

# A Generating Algorithm of the Complete Cover of a Compatible Relation on Nonempty Finite Sets

Kaiyou YUAN<sup>1</sup>, Heng LI and Feng TAN

*Software Department, Chongqing Institute of Engineering, Chongqing, China*

**Abstract.** Searching for the complete cover of a compatible relation on non-empty finite sets plays an important role in data mining and management decision-making. In this paper, the complete cover theory of a compatible relation is discussed, and the relevant theorem of merging compatible classes into complete cover is put forward and proved, and a generating algorithm of the complete cover of a compatible relation is proposed, and apply it to find the optimal solution for resource scheduling and allocating and so on. The application results show that the algorithm has the close efficiency to dynamic programming method, and the practical application is simple, feasible and steady.

**Keywords.** Compatible relation, maximal compatible class, complete cover, algorithm.

## 1. Introduction

Set theory is not only an important part of discrete mathematics, but also the basis of modern mathematics. Compatible relations is an important binary relation between set elements. Its only complete cover on set plays an important role in the fields of computer theory, data mining and management decision-making [1, 2]. At present, researches on the complete cover on nonempty finite sets mainly focused on relational theory, complete cover algorithm, maximal clique mining and clustering algorithm.

Feng Shuxia et al. gave the definitions of compatible complete cover, maximal compatible complete cover, minimal compatible complete cover and appropriate complete cover, and obtained the relationship between them [3]. Using the inclusion exclusion theorem and set theory, Chen Renrong gave the relationship between partition and equivalence, complete cover and compatible relations on finite sets, as well as the calculation formulas of partition fraction and complete cover number on finite sets [4]. Although the above theoretical research involved the relationship between compatible relations, minimal compatible complete cover and maximal compatible complete cover, there is insufficient discussion on the atomic cover of a compatible relation and the transformation between atomic cover and complete cover. On the basis of the original method of finding the complete cover on a set by using a compatible relation matrix, Tian Hemin improved the comparison and consolidation of the existing compatible

---

<sup>1</sup> Corresponding Author, Kaiyou YUAN, Software Department, Chongqing Institute of Engineering, Chongqing, China.; E-mail: 314104670@qq.com.

classes, and obtained a new method of constructing the complete cover by using the simplified compatible relation matrix [5]. However, neither the original matrix method nor the improved matrix method gave the theoretical basis and specific algorithm of merging. Chen Xing gave a theorem about the path length of the maximal complete subgraph by using the corresponding relationship between the maximal compatible class of a compatible relation and the maximal complete subgraph of a simple graph, and constructed an algorithm to search the maximal complete subgraph of a simple graph by using the path length of a compatible relation matrix corresponding graph, so as to find out the complete cover of the compatible relation [6]. Yin Jianhong used graph theory to optimize the breadth first search of the simplified undirected graph corresponding to a compatible relation, and obtained the set cluster (the complete cover) of the maximal compatible classes [7]. Tang Xiaochun et al. proposed a parallel mining algorithm for searching all maximal cliques in large graphs by improving the filtering and detection of repeated maximal cliques and non maximal cliques [8]. Michael Svendsen et al. designed the sorting parallel enumeration algorithm of maximum cliques under the MapReduce framework to search the maximum subgroups in the large graph by means of load balancing and enumerating non repeated maximum cliques [9]. According to the dynamic characteristics of the actual information system, Zhang Yingying proposed an algorithm for incrementally updating the maximal compatible classes based on the maximal cliques Bron-Kerbosch algorithm in graph theory, which was compared and verified on multiple data sets [10]. Ren Xia Wan et al. proposed an clustering method based on a compatible relation, which can effectively compress the search space. It is an improvement and supplement to the existing compatible relation clustering algorithm [11]. Dai T. put forward a clustering algorithm based on a compatible relation on object set, and made an empirical analysis on the evaluation of foreign trade sustainable development in 15 regions, which verified the feasibility and practicability of this method [12]. The above algorithms for solving complete subgraphs or maximal compatible clusters based on graphs have the defects of large backtracking scale and relatively high time complexity.

