Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080531

Solving the Maximum Independent Set Problem based on Molecule Parallel Supercomputing

Zhaocai Wang[1], *Jian Tan*[2,*], *Lanwei Zhu*[2] *and Wei Huang*[3]

[1] College of Information, Shanghai Ocean University, Shanghai 201306, P. R. China
[2] Key Laboratory of Digital Earth, Center for Earth Observation and Digital Earth, Chinese Academy of Sciences, Beijing 100094, P. R. China
[3] College of Mathematics and Information Science ,Guiyang University, Guizhou 550005 , P. R. China

Received: 3 Sep. 2013, Revised: 1 Dec. 2013, Accepted: 2 Dec. 2013
Published online: 1 Sep. 2014

Abstract: The maximum independent set Problem is to find a biggest vertex independent set in a given undirected graph. It is a vitally important NP problem in graph theory and applied mathematics, having numerous real life applications. It can be difficultly solved by the electronic computer in exponential level time. Simultaneity in previous studies DNA molecular computation usually be used to solve NP-complete continuous path search problems (for example HPP, traveling salesman problem), rarely for NP problems with discrete vertex or path solutions result, such as the maximum independent set problem, graph coloring problem and so on. In this paper, we present a new algorithm for solving the maximum independent set problem with DNA molecular operations. For an undirected graph with n vertices, We reasonably design fixed length DNA strands representing the vertices and edges of graph, take appropriate steps and get the solutions of the problem in proper length range using $O(n^2)$ time. We extend the application of DNA molecular operations and simultaneity simplify the complexity of the computation.

Keywords: DNA computation; The maximum independent set problem; Adleman-Lipton model; NP-complete problem

1 Introduction

DNA computing is a newly emerging crossdisciplinenary science that uses DNA molecular biotechnologies to solve conundrum problems of computer science and computational mathematics. Huge storage capacity and massive parallelism are two important advantages of DNA computation. DNA computing can execute billions of operations simultaneously. The massive parallelism of DNA computing comes from the large number of molecules which chemically interact in a small volume. DNA also provides a huge storage capacity since they encode information on the molecular scale. Meanwhile DNA has a great application prospect for having wide range of abundant resources. NP (nondeterministic polynomial time) problems are a class of mathematical problems which have most likely exponential complexity, for which no efficient solution has been found yet [1]. As the pioneering work for DNA computing, Adleman [2] presented an idea of solving the Hamiltonian path problem of size n in $O(n)$ steps using DNA molecules. Lipton [3] demonstrated that Adleman's experiment could be used to figure out the NP-complete satisfiability (SAT) problem (the first NP-complete problem). In recent years, lots of papers have occurred for designing DNA procedures and algorithms to solve various NP-complete problems [4,5,6,7,8,9,10]. However, most of the previous works in DNA computing are concentrated on solving the path search problems that the optimum results are continuous head-to-tail ligation edges or vertices sets. For example, Lee [11] first designs different length's strands representing paths values and cities, takes molecular operations to generate strands standing for all possible paths, then uses biochemical techniques, such as denaturation temperature gradient polymerase chain reaction and temperature gradient gel, to get the optimum solutions of the traveling salesman problem. To solve the shortest path problem, Narayanan [12] respectively carries out DNA reaction to get the strands for a list of series paths, then chooses the shortest length strands as the solution through DNA biotechnologies. The previous researches have some insufficient factors. One is that the

* Corresponding author e-mail: jtan1980@163.com

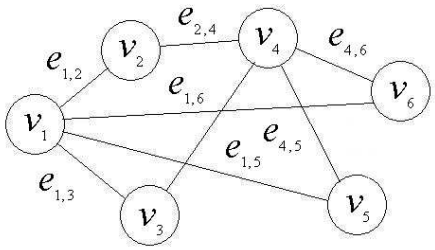


Fig. 1. An undirected graph G with 6 vertices

strands for the possible paths are usually very long, while too long DNA strands can lead to error-prone in annealing and separation procedures using modern biotechniques. The other is that previous research problems are all optimum path search problems, so that the possible solutions can be relatively easily represented by DNA strands. While the maximum independent set Problem is a discrete vertices set problem with discontinuous path. So representation of discrete data in DNA strands is an important issue toward expanding the capability of DNA computing to solve many optimization problems.

