Solution to the 0-1 Knapsack Problem based on DNA Encoding and Computing Method

Lian Ye
Department of Computer, Chongqing University, Chongqing, China
Email: ylredleaf@cqu.edu.cn

Min Zhang
Department of Computer, Chongqing University, Chongqing, China
Email: juneyl@sina.com

***Abstract*—DNA computing is a new computational paradigm that executes parallel computation with DNA molecules. Some researches in DNA computing have been presented to solve computational problems such as NP-complete problems in polynomial increasing time by using its super parallel and high density power. Among them knapsack problem is one of the most common problems which have been studied intensively in the last decade attracting both theorists and practicing. This paper proposes an encoding and computing method to solve 0-1 knapsack problem. The encoding method is described to generate superior DNA strands with fewer errors according to the characteristics of DNA sequences. Then the computing algorithm replicate the strands which expressed as the weight of items and take the combination of every DNA strand to form double stranded DNA sequences in order to find out the optimal solution. The results demonstrate the superiority of our approach which may be used to resolve different NP-hard problems by adjusting the DNA-based procedures.**

***Index Terms*—0-1 knapsack problem, DNA computing, Optimization**

I. Introduction

Bioinformatics studies the store, process, distribution and analysis biological information so as to understand the meanings of biological data by means of mathematic, computer science and biological techniques. Some researches on bioinformatics like the properties of DNA and the Watson-Crick's law has provided a probability of computing with DNA molecules to make DNA computing as an applied branch of bioinformatics, and the results of researches of bioinformatics will improve the capabilities of DNA computing.

Based on the massive parallelism of DNA computing, many researchers tried to solve a large number of difficult problems. In 1994, Adleman has solved a seven vertex instance of the directed Hamiltonian path problem by means of the techniques of molecular biology [1]. A seven vertex graph was encoded in DNA strands and the operations were performed with standard DNA protocols and enzymes. This experiment demonstrates the feasibility of carrying out computations at the molecular level and the result showed that parallelism of DNA computers to solve painstaking problems such as NP-complete problems with linearly increasing time. Then in 1995, Lipton proposed molecular biology experiments to solve the 3-SAT problem [2]. The advantage of the results is the huge parallelism inherent in DNA computing, which has the potential to yield vase speedups over conventional silicon-based computers. In 1997, Ouyang solved the maximal clique problem using the techniques of molecular biology [3]. A pool of DNA molecules corresponding to the ensemble of six vertex cliques was built followed by a series of selection processes.

Knapsack problem is a typical NP-complete problem, mainly used in the management of resource allocation, investment decision-making and modeling of loading problems. Its solution mainly depends on a number of heuristic algorithms, such as greedy algorithms, approximation, simulated annealing algorithm. In solving large problem of multiple operational research algorithms, knapsack problem is also regarded as sub-problem to deal with. It is beneficial to complicated algorithm of multiple operational researches if we make development of knapsack problem.

In [4], Majid proposed a surface-based method to solve 0-1 knapsack problem. In this method, combinatorial set of single-stranded DNA molecules representing all possible solutions to a given computational problem is synthesized and immobilized on a surface via a reactive functional group. In each of successive cycles of the DNA computation, subsets of the surface-bound combinatorial mixture are tagged by hybridization to their complements, rendering them double stranded. At the end of cycles, only those strands which are solutions to the problem remain. Christian described a parallel search algorithm for knapsack problem in [5], which demonstrates how the computations can be extended by in vivo translation of the DNA library into protein. This combination of DNA and protein allows for multi-criterion optimization. Zhu Ying proposed a DNA algorithm based on biochips to solve knapsack problem [6]. The DNA fragments design was based on reactions and the computational procedure is high-efficiency DNA

doi:10.4304/jcp.8.3.669-675

ligation, quantitative analysis and qualitative analysis were used to detect the result.

A DNA encoding and computing algorithm to solve 0-1 knapsack problem is proposed in this paper. The new method conclude two major procedures: encoding process and computing process. In former process, the DNA sequences were generated in consideration of the characteristics of DNA sequences. This model avoids hybridizations and prevents unwanted secondary structures by adapting its variable's environmental factors occurring in the process of sequence generation. In latter process, two sets of DNA fragments: double stranded fragment that represent each item's weight and the linkers to combine the items randomly. The computing model is based on different combination of biological molecule operations such as amplify, separate and so on.

