Detecting the solution space of vertex cover by mutual determinations and backbones

Wei Wei, Renquan Zhang,* Binghui Guo, and Zhiming Zheng
LMIB and School of Mathematics and Systems Sciences, Beihang University, 100191, Beijing, China
(Received 15 November 2011; revised manuscript received 23 March 2012; published 25 July 2012)

To solve the combinatorial optimization problems, especially the minimal Vertex-cover problem with high efficiency, is a significant task in theoretical computer science and many other subjects. Aiming at detecting the solution space of Vertex-cover, a new structure named mutual-determination is defined and discovered for Vertex-cover on general graphs, which results in the emergence of strong correlations among the unfrozen nodes. Based on the backbones and mutual-determinations with node ranks by leaf removal, we propose a Mutual-determination and Backbone Evolution Algorithm to achieve the reduced solution graph, which provides a graphical expression of the solution space of Vertex-cover. By this algorithm, the whole solution space and detailed structures such as backbones can be obtained strictly when there is no leaf-removal core on the given graph. Compared with the current algorithms, the Mutual-determination and Backbone Evolution Algorithm performs as well as the replica symmetry one in a certain interval but has a small gap higher than the replica symmetric breaking one and has a relatively small error for the exact results. The algorithm with the mutual-determination provides a new viewpoint to solve Vertex-cover and understand the organizations of the solution spaces, and the reduced solution graph gives an alternative way to catch detailed information of the ground/steady states.

DOI: 10.1103/PhysRevE.86.016112 PACS number(s): 89.90.+n, 89.20.Ff

I. INTRODUCTION

The minimal vertex-cover problem (Vertex-cover) belongs to one of Karp's 21 NP-complete problems [1] and the six basic NP-complete problems [2,3], and is considered as one of the classical problems in theoretical computer science. The aim of this problem is to mark a minimum subset of vertices such that there are at least one vertex of each edge in the subset. There are a large number of applications of this problem in the related real networks, such as immunization strategies in networks [4] and monitoring of internet traffic [5].

There is a threshold behavior of the minimum vertex-cover problem on the Erdös-R*é*nyi random graph [6], which reveals that the typical running time of solving algorithms for Vertex-cover changes from polynomial to exponential when the order parameter becomes larger than the Euler number e [7,8]. This phase transition phenomenon is considered to have an intrinsic correspondence with the clustering structure of solution space, which has already been observed in statistical physics when studying spin glasses [9,10]. Although most statistical physicists believe that the clustering structure leads to the failure of replica symmetry, details of the relationship between searching solutions and the structure are not well established, and how the clustering structure looks is far from being clear for most models [11–13]. From an algorithmic point of view, the solutions' structure makes a great effect on the algorithm to find the solutions, which sets barriers to local searching algorithms and makes the computation expensive [14,15]. So the features of solutions' structure are explored by different approaches. Till now, some typical structures such as clustering, backbone, backdoor [16], and frustration [17] have been widely investigated to understand the structure of solutions more clearly. In particular, Zhou [17,18] has proposed the *long-range frustration* structure, and Krzakala [19] has provided a formal definition as *long-range correlation*. The long-range correlation and backbone structures are treated as the origin of the replica symmetric breaking and the high computational complexity. And, based on the analysis of these typical characteristics of the solution space, many efficient searching algorithms are proposed to solve NP-complete problems, such as Belief Propagation and Survey Propagation [20–22].

In this paper, a *mutual-determination* structure is proposed by statistical mechanic approach to investigate the solution space of the minimum vertex-cover problem. This structure reflects the feature of the Hamming distance [23,24] among solutions and describes how tight the correlations among unfrozen variables are. By this structure, we can detect the equivalent variables in the solution space [25], i.e., the variables which must take the same or the opposite Boolean values. Furthermore, based on the existence of the mutual-determination in the solution space of Vertex-cover, the node ranks of a given graph by the leaf-removal process are provided to describe the influence orders of leaves in different levels. Taking advantage of the leaf-removal ranks and the relationship of mutual-determination with the backbone and unfrozen variables, we can have a much clearer understanding of the evolution of the states in the solution space when a new node is added, and a *reduced solution graph* is defined to exactly express the structural information of the solution (sub-)space. Finally, an algorithm named the *Mutual-determination and Backbone Evolution Algorithm* is proposed by the evolution of the mutual-determinations and backbones on the reduced solution graph, and some analysis and numerical experiments are given to verify its efficiency and adaptability. This algorithm is complete to find the whole solution space of Vertex-cover when there is no leaf-removal core on the graph, otherwise an approximated one with a small error to the exact results.

II. DEFINITION OF MUTUAL-DETERMINATION

A vertex cover on an undirected graph $G(N,M)$ with N nodes and M edges is a subset $S = \{i_1, i_2, \ldots, i_m\}$ of its nodes such that every edge has at least one endpoint in S. The

*zhangrenquan09@smss.buaa.edu.cn

minimum vertex-cover problem is an optimization problem to find the minimum size of a vertex cover on a given graph. Mapped to a spin-glass model, the energy function of the minimum vertex-cover problem can be written as

$$E[\{\sigma_i\}] = -\sum_{i=1}^{N} \sigma_i + \sum_{(i,j)\in E(G)} (1+\sigma_i)(1+\sigma_j), \qquad (1)$$

where $E(G)$ denotes the edge set and (i,j)s are edges in it, and spin/variable $\sigma_i = -1$ if node $i \in S$ (covered) and $\sigma_i = 1$ otherwise. Then, different energy levels are produced by different assignments or *configurations* in the terminology of spin-glass theory. The assignments with the lowest energy are named *solutions*/*ground states*, and the set of all these solutions achieving the lowest energy (minimum vertex cover) is named the *solution space* $\mathcal{S}$.

Backbones [16] and long-range correlations [17,19] are both the typical structures of solution space of combinatorial optimization problems, which have been well studied in algorithmic and statistical analysis. In the solution space $\mathcal{S}$, spin σ_i is *frozen* or called a *backbone* if it takes the same value in all solutions; otherwise it is *unfrozen*. For an unfrozen spin σ_i, if its taking some value will influence an infinite number of other spins [assumed $O(N)$ with the total number of N spins], its assignment can lead to *long-range correlations* on distant nodes [18,19]. Recent research suggests that the complicated organizations of the solutions of combinatorial optimization problems, e.g., backbones and long-range correlations, would be the kernel reason for the algorithmic hardness to find a solution for large-scale combinatorial optimization problems with massive constraints and variables [26]. To study the solution space $\mathcal{S}$ of Vertex-cover, we classify the variables as unfrozen, *positively frozen* (frozen to +1), and *negatively frozen* (frozen to −1) variables.

As a generalization of the backbone and long-range correlation, a new structure named *mutual-determination* is proposed to achieve a better understanding of the solution space, which can be viewed as an interactive relation of unfrozen variables in the solution space. If two unfrozen variables form a mutual-determination, the fixation of the assignment of any one will result in the fixation of the other in the solution space. Indeed, it is a special relation implied by the constraints that two unfrozen variables can be mutually determined by each other, i.e., if two unfrozen variables σ_i,σ_j form a mutual-determination, then $\sigma_i + \sigma_j = 0$, and these mutual-determination relations can exist only among the connected nodes on the graph for Vertex-cover, which will be indicated in the following sections. When two unfrozen nodes which are connected by a link form a mutual-determination for Vertex-cover, it means that if one of them is covered, the other should be uncovered, and there is one and only one that should be covered for this pair of nodes.