The maximum independent set problem is a problem of central importance in graph theory and computational Sciences. It is intractable to solve. Given a graph G, an independent set is a subset S of vertices in G such that no two vertices in S are adjacent (connected by an edge). The maximum independent set problem is to find an independent set with the largest number of vertices in a given graph. This problem is NP-hard, and it is considered unlikely to exist an efficient algorithm for solving it up to now. In this paper, a new biocomputing procedure based on the research of Adleman [2] and Lipton [3] is introduced for figuring out solutions of the maximum independent set problem: Given an undirected graph $G = (V,E)$ with vertex set $V = \{v_1, v_2, \cdots, v_n\}$ and edge set $E = \{e_{i,j} | 1 \leq i < j \leq n\}$, an independent set is a subset $V^{'} \subseteq V$ such that for any vertices in subset $V^{'}$ aren't adjacent (connected by an edge). A independent set $V^{'}$ is to be a maximum independent set of graph G, if for any vertex independent set $V^{''} \subseteq V$ with $|V^{'}| \geq |V^{''}|$. For instance, the undirected graph G in Fig. 1 defines such a problem. It is not difficult to find that the vertex set $\{v_2, v_3, v_5, v_6\}$ is the solutions to the maximum independent set 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 uses a DNA molecular algorithm for solving the maximum independent set problem and the complexity of the proposed algorithm is described. We get conclusions in Section 4.

2 The Adleman-Liption Model

Bio-molecular computers work at the molecular level. Because biological and mathematical operations have some similarities, DNA, the genetic material that encodes for living organisms, is stable and predictable in its reactions and can be used to encode information for mathematical systems.

DNA is the major information storage molecule in living cells, and billions of years of evolution have tested and refined both this wonderful informational molecule and highly special enzymes that can either duplicate the information in DNA molecules or transmit this information to other DNA molecules.

A DNA(deoxyribonucleic acid) is a polymer, which is strung together from monomers called deoxyribonucleotides [14]. 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 [2] and Lipton [3] are described below. These operations will be used for figuring out solutions of the maximum independent set 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:

(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) *Ligation (T)*: for a given tube T, the operation is used to ligate together the strands in T;

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

(11) *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;

(12) *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 twelve manipulations are implemented with a constant number of biological steps for DNA strands [14], we assume that the complexity of each manipulation is $O(1)$ steps.

3 DNA algorithm for the maximum independent set problem

For a given undirected graph $G=(V,E)$, $V=\{v_k|k=1,2,\ldots,n\}$ is vertex set and $E=\{e_{i,j}|1\leqslant i<j\leqslant n\}$ is edge set. Some vertices v_i and v_j can be connected by the edge $e_{i,j}$ $(i<j)$ in graph. We let $|E|=m$ and $m\leqslant n(n+1)/2$. At the same time, the graph processed in this paper has no self-loops.

In the following, the symbols $\#,\overline{\#},0,1,A_k,B_k$ $(k=1,2,\ldots,n)$ denote distinct DNA singled strands with same length, say t-mer (t can choose a small integer). Obviously the length of the DNA singled strands greatly depends on the size of the problem involved in order to distinguish all above symbols and to avoid hairpin formation [15]. Then in the below operations, we use the distinct DNA singled strands symbols A_k0B_k, $A_k1B_k(k=1,2,\ldots,n)$ to denote the vertex v_k, with A_k1B_k for v_k including in the vertex set, while A_k0B_k for not. Simultaneity the symbol $\#,\overline{\#}$ is the signal of division between different vertex sets. We denote DNA singled strands $y_{i,j}$ to encode the edge $e_{i,j}$ connecting by the vertices v_i and v_j with length of t-mer if there exists the edge $e_{i,j}$ in graph G. For distinguishing whether some vertices belonging to a vertex set or not, we meantime design DNA string X with t-mer length. Let

$$P=\{0,1,\#A_1,B_n\#,A_kB_{k-1}|k=2,3,\cdots,n\},$$

$$Q=\{\overline{\#},\overline{B_k1A_k},\overline{B_k0A_k}|k=1,2,\cdots,n\},$$

$$R=\{y_{i,j}|1\leqslant i<j\leqslant n\},$$

$$S=\{X\}.$$

For a graph with n vertices, every possible subset of the vertex set V can be expressed by a n-digit binary number. A bit set to 1 represents the vertex in the subset, and a bit set to 0 represents the vertex out of the subset. For example in Fig. 1, the subset $\{v_2,v_4,v_5\}$ can be expressed by the binary number 010110. In this way, we transform all possible subsets of V in a n-vertex graph into an ensemble of all n-digit binary numbers. We call this the data pool.