Therefore, it is necessary to discuss the relevant theory of maximal compatible classes of a compatible relation on nonempty finite sets, and design a practical algorithm to generate the complete cover on nonempty finite sets, so as to guide the solution of practical problems.

## 2. Basic Definitions and Theorems

**Definition 1.** Let  $R$  be a binary relation on a given nonempty finite set  $A$ . if  $R$  is reflexive and symmetric,  $R$  is called a compatible relation on set  $A$  [1, 2].

**Definition 2.** Let  $R$  be a compatible relation on a nonempty finite set  $A$ , and  $B$  be a subset of  $A$ . if there is  $xRy$  for any element  $x, y \in B$ ,  $B$  is called a compatible class generated from the compatible relation  $R$  [1, 2].

**Definition 3.** Let  $R$  be a compatible relation on a nonempty finite set  $A$ . For any element  $x, y \in A$ , and  $x \neq y$ , there is  $xRy$ , so the set  $\{x, y\}$  is called an atomic compatible class of  $R$ ; For any element  $z$  with only autocorrelation, and  $z \in A$ , the set  $\{z\}$  is called the atomic compatible class of  $R$ .

**Definition 4.** If  $R$  is a compatible relation on a nonempty finite set  $A$ , then the set composed of all atomic compatible classes generated from  $R$  is called the atomic cover on  $A$ . Atomic cover is called *level-1* cover.

**Definition 5.** Let  $R$  be a compatible relation on a nonempty finite set  $A$  and  $C$  be a compatible class generated from  $R$ . if any element in  $A-C$  cannot have a compatible relation with all elements in  $C$ ,  $C$  is called the maximal compatible class (maximal compatible block) of  $R$  [1, 2].

**Definition 6.** If  $R$  is a compatible relation on a nonempty finite set  $A$ , then the set composed of all maximal compatible classes generated from  $R$  is called the complete cover on  $A$  [1, 2].

**Theorem 1.** If  $R$  is a compatible relation on a nonempty finite set  $A$  and  $b_i$  ( $1 \leq i \leq m$ ) is an atomic compatible class of relation  $R$  on  $A$ , then the atomic cover  $B = \{b_1, b_2, \dots, b_m\}$  is the only cover on  $A$ .

**Proof:**  $b_i$  ( $1 \leq i \leq m$ ) is an atomic compatible class generated from any two unequal elements or any autocorrelation only element on set  $A$  according to relation  $R$ , and any atomic compatible class  $b_i \subseteq A$  ( $1 \leq i \leq m$ ). So  $b_1 \cup b_2 \cup \dots \cup b_m \subseteq A$ .

In addition, for any  $a \in A$ , there must be an atomic compatible class  $b_i$  formed by  $a$ , so that  $a \in b_i$ . So  $a \subseteq b_1 \cup b_2 \cup \dots \cup b_m$ , that is,  $A \subseteq b_1 \cup b_2 \cup \dots \cup b_m$ .

Therefore, the atomic cover  $B$  on  $A$  is a cover on  $A$ .

Let  $c_i$  ( $1 \leq i \leq n$ ) be an atomic compatible class of relation  $R$  on set  $A$ , and the set composed of all atomic compatible classes on  $A$  is another cover on  $A$ . For  $c_i \in \{b_1, b_2, \dots, b_m\}$ , there is  $b_k$  ( $1 \leq k \leq m$ ), so that  $c_i = b_k$ , otherwise  $c_i \neq b_k$ , that is,  $\{b_1, b_2, \dots, b_m, c_i\}$  is a cover on  $A$ . This contradicted that  $\{b_1, b_2, \dots, b_m\}$  is all atomic compatible classes of relation  $R$  on  $A$ . So  $\{c_1, c_2, \dots, c_n\} \subseteq \{b_1, b_2, \dots, b_m\}$ ,  $n \leq m$ . Similarly,  $\{b_1, b_2, \dots, b_m\} \subseteq \{c_1, c_2, \dots, c_n\}$ ,  $m \leq n$ . Therefore  $m = n$ , the set  $\{b_1, b_2, \dots, b_m\}$  and  $\{c_1, c_2, \dots, c_n\}$  are the same cover on  $A$ . Q.E.D.