For the well-known Survey Propagation algorithm [21], it takes advantage of the idea of the backbone and long-range correlation to gradually eliminate variables and constraints of the original problem in size, and achieves excellent performance for solving 3-SAT, Vertex-cover, etc. For the special formulation of mutual-determination structure in Vertex-cover and that the variables in mutual-determination relations are equivalent variables, we can use variable substitution to decrease the number of variables in the original problem to obtain new algorithmic strategies and have an overall observation of the solution (sub-)spaces. In the following sections, we will use the backbones and mutual-determinations to analyze the solution space of Vertex-cover.

III. REDUCED SOLUTION GRAPH OF VERTEX-COVER

To study the solution space of Vertex-cover, the leaf removal [29] should be mentioned as inspiration. Given a graph G, a *leaf* is a couple of nodes $\{v,w\}$ in which the first one has degree 1 and the second one is the only neighbor of it. Here, node v is a pendant point in the graph, node w acts as a petiole, and for the same petiole there may be more than one pendant points connecting it. To define the *leaf removal*, if the nodes pair $\{v,w\}$ is a leaf in graph G, remove the two nodes with the edges touching them. It is very interesting that the leaf removal process can destroy all the leaves in graph G and can produce new leaves for the rest of the graph. In Fig. 1 a leaf removal process for a simple graph is shown.

By this leaf removal process, we can find that there exist some graphs which have no leaf-removal core until the termination of this process, which means that each node belongs to a leaf at a certain stage of the iterated leaf-removal process for the graphs. For the Vertex-cover of these graphs, to obtain the minimum vertex cover for the graph, there is one and only one node that should be covered for each leaf. By the results in Ref. [29], a trivial minimum vertex cover can be obtained by making all the petioles covered with all the pendant points uncovered in different levels of leaves, e.g., making the nodes $\{c,e,g\}$ or $\{c,e,f\}$ covered and the rest uncovered leads to minimum vertex covers in Fig. 1. Here, we will take use of this trivial solution to construct a relationship between/among nodes in a leaf.

In Fig. 1 the first subgraph (1) is the original graph. For the first leaf $\{a,b,c\}$ in this graph which lies in the red rectangle of

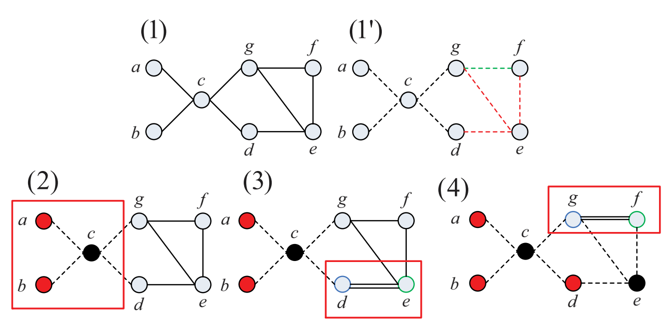

FIG. 1. (Color online) The subgraph (1) is the original graph and the subgraph (1′) reflects the leaf removal process. For the original graph (1), nodes $\{a,b,c\}$ form a multiple leaf with the common petiole c, and the corresponding edges are marked by black dashed lines; after removing this leaf, a new leaf with nodes $\{d,e\}$ appears, and the corresponding edges are marked by red dashed lines; when leaf $\{d,e\}$ is removed, the last new leaf $\{f,g\}$ is produced, and the corresponding edge is marked by a green dashed line. The subgraphs (2–4) reveal the relationship of nodes in the leaves of a graph at different leaf-removal stages. There are three such leaves in the graph, which are marked by the red rectangles in subgraph (2–4), and the relations among the nodes are revealed by the underlying constraints in the leaves for Vertex-cover.

subgraph (2), there are two pendant points with one petiole, and to ensure the minimum coverage of the subgraph of $\{a,b,c\}$, the only way is to cover the petiole node c and make the two pendant points $\{a,b\}$ uncovered. In this case, the node c acts as a negatively frozen node (backbone), which is marked by a solid black circle, the nodes a,b act as positively frozen nodes, which are marked by solid red circles, and the edges connecting them are marked by dashed ones. For the second leaf $\{d,e\}$ in the red rectangle of subgraph (3), there is only one pendant point d and one petiole e, and to ensure the minimum coverage of the subgraph of $\{a,b,c,d,e\}$, covering any node of $\{d,e\}$ with the other one uncovered will make an optimization solution. In this case, the assignments of σ_d,σ_e must be opposite, and they are *mutually determined* in the solution space of Vertex-cover, which is denoted by a double edge and two nodes with different colors. For the third leaf $\{g,f\}$ which is similar as the leaf $\{d,e\}$, their relation is also mutually determined. However, the relation of $\{g,f\}$ has influence on the leaf $\{d,e\}$, which makes the states of nodes $\{d,e\}$ changed to be backbones, and the detailed techniques for this case will be discussed in the following sections.

Based on the analysis inspired by the example in Fig. 1 and the backbone and mutual-determination structures, we can construct an expression of the solution space of Vertex-cover which is named the *reduced solution graph* $R(G)$ to reveal the underlying structures by modifying the formations of graph G. To achieve the reduced solution graph, what we should do is to identify the backbones and mutual-determinations by different marks and retain all the unfrozen nodes with edges among them on a given graph G. To show different minimum vertex covers of a given graph G by its reduced solution graph $R(G)$, the positive/negative backbones on it are marked by solid red/black circles, and *double edges* connecting unfilled hollow nodes suggest that the relations between the nodes are mutual-determinations; the edges connecting the backbones will be changed to dashed ones, and any mutual-determination connects two unfrozen nodes with different colors (blue or green), which cannot take the same value simultaneously; the edges connecting two unfrozen nodes except mutual-determinations are retained unchangeably. By the leaf-removal process and the strong correlation among/between nodes in the leaves, the mutual-determination can only be in the pendant point and its petiole for each leaf. When there is no leaf-removal core for the given graph, each node belongs to a leaf at a certain stage of the iterated leaf-removal process, and the role of each node can only be backbone or belong to some mutual-determination. Then, it can be obtained that the reduced solution graph can express the solution space of Vertex-cover strictly, and whether this expression is also effective for general graphs with leaf-removal cores and how to obtain the reduced solution graph of Vertex-cover will be discussed in the following.

In order to have a convenient analysis of the reduced solution graph, the leaf-removal [29] sequence is very important. Here, we take advantage of the sequence of the leaf removal to define the rank of nodes:

Step 1: All the pendant points in the graph are assigned to sequence order/rank 1, and their neighbors (the petioles) have rank 2.

Step 2: Remove the leaves with edges connecting them from the graph. After the first-level leaf removal, all the new produced pendant points are assigned to rank 3, and their corresponding petioles have rank 4.

Step 3: Repeat the steps 1–2 and assign increasing ranks until there are no new leaves produced. If there is still a leaf-removal core after the above two repeated steps, assign the nodes in the core with ranks according to their already ranked neighbors by a gradual increase.

IV. ANALYSIS OF MUTUAL-DETERMINATION IN THE SOLUTION SPACE OF VERTEX-COVER

In this section, we are concerned with achieving the reduced solution graph $R(G)$ by determining the states of the nodes one by one following the leaf-removal sequence/ranks. This process is fulfilled by a method similar to the cavity method, and for each node its state is determined by the local environment of itself. Assuming that we have already obtained the reduced solution graph $R(G)$ and considering a new node i connected to a graph G with k edges, the newly produced graph is denoted by G', then what we aim for is to determine the state of node i in G' and obtain $R(G')$. For the neighborhood of node i, there are three kinds of neighbors: positively frozen ones, negatively frozen ones, and unfrozen ones in $R(G)$.