(1)We choose all possible sets of vertex in graph.
(1-1)$Merge(P,Q)$;
(1-2)$Aneealing(P)$;
(1-3)$Ligation(P)$;
(1-4)$Denaturation(P)$;
(1-5)$Separation(P,\{\#A_1\},T_1)$;
(1-6)$Discard(P)$;
(1-7)$Separation(T_1,\{B_n\#\},P)$.
After the above seven steps of manipulations, the singled strands in tube P will encode all possible sets of vertex. For example, for the graph in Fig. 1, we have singled strands:

$$\#A_11B_1A_21B_2A_30B_3A_41B_4A_50B_5A_61B_6\#\in P$$

which denotes the set of vertex $\{v_1,v_2,v_4,v_6\}$ corresponding to the binary number 110101. 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 vertex set after the first step. While the maximum independent set problem require that any vertex shouldn't be connected with other vertices in a same set by an edge. So we should check all the vertex sets whether to satisfy the above restrictive condition. If $e_{i,j}\in E$ in graph, We should discard the strands which both vertices v_i and v_j are in the same set. For example in Fig. 1, the singled strands $\#A_11B_1A_20B_2A_30B_3A_41B_4A_51B_5A_60B_6\#$ (representing the set of vertex $\{v_1,v_4,v_5\}$) should be discarded for the vertices v_4 and v_5 in the set can be connected by edge $e_{4,5}\in E$. Only so we can choose all possible vertex independent sets in graph.

For $i=1$ to $i=n-1$
  For $j=i+1$ to $j=n$
(2-1)$Separation(R,\{y_{i,j}\},T_2)$;
(2-2)If($Detect(T_2)$)
  Then

(2-3)$Discard(T_2)$;
(2-4)$Separation(P,\{A_i1B_i\},T_3)$;
(2-5)$Separation(T_3,\{A_j1B_j\},T_4)$;
(2-6)$Discard(T_4)$;
(2-7)$Merge(P,T_3)$;
(2-8)$Discard(T_3)$.
End for
End for

After the above operations, the singled strands in tube P are all vertex independent sets. Meanwhile we use two "For" clauses, thus this operation can be finished in $O(n^2)$ steps since each single manipulation above works in $O(1)$ steps.

(3)The maximum independent set problem should be a biggest vertex set which satisfies the independent set restrictive condition. So we choose the maximum vertex independent set in all vertex independent sets. If a vertex v_i in the vertex set, we append additional strand X at the end of previous strand in order to find the optimum strand solutions. For example, for the graph in Fig. 1, the singled strands

$$\#A_10B_1A_21B_2A_30B_3A_40B_4A_51B_5A_61B_6\#$$

in P represent containing the vertices $\{v_2,v_5,v_6\}$, So we append strand X three times at the previous strands to

$$\#A_10B_1A_21B_2A_30B_3A_40B_4A_51B_5A_61B_6\#XXX$$

This is done by the following manipulations:
For $k=1$ to $k=n$
(3-1)$Separation(P,\{A_k1B_k\},T_5)$;
(3-2)$Append(T_5,X)$;
(3-3)$Merge(P,T_5)$;
(3-4)$Discard(T_5)$.
End for
In the above operation, we use one "For" clause, thus this operation can be finished in $O(n)$ steps since each single manipulation above works in $O(1)$ steps.

(4)We take out those singled strands in P with biggest length, which represent the solutions to maximum independent set problem. For example, for the graph in Fig. 1, the singled strands in P with largest length are

$$\#A_10B_1A_21B_2A_31B_3A_41B_4A_51B_5A_60B_6\#XXXX$$

Therefore, solutions to maximum independent set problem for the graph in Fig. 1 are $\{v_2,v_3,v_5,v_6\}$ which contain four vertices.
For $k=1$ to $k=n$
(4-1)$Selection(P,(3n+2)t+kt,T_6)$;
(4-2)If($Detect(T_6)$)
Break;
End for
In the above operation, we use one "For" clause, the worst conditions is that the algorithm stop at $k=n$, thus this operation can be finished less than $O(n)$ steps since each single manipulation above works in $O(1)$ steps.