The rest of paper is organized as follows: Section II gives the mathematical description of 0-1 knapsack problem. Section III introduces the inherent characteristics of sequences and the method of generating the sequences according to the characteristics. The DNA sequence encoding method for knapsack problem and the basic biological operations are described in section IV. In section V, the DNA algorithm for solving the knapsack problem is proposed and the experiment result shows the correctness of the algorithm. The conclusion is given in section VI, along with a summary of the results.

II. 0-1 Knapsack Problem

Knapsack problem allows the exploitation of a number of combinatorial properties. Suppose that we want to fill up a knapsack by selecting some objects among various objects (generally called items). There are *n* different items available and each item *j* has a weight of w_j and a profit of p_j. The knapsack can hold a weight of at most W. The problem is to find an optimal subset of items so as to maximize the total profits subject to the knapsack's weight capacity. The profits, weights, and capacities are positive integers. Let x_j be binary variables given as follows:

$$x_j = \begin{cases} 1 & if\ item\ j\ is\ selected \\ 0 & otherwise \end{cases} \tag{1}$$

The knapsack problem can be mathematically formulated as follows:

$$\begin{aligned} & \max \quad \sum_{j=1}^{n} p_j x_j , \\ & s.t. \quad \sum_{j=1}^{n} w_j x_j \le W , \\ & \quad x_j = 1 \ or \ 0 , \qquad j = 1, 2, \ldots, n . \end{aligned} \tag{2}$$

This is known as the 0–1 knapsack problem, which is pure integer programming with a single constraint and forms a very important class of integer programming. That is to say, the purpose is to find the best way to put items with different sized and weights into a limited volume knapsack to meet the maximum total value.

III. DNA Computing Encoding

In DNA computing, the hybridization between a DNA strand and its base-pairing complement is crucial to retrieve the information stored in DNA sequences and operates the computation process. For this reason, the desired set of good DNA sequences, which have a stable duplex with their complement are more needed. Here a hierarchy evolutionary searching algorithm is adopted to obtain good DNA encoding sequences [7], this approach is based on combinatorial constraints which affect the molecular reaction process, and can provide some reliable and effective encoding sequences for controllable DNA computing.

A. Characteritic of Sequence

The basic notations will be defined. An alphabet to consist of each single nucleotide is defined as Λ={A, C, G, T}, then a single strand is denoted as x, and the set of all DNA sequences is denoted as Λ*, x∈Λ*. The length of x is denoted as |x|, and x_i $_($1≤i≤|x|) means *i*-th nucleotide from 5'-end of sequence x. A set of sequences with the same length *l* is denoted by Σ, where *j*-th member of Σ is denoted as Σ_j. $\overline{X}$ is represented as the complementary of x.

Seven constraints such as continuity, GC-content, hairpin, melting temperature, free energy, H-measure and similarity are considered in the method.

- Continuity: If the same bases occur continuously in a sequence, the sequence can show the unexpected structures. Continuity (x_i) calculates the degree of successive occurrence of the same bases in the strand.

$$\text{Continuity}(x_i) = \max(N_a^{\ i})\, a \in \Lambda \tag{3}$$

  Where N_a^i denotes the number of times to which the same base appear continuously in DNA sequence $x_{I.}$ The value of *a* may be A, C, G or T.

- GC-content: The GC-content is the percentage of G and C in a sequence. The GC content affects the thermodynamic properties of a DNA molecule. Therefore, if all these sequences will ensure similar GC content, they must have similar thermodynamic characteristics which can effectively reduce the probability of the occurrence of non-specific hybridization. GC-content's formulation as:

$$GC_{content}= (yG+zC)/ (wA+xT+yG+zC) \tag{4}$$

- Hairpin: Hairpin is an undesirable DNA secondary structure, because it can hybridize its self. In order to make the hybridization between a DNA strand and its Watson-Crick complement more efficient, the single DNA strand should be hairpin structure-free. Hairpin structure consists of the ring part and the stem part as shown in Fig.1.

$$Hairpin = \sum_{r=R\min}^{l-2*P\min} \sum_{P=P\min}^{l-P\min-r} T(\sum_{i=1}^{pinlen(p,r)} bp(x_{p+1-i}, y_{p+r+i}), \frac{pinlen(p,r)}{2}) \tag{5}$$

where pinlen (p, r, i)=min (p+i, l-r-i-p).