A. Local evolution of mutual-determinations and backbones

In this subsection, we consider different local environments of a new node i, and investigate the state determination and evolution of it with associated nodes. Taking advantage of the analysis in Ref. [18], we first study the following four cases:

Case A: Only one of its neighbors is positively frozen in G; some other neighbors are unfrozen nodes which can take spin value -1 simultaneously.

In case A, energy increase is unavoidable when node i is added. When σ_i takes value -1 (covered), its neighbors of nodes are free to take their spin values in the original G, and new covers of the new graph G' come out with the lowest energy; when σ_i takes value 1, the positively frozen neighbor should be changed to an unfrozen node taking -1 now, e.g., adding the node e to the subgraph of $\{a,b,c,d\}$ leads to a mutual-determination of $\{d,e\}$ in Fig. 1, and the above subgraph in Fig. 3 also shows this process. Then, mutual-determination of the new added node i with the original positively frozen node j is formed.

Case B: There are two or more neighbors which are positively frozen.

In case B, energy increase is unavoidable when node i is added. To obtain a coverage of the new graph G' from the original one G, the new added node i must be covered without another choice. Then, node i is negatively frozen. (There will be a supplementary and additional adjustment for this case in the following case E when the positively frozen neighbors have some common properties.)

Case C: There is no neighbor which is positively frozen, but all the unfrozen neighbors can take spin value -1 simultaneously.

In case C, energy increase is avoidable when node i is added, and the new added node i should be uncovered. When

σ_i takes value $+1$, all the unfrozen neighbors should take -1 simultaneously. Then, by the mutual-determinations and coverage of each edge in the reduced solution graph $R(G)$, that these unfrozen neighbors change to be negatively frozen will lead a number of associated unfrozen nodes to be frozen.

Case D: There is at least one pair of unfrozen neighbors that cannot take spin value -1 simultaneously.

In case D, as the two neighbors can not take -1 simultaneously, energy increase is inevitable. Then, the new added variable i should take -1 to ensure the coverage. On the reduced solution graph $R(G')$, this incompatibility will lead to incompatible cycles (like that in the bottom subgraph of Fig. 3) and make great trouble for the exactness of the local evolution. However, for the incompatible cycles, making any other node except i frozen to -1 and the rest unfrozen nodes connected by alternatively existing double edges will also have the same effect and ensure the coverage. Thus, in case D freezing the new added node i to -1 will reduce the whole solution space to a partial solution subspace, and even make further incorrectness in adding the sequential nodes. Nevertheless, as takeing any one to be frozen on the incompatible cycle leads to a solution subspace with the same size, we have the convenient way to make i a negative backbone.

By the above analysis, the incompatible cycle in the reduced solution graph makes a possible inaccurate choice of the negatively frozen backbone. Thus, the hardness of solving Vertex-cover mainly stems from the incompatible cycles. Therefore, the operation in case D may not only reduce the solution space, but also lead to nonoptimal vertex covers at later stages.

There exists an interesting entanglement between case A and case C. In case C, some added node i_1 has a positively frozen state, and its unfrozen neighbors are forced to be frozen with some associated nodes. In case A, if some new added node i_2 is connected to i_1 and forms mutual-determination with i_1 by the rule of case A, the nodes that have been frozen by i_1 should be released to their original unfrozen states. To fulfill this releasing steps, an *additional mark* should be sticken to the node number, e.g., a node (4,7) means that node 4 is influenced to be frozen by the operation of adding node 7. Indeed, this *freezing influence* happens only in case C with node 7 positively frozen. Then, if the state of node 7 is changed to be unfrozen by adding a new node in case A, we can release all the nodes with marks $(*,7)$ and change the corresponding numbers to $(*,0)$, in which 0 means the current state of the node is unfrozen. This operation is named *releasing operation*.

For the releasing operation, there is a special case which should be considered for adding a new node i:

Case E: There are more than two neighbors which are positively frozen and have the same additional mark, and the unfrozen neighbors can take spin value -1 simultaneously. This case is a supplementary and additional adjustment of case B.

In this case, the current node should form mutual-determinations with the positively frozen nodes whose additional marks are the same, and the releasing operation is operated for these positively frozen neighbors (a simple example can be seen in the bottom subgraph of Fig. 7).

B. Some supplementing techniques for the states evolution

In the above cases A–E, there is an important condition which should be determined frequently, that is, whether the unfrozen neighbors can take spin value -1 simultaneously or not. To determine this condition, what we should do is to consider the freezing influence by fixing all these unfrozen neighbors to be covered simultaneously, and this operation may lead to the freezing of a large number of associated unfrozen nodes. If there is contradiction in this process (some node k is required to be negatively frozen by the freezing influence of some unfrozen neighbor i taking -1, but node k is also required to be positively frozen by another unfrozen neighbor j taking -1), these unfrozen neighbors cannot take -1 simultaneously; otherwise they can take -1 simultaneously.

First, a depth-first directed algorithm is used to fix the unfrozen neighbors and return the contradiction that one node is required to be different value by two neighbors. In this process, the $state[i]$ denotes the status of node i and $R(G)$ the recent reduced solution graph. Furthermore, the $state[i] = -1, 0$ and 1 denote the node i is covered, undetermined, and uncovered. The algorithm of *depth-first directed* is shown as follows:

```
algorithm depth-first directed(i,R(G))
begin
    if(state[i] = 1)
    begin
        for(all the unfrozen neighbors j of i in R(G)) do
            if(state[j] = 0)
                state[j] = −1;
                depth-first directed(j,R(G));
            else if(state[j] = 1)
                return(false);
            else
                continue;
    end
    if(state[i] = −1)
    begin
```

```
                for(all the unfrozen neighbors j of i with E_ij = 2 in R(G)) do
                    if(state[j] = 0)
                        state[j] = 1;
                        depth-first directed(j,R(G));
                    else if(state[j] = -1)
                        return(false);
                    else
                        continue;
        end
        return(true);
end
```

With the algorithm above, we can determine whether the unfrozen neighbors of node i can take spin value -1 simultaneously or not. Now taking i into the $R(G)$, we obtain the algorithm *determine*$(i,R(G))$ as follows:

```
algorithm determine(i,R(G))
begin
    initialize b=true and state[i] = 0 for all nodes in R(G);
    for(all the unfrozen neighbors j of i in R(G)) do
    begin
        state[j]=-1;
        b=depth-first directed(j,R(G));
    end
end
```

It is easy to see that only all $b = true$, the unfrozen neighbors of node i can take spin value -1 simultaneously, otherwise, there must be contradiction, which makes some $b = false$.

In the releasing operation for case A, to avoid some possible mistakes, a *checking technique* should be considered. When releasing the negatively frozen backbones in the releasing operations for case A, its local environment should be considered, and if there are positively frozen neighbors for the current negatively frozen backbone whose additional mark is not the same as itself, the releasing process should be stopped [e.g., the node b in the process of subgraphs (3–4) in Fig. 2].

Be specific to this checking item, after the operations of case A–D, a *rechecking technique* should be added: when the freezing influence and releasing operations of adding a new node have been done, we should check any of the negatively frozen backbone whose additional mark is not 0, and if there is only one positively frozen neighbor for itself, release the negatively frozen backbone with the only positively frozen neighbor and the nodes which have the same additional mark with it [e.g., the node b in the process of subgraphs (4–5) in Fig. 2].