(5)Finally the "*Read*" operation is applied to giving the exact solutions to the maximum independent set problem. For example, for the graph in Fig. 1, the maximum independent set is $\{v_2,v_3,v_5,v_6\}$. This operation works in O(1) steps.
(5-1)$Read(T_6)$;

The following theorem tells that the algorithm proposed above really can get solutions of the maximum independent set problem in $O(n^2)$ steps using DNA molecules.

Theorem 1. *The solutions of maximum independent set problem for a graph with n vertices can be figured out in $O(n^2)$ steps using DNA molecules parallel supercomputing.*

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

$$\#A_1z_1B_1A_2z_2B_2\cdots A_kz_kB_k\cdots A_nz_nB_n\# \quad z_k=0 \ or \ 1$$

.
After the operations of second step, all the strands in P denotes one possible vertex independent set. In the first instance we reasonably design the length of $\#,A_k,B_k,0,1$ and X, For

$$||A_k||=||B_k||=||\#||=||0||=||1||=||z_k||=||X||=t$$

So we define S as the strands after the third step. Then S can be described:

$$\#A_1z_1B_1A_2z_2B_2\cdots A_kz_kB_k\cdots A_nz_nB_n\#X\cdots X$$

The number p of appending X times is decided by the existing vertices information on the strands. Due to the possible times p of containing vertices v_k information is between 1 and n, So

$$\begin{aligned}
&||S||\\
&=||\#||+||A_1||+||z_1||+||B_1||+||A_2||+||z_2||+||B_2||+\cdots+\\
&\quad ||A_n||+||z_n||+||A_n||+||\#||+||X||+\cdots+||X||\\
&=2||\#||+\sum_{k=1}^{n}||A_k||+\sum_{k=1}^{n}||z_k||+\sum_{k=1}^{n}||B_k||+p||X||\\
&=(3n+2)t+pt\\
&\because 0\le p\le n\\
&\therefore (3n+2)t\le||S||\le(3n+2+n)t
\end{aligned}$$

So the length of strands which denote containing all the vertices information must be between $(3n+2)t$ and $(3n+2+n)t$. So we can get the solution in step 4 in appropriate length range.

Besides,the manipulates of algorithm can be entirely finished in finite operations. Such as step (1), (5) in $O(1)$, step (4) less than $O(n)$, Simultaneity step (2) in $O(n^2)$ and step (3) in $O(n)$. In conclusion, We can get the solution of maximum independent set problem with n vertices in $O(n^2)$.

4 Conclusion

In this paper, we present DNA algorithms for solving the maximum independent set problem based on biological operations in the Adleman-Lipton model. The proposed algorithms have two advantages. Firstly, the proposed algorithm actually has a lower rate of errors for hybridization because we generate fixed reasonable DNA sequences for generating the solutions of the problem. Secondly, the proposed algorithms can works in $O(n^2)$ steps for the maximum independent set problem of an undirected graph with n vertices, Comparing exponential level time by electronic computer. All our results in this paper are based on a theoretical model. However, the ability to perform complex operations in solution might help us learn more about the nature of computation and lead to the development of better DNA based computation, capable of solving a wide range of complex problems.

Acknowledgement

The first author acknowledges the financial support by Doctor fund of Shanghai Ocean University (A-2400-12-0000351). The second and third authors acknowledge the financial support by CNSF (NO.41201400).
The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