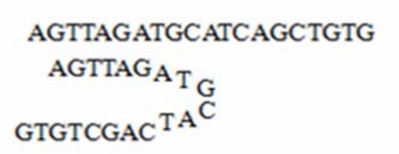


Figure 1. A DNA sequence with 20-mer in forming the loop with p and r=6.

- Melting Temperature: Melting temperature is one of the most important features of DNA that have to be considered for laboratory experiment. It is defined as a temperature, where half of double-stranded DNA starts to break into its single-stranded form.

$$T_m(x) = \frac{\Delta H}{\Delta S + R\ln C_T} + 16.6\log(Na^+) \quad (6)$$

  ΔH and ΔS are enthalpy and entropy changes of the annealing reaction. The universal gas constant, R is 1.987 cal/mol°C. And C_T is total oligonucleotide strand concentration.

- Free Energy: Free energy is the necessary energy to make a duplex, it is defined as the energy required to breaking a duplex. Usually, the free energy is calculated to detect the hybridization fidelity. The formula of free energy is:

$$\begin{aligned}\Delta G(total) = {} & \sum_i n_i \Delta G(i) + \Delta G(init\ w/termG-C) \\ & + \Delta G(sym) + \Delta G(init\ \ w/termG-C)\end{aligned} \quad (7)$$

  where ΔG (i) the standard free energy changes for 10 possible Watson-Crick nearest neighbors, ni is the number of occurrences of each nearest neighbor.

- H-measure: The Hamming distance between two strands is the number of corresponding places where two characters differ. Thus the Hamming distance is not an adequate measure of hybridization likelihood. The H-measure, a more appropriate measure of hybridization repulsion, takes the minimum of all Hamming distances obtained by successively shifting and lining up the complement of x_j against $x_{I.}$ Denoted as:

$$\text{H-measure }(x_i, x_j) = \min_{-l<k<l} H(x_i, \sigma^k(\overline{x}_j)) \quad (8)$$

  Where H (x_i, x_j) denotes the Hamming distance. σ^k denotes the right (left) shift in case of k>0 (k<0), *k* denotes the number of the shift, and $\overline{x}_j$ denotes the complementary pair.

- Similarity: The similarity measure computes the similarity in the same direction of two given sequences to keep each sequence as unique as possible including position shift. The similarity between the sequences should as small as possible to prevent the sequence from hybridizing with wrong sequence.

$$\text{Similarity }(x_i, x_j) = \max_{-l<k<l}\{l - H(x_i, \sigma^k(x_j))\} \quad (9)$$

  H (x_i, σ^k (x_j)) calculates the number of the different bases in sequence x_i and x_j.

B. Hierarchy Evolutionary Searching Algorithm

The hierarchy evolutionary searching algorithm is used to generate candidate resolutions. The constraints are separated into two parts according to different objects, means that some constraints are about single strand, and the others are for double strands. Constraints-I include continuity, GC-content, hairpin, Melting Temperature Free Energy, and Constraints-II include H-measure and similarity. The first procedure the single strands satisfy the constraint-I have been found, then based on the strands set obtained by the first hierarchy, the strands were checked to select the strands which satisfies constraint-II. Therefore, the strands obtained finally will have similar physical and chemical properties for the reliable and efficient experiments.

The steps of the algorithm are as follows:

- Step1: Generate initial single strand set P randomly.
- Step2: Test whether the sequences in P satisfy the constraint-I, then select the satisfying strands into set P’, the rest of P is called P’’.
- Step3: Add P’ to S’, if |S’|>n, n is a given value, then go to step 5, else go to step 4.
- Step4: Carry out mutation to strands in P’, Perform crossover and mutation to strands in P’’. Then merge P’ and P’’ into P, back to step 2.
- Step5: Checkout the strands in S’, filtrate the strands if it doesn’t satisfy the constraint-II with each other, select the satisfying sequence into S, if |S|>m, m is a given value, output S, else back to step 4.

In step 2, constraint-I compose several conditions restricted the single strands, check them by the order with considering the time complexity and strength of the constraints.