At last, by the process of above analysis, a complicated structure-odd cycles on the reduced solution graph could come into view, which makes conflicts for the relations among the unfrozen nodes. For the example in Fig. 2, the subgraphs (1–7) provide the process from the original graph to the reduced solution graph by adding the nodes one by one using our techniques above, but unfortunately the unfrozen nodes b,c,d,e,f,g,h in subgraph (7) form an odd cycle. In the odd cycle, we find that any node except b taking any value will force node b to be negatively frozen, and it is an incompatible cycle. To break this disharmony, the only way is to change the state of b to be a negative backbone and make corresponding changes for its neighbors with lower ranks [e.g., the process from subgraph (7) to (8) in Fig. 2]. This technique is named *odd cycles breaking*.

C. Global characteristics of mutual-determinations

In the following, we will have an explicit discussion of the mutual-determinations and unfrozen nodes structure of Vertex-cover. As mentioned above, if some node i forms mutual-determination with a node j, i.e., $\sigma_i = -1$ forces σ_j to take $+1$, correspondingly by the Vertex-cover, we have that $\sigma_i = +1$ requires $\sigma_j = -1$ to satisfy the coverage. If a node j forms a mutual-determination chain with some other nodes $j_0, j_1, \ldots, j_k$, a possible way is that the edges $(j, j_0), (j_1, j_2), \ldots$ on the reduced solution graph are all double edges, i.e., all these pairs of nodes form mutual-determinations, which is shown in the Cycle 2-Compatible Cycle in Fig. 3. When the node j takes value -1 (covered), the nodes $j_0, j_2, \ldots, j_{2l}, \ldots$ must take $+1$ by the mutual-determination relations and coverage of the edges connecting them, and the nodes $j_1, j_3, \ldots, j_{2l+1}, \ldots$ must take -1. Therefore, the alternatively existing double edges on the reduced solution graph lead to the emergence of the strong correlation for nodes of long distance.

Without confusion, we neglect the backbones in the reduced solution graph but keep the unfrozen nodes. As we know, there are almost no local cycles on random graphs and the cycle sizes on random graphs are of $O(\log N)$. By this characteristic of random graph, the emergence of the Cycle 2-Compatible Cycle in Fig. 3 leads to the long-range correlation structure, and indeed the alternating mutual-determination chain is the only way to produce the long-range correlation in Vertex-cover.

When the unfrozen neighbors of the new added node i only have influence range over tree structures, which means that the

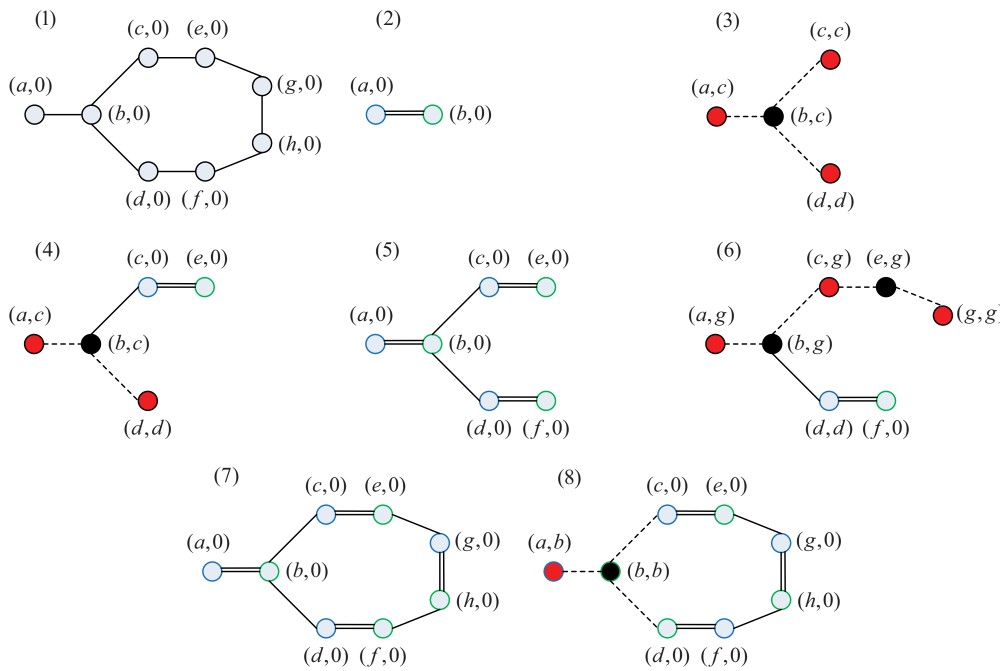

FIG. 2. (Color online) An example for the emergence of odd cycles of unfrozen nodes and the way to break this conflicted cycle. Subgraph (1) provides the original graph for Vertex-cover; subgraph (2) describes the process of adding the nodes $\{a,b\}$, which consists of a procedure of case A to produce a mutual-determination; subgraph (3) describes the process of adding the nodes $\{c,d\}$, which consists of a procedure of case C to produce positively frozen backbones; subgraphs (4–5) are for adding nodes e and f separately, which consists of the procedure of case A again, and the checking technique works when adding e and the rechecking technique works when adding f; subgraphs (6–7) correspond to the process of adding nodes g,h, which can be followed by case A and C, and the freezing influence works when adding g, and the releasing operation works when adding h; subgraph (8) is obtained by breaking the odd cycle to obtain the real reduced solution graph, which changes the node b with lowest rank on the cycle to be negatively frozen.

double edges belonging to different unfrozen neighbors are disconnected except i. At this time, these unfrozen neighbors cannot propagate their influence to each other, and they can take -1 simultaneously. Similar to that in a random graph, as the increasing of the number of nodes and edges by the strategies of case A–E, the unfrozen nodes with double edges in the reduced solution graph may connect together and form cycles and even a giant connected component. Especially, for random graphs, the cycles connected by unfrozen nodes in the reduced solution graph must be with size of $O(\log N)$. Therefore, if the unfrozen nodes connect together to form a giant connected component [28], some of them taking value -1 will cause a percolation phenomenon [27] that many other nodes ($O(N)$) in this giant connected component will be forced to be frozen. As a result, the long-range correlation phenomenon emerges. In the work of Zhou [18], the long-range correlation of Vertex-cover for random graph appears at $c = e$. Indeed, by the literature of statistical mechanics, the existence of long-range correlation has close connection with the replica symmetric breaking of the solution space. As the correlation is formed by mutual-determinations, the long-range correlation can also provide an explicit explanation of the clustering phenomenon of solution space.

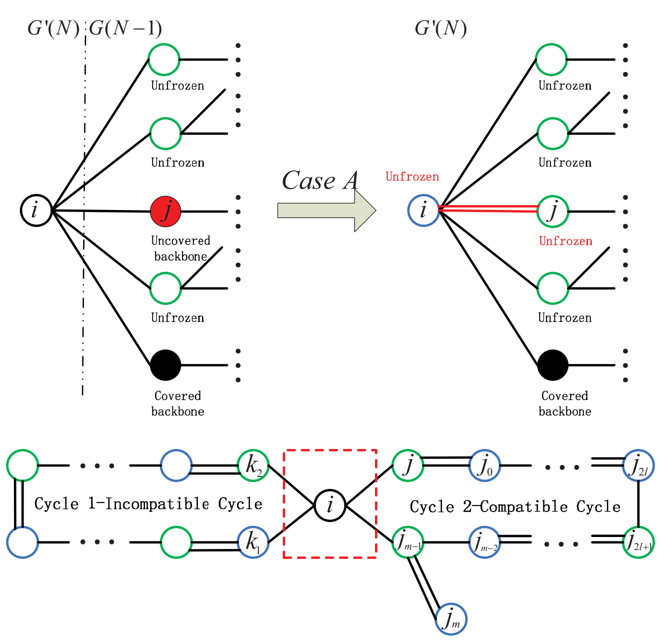

FIG. 3. (Color online) The top subgraph provides the process of case A to produce the mutual-determination structure, and the bottom subgraph reveals the formation of the incompatible cycle and the compatible cycle by mutual-determinations.