**Theorem 2.** If  $R$  is a compatible relation on a nonempty finite set  $A$ ,  $B$  and  $C$  are any two compatible classes generated from  $R$ , and  $|B-C| = |C-B| = 1$  ( $|B|$  is the cardinality of set  $B$ ), and there is an  $R$  relation on the symmetric difference between  $B$  and  $C$ , then  $B \cup C$  must be a larger compatible class.

**Proof:** Let  $B = \{b_1, b_2, \dots, b_n\}$ ,  $C = \{c_1, c_2, \dots, c_m\}$ . Since  $|B-C| = |C-B| = 1$ , then  $|B| = |C|$ , that is,  $n = m$ ,  $B$  and  $C$  have  $n-1$  elements in common.

Without losing generality, assuming that the first  $n-1$  elements in  $B$  and  $C$  are equal, that is,  $b_1 = c_1, b_2 = c_2, \dots, b_{n-1} = c_{n-1}$ , then  $b_n$  is the element of  $B-C$  and  $c_n$  is the element of  $C-B$ . According to the  $R$  relation on the symmetric difference between  $B$  and  $C$ , there is  $b_n R c_n$ .

Since  $B$  and  $C$  are any compatible classes generated from  $R$ , there is an  $R$  relation between any two elements in  $B$  or  $C$ , and there is an  $R$  relation between  $c_n$  and the first  $n-1$  elements in  $C$ . Since the first  $n-1$  elements in  $B$  and  $C$  are equal, there is an  $R$  relation between  $c_n$  and the first  $n-1$  elements in  $B$ . And because  $b_n R c_n$ , there is an  $R$  relation between  $c_n$  and all elements in  $B$ . Therefore, all elements in  $B \cup C = \{b_1, b_2, \dots, b_n, c_n\}$  have an  $R$  relation each other, which is a larger compatible class. Q.E.D.

**Theorem 3.** If  $R$  is a compatible relation on a nonempty finite set  $A$ , then the unique atomic cover and the unique complete cover generated from  $R$  can be transformed into each other.

**Proof:** According to theorem 1, the atomic cover  $B$  (*level-1* cover) generated from the compatible relation  $R$  on a nonempty finite set  $A$  is unique.

According to theorem 2, any two elements ( $b_i, b_j$  are the compatible classes generated from  $R$ ) in the *level-1* cover  $B = \{b_1, b_2, \dots, b_n\}$  that meet the condition ( $|b_i - b_j| = |b_j - b_i| = 1$ ), and there is an  $R$  relation on the symmetric difference between  $B$  and  $C$  are merged to obtain a larger compatible class to store into the set  $C$ , and the non merged elements in the *level-1* cover are transferred to  $C$ , which is composed of a larger compatible classes (*level-2* cover). *Level-2* cover  $C$  is unique.

Iteratively and repeatedly apply theorem 2 to merge the elements that meet the conditions of theorem 2 in *level-i* cover into set  $C$ , and transfer the non merged elements in *level-i* cover into  $C$  to form *level-i+1* cover. *Level-i+1* cover is unique.

Repeat merging and transferring until all elements in  $C$  cannot be merged (all elements in  $C$  are maximal compatible classes), then  $C$  is the complete cover on  $A$ . The complete cover  $C$  is unique.

Therefore, the atomic cover can be transformed into the complete cover, and the complete cover generated from the compatible relation  $R$  is unique.

The proof of the complete cover conversion to the atomic cover is omitted. Q.E.D.