In the next process, the evolution operation including mutation and crossover has been used to generate the new populations. The mutation operator changes DNA base at random position. The crossover is regarded as an exchange of member sequence between two individuals. Using both operators showed empirically better results than using one of operators.

Lastly, the DNA strands are eliminated which may create error hybridize by H-measure and similarity to obtain the final solution. The algorithm ends when the number of iterations reaches a maximum value, or the solution set has got enough DNA sequence.

Fig. 2 shows the flowchart of the hierarchy evolutionary searching algorithm.

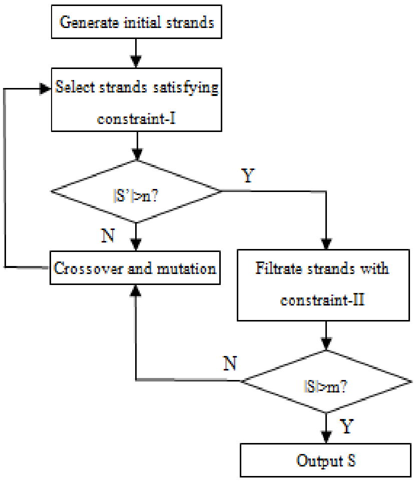

Generate initial strands
Select strands satisfying constraint-I
|S'|>n?
Y
N
Crossover and mutation
Filtrate strands with constraint-II
N
|S|>m?
Y
Output S

Figure 2. The flowchart of hierarchy evolutionary searching algorithm.

IV. DNA Computing Model for 0-1 Knapsack Problem

The encoding method for 0-1 knapsack problem is introduced firstly. Then the basic biological operations, such as annealing, are described.

A. DNA Encoding Method

For 0-1 knapsack problem, the set of n items denotes as I = {1, 2, ... , n}, the value of items denotes as C = {c_1, c_2, ... , c_n}, the weight W = {w_1, w_2, ... , w_n} , and the capacity of backpack denotes as M. Assume that the value and weight of each item are integer.

An oligonucleotide corresponding to fragment D_i is shown in Fig.3.

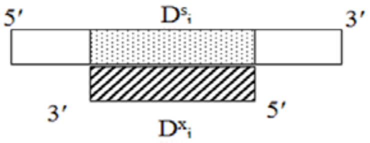


Figure 3. The oligonucleotide fragment of item *i*.

D_i is a double-stranded DNA fragment composed of two single-strand D^s_i and D^x_i which has sticky ends on both sides. The length of longer single-stranded D^s_i denotes the weight of items *i*, and the shorter single-stranded D^x_i denotes the item value of items *I*. Therefore, the numeric of weight must greater than the numeric of value. In the specific problem, if the condition is not satisfied, the weight or the value needs some process before encoding. For example, all of the weight and the capacity multiplied by a certain integer.

The following is an example of 0-1Knapsack problem to illustrate the computing model.

Given a set of items S, there are n=6 items, each item i has weight w_i and value $v_{I.}$ We take some items from S to put in the knapsack, maximizing the total value, but within the limitation of total weight thirty. The item features are shown in Table I.

TABLE I.
WEIGHT AND PRICE OF EACH ITEM *I*

Item list	Weight w_i	Value c_i
1	5	7
2	12	10
3	10	12
4	6	8
5	7	11
6	8	10

Because most weights of items are smaller than values of them, all of the weight and the capacity multiplied by three, the new weights are shown in Table II. The capacity of knapsack is ninety.

TABLE II.
MULTIPLIED WEIGHT OF EACH ITEM *I*

Item list	Weight w_i	Value c_i
1	15	7
2	36	10
3	30	12
4	18	8
5	21	11
6	24	10

The procedure of the weight and value of items mapped to DNA sequence could be divided into the following steps:

- Determine the length of D^s_i according to the weight of item.
- Determine and construct D^x_i according to the value of item.
- Construct $D^s_{I.}$ D^s_i contains three parts. The middle part D^x_i is complement sequence, the left and right parts are always equal to $(w_i-v_i)/2$ individually.

For example, the first item in Table II whose weight is 10 and value is 7. The DNA fragment of first item obtained by the hierarchy evolutionary searching algorithm describe in section III is shown in Fig.4.

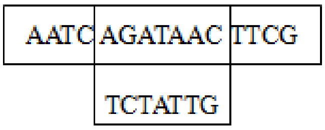


Figure 4. The DNA fragment D_1 of first item.

