensurable. The DNA singled strands A_i and $\overline{A_i}$ are both used to denote the weights k_i on the vertices $v_i\in V$ with $||A_i||=||\overline{A_i}||=k_i$, e.g., take $||A_1||=||\overline{A_1}||=5mer$ in Fig. 1, Then Let $m=\max_{v_i\in V}k_i$ and $||\#||=||\overline{\#}||=n*m=t$. For example, for the graph in Fig. 1, We can let $m=\max\{5,2,9,10,7,3,6\}=10mer$, Then $||\#||=||\overline{\#}||=n*m=t=7*10=70mer$. Let

$$P=\{\#A_i\overline{\#},\#\overline{A_i\#}|v_i\in V,i=1,2,\cdots,n\}$$

$$Q=\{x\}$$

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

(1)We choose all possible subsets of vertices.
(1-1)$Annealing(P)$;
(1-2)$Denaturation(P)$;
(1-3)$Cleavage(P)$;
After the above three steps of manipulations, the singled strands in tube P will encode all sets of vertices. For example, for the graph in Fig. 1, we have singled strands: $\#A_1\overline{\#A_3}\#A_5\overline{\#A_7}\#A_2\overline{\#A_6}\#$ which correspond to set of vertices $\{v_1,v_3,v_5,v_7,v_2,v_6\}$ with the weight sum 32 respectively. This operation can be finished in $O(1)$ steps since each manipulation above works in $O(1)$ steps.

(2)Each singled strand in tube P denotes one possible vertice set. Even because the effort of biotechnologies, we meantime get some strands which denote beyond subsets of vertices. For example, for the graph in Fig. 1, we have singled strands: $\#A_1\#\overline{A_2}\#A_4\#\overline{A_3}\#A_2\#\overline{A_5}\#A_4\#$ which correspond to set of vertices $\{v_1,v_2,v_4,v_3,v_2,v_5,v_4\}$. Of course, it can not be the optimum solution because the set contains the vertice v_2 twice. To find proper length subset sum, we append the vertice length information on each strand. So we can get all possible strands which contain whole vertices length information at least once.
For $k=1$ to $k=n$
(2-1)$Separation(P,\{A_k\},T_1)$;
(2-2)$Separation(P,\{\overline{A_k}\},T_2)$;
(2-3)$Append(P,x)$;
(2-4)$Merge(P,T_1)$;
(2-5)$Merge(P,T_2)$;
(2-6)$Discard(T_1)$;
(2-7)$Discard(T_2)$;
End for
In the above operations, we get the strands that contain all vertices length information at least once. For example, for the graph in Fig. 1, we have singled strands: $\#A_1\#\overline{A_5}\#A_2\#\overline{A_4}\#A_3\#xx\in P$ which denote the set $\{v_1v_5v_2v_4v_3\}$. We append x twice on the strands for that original strands omit vertices v_6,v_7 length information . In the above operation we use a "For" clause. Thus this operation can be finished in $O(n)$

steps since each single manipulation above works in $O(1)$ steps.

(3)First of all, We set the length of DNA stands as following: $m = \max_{v_i \in V} k_i$ and $||\#|| = ||\overline{\#}|| = ||x|| = n * m = t$. If the veryice $v_i \overline{\in} E_1$, we append x with length t on the correspond strands. Consequently the length of DNA strands which denote containing every vertice length information only once in tube P must be between $(n+1)t$ and $(n+2)t$. We can get the strands denoting the optimum solution in this length range. This is done by the following manipulations.

(3-1)$Selection(P,(n+1)t+L,T)$;

(3-2)$Detect(T)$

If $Detect(T)$ is "yes", then the solution of subset sum problem is obtained. This operation can be finished in O(1) steps since each manipulation above works in O(1) steps.

(4)Finally the *Read* operation is applied to giving the exact vertices subset in the subset sum problem. For example, for the graph in Fig. 1, the vertices subsets are $\{v_1,v_2,v_5,v_6,v_7\}$,$\{v_1,v_2,v_4,v_7\}$, $\{v_1,v_3,v_6,v_7\}$, $\{v_1,v_2,v_3,v_5\}$ and $\{v_4,v_5,v_7\}$ with total weights sum 23.

(4-1)$read(T)$;

The following theorem tells that the algorithm proposed above really can get solutions of the subset sum problem in $O(n)$ steps using DNA molecules.