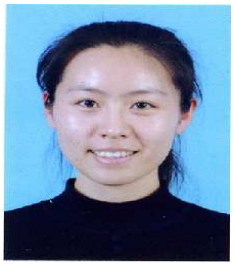
References

[1] Garey, M.R., Johnson, D.S, Computers and Intractability: A Guide to the Theory of NP-completeness. W. H. Freeman and Company, (1979).
[2] L.M. Adleman, Molecular computation of solution to combinatorial problems, Science, **266**, 1021-1024 (1994).
[3] R.J. Lipton, DNA solution of HARD computational problems, Science, **268**, 542-545 (1995).
[4] A. Fujiwara, K. Matsumoto, Wei Chen, Procedures for logic and arithmetic operations with DNA molecules, International Journal of Foundations of Computer Science, **15**, 461-474 (2004).
[5] F. Guarnieri, M. Fliss, C. Bancroft, Making DNA add, Science, **273**, 220-223 (1996).
[6] H. Hug, R. Schuler, DNA-based parallel computation of simple arithmetic, in: Proceedings of the 7th International Meeting on DNA Based Computers, 159-166 (2001).
[7] S. Kamio, A. Takehara, A. Fujiwara, Procedures for computing the maximum with DNA strands, in: Humid R. Arabnia, Youngsong Mun (Eds.), Proceedings of the International Conference on DNA Based Computers, (2003).
[8] W.X. Li, D.M. Xiao, L. He, DNA ternary addition, Applies Mathematics and Computation, **182**, 977-986 (2006).
[9] 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**, 207-216 (2006).
[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**, 117-125 (2008).
[11] J.-Y. Lee, S.-Y. Shin, T.-H. Park, B.-T. Zhang, Solving traveling salesman problems with DNA molecules encoding numerical values, BioSystems, **78**, 39-47 (2004).
[12] 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, 718-723 (1998).
[13] R.M. Karp, Reducibility among combinatorial problems, in: R.E.Miller, J.W. Thatcher (Eds.), Complexity of Computer Computation, Plenum Press, New York, 85-103 (1972).
[14] M. Yamamura, Y. Hiroto, T. Matoba, Solutions of shortest path problems by concentration control, Lecture Notes Computer Science, **2340**, 231-240 (2002).
[15] 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, 994-1000 (1999).
[16] G. Paun, G. Rozeberg, A. Salomaa, DNA Computing, Springer-Verlag, (1998).
[17] Adam J. Ruben, Laura F. Landweber, The past, present and future of molecular computing, Nature Reviews Molecular Cell Biology, **1**, 69-72 (2000).
[18] Li, D., Huang, H., Li, X., Li, X. Hairpin formation in DNA computation presents limits for large NP-complete problems. BioSystems, **72**, 203-207 (2003).
[19] GONG Zai-wu, LIU Si-feng, Research on Consistency and Priority of Interval Number Complementary Judgment Matrix, Chinese Journal of Management Science, **4**, (2006).
[20] Q. Ouyang, Peter D. Kaplan, S. Liu, A. Libchaber, DNA solution of the maximal clique problem, Science, **278**, 446-449 (1997).
[21] 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**, 71-82 (2005).
[22] Z.C. Wang, Y.M. Zhang, W.H. Zhou, H.F. Liu, Solving traveling salesman problem in the Adleman-Lipton model, Applied Mathematics and Computation, **219**, 2267-2270 (2012).
[23] Z.C. Wang, D.M. Huang, H.J. Meng, C.P Tang, A new fast algorithm for solving the minimum spanning treeproblem based on DNA molecules computation, Biosystems, **114**, 1-7 (2013).
[24] Kensaku Sakamoto, Hidetaka Gouzu, Ken Komiya, Daisuke Kiga, Shigeyuki Yokoyama, Takashi Yokomori, Masami Hagiya, Molecular Computation by DNA Hairpin Formation, Science, **288**, 1223-1226 (2000).
[25] Wang Y, Yang T, Ma Y, et al. Mathematical modeling and stability analysis of macrophage activation in left ventricular

remodeling postmyocardial infarction[J]. BMC genomics, 2012, 13(Suppl 6): S21.
[26] 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.
[27] 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.

Zhaocai Wang was born in Shandong, China in 1979. He received the BS degree in Applied Mathematics from Shanghai Jiaotong University and the PhD degree from Fudan University in 2006 and 2012 respectively. His research interests include biology mathematics and biology computation. Dr Wang has published over 15 papers in leading journals and conferences.

Jian Tan was born in Hubei, China in 1980. His research interests include 3D spatial information system, theory of digital earth and web GIS. Dr Tan has published over 20 papers in leading journals and conferences.



Lanwei Zhu was born in Neimenggu, China. She is a research associate of Key Laboratory of Digital EarthCenter for Earth Observation and Digital Earth, Chinese Academy of Sciences. Her research field are GIS and remote sensing. Mrs Zhu has published over 20 papers in leading journals and conferences.

Wei Huang was born in Guizhou, China in 1977. He received the BS degree in School of Communication and Information Engineering from Guizhou University and the PhD degree from Chengdu University of Technology. His research interests include Wireless sensor and biology computation. Dr Huang has published over 10 papers in leading journals and conferences.