Encode the other items by the same method. The DNA fragment D_2-D_6 is shown in Fig.5.

D_2: TCATTGTACGCTA|GCGGTGAGTG|AATCGTTGCAGTG
CGCCACTCAC

D_3: CTCATGCGT|TAGGCAGATTCT|GCATAGCCT
ATCCGTCTAAGA

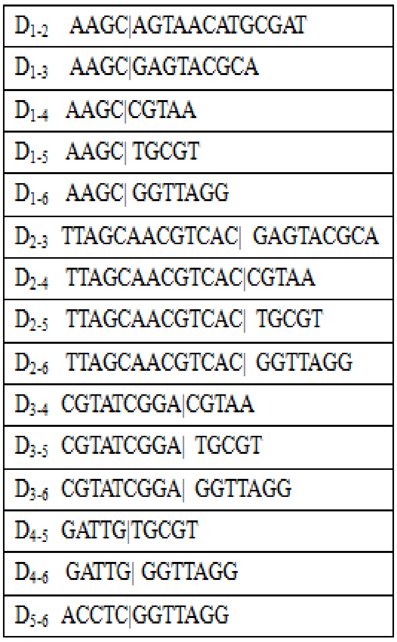
D_4: GCATT|CACGTCCT|CTAAC
GTGCAGGA

D_5: ACGCA|ATGACGGTCAA|TGGAG
TACTGCCAGTT

D_6: CCAATCC|TATTGGCACG|CTTCGGC
ATAACCGTGC

Figure 5. The DNA fragments of items.

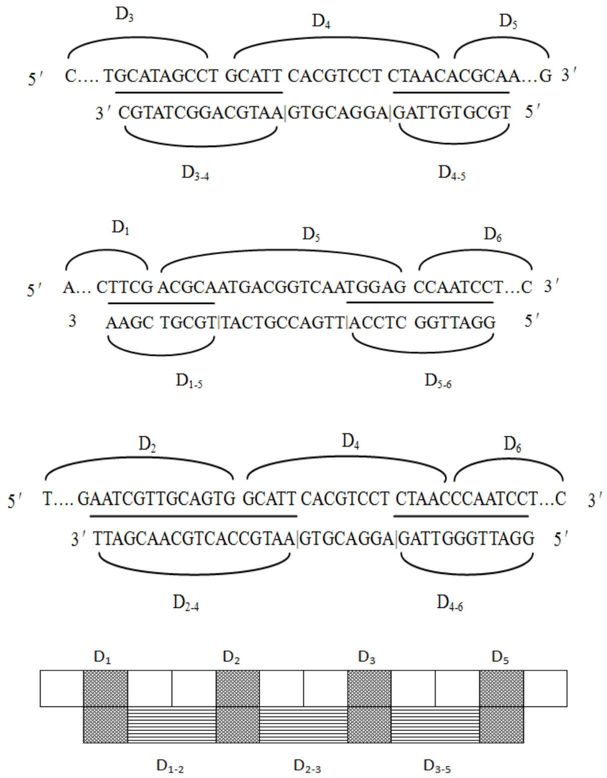
Then construct the connection fragments called connection code of two different items D_i and D_j. In order to ensure that each item could only load into the bag at most once, the connection code $D_{i\text{-}j}$ should satisfy one condition, 1≤ i< j ≤ 6. Synthesis different 15 connection codes which shown in Fig. 6.



D_{1-2}	AAGC|AGTAACATGCGAT
D_{1-3}	AAGC|GAGTACGCA
D_{1-4}	AAGC|CGTAA
D_{1-5}	AAGC| TGCGT
D_{1-6}	AAGC| GGTTAGG
D_{2-3}	TTAGCAACGTCAC| GAGTACGCA
D_{2-4}	TTAGCAACGTCAC|CGTAA
D_{2-5}	TTAGCAACGTCAC| TGCGT
D_{2-6}	TTAGCAACGTCAC| GGTTAGG
D_{3-4}	CGTATCGGA|CGTAA
D_{3-5}	CGTATCGGA| TGCGT
D_{3-6}	CGTATCGGA| GGTTAGG
D_{4-5}	GATTG|TGCGT
D_{4-6}	GATTG| GGTTAGG
D_{5-6}	ACCTC|GGTTAGG

Figure 6. Connection code.