### 3. Set Merging Algorithm

Based on the above theory, this paper proposed a generating algorithm (set merging method) of the complete cover of a compatible relation  $R$  on nonempty finite set  $A$ .

Basic idea of set merging method:

(1) The atomic cover  $B$  (*level-1* cover) is obtained from the compatible relation  $R$  on the nonempty finite set  $A$ .

(2) Applying theorem 2 and theorem 3, the elements in *level-i* cover are merged and transferred to  $C$  to form *level-i+1* cover ( $i = 1, 2, 3, \dots$ ).

(3) Repeat step (2) until all elements in  $C$  cannot be merged (all elements in  $C$  are maximal compatible classes), then  $C$  is the complete cover on  $A$ .

Steps of set merging algorithm:

Objective: generation of the complete cover  $stC$  on set  $A$

Input: atomic cover  $stB$  on set  $A$

Output: the complete cover  $stC$  on set  $A$

Process: CompleteCover (atomic cover  $stB$  on set  $A$ )

Copy atomic cover  $stB$  into set  $stC$ .

Initialize the  $mgFlag$  of the corresponding element merging tag list in  $stC$ , and the initial values are 0.

Define an empty list  $stNew$ , which is used to store the new merged compatible classes.

Define the merging operation flag  $bMerge$  (initial value is true), which was used to determine whether to perform the next round merging.

while  $bMerge$ :

$bMerge = \text{False}$

    for  $i$  in range( $\text{len}(stC)-1$ ):

        for  $j$  in range( $i+1, \text{len}(stC)$ ):

            if ( $|\text{stC}[i] - \text{stC}[j]| = |\text{stC}[j] - \text{stC}[i]| = 1$  and

            (Symmetric difference of  $\text{stC}[i]$  and  $\text{stC}[j] \in stB$ ):

                Save  $\text{stC}[i] \cup \text{stC}[j]$  into  $stNew$

                Corresponding merging flag  $mgFlag[i] = mgFlag[j] = 1$

The next round of merging operation flag  $bMerge = True$   
 for  $k$  in range(len(mgFlag)): #traverse the index of  $mgFlag$ .  
   if  $mgFlag[k] == 0$ :  
     Transfer the element corresponding to index  $k$  in  $stC$  to  $stNew$ .  
     Copy the higher level cover  $stNew$  to  $stC$ .  
     Reset the merging flag  $mgFlag$  and  $stNew$ , and proceed to the next iteration.  
 Return  $stC$

## 4. Application Case

The complete cover on sets can be used to find the optimal solution of resource scheduling and allocation problems, such as, how to schedule or allocate multiple meetings scientifically, reasonably and orderly, how to assign or arrange different personnel to participate in a variety of activities, and so on.

### 4.1. Resource Scheduling Problem

There are 8 meetings to be held one morning, and the specific meeting time is shown in table 1.

**Table. 1** Meetings Held in One Morning and its Schedules

Meetings	Meeting1	Meeting2	Meeting3	Meeting4	Meeting5	Meeting6	Meeting7	Meeting8
Times	[8:00-9:00)	[8:30-9:40)	[8:40-11:50)	[9:00-10:00)	[9:00-12:00)	[9:10-10:40)	[10:00-11:00)	[11:10-12:30)

For the above meetings:

(1) How many kinds of meetings can be held in the same conference room that morning?

(2) How many meetings can be held at most in only one conference room available?

The relation  $R$  on the above meeting time set is: there is no conflict among meeting times. Obviously, any meeting  $i$  does not conflict with its own time, so  $R$  is reflexive. At the same time, if any meeting  $i$  and meeting  $j$  have no time conflict, there must be no time conflict between meeting  $j$  and meeting  $i$ , so  $R$  is symmetrical. Therefore,  $R$  is a compatible relation.