Theorem 1. *The solutions of subset sum problems for a graph with n vertices can be figured out in O(n) steps using DNA molecules.*

Proof. After the operations of first step, all the singled strands in tube P denote all sets of vertices. Then strands can be described:

$$\#A_{i_1}\overline{\#A_{i_2}}\cdots\#A_{i_{j-1}}\overline{\#A_{i_j}}\#$$

To eliminate the variance of vertices number in sets, We append x with length t one time on the strands if we detect the strands missing the vertice v_k length information once. So After the operations of second step, all the strands in P contain all the vertices length information at least once. we reasonably design the length of $\#,\overline{\#},A_k,\overline{A_k}$ and x, For $||A_i|| = ||\overline{A_i}|| = k_i$ (k_i is the weights of vertice $v_i \in E$)
$m = \max\{k_i\} = \max\{||A_i||,||\overline{A_i}||\}$
$||\#|| = ||\overline{\#}|| = ||x|| = n * m = t$
So we define S as the strands after the second step. Then S can be described:

$$\#A_{i_1}\overline{\#A_{i_2}}\cdots\#A_{i_{j-1}}\overline{\#A_{i_j}}\#x\cdots x$$

The number λ of appending x can be decided by the missing vertices information on the strands. Due to the possible of containing vertice v_k information more than once, then $i_j + \lambda \geq n$. So

$$\begin{aligned}
||S|| &= ||\#|| + ||A_{i_1}|| + ||\overline{\#}|| + ||\overline{A_{i_2}}|| + \cdots + ||\overline{\#}|| + ||\overline{A_{i_j}}|| \\
&\quad + ||\#|| + ||x|| + \cdots + ||x|| \\
&= (i_j+1)||\#|| + \sum_{v_{i_k}\in V} ||A_i|| + \lambda||x|| \\
&= (i_j+\lambda+1)t + \sum_{v_{i_k}\in V} ||A_i|| \\
&\geq (n+1)t + \sum_{v_{i_k}\in V} ||A_i|| \\
&\because 0 \leq \sum_{v_{i_k}\in V} ||A_i|| \leq t
\end{aligned}$$

So the length of strands which denote containing all the vertices information only once must be between $(n+1)t$ and $(n+2)t$. The length of strands which denote containing a same vertice information more than once must be longer than $(n+2)t$. So we can get the solution in step (3) in appropriate length range.

Besides,the manipulates of algorithm can be entirely finished in finite operations. Such as step (1), (3), (4) in $O(1)$, Simultaneity step (2) in $O(n)$. In conclusion, We can get the solution of subset sum problem with n vertices in $O(n)$.

4 Conclusion

In this paper, we propose a procedure for the subset sum NP-complete problem in the genetic algorithm model. The procedure works in $O(n)$ steps for the subset sum problem of an vertice-weighted graph with n vertices. All our results in this paper are based on a theoretical model. However, to solve some complex problems using the new method can help us understand more about the characteristic of problem and promote the development of DNA computing research, for that DNA-based computers may be a good choice for performing massively parallel computations. Up to now, there are still many unsolved mathematical NP-complete problems because they are difficult to support basic biological experiment operations. We hope that our results can have certain help and effect to the DNA-based computing development.

Acknowledgement

The project was supported by CNSF (Grant No. 51108376) and SSF (Grant No. 2014JQ7231). The corresponding author thanks the fund support by the Science and Technology Project of Henan Province (Grant No. 142102210225). This research was also supported by the Open Research Fund of State Key Laboratory of Simulation and Regulation of Water Cycle in River BasinChina Institute of Water Resources and Hydropower Research (Grant No. 2014ZY05) and 12th