Since the complementary characteristics of DNA sequence, different item code combined to form different DNA chains under the action of the connection code. Each chain represents a combination of several items, which contains optimal solution of the problem. Some possible combinations are shown in Fig.7.


D_3
D_4
D_5
5′ C.... TGCATAGCCT GCATT CACGTCCT CTAACACGCAA...G 3′
3′ CGTATCGGACGTAA|GTGCAGGA|GATTGTGCGT 5′
D_{3-4}
D_{4-5}
D_1
D_5
D_6
5′ A... CTTCG ACGCAATGACGGTCAATGGAG CCAATCCT...C 3′
3 AAGC TGCGT|TACTGCCAGTT|ACCTC GGTTAGG 5′
D_{1-5}
D_{5-6}
D_2
D_4
D_6
5′ T.... GAATCGTTGCAGTG GCATT CACGTCCT CTAACCCAATCCT...C 3′
3′ TTAGCAACGTCACCGTAA|GTGCAGGA|GATTGGGTTAGG 5′
D_{2-4}
D_{4-6}
D_1
D_2
D_3
D_5
D_{1-2}
D_{2-3}
D_{3-5}

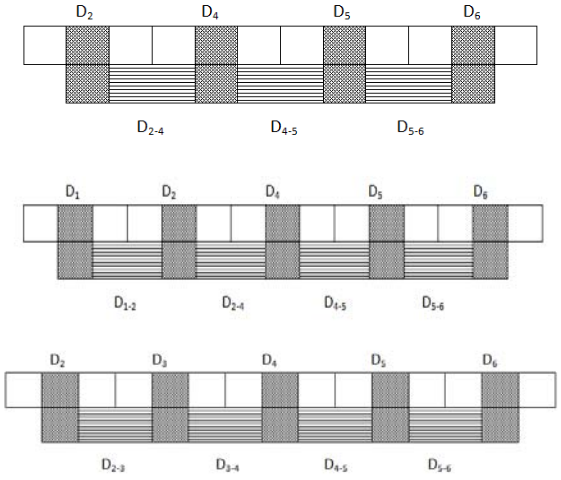


Figure 7. DNA chain composed of item code and connection code.

B. Biological Tools for DNA Computing

The basic assumptions are that the data can be encoded in DNA strands and are error-free, and that molecular biologic technologies can perform all computational operations. The models of DNA computing are based on different combinations of the following biological operations on DNA strands:

- Melting/annealing: break apart/bond together two single DNA strands with complementary sequences.
- Synthesis of a desired DNA strand of polynomial length.
- Separation of the strands by length.
- Merging: pour two (or more) test tubes into one.
- Extraction: extract the strands that contain a given pattern as a substring.
- Amplifying: make copies of DNA strands by using the polymerase chain reaction (PCR).
- Polymerization: transform a single strand that has a portion of double-stranded subsequence into an entire double-stranded molecule.
- Cutting: cut DNA strands by using restriction enzymes.
- Ligation: paste DNA strands with complementary sticky ends by using ligases.
- Substitution: substitute, insert, or delete DNA sequences by using PCR site-specific oligonucleotide mutagenesis.
- Marking single strands by hybridization.
- Destroying the marked strands.
- Detection: given a tube, check if it contains at least one DNA strand.
- Number: given a tube, count many DNA strands in it.

V. The Computing Algorithm for the 0-1 Knapsack Problem

A. Algorithm Framework

The computing algorithm for the 0-1 knapsack problem composes two major stages: computation and detection.

In the computation procedure, the DNA fragments which represent items and linkers in the knapsack are mixed, and the reaction results will be DNA strands, whose mobility ratio is in reverse logarithm relationship with its quantities of base pairs in the gel substance.

In the detection procedure, the feasible solution is from biological detection. Beside taking the double-stranded DNA items and linkers as the reaction materials, which is mentioned in the previous section, it is necessary to put in high efficient enzymes and corresponding detection methods in order to use DNA solution ligation reaction to realize the algorithm.