By the emergency of the long-range correlation nodes, the local environment of a new added node i becomes much more complicated. As unfrozen neighbors of the new added node i can be connected together by other unfrozen nodes, their values can not be assigned arbitrarily. In Fig. 3 a schematic view of the compatible and incompatible cycles of unfrozen nodes is shown. In Cycle 1, the unfrozen nodes k_1,k_2 are connected by unfrozen nodes with alternatively existing double edges, and it is easy to find that the nodes k_1,k_2 form a strong correlation relation and cannot take spin value -1 simultaneously. To the contrary, in the compatible Cycle 2, though the relation between j, j_{m-1} is also strong correlation, they can take spin value -1 simultaneously by the mutual-determination chain in Cycle 2.

V. MUTUAL-DETERMINATION AND BACKBONE EVOLUTION ALGORITHM FOR VERTEX-COVER

In this section, we will introduce an algorithm for solving Vertex-cover based on cases A–E and the node ranks. By the analysis in cases A–E, we consider to update the states of the original graph $G(N-1)$ after adding a new node i. As the node states are classified by mutual-determinations and backbones, we can get an algorithm to find the reduced solution graph $R(G)$ of Vertex-cover, this algorithm is named the *Mutual-determination and Backbone Evolution Algorithm*, shown as follows:

Let us take $G[v]$ as an induced subgraph of original graph G by adding node v and $E_{ij} = 1/0$ represents nodes i and j are connected/unconnected. Especially, when nodes i and j form the mutual-determination, we take $E_{ij} = 2$. $backbone[i] = -1,0$ or 1 means the node i is negatively frozen, unfrozen, or positively frozen node, respectively; $root[i]$ is the additional mark of node i.

The algorithm of *Releasing Operation* is shown as follows:

```
algorithm Releasing(i,G,ROOT)
begin
      backbone[i] = 0;
      root[i] = 0;
      for all neighbor j of i in G with root[j] = ROOT do
      begin
            if(backbone[j] = -1 and j has at least one positively frozen
            neighbor k with root[k] ≠ ROOT) then
                  continue;
            else do
                  Releasing(j,G,ROOT);
      end
end
```

The algorithm of *Freezing Influence* is shown as follows:

```
algorithm Freezing(i,G)
begin
      if(backbone[i] = 1)  do
            for all unfrozen neighbor j of i in G do
                  root[j] = root[i];
                  backbone[j] = -1;
                  Freezing(j,G);
      if(backbone[i] = -1)  do
            for all unfrozen neighbor j of i in G with E_ij = 2 do
                  root[j] = root[i];
                  backbone[j] = 1;
                  Freezing(j,G);
end
```

Now, we have the algorithm of Mutual-determination and Backbone Evolution Algorithm:

```
begin
   calculate the leaf-removal sequence layer[i] of G;
   vertex set v = ∅;
   initialize backbone[i] = 0 and root[i] = 0 for all i in G;
   for l = 1 to Max(layer[i],1 ≤ i ≤ N) do
   begin
      for all vertex i with layer[i] = l do
      begin
         v = v ∪ i;
         G' = G[v];
         calculate num is the number of positively frozen neighbors of i in G'
         if(num = 1)  do
         begin
            Pos is the positively frozen neighbor of i;
            determine(i,R(G'));
```

```
          if(all the b in determine algorithm is true) then
             E_{i,Pos} = 2;
             Releasing(Pos,G',root[Pos]);
             Rechecking technique and Odd cycles breaking;                    *Case A*
          else do
             backbone[i] = -1;
             root[i] = i;                                                     *Case D*
       end
       if(num >= 2)  do
       begin
          if(all frozen neighbors of i in G' have same additional mark) then
             determine(i,R(G'));
             if(all the b in determine algorithm is true) then
             Pos is one of neighbors of i randomly;
             E_{i,Pos} = 2;
             for all the neighbors j of i in G' do
                Releasing(j,G',root[j]);
                Rechecking technique and Odd cycles breaking;                 *Case E*
          else do
             backbone[i] = -1;
             root[i] = i;                                                     *Case B*
       end
       if(num = 0)  do
       begin
          determine(i,R(G'));
          if(all the b in determine algorithm is true) then
             backbone[i] = 1;
             root[i] = i;
             Freezing(i,G');                                                  *Case C*
          else do
             backbone[i] = -1;
             root[i] = i;                                                     *Case D*
       end
    end
 end
end
```

A. Some numerical results of the Mutual-Determination and Backbone Evolution Algorithm

In this section, some numerical experiments will be performed to verify the efficiency and performance of the Mutual-determination and Backbone Evolution Algorithm on random graphs.

To reflect the solution space structures of Vertex-cover, the ratios of negatively frozen backbones and positively frozen backbones are detected by the algorithm, in which one is monotonely increasing and the other is monotonely decreasing. In Fig. 4 the ratios of the backbones are shown by the solid triangles, and the unfrozen nodes have its ratio with the residual part of 1. Our results on the frozen nodes in one macroscopic state are higher than that in Ref. [17] mainly by the freezing influence and case D. Besides, in Fig. 5 the coverage of the Vertex-cover which is the size of the minimal vertex-cover is approximated by our algorithm, which is shown by the blue cycles with error bars and compared with the results of replica symmetry, replica symmetric breaking theory, and Survey Propagation. The results on coverage of our algorithm perform as well as that of replica symmetry when the average degree c is not very large (at least when $c <\sim 6$), but still have a small gap with the results of replica symmetric breaking theory and Survey Propagation. By the proof of the strictness of Mutual-determination and Backbone Evolution Algorithm when $c < e$ in the next section, our numerical results should be exact ones for the corresponding interval on random graphs.

As a comparison to the complete algorithm and the exact coverage, some experiments are made to verify the performance of the Mutual-determination and Backbone Evolution Algorithm which is an incomplete algorithm. In Fig. 6, average errors of minimal vertex cover between experimental and exact results are plotted to provide the difference between the exact results and our results on the coverage, and it is evident to see that these differences are not very big and have their scales no more than 0.04 for $c = 2,4,6$ with increasing sizes. As numerical experiments by complete algorithms for Vertex-cover can only be for instances with not very large scale, our experiments are realized only for $N < 160$ in Fig. 6. For finite N there is a small but still nonzero probability that a graph with a core appears also at $c = 2$. In that case, the algorithm would report an error with respect to the exact algorithm.