(1) According to the compatible relation  $R$  on the conference set  $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$ , the atomic cover (level-1 cover)  $B = \{\{1, 4\}, \{1, 5\}, \{1, 6\}, \{1, 7\}, \{1, 8\}, \{2, 7\}, \{3\}, \{4, 7\}, \{4, 8\}, \{6, 8\}, \{7, 8\}\}$  can be obtained. Copy  $B$  into  $C$ , and the complete cover on set  $A$  can be calculated by the set merging method.

(2) Compare the elements in  $C$  (level-1 cover) according to the conditions in theorem 2 to complete the first round of merging and transfer:

The symmetric difference between  $\{1, 4\}$  and  $\{1, 5\}$  is  $\{4, 5\} \notin B$ , so  $\{1, 4\}$  and  $\{1, 5\}$  cannot be merged. Similarly,  $\{1, 4\}$  and  $\{1, 6\}$  cannot be merged. The symmetric difference between  $\{1, 4\}$  and  $\{1, 7\}$  is  $\{4, 7\} \in B$ , so  $\{1, 4\}$  and  $\{1, 7\}$  are merged to  $\{1, 4, 7\}$  to store in set  $N$ . similarly,  $\{1, 4\}$  and  $\{1, 8\}$  are merged to  $\{1, 4, 8\}$  to store in set  $N$ .  $\{1, 4\}$  and  $\{2, 7\}$ ,  $\{2, 8\}$ ,  $\{3\}$  cannot be merged respectively.  $\{1, 4\}$  and  $\{4, 7\}$ ,  $\{4, 8\}$  are merged respectively to  $\{1, 4, 7\}$ ,  $\{1, 4, 8\}$ , which have been stored in  $N$  (due to the non repeatability of set elements, only one copy of the same element can be stored in  $N$ ).  $\{1, 4\}$  and  $\{6, 8\}$ ,  $\{7, 8\}$  cannot be merged respectively.  $\{1, 5\}$  and other elements behind cannot be merged.  $\{1, 6\}$  and  $\{1, 8\}$ ,  $\{6, 8\}$  are merged respectively to  $\{1, 6, 8\}$  to store in  $N$ .

$\{1, 6\}$  and other elements behind cannot be merged.  $\{1, 7\}$  and  $\{1, 8\}$ ,  $\{7, 8\}$  are merged respectively to  $\{1, 7, 8\}$  to store in  $N$ .  $\{1, 7\}$  and  $\{4, 7\}$  are merged to  $\{1, 4, 7\}$  which have been stored in  $N$ .  $\{1, 7\}$  and other elements behind cannot be merged.  $\{1, 8\}$  and  $\{4, 8\}$ ,  $\{6, 8\}$ ,  $\{7, 8\}$  are merged respectively to  $\{1, 4, 8\}$ ,  $\{1, 6, 8\}$ ,  $\{1, 7, 8\}$  which have been stored in  $N$ .  $\{1, 8\}$  and other elements behind cannot be merged.  $\{2, 7\}$  and  $\{2, 8\}$ ,  $\{7, 8\}$  are merged respectively to  $\{2, 7, 8\}$  to store in  $N$ .  $\{2, 7\}$  and other elements behind cannot be merged.  $\{3\}$  and other elements behind cannot be merged.  $\{4, 7\}$  and  $\{4, 8\}$ ,  $\{7, 8\}$  are merged respectively to  $\{4, 7, 8\}$  to store in  $N$ .  $\{4, 7\}$  and  $\{6, 8\}$  cannot be merged.  $\{4, 8\}$  and  $\{6, 8\}$  cannot be merged.  $\{4, 8\}$  and  $\{7, 8\}$  are merged to  $\{4, 7, 8\}$  which have been stored in  $N$ .  $\{6, 8\}$  and  $\{7, 8\}$  cannot be merged. At this time,  $\{1, 5\}$  and  $\{3\}$  remained in *level-1* cover  $C$  are not involved in merging, and transfer them to  $N$ , so  $N = \{\{1, 4, 7\}, \{1, 4, 8\}, \{1, 6, 8\}, \{1, 7, 8\}, \{2, 7, 8\}, \{4, 7, 8\}, \{1, 5\}, \{3\}\}$ , which is the *level-2* cover. Copy  $N$  to  $C$  for the next round of merging and transfer.