B. Algorithm Description

The computing algorithm for solving the 0-1 knapsack problem is presented below:

- Step1: Encoding the items, synthetic the oligonucleotide fragments and the connection fragments corresponding to each item.
- Step2: The DNA fragments were mixed together. Based on the complementary W-C rule, various double strands are generated randomly.
- Step3: Determine the DNA fragments corresponding to solution by gel electrophoresis. Remove these sequences whose lengths are greater than the capacity M of knapsack.
- Step4: The remaining double strands are converted into single strands by melting operation. Discard the strand which does not contain connection code and reserve strands which have value information and connection fragment.
- Step5: Separate the DNA strands without connection code through the restriction enzymes. For example, a strand with connection code from the double-stranded DNA in Fig.7 is added complementary strands D^x_3, D^x_4 and D^x_5, after annealing, these strands combine to a double strands, the restriction enzymes are added to delete the connection code. The implementation of the operation is shown in shown in Fig.8.

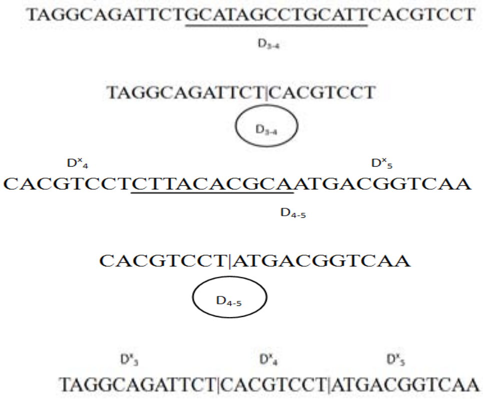


Figure 8. Remove the link segment.

- Step 6: Analysis the solutions. The maximum is the value of the items could into the knapsack.

The results for knapsack problem obtained by the algorithm: Choose four items (1, 3, 5, 6) into the bag, the total weight is ninety and the highest profit is fourty.

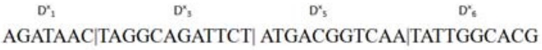


Figure 9. Optimal solution.

C. Algorithm Evaluation

These computing methods generate alternant single-stranded or double stranded DNA in a single legation reaction. However, the part of DNA single strand can easily combine with other molecules through hydrogen bonds based on complementary Watson-Crick law. Therefore these methods do not guarantee the generation of the optimal solution.

The proposed DNA encoding method is an improvement on the previous ones. In the new algorithm, each item is encoded using two DNA strands D^x_i and D^x_j of different length. The shorter DNA strand D^x_j can combine with the centre part of the longer DNA strand D^x_i to form double DNA sequence. So the random solution generated in a sing ligation reaction using the proposed DNA encoding method are stable DNA sequences instead of alternant single strand or double strands. Moreover, the proposed encoding method can be generalized, and it also has characteristics of easy encoding and low error rate.

Here we use six items in this small scale example. It is the same procedure to solve large scale 0-1 knapsack problem. The importance of the algorithm is the encoding of items, especially the relations between weight and value. Sometimes, the weight and value of items may both be transformed into new integer before encoding to ensure correct computing.

VI. CONCLUSIONS

DNA performs millions of operations simultaneously, generates a complete set of potential solutions, conduct large parallel searches and efficiently handle massive amounts of working memory.

In this paper, we presented a DNA encoding and computing model to solve the 0-1 knapsack problem. Compared with the previous methods, the new algorithm has two advantages. First, the hierarchy evolutionary searching algorithm is adopted to obtain good DNA encoding sequences, which consider the thermodynamic properties of DNA molecules in the encoding process in order to ensure the accuracy and reliability of results, such as the GC content of molecules remains at 50% as much as possible, which can effectively control various parameters of biochemical reactions. Second, the new coding strategy make the weight and quality of the item correspond with the DNA fragments. The length of the strand D^s_i which used to denote item *i* is equal to the weight w_i, and its center part is a length of the value $v_{I.}$ By means of the separation operation, all the sequence whose length is above to the knapsack capacity can be discarded. Feasible solutions are obtained through a series of biological reactions, and the solutions are analysis of the molecular composition to get the optimal solution.

Generally, the time complexity of the proposed algorithm is O (n^2) and space complexity is O (2^n). As a result, the DNA computing model proposed is effective in solving the 0-l Knapsack problem. It is plausible that the benefits may be expected to other combinatorial optimization problems.