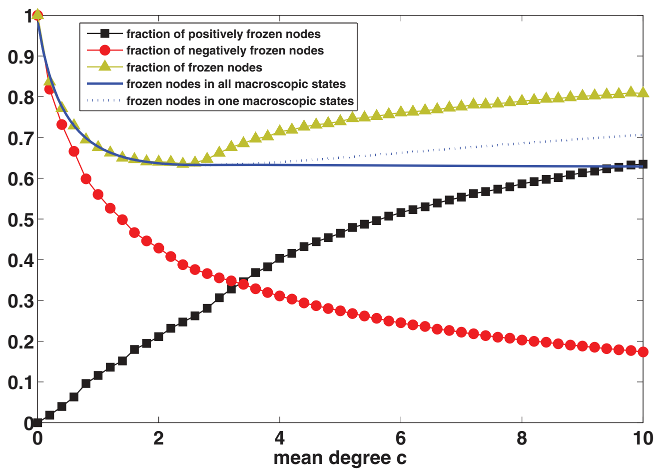

FIG. 4. (Color online) Numerical results by our algorithm of fraction of positively frozen (red solid cycle) and negatively frozen (black solid rectangle) nodes with different mean degrees, which are obtained by 1000 random instances with $N = 5000$ nodes; fraction of frozen nodes in one macroscopic state (yellow solid triangle) and its comparison with results of Ref. [17] (blue line for the fraction in all macroscopic states and blue dashed line for the fraction in one macroscopic state).

B. Mutual-Determination and Backbone Evolution Algorithm with some examples

To detect the reduced solution graph and provide a primary analysis of efficiency of our algorithm on the leaf-removal core, we will discuss the Vertex-cover on the complete graphs and cycles with even number of nodes for inspiration.

For the complete graphs, the process of our algorithm to obtain the reduced solution graph is rather simple: when the second node is added with an edge [subgraph (1) in Fig. 7], the mutual-determination emerges; for the following added nodes, their local environment satisfies the case D, and they

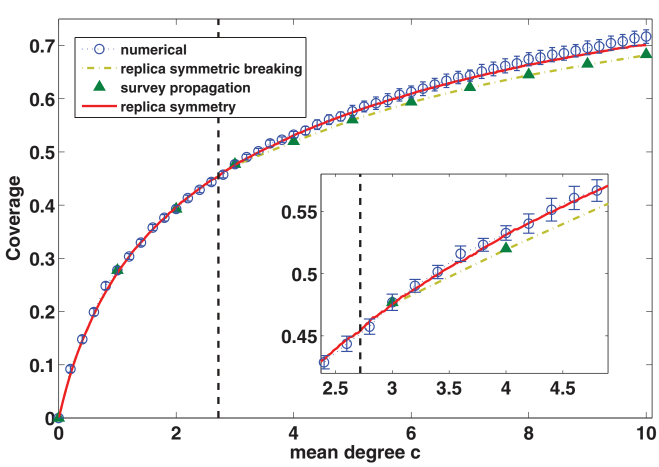

FIG. 5. (Color online) Numerical results for Vertex-cover by our algorithm. The blue cycles with error bar denote the minimal coverage ratios by our algorithm with $N = 5000$ and 1000 random instances; the yellow dashed line and green triangles denote the results of replica symmetric breaking theory and survey propagation, respectively; the red line denotes the results by replica symmetry theory; the vertical dashed line represents mean degree $c = e$.

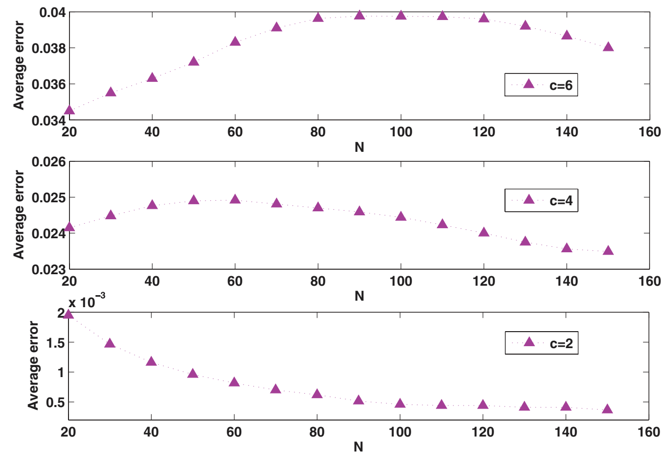

FIG. 6. (Color online) Average error of minimal vertex cover between experimental and exact results. All results are obtained by 1000 random instances with $c = 2$, $c = 4$, and $c = 6$ with different sizes $N = 20, 30, \ldots, 150$.

can only be negative backbones. This process is shown in Fig. 7 by a typical graph K_5. It is easy to know that for the complete graph K_N there must be $N - 1$ nodes covered, and our results of reduced solution graph correspond to a solution subspace of the Vertex-cover of K_N. The whole solution space possesses N solutions, and by our algorithm we can obtain two solutions. Therefore, the mutual-determination and Backbone Evolution Algorithm is an incomplete algorithm for the solution space, but it may be efficient for finding one solution of Vertex-cover. Certainly, as analyzed in the above section, the incompatible cycles of the unfrozen nodes will bring the intrinsic difficulty for solving and our algorithm can only obtain some approximated solutions for the original problem.

Then, the cycles C_{2N} with even number of nodes are considered. The process of our algorithm to obtain its reduced solution graph is a regular process: When an odd number of nodes are added, there are no unfrozen nodes in the reduced solution graph; when an even number of nodes are added, the releasing operations should be considered and all the nodes are unfrozen with double edges/mutual-determinations alternatively connected together; for the last node, it connects with two positively frozen nodes which have the same additional mark $2N - 1$, and by the case E the last node forms mutual-determination with node $2N - 1$ and the rest are released. The whole process is schematically shown in the lower subgraphs $(a - d)$ in Fig. 7 by a typical graph C_4. Thus, the solution space of C_{2N} can be obtained, and it is easy to verify that the result is strict by our algorithm.

VI. ANALYSIS OF MUTUAL-DETERMINATION AND BACKBONE EVOLUTION ALGORITHM FOR VERTEX-COVER

The Mutual-determination and Backbone Evolution Algorithm aims to obtain the whole solution space, and it is easy to find that it is an algorithm of polynomial time. As the Vertex-cover problem is a typical NP-complete problem, this algorithm cannot be a complete one and will lose its efficiency

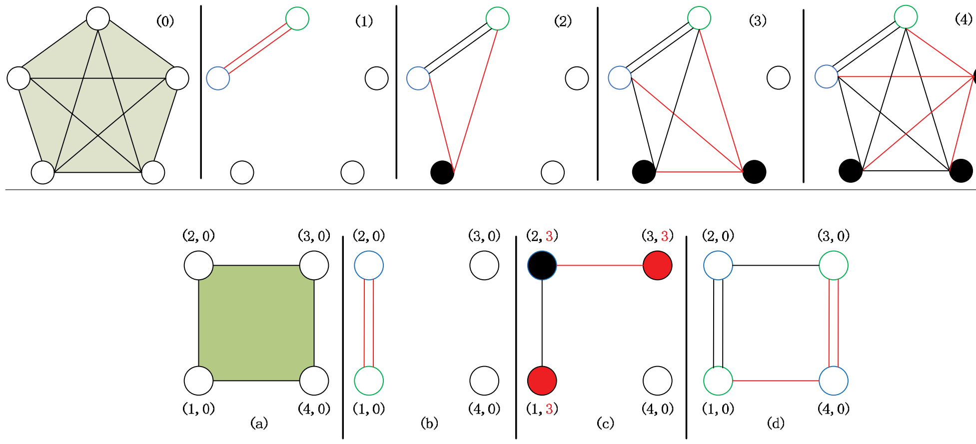

FIG. 7. (Color online) The top subgraphs provide the process of obtaining the reduced solution graph of complete graphs, which is shown by the complete graph with five nodes in this figure. The bottom subgraphs provide the process of obtaining the reduced solution graph of even cycles.

in some case. In this section, some detailed analysis on the algorithm will be provided.