(3) Compare the elements in  $C$  (*level-2* cover) according to the conditions in theorem 2 to complete the second round of merging and transfer:

The symmetric difference between  $\{1, 4, 7\}$  and  $\{1, 4, 8\}$  is  $\{7, 8\} \in B$ , so  $\{1, 4, 7\}$  and  $\{1, 4, 8\}$  are merged to  $\{1, 4, 7, 8\}$  to store in set  $N$ . similarly,  $\{1, 4, 7\}$  and  $\{1, 7, 8\}$ ,  $\{4, 7, 8\}$  are merged respectively to  $\{1, 4, 7, 8\}$  which have been stored in  $N$ .  $\{1, 4, 7\}$  and other elements behind cannot be merged.  $\{1, 4, 8\}$  and  $\{1, 7, 8\}$ ,  $\{4, 7, 8\}$  are merged respectively to  $\{1, 4, 7, 8\}$  which have been stored in  $N$ .  $\{1, 4, 8\}$  and other elements behind cannot be merged.  $\{1, 6, 8\}$  and other elements behind cannot be merged.  $\{1, 7, 8\}$  and  $\{4, 7, 8\}$  are merged to  $\{1, 4, 7, 8\}$  which have been stored in  $N$ .  $\{1, 7, 8\}$  and other elements behind cannot be merged.  $\{2, 7, 8\}$  and other elements behind cannot be merged.  $\{4, 7, 8\}$  and other elements behind cannot be merged.  $\{1, 5\}$  and other elements behind cannot be merged. At this time,  $\{1, 6, 8\}$ ,  $\{2, 7, 8\}$ ,  $\{1, 5\}$  and  $\{3\}$  remained in *level-2* cover  $C$  are not involved in merging, and transfer them to  $N$ , so  $N = \{\{1, 4, 7, 8\}, \{1, 6, 8\}, \{2, 7, 8\}, \{1, 5\}, \{3\}\}$ , which is the *level-3* cover. Copy  $N$  to  $C$  for the next round of merging and transfer.

(4) Compare the elements in  $C$  (*level-3* cover) according to the conditions in theorem 2 to complete the third round of merging and transfer:

At this time, the elements of  $C$  in pairwise comparison cannot be merged. *Level-3* cover  $C$  is returned and the algorithm ends.

The complete cover  $C$  has five maximal compatible classes, so there are five types of meetings. Meetings in each maximal compatible class can be held in the same conference room that morning. When there is only one conference room available, the first maximal compatible class  $\{1, 4, 7, 8\}$  in  $C$  with up to 4 meetings is the best select.

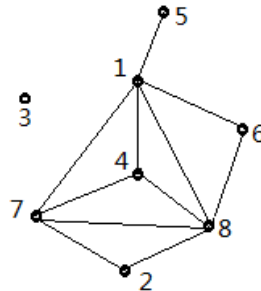


Figure 1. Simplified Diagram of a Compatible Relation on Meeting Set

According to the relation  $R$  of 8 conferences set  $A$  in the above example, a simplified relation graph of compatible relations can be drawn (as shown in figure 1). Using the maximal complete polygon method based on direct observation (which is difficult to be applied to simplified graphs with complex relation), it can be concluded that  $\{\{1,4,7,8\}, \{1,6,8\}, \{2,7,8\}, \{1,5\}, \{3\}\}$  is the complete cover of conferences set  $A$ , which can be mutually confirmed with the results obtained by set merging method.

#### 4.2. Finding the Optimal Solution Problem

Adding value weight to meetings in Section 4.1 makes it a problem of finding the optimal solution. The specific meeting weights are shown in table 2.