ACKNOWLEDGMENT

This work was supported by Chongqing Natural Science Foundation, CSTC2011BB2063.

REFERENCES

[1] L.M. Adleman. Molecular Computation of Solutions to Combinatorial problems. Science, 266: p. 1021-1024, 1994.

[2] R.J. Lipton. DNA solution of hard computational problems. Science, 268: p. 542-545, 1995.

[3] Ouyang Q, Kaplan P D, Liu S, DNA solution of the maximal clique problem.Scienec 278: 446-449, 1997.

[4] Majid D, Hasan MN. A surface-based DNA algorithm for the solving binary kanpsack problem. Mathematics and Computation 188: p. 1991-199, 2007.

[5] Christiaan Henkel. DNA computing of solutions to knapsack problems. Biosystems, (88): p. 156-162, 2007.

[6] Zhu Ying, Ren Li Hong Ding Yongsheng, DNA igation design and biological realization of knapsack problem, Chinese Journal of Computers, vol 31 No. 12 2207-2214, 2008.

[7] Ye Lian, Chen Jing, Xing Yong-Kang, Hierarchy evolutionary searching algorithm for DNA encoding sequence, 2010 International Conference on Information Technology for Manufacturing Systems, 94-98, 2010.

[8] Wu keerong, Yeh chungwei, Solution to the 0-1 multidimensional knapsack problem based on DNA computation, International Conference on Information Technology for Manufacturing Systems, ITMS 1767-1772, 2011.

[9] Kim Eungyeong, Ahn, Chang Wook, DNA sequence generation algorithm using DNA Coding method for 0/1 Knapsack Problem, Proceedings 2010 IEEE 5th International Conference on Bio-Inspired Computing, 168-173, 2010.

[10] Yong Jiang, Kenli Li, Zhishui Zhong, Goodman Makojoa , An 0 (1.414) Volume Molecular Solution for the 0-1 Knapsack Problem on DNA-Based Supercomputing , 978-984, 2010.

[11] Bhalgat, Anand1, Goel, Ashish2 ; Khanna, Sanjeev Improved approximation results for stochastic knapsack problems, Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 1647-1665, 2010.

[12] Yusof Yuhani, Sarmin Nor Haniza, Goode T. Elizabeth, Mahmud Mazri, Heng Fong Wan, An extension of DNA splicing system, 6th International Conference on Bio-Inspired Computing: Theories and Applications, 246-248, 2011.

[13] Kuruppu, S , Beresford-Smith, B , Conway, T , Zobel, J , Iterative Dictionary Construction for Compression of Large DNA Data Sets, Tranaction on computational biology and bioinformatics, 9 : 1 : 137-149 , 2011.

[14] Taha Ghasemia, Mohammadreza Razzazi Development of core to solve the multidimensional multiple-choice knapsack problem, Computers & Industrial Engineering Volume 60, Issue 2, 349-360, 2011.

[15] M. Darehmiraki, H. M. Nehi, "Molecular solution to the 0-1 knapsack problem based on DNA computing, " Applied mathematics and computation, Vol 187, Issue. 2, pp. 1033-1037, 2007.

[16] E-G. Kim, S.-Y. Lee, A DNA Sequence Generation Algorithm for Traveling Salesman Problem using DNA Computing with Evolution Model, " International Journal of Fuzzy Logic and Intelligent Systems, Vol. 16, No. 2, pp. 222-227, 2006.

[17] W.L. Chang, Fast parallel molecular solutions for DNA-based supercomputing: Factoring Integers [J]. IEEE Transactions on Nano bioscience, 4 (2): 149-163, 2005.

[18] X. L. Wang, Z. M. Bao , "Solving the SAT problem using a DNA computing algorithm based on ligase chain reaction, " Bio Systems, vol.91, pp.l7-125, 2008.

Lian Ye received Bachelor degree of Computer Science and Technology from Chongqing University, Master degree of Applied Computer Technology from Chongqing University. She is currently working toward a Ph.D. degree in Computer Software and Theory, Chongqing University. She is currently a Lecturer with the School of Computer Science, Chongqing University. Her research interests include Biological computing, Intelligent Systems.

Min Zhang received the Master of Computer Science and Technology in computer department in Chongqing University, and the Ph.D. degree in computer systems engineering from Chongqing University. Her research interests include imaging processing and pattern recognition.