A. The time complexity of Mutual-Determination and Backbone Evolution Algorithm

By the algorithm in the above section, the process of determining the ranks of the nodes in the graph is intrinsically a leaf removal process, and it will cost at most $O(N)$ steps to obtain the whole ranks of all the nodes.

By considering the nodes sequentially according to their ranks, when adding a new node to the original graph, first we should consider its local environment, which will cost at most constant C steps for random graph. Then, in different cases A–E, there may be additional time cost. In case B and D, the current state of the new added node is only determined by its neighbors and it causes no influence to others; In case C, the freezing of the current node will cause an influence to the unfrozen neighbors and those related to them, and thus the influence propagation will cost at most $O(N)$ steps. In case D, the releasing operation with the checking technique will cost at most $O(N)$ steps. At last, the rechecking technique and odd cycle breaking will cost at most $O(N)$ steps for changing the states of some nodes. In sum, when a new node is added to the graph, there are at most $C + O(N) + O(N) = O(N)$ steps to obtain the new reduced solution graph.

Besides, by the node ranks, there are N nodes to be added in total. Therefore, the total time cost for the Mutual-determination and Backbone Evolution Algorithm is at most $O(N) + N * O(N) = O(N^2)$ steps for random graphs.

B. The strictness of Mutual-Determination and Backbone Evolution Algorithm

In this subsection, we will discuss the strictness of the Mutual-determination and Backbone Evolution Algorithm. By the analysis in the above sections, Vertex-cover can be solved in polynomial time by assigning the pendants $+1$ and their petioles -1 for all leaves at different leaf-removal stages when there is no leaf removal core. Indeed, if all the nodes can be assigned ranks by the leaf-removal, i.e., the leaf-removal core is null, the reduced solution graph can reveal the whole solution space of Vertex-cover strictly, and our algorithm is a complete one to obtain the whole solution space in this case. The proof is given in the following.

Proof: For each pair of leaf, they form mutual-determination or both are backbones with one positive and the other negative. Our algorithm is intrinsically an evolution process for the two kinds of states of leaves.

When the reduced solution graph is with unfrozen-node structures of trees or forests for each step of the algorithm, this evolution guarantees that each step of adding a leaf will obtain the whole solution space of the enlarged graph. The strictness of operation in case B is trivial. Mainly by the case A and C, the releasing operation and freezing influence alternatively change the states on the trees or forests and have no cross-influence among different branches, which leads to the strictness of our algorithm.

When the reduced solution graph is with unfrozen-node structures of odd cycles of alternatively connected mutual-determinations in some steps, the odd cycles breaking technique ensures the correctness of the algorithm and the resulted reduced solution graph can be reduced to the case of unfrozen-node structures of trees or forests above.

The reduced solution graph can never have even cycles with alternative existing mutual-determinations when there is no leaf-removal core. In the right subgraph of Fig. 8, a schematic view for the leaf removal is provided. All the nodes in the red circles will be removed in pairs by leaf-removal process, and all nodes $\{a,b,c,d,e,f,g,h\}$ can have only their own leaf partners on their own cycle, which means that there are no new leaves after the nodes in the red circles are removed and the even cycle formed by $\{a,b,c,d,e,f,g,h\}$ survives at last. Evidently, these even cycles will be in the leaf-removal core.

Cycles of unfrozen nodes without alternative existing mutual-determinations can survive on the reduced solution

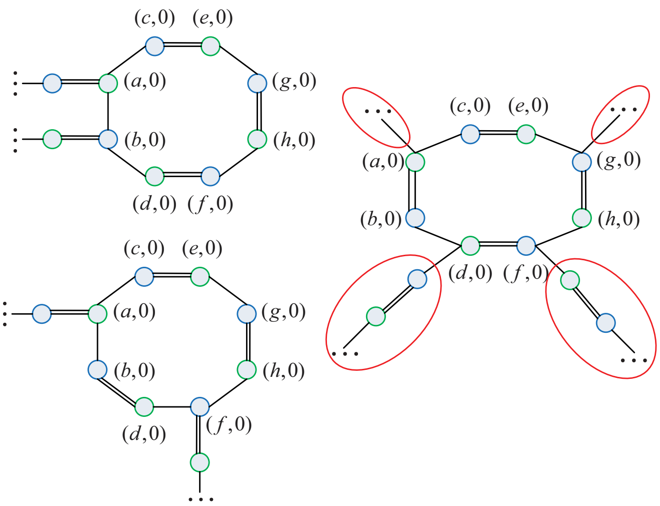

FIG. 8. (Color online) Then existence of cycles with nonalternatively mutual-determinations and the nonexistence of the even cycles with alternatively mutual-determinations when leaves are removed.

graph. In the left above and below subgraphs of Fig. 8, two simple examples are given to reveal the existence of ordinary cycles of unfrozen nodes on the reduced solution graph. By simple logic, we can find that each node on the graphs can have both covered and uncovered states. In this situation, the strictness of our algorithm is guaranteed by the checking and rechecking techniques and the case E, which ensure that the influence of the freezing operation and releasing operation can be controlled in a correct way.

At last, considering case D, when the leaf-removal core is null, this case can be reduced to that of odd cycles breaking, and it will bring the kernel difficulty when the leaf-removal core exists.

Therefore, the reduced solution graph obtained by our algorithm can reveals the exact solution space when there is no leaf-removal core. ■

By the above analysis, we have shown that the Mutual-determination and Backbone Evolution Algorithm is strict when there is no leaf-removal core in the graph. By the results in Ref. [29], there is no leaf-removal core in the random graph with high probability when the average degree c is less than e. Then, our algorithm is strict with high probability when $c < e$.

C. Cycles in the reduced solution graph

The even and odd cycles of unfrozen nodes will be analyzed in this subsection. In the right subgraph of Fig. 8, the nodes $\{a,b,c,d,e,f,g,h\}$ with four mutual-determination $(a,b),(c,e),(d,f),(g,h)$ construct an even cycle of unfrozen nodes. Indeed on this cycle all the nodes have a mutual-determination relation, that is to say, that any node is covered or uncovered will lead to the fixation of all the other seven nodes. Then, the double edges can also be drawn on $(a,c),(b,d),(f,h),(e,g)$ or all the edges, all these expressions on the reduced solution graph correspond to the same solution space, and there are only two solutions on the even cycles of alternative mutual-determinations.

For the odd cycles of alternative mutual-determinations, e.g., subgraph (7) of Fig. 2, what we can do is to perform the odd cycles breaking, which keeps the strictness of our algorithm. Unfortunately, there is the other way to produce an odd cycle of alternative mutual-determinations, just like the Incompatible Cycle in the below subgraph of Fig. 3. This kind of odd cycle structure emerges when the leaf-removal core exists and is hard to be broken for the lowest energy configuration of Vertex-cover. As the incompatible cycle brings obstacle for obtaining the real reduced solution graph, changing any unfrozen node on it to be negatively frozen is a possible choice for the reduced solution graph. In our algorithm, we can choose only one way to proceed, which makes the solution space collapse to a subspace. Many steps of the collapsing may lead to unnecessary energy increase and superfluous cover of the graph.

