Int J Software Informatics, Volume 5, Issue 4 (2011), pp. 595–606
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

Color-Coding and its Applications: A Survey

Jianxin Wang[1], Qilong Feng[1], and Jianer Chen[1,2]

[1] (School of Information Science and Engineering, Central South University, Changsha 410083, China)

[2] (Department of Computer Science and Engineering, Texas A&M University, College Station, Texas 77843-3112, USA)

Abstract Color-Coding is an important algorithmic technique in solving many NP-hard problems. In this paper, we give a survey on Color-Coding technique and its applications. We first give brief introduction on three Color-Coding methods: random Color-Coding, Color-Coding based on perfect hash function, and Color-Coding for $n \leq 2k$. Then, applications of Color-Coding technique in various fields are presented, such as Bioinformatics, Networks, etc. Finally, we give future research topics of Color-Coding technique.
Key words: color coding; perfect hash function; k-Path problem; matching and packing problem

Wang JX, Feng QL, Chen JE. Color-Coding and its applications: A survey. *Int J Software Informatics*, **Vol.5, No.4 (2011): 595–606. http://www.ijsi.org/1673-7288/5/i113.htm**

1 Introduction

In the field of computer science, many problems can be depicted as subset selection problem, i.e., given an universal set U of size n, to find a subset $W \subseteq U$ of size k satisfying specific property R. For the subset selection problem, it is easy to get solution by enumerating all possible subsets of U with size k. Obviously, the above enumeration process is of time $O(\binom{n}{k}) = O(n^k)$, which is unpractical for many applications.

Color-Coding technique was first proposed by Alon[1], which is an efficient method dealing with subset selection problem. The general idea of Color-Coding technique is to use k colors to color the elements of U, aiming at finding a coloring such that any two elements of W are in different colors. For Color-Coding technique, the following two questions need to be answered:

(1) How many colorings are needed to guarantee that there exists a coloring making any two elements of W have different colors?

(2) How to find objective solution W based on the coloring on U?

The first question is about coloring scheme of Color-Coding. In fact, different Color-Coding methods have different coloring scheme size. Generally, coloring

This work is supported by the National Natural Science Foundation of China under Grant Nos.61073036, 61070224, 70921001, the Doctoral Discipline Foundation of Higher Education Institution of China under Grant No.20090162110056.
Corresponding author: Jianxin Wang, Email: jxwang@mail.csu.edu.cn
Received 2010-11-25; Revised 2011-03-05; Accepted 2011-05-03.

schemes of all Color-Coding methods have form of $O^*(c^k)$, i.e., for $O^*(c^k)$ colorings, there must exist a coloring f such that any two elements of W have different colors under coloring f, where c is a constant. Currently, there are three popular Color-Coding methods: random Color-Coding, Color-Coding based on perfect hash function, and Color-Coding for $n \leq 2k$. In section 3, the above three Color-Coding methods are presented in detail and several examples are given to illustrate how to use those Color-Coding methods to solve problems.

The second question is about how to use Color-Coding technique to solve problems. In fact, Color-Coding technique divides elements of U into k classes, each of which is colored by one color, such that objective solution W can be obtained in a more efficient way. Generally, Color-Coding technique is combined with dynamic programming technique to solve problems. In the literature, Color-Coding technique has been used to solve many NP-hard problems, such as k-Path problem, Subgraph Isomorphism problem, Matching and Packing problems, etc. Particularly, Color-Coding technique has been used to solve many important problems in the fields of Bioinformatics and Networks. In section 4, we give brief introduction on applications of Color-Coding technique.

2 Related Terminology

For Color-Coding technique, there are many ways to define a coloring. A coloring can be defined as a function f, i.e, $f : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, k\}$, where $k \leq n$. Moreover, a coloring can also be described as a dividing of universal set, i.e., divide the elements of universal set into k classes. In this paper, we use function to define coloring. In the following, universal set is denoted by $U = \{e_1, e_2, \cdots, e_n\}$ and color set is denoted by $C = \{c_1, c_2, \cdots, c_k\}$. A subset W of U is called a k-subset if W contains exactly k elements.

Definition.[9,26] (n, k)-coloring: Given an universal set $U = \{e_1, e_2, \cdots, e_n\}$ and a color set $C = \{c_1, c_2, \cdots, c_k\}$. A (n, k)-coloring is a function $f : U \to C$, satisfying $\bigcup_{e_i \in U} f(e_i) = C$, i.e, each element in U is colored with one color and each color in C is used at least once.

Definition.[9,26] Given a (n, k)-coloring f on U and a k-subset W of U, for any two elements $e_i, e_j \in W$ $(i \neq j)$, if $f(e_i) \neq f(e_j)$, then W is *properly colored* by f.