**Table 2.** Meetings, Schedules and Value Weights

Meetings	Meeting1	Meeting2	Meeting3	Meeting4	Meeting5	Meeting6	Meeting7	Meeting8
Times	[8:00-9:00)	[8:30-9:40)	[8:40-11:50)	[9:00-10:00)	[9:00-12:00)	[9:10-10:40)	[10:00-11:00)	[11:10-12:30)
Weights	1	4	5	2	2	10	7	3

In view of the above information, what is the maximum value of meetings that can be held that morning? What meetings were held?

By analyzing the meeting information, it is found that the maximum value of meetings that can be held that morning must appear in the maximum compatible class on meeting set  $A$ . The counter proof proves that if  $\exists C$  is a non maximal compatible

class in which the sum of the meetings values  $\sum_{i=0}^n V(a_i)$  is the maximum value, there

must be a maximal compatible class  $C' \supset C$ , so that the sum of the meetings values in

$C'$  is  $\sum_{i=0}^{n'} V(a_i) = \sum_{i=0}^n V(a_i) + \sum_{i=n+1}^{n'} V(a_i) > \sum_{i=0}^n V(a_i), (n' > n)$ , which is contrary to that

the sum of the meetings values in  $C$  is the maximum value, so the proposition is proved. Therefore, solving the maximum value of meetings that can be held that morning is transformed into finding the maximal compatible class set on meeting set  $A$ , and then finding the maximum value in these maximal compatible classes.

According to Section 4.1, the complete cover of the meeting set  $A$  is  $\{1,4,7,8\}, \{1,6,8\}, \{2,7,8\}, \{1,5\}, \{3\}$ , then the corresponding maximum values are  $\{1+2+7+3 = 13, 1+10+3 = 14, 4+7+3 = 14, 1+2 = 3, 5\}$ , so the maximum value of meetings that can be held that morning is 14, which is in two maximal compatible classes  $\{1,6,8\}$  and  $\{2,7,8\}$ .

The maximum value solution of the above meetings can be divided into several interrelated stages according to the end time of the meetings. In each stage, you can decide which meeting to hold and the current maximum value of the meetings, so as to fully determine the meetings and its maximum value held in the final stage. It has the characteristics of multi-stage decision-making and meets the requirements of dynamic programming algorithm, so the process and results of solving the problem by dynamic programming algorithm are analyzed as follow.

According to the end time of 8 meetings, the decision-making process of the maximum value of the meeting is made in stages, as shown in table 3. First, when only the 1st meeting is held, the corresponding value is 1 on 08:00 to 09:00. There is no meeting in the subsequent time, and the maximum value remains 1. When only 1st and 2nd meetings are held, the corresponding value of the 2nd meeting is 4 from 08:30 to 09:40. There is no maximum meeting value of previous decisions before 08:30, and  $4 > 1$ , so the maximum value that can be generated when the 1st and 2nd meetings are held remains 4. By analogy, the 8th meeting is held from 11:10 to 12:30, its value is 3. And the maximum value (optimal substructure) generated by the meetings before 11:10 is 11, so the maximum value generated at this time is  $3+11 = 14 > 11$ , so the maximum value of the meeting that can be held that morning is 14, which is consistent with the solution result of the set merging method.

Table 3. Phased Decision Table on Meetings Maximum Value

Meetings	[08:00-09:00)	[08:30-09:40)	[09:00-10:00)	[09:10-10:40)	[10:00-11:00)	[08:40-11:50)	[09:00-12:00)	[11:10-12:30)
1	1	1	1	1	1	1	1	1
2	1	4	4	4	4	4	4	4
3	1	4	4	4	4	5	5	5
4	1	4	4	4	4	5	5	5
5	1	4	4	4	4	5	5	5
6	1	4	4	11	11	11	11	11
7	1	4	4	11	11	11	11	11
8	1	4	4	11	11	11	11	14