Five-Year Plan to Support Science and Technology Project (Grant No. 2012BAB04B02) respectively. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] L.M. Adleman, Molecular computation of solution to combinatorial problems, Science **266** (1994) 1021-1024.
[2] R.J. Lipton, DNA solution of HARD computational problems, Science **268** (1995) 542-545.
[3] A. Fujiwara, K. Matsumoto, Wei Chen, Procedures for logic and arithmetic operations with DNA molecules, International Journal of Foundations of Computer Science **15** (2004) 461-474.
[4] F. Guarnieri, M. Fliss, C. Bancroft, Making DNA add, Science **273** (1996) 220-223.
[5] H. Hug, R. Schuler, DNA-based parallel computation of simple arithmetic, in: Proceedings of the 7th International Meeting on DNA Based Computers, 2001, pp. 159-166.
[6] Wang Z, Huang D, Pei R. Solving the Minimum Vertex Cover Problem with DNA Molecules in Adleman-Lipton Model[J]. Journal of Computational and Theoretical Nanoscience, 2014, **11**: 521-523.
[7] W.X. Li, D.M. Xiao, L. He, DNA ternary addition, Applies Mathematics and Computation **182** (2006) 977-986.
[8] D.M. Xiao, W.X. Li, J. Yu, X.D. Zhang, Z.Z. Zhang, L. He, Procedures for a dynamical system on 0,1n with DNA molecules, BioSystems **84** (2006) 207-216.
[9] Wang Z, Xiao D, Li W, et al. A DNA procedure for solving the shortest path problem[J]. Applied mathematics and computation, 2006, **183**: 79-84.
[10] X.L. Wang, Z.M. Bao, J.J. Hu, S. Wang, A. Zhan, Soling the SAT problem using a DNA computing algorithm based on ligase chain reaction, BioSystems **91** (2008) 117-125.
[11] A. Narayanan, S. Zorbalas, et al., DNA algorithms for computing shortest paths, in: J.R. Koza (Ed.), Proceedings of the Genetic Programming 1998, Morgan Kaufmann, 1998, pp. 718-723.
[12] S.-Y. Shin, B.-T. Zhang, S.-S. Jun, et al., Solving traveling salesman problems using molecular programming, in: P.J. Angeline (Ed.), Proceedings of the Congress on Evolutionary Computation 1999, IEEE Press, 1999, pp. 994-1000.
[13] M. Yamamura, Y. Hiroto, T. Matoba, Solutions of shortest path problems by concentration control, Lecture Notes Computer Science, **2340**, 2002, pp. 231-240.
[14] Wang Z, Huang D, Meng H, et al. A new fast algorithm for solving the minimum spanning tree problem based on DNA molecules computation[J]. Biosystems, 2013, **114**: 1-7.
[15] J.-Y. Lee, S.-Y. Shin, T.-H. Park, B.-T. Zhang, Solving traveling salesman problems with DNA molecules encoding numerical values, BioSystems **78** (2004) 39-47.
[16] Wang Y, Yang T, Ma Y, et al. Mathematical modeling and stability analysis of macrophage activation in left ventricular remodeling post-myocardial infarction[J]. BMC genomics, 2012, 13(Suppl 6): S21.
[17] Z.C. Wang, Y.M. Zhang, W.H. Zhou and H.F Liu, Solving traveling salesman problem in the Adleman-Lipton model, Applied Mathematics and Computation (2012), **219**, pp 2267-2270.
[18] Wang Y, Han H C, Yang J Y, et al. A conceptual cellular interaction model of left ventricular remodelling post-MI: dynamic network with exit-entry competition strategy[J]. BMC systems biology, 2010, 4(Suppl 1): S5.
[19] Yang T, Chiao Y A, Wang Y, et al. Mathematical modeling of left ventricular dimensional changes in mice during aging[J]. BMC systems biology, 2012, 6(Suppl 3): S10.
[20] Wang Z, Huang W, Ye C, et al. Algorithm of Solving the Maximum Edges Independent Set Problem Based on DNA Molecules Computation[J]. Journal of Computational and Theoretical Nanoscience, 2014, **11**: 961-963.
[21] Z.C. Wang, J. Tan, D.M. Huang, et al. A biological algorithm to solve the assignment problem based on DNA molecules computation, Applied Mathematics and Computation (2014), **244**, pp 183-190.

Lei Li received the PhD degree in structural engineering from the Xi′an University of architecture and technology China in 2010. Currently he is a lecture in civil engineering department of Xi′an University of architecture and technology, China. His research interests include computational mathematics and mechanics, numerical simulation and digital image processing.

Kai Zhao received a master's degree in Computer Engineering from Xidian University China in 2010. Currently he is a lecturer in Pingdingshan University of computer science and technology, China. His research interests include computational mathematics and SVM.

Zuwen Ji was born in Jiangsu, China in 1967. He received the BS degree from Wuhan University of Hydraulic and Electrical Engineerin. His research interests include parallel computing and large scale data processing.