Definition.[9,26] (n, k)-coloring scheme: A (n, k)-coloring scheme is a set of (n, k)-colorings satisfying that for any k-subset W of U, W is properly colored by at least one (n, k)-coloring in (n, k)-coloring scheme.

For a (n, k)-coloring scheme F, the size of F is the number of colorings in F.

3 Algorithms for Constructing Coloring Scheme

The time complexity of using Color-Coding to solve problems is mainly determined by the size of coloring scheme. For the case when $k \ll n$ (k is a small parameter), there are two available methods for constructing coloring scheme: random method and method based on perfect hash function. However, in many practical applications, problem parameter is not very small. For example, for Motif Finding problem in Bioinformatics, $k = 16$, $n = 20$. Obviously, the random Color-Coding and the Color-Coding based on perfect hash function are not workable any more.

For problems with $n \leq 2k$, a Color-Coding method based on dividing is available, which can be used to solve many problems in Bioinformatics and Networks. In this section, we give detailed introduction on random Color-Coding, Color-Coding based on perfect hash function, and Color-Coding for $n \leq 2k$.

3.1 Random color-coding

The general idea of random Color-Coding is that for any element e of U, randomly choose a color from C to color e.

For any k-subset W of U, in the following, we analyze the probability that W is properly colored. For any element x of W, x can be colored by any color in C, i.e., x has k possible colors. Therefore, the total number of possible colorings for the elements of W is k^k. It is easy to see that there are $k!$ ways to color the elements of W such that any two elements of W are in different colors and each color in C must be used at least once, which is the number of permutations for the k elements of W. Therefore, for a random coloring, W is properly colored with probability $k!/k^k \approx 1/e^k$.

In order to color W with higher probability, repeat the above random coloring process e^k times. In the following, we take k-Path problem as an example to illustrate how random Color-Coding is applied to solve problems[1].

Definition.[1] k-Path: Given a graph $G = (V, E)$ and a parameter k, find a simple path in G of length k, or report that no such path exists in G.

The general idea solving k-Path problem by random Color-Coding is as follows: Color the vertices of G randomly. Then, apply dynamic programming technique to find a properly colored k-path.

Assume that G contains a k-path P. For each random coloring, it is easy to get that P is properly colored with probability $k!/k^k \approx 1/e^k$. The remaining problem is how to apply dynamic programming technique to find properly colored k-path.

In the colored graph G, add a new vertex s with assigned color 0. For each vertex v of G, add edge (s, v) to E. Denote the new graph by G'. It is easy to see that there exists a properly colored k-path in G if and only if there is a properly colored $(k+1)$-path in G'. In dynamic programming process, additional information is saved. For example, for any vertex v in G', all the possible color sets used by paths from s to v should be saved. The general idea of applying dynamic programming to find a $(k+1)$-path starting from s is as follows.

For any vertex v in G', if there exists simple path from s to v of length i, all the color sets used by the paths from s to v with length i are saved. For simple path of length i, the number of color sets saved is at most $\binom{k}{i}$. Assume that $Q_{v,i} = \{C_1, C_2, \cdots, C_h\}$ $(1 \leq h \leq \binom{k}{i})$ is a set of color sets saved for the paths from s to v with length i. Now we analyze how to get a simple path of length $i+1$ from vertex s passing through v based on the color sets in $Q_{v,i}$. For each neighbor u of v and for each color set C_j $(1 \leq j \leq h)$ of $Q_{v,i}$, if the color of u is not contained in C_j, a color set $C' = C_j \cup \{f(u)\}$ of size $i+1$ can be constructed, where $f(u)$ is the color of vertex u. Therefore, color set C' is saved to denote that there exists a simple path of length $i+1$ from s to u using the colors of C'.

Now we analyze the time complexity of above dynamic programming process. For a simple path of length i through vertex v, in order to get a simple path of length

$i+1$, at most $|E|$ vertices should be considered, and at most $\binom{k}{i}$ color sets are saved to denote the simple paths from s to v. Therefore, the running time of above dynamic programming process is bounded by $O(\sum_{i=1}^{k} i \cdot \binom{k}{i} \cdot |E|) = O(|E| \cdot k \cdot 2^k)$.

For the k-Path problem, if G contains k-path, in order to find a k-path with high probability, repeat the above random coloring and dynamic programming re^k times, where r is a positive integer. Then, in time $O((2e)^k \cdot rk|E|)$, a k-path of G can be found with probability at least $1 - e^{-r}$.