5. Discussion and Prospect

- (1) The set merging algorithm is simple and practical, avoids the scale of repeated search by exhaustive method and the complexity of recursive search of depth, breadth or path of a graph, and is suitable for solving the complete cover of compatible relation, the complete subgraphs of a graph and the optimal solution of weighted activities with compatible relation, etc.
- (2) In solving the optimal combination problem of weighted activities with compatible relationship, the set merging algorithm first obtains all the maximal compatible classes, and then finds the optimal solutions in the maximal compatible classes. It can directly obtain the multiple specific optimal schemes in polynomial time complexity. The dynamic programming algorithm can also obtain the optimal solution of a problem in polynomial time complexity, but when the problem has multiple optimal schemes, the dynamic programming algorithm either saves the current optimal scheme when calculating the optimal substructure, or traces back to the phased decision matrix to search for the optimal scheme. Both results of solving the optimal scheme involve traversal of a tree, which will affect the final computational efficiency of the algorithm.
- (3) The set merging algorithm can merge the subsets with the largest cardinality only in each round, which can improve the algorithm merging efficiency.



(4) After solving the complete cover  $C$  on set  $A$ , the set merging algorithm can further discuss how to find the minimum subset set covering  $A$  in  $C$ . It is likely to be a key problem in SCP. So a calculation model can be established according to the relevant characteristics of complete cover of a compatible relation, which provides support for the follow-up research on minimum cover of a set.

## Acknowledgments

Fund project. 2020 Chongqing science and technology project "Research on Software Test Case Design Method Based on C4.5 Decision Tree Algorithm" (No.: KJQN202001905)

## References

- [1] Wu Mingfen. Discrete Mathematics and Its Applications [M]. Beijing: Tsinghua University Press, 2020,11. In Chinese.
- [2] Richard Johnson Baugh. Discrete Mathematics (8th Edition) (English Version) [M]. Beijing: Publishing House of Electronic Industry, 2018,07.
- [3] Feng Shuxia, Wang Jie, Dong Daozhen. Compatible Relations and the Complete Cover on Finite Sets [J]. Journal of Henan University (Natural Sciences), 2011,41 (05): 441-446. In Chinese.
- [4] Chen Renrong. Partition and Cover on Finite Sets [J]. Journal of Changzhou Institute of Technology, 2006, (01): 5-8. In Chinese.
- [5] Tian Hemin, Zhu Limei. Improvement of "Finding the Complete Cover on Sets" [J]. Journal of Shenyang University of Technology, 2012,31 (01): 66-69. In Chinese.
- [6] Chen Xing, Lin Qiong, Xue Ying, et al. A method for Finding Maximal Compatible Classes Using Matrices [J]. Journal of Logistical Engineering University, 2010,26 (04): 92-96. In Chinese.
- [7] Yin Jianhong. A Generating Algorithm of Maximal Compatible Class of Compatible Relations [J]. Journal of Hefei University of Technology (Natural Sciences), 2004, (06):702-705. In Chinese.
- [8] Tang Xiaochun, Zhou Jiawen, Tian Kaifei, et al. Research on Parallel Mining Algorithm of All Maximal Cliques in Large Map [J]. Chinese Journal of Computers, 2019,42 (03): 513-531. In Chinese.
- [9] Michael Svendsen, Arko Provo Mukherjee, Srikantha Tirthapura. Mining Maximal Cliques from a Large Graph Using MapReduce: Tackling Highly Uneven Subproblem Sizes [J]. Journal of Parallel and Distributed Computing, 2015, 79(C):104-114.
- [10] Zhang Yingying, Li Tianrui. An Incremental Update Algorithm of Maximal Compatible Class in Interval Value Information System [J]. Journal of Chinese Mini-Micro Computer Systems, 2017,38 (07): 1573-1579. In Chinese.
- [11] Ren Xia Wan, Xiao Ya Yan, Cai Xia Li. Compatible Clustering Algorithm with Convex Space Partitioning [J]. Advanced Materials Research, 2012(03):147-151.
- [12] Dai T. Characteristics of International Trade Benefits Based on the Cluster Analysis of Compatibility Relation [C]. BDCPS 2019(12):28-29.