In fact, we can keep all the incompatible cycles of alternative mutual-determinations without breaking choices for each step in the leaf-removal core and deal with them for the final reduced solution graph. All the backbones have no influence on the solution space, but breaking the incompatible cycles of alternative mutual-determinations on the reduced solution graph is an urgent task for achieving the proper solution subspaces. Many these incompatible cycles are coupled together and should be broken by making some nodes on them negatively frozen. The fewer the number of negatively frozen nodes are chosen, the better covers we can obtain. Therefore, this problem can be reduced to the MAX-CUT [30] problem for the unfrozen nodes of the reduced solution graph. By the results of MAX-CUT, breaking the edges of unfrozen nodes which do not belong to the max-cut will lead to totaly compatible cycles. However, the MAX-CUT problem is also a NP-complete problem which is hard to solve.

VII. CONCLUSION AND DISCUSSION

A new solution space structure, mutual-determination between unfrozen nodes, is defined and discovered in the detailed case of the Vertex-cover. Based on the mutual-determinations and backbones, we construct the reduced solution graph to reveal the solution space of Vertex-cover. And, inspired by the leaf-removal process and introducing node ranks, a dynamical process for the evolution of the node states is studied to achieve the current states of nodes in the reduced solution graph. Combing the mutual-determinations, backbones, and the node ranks, an algorithm named the Mutual-determination and Backbone Evolution Algorithm is proposed to obtained the accurate reduced solution graph. To ensure the accuracy of the algorithm, the releasing operations, checking and rechecking techniques, and odd cycles breaking operation are defined by considering the influence propagation. Then, the numerical results and some examples are given to verify the validity of the algorithm. Besides, we have proved that this algorithm is an $O(N^2)$ algorithm and performs strict when there is no leaf-removal core for the graph. The influence of incompatible cycles of unfrozen nodes to the algorithm is given, which can be reduced to the MAX-CUT problem.

The Mutual-determination and Backbone Evolution Algorithm can be applied to a wide range of graphs. Though the difficulties are brought to by the incompatible cycles in case D on the reduced solution graph, choosing proper strategies to break the cycles will be helpful to obtain a solution subspace, which will be beneficial to solve the Vertex-cover problem in different topologies. Besides, in order to break the incompatible cycles of unfrozen nodes in case D on the reduced solution graph, we should design better heuristic strategies to check the key unfrozen nodes on it, such as taking advantage of the centrality or clustering coefficient [33,34]. However, as the intrinsic character of Vertex-cover is NP-complete, the Mutual-determination and Backbone Evolution Algorithm will still be an approximated one, and what we aim at is to improve the accuracy of solving different graphs of Vertex-cover.

The reduced solution graph of Mutual-determination and Backbone Evolution Algorithm can correspond to the whole solution space of Vertex-cover in some cases, which is a great help to count the number of solutions. Similar to #CSP [31,32], #Vertex-cover can be analyzed based on the reduced solution graph. However, calculating the exact entropy of the solution space needs a much detailed analysis of the constructions of the reduced solution graph, and there should be many techniques to be introduce on counting the solutions on the reduced solution graph. Besides, the reduced solution graph can help explicitly determine the role of every node and calculate the partition functions and marginal probabilities of the nodes/variables. Some of the related results will proceed in our future work.

The principle of our algorithm is related to the replica symmetry theory but not restricted to it. Most recent algorithms solving combinatorial optimization problems always concentrate on finding one solution, such as the searching algorithms and some heuristic algorithms, which assign values to the nodes/variables according to some strategies and do backtracking to reach the optimal solution. The Belief/Survey Propagation algorithms [15,21] provide marginal distributions and via additional heuristic procedures (decimation, reinforcement) to construct a single solution, but we can hardly have a visible and global understanding on the detailed underlying structures of the solution space. The Mutual-determination and Backbone Evolution Algorithm collects as more solutions as possible for the initial subgraphs, and aims to find solutions by contracting the solution space. At least, algorithms of detecting the solution space provide a strategy of reducing the complexity of finding solutions, and combing our algorithm with other searching and heuristic algorithm may be an interesting research direction for accelerating the solving process.

ACKNOWLEDGMENTS

This work is supported by the Fundamental Research Funds for the Central Universities.

[1] R. M. Karp, in *Proc. Sympos. IBM Thomas J. Watson Res. Center* (Plenum, New York, 1972), pp. 85–103.

[2] S. A. Cook, in *Proc. 3rd Ann. ACM Symp. on Theory of Computing* (Association for Computing Machinery, New York, 1971), pp. 151–158.

[3] C. H. Papadimitriou, *Computational Complexity* (John Wiley and Sons, Chichester, 2003).

[4] J. Gomez-Gardenes, P. Echenique, and Y. Moreno, Eur. Phys. J. B **49**, 259 (2006).

[5] Y. Breitbart, C. Y. Chan, M. Garofalakis, R. Rastogi, and A. Silverschatz, in Proc. IEEE INFOCOM (IEEE Communication Society, Anchorage, Alaska, 2001), pp. 933–942.

[6] B. Bollobàs, *Random Graphs* (Academic Press, New York, 1985).

[7] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000).

[8] M. Weigt and A. K. Hartmann, Phys. Rev. E **63**, 056127 (2001).

[9] M. Mézard, G. Parisi and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).

[10] S. Mertens, M. Mézard, and R. Zecchina, Random. Struc. Algor. **28**, 340 (2005).

[11] W. Barthel and A. K. Hartmann, Phys. Rev. E **70**, 066120 (2004).

[12] W. Wei, B. Guo, and Z. Zheng, J. Stat. Mech.: Theor. Exp. (2009) P02010.

[13] B. Guo, W. Wei, Y. Sun, and Z. Zheng, Phys. Rev. E **81**, 031122 (2010).

[14] V. Kumar, AI Magazine **13**(1), 32 (1992).

[15] M. Guidetti and A. P. Young, arXiv:1102.5152v1.

[16] P. Kilby, J. Slaney, S. Thiebaux, and T. Walsh, in *Proc. of the 20th Ntl. Conf. on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conf.* (AAAI/MIT Press, Menlo Park, 2005).

[17] H. Zhou, New J. Phys. **7**, 1 (2005).

[18] H. Zhou, Phys. Rev. Lett. **94**, 217203 (2005).

[19] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborova, Proc. Natl. Acad. Sci. USA **104**, 10318 (2007).

[20] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).

[21] A. Braunstein, M. Mézard, and R. Zecchina, Random. Struct. Algor. **27**(2), 201 (2005).

[22] M. Weigt and H. Zhou, Phys. Rev. E **74**, 046110 (2006).

[23] M. Mézard, T. Mora, and R. Zecchina, Phys. Rev. Lett. **94**, 197205 (2005).

[24] D. Achlioptas, A. Coja-Oghlan, and F. Ricci-Tersenghi, Rand. Struct. Algor. **38**(3), 251 (2011).

[25] W. Wei, B. Guo, and Z. Zheng, in *Proc. of the Intl. Conf. on Internet Technology and Security* (American Society of Mechanical Engineers, Wuhan, China, 2010), pp. 169–173.

[26] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, Rand. Struct. Algor. **15**, 414 (1999).

[27] D. Achlioptas, R. D'Souza, and J. Spencer, Science **323**, 1453 (2009).

[28] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, Phys. Rev. E **64**, 025101 (2001).

[29] M. Bauer and O. Golinelli, Eur. Phys. J. B **24**, 339 (2001).

[30] M. X. Goemans and D. P. Williamson, J. ACM **42**, 1115 (1995).

[31] A. A. Bulatov, Lect. Notes Comput. Sci. **5125**, 646 (2008).

[32] J. Cai and V. Choudhary, Theor. Comp. Sci. **384**, **1**, 22 (2007).

[33] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, Nature (London) **453**, 479 (2008).

[34] J. Reichardt and S. Bornholdt, Phys. Rev. E **74**, 016110 (2006).