3.2 Color-Coding based on perfect hash function

For a k-subset W of U, random Color-Coding can color W with probability around $1/e^k$. In order to color W properly in a deterministic way, a deterministic coloring scheme should be constructed, i.e., construct a (n, k)-coloring scheme of certain size such that W can be properly colored by at least one coloring in the (n, k)-coloring scheme.

How to construct a deterministic coloring scheme efficiently has attracted lots of attention. Currently, the most popular method for constructing deterministic coloring scheme is perfect hash function.

Definition.[1,26] perfect hash function: Given an universal set $U = \{1, 2, \cdots, n\}$ and a set $C = \{1, 2, \cdots, k\}$, g is a function from U to C, i.e., $g : U \rightarrow C$. For a subset $W \subseteq U$, if $g(i) \neq g(j)$, then g is called a perfect hash function on W.

Given a collection F of perfect hash functions, for any k-subset W, if there exists a function f in F such that f is a perfect hash function on W, then F is called a k-collection of perfect hash functions. It is easy to see that a k-collection of perfect hash functions is a (n, k)-coloring scheme.

The hash function for constructing deterministic coloring scheme generally has the following form:

$$g_{a,b,s}(x) = ((ax + b) \bmod p_n) \bmod s$$

where a, b, s are integers, and p_n is the smallest prime number between n and $2n$.

The method for constructing deterministic coloring scheme is based on the study on hash function in Ref. [16]. Schmidt and Siegal[22] gave a method to construct k-collection of perfect hash functions, in which each hash function can be constructed using $O(k) + 2\log\log n$ bits and is an injective function from Z_n to Z_{3k}. Then, (n, k)-coloring scheme can be obtained based on the $(3k, k)$-coloring scheme, and the size of the coloring scheme in Ref. [22] is bounded by $2^{O(k)}\log^2 n$. The above result was reduced by Ref. [20], in which a k^2-collection of hash functions from $\{1, 2, \cdots, n\}$ to $\{1, 2, \cdots, k^2\}$ is constructed, then a (n, k)-coloring scheme can be obtained by getting a k-collection perfect hash functions from $\{1, 2, \cdots, k^2\}$ to $\{1, 2, \cdots, k\}$, which is of size $2^{O(k)}\log n$.

For the methods used in Ref. [22], at least 12bits are needed to construct (n, k)-coloring scheme, i.e., the number of hash functions in (n, k)-coloring scheme is at least $2^{12k} > 4000^k$, which is not practical even if k is very small.

Chen *et al.*[9] constructed a k-collection of perfect hash functions through three steps: $Z_n \rightarrow Z_{k^2} \rightarrow Z_{k/4} \rightarrow Z_{c_j(c_j-1)}$, and obtained a (n, k)-coloring scheme of size $O^*(6.1^k)$. Recently, using conditional expectations and the method in Ref. [19], (ε, k)-balanced families of hash functions are constructed[2], resulting in a deterministic coloring scheme of size $e^{k+O(log^3 k)}\log n$, which is the current best result.

In the following, we use 3-Set Packing problem as an example to show how deterministic Color-Coding technique is used to solve problems.

We first give some related terminology and notions. A set of size three is called a 3-set. For a 3-set $\sigma = (a, b, c)$, let $Val(\sigma)$ denote the set of elements contained in σ, i.e., $Val(\sigma) = \{a, b, c\}$. Assume that S is a set containing n 3-sets. Let $Val(S) = \bigcup_{\sigma \in S} Val(\sigma)$. For any subset $P \subseteq S$, if any two 3-sets of P have no common element, then P is called a *packing*. If P is a packing containing exactly k 3-sets, then P is called a k-Packing.

Definition.[15] 3-Set Packing: Given a set S of 3-sets and a positive integer k, find a k-Packing in S, or return that no such packing exists.

In order to solve 3-Set Packing problem efficiently, the following problem is introduced.

Definition.[15] 3-Set Packing Augmentation: Given a set S of 3-sets and k-Packing P_k of S, find a $(k+1)$-Packing in S, or return that no such packing exists.

In fact, 3-Set Packing problem is equivalent to 3-Set Packing Augmentation problem, i.e., 3-Set Packing problem can be solved in $O^*(c^k)$ time if and only if 3-Set Packing Augmentation problem can be solved in $O^*(c^k)$ time[15].

For an instance of 3-Set Packing Augmentation problem (S, P_k), assume that S contains a $(k+1)$-Packing P_{k+1}. There exists a special structure relationship between P_k and P_{k+1}, as follows.

Lemma 3.1.[15] Given an instance of 3-Set Packing Augmentation problem (S, P_k), if S contains $(k+1)$-Packing, then there exists a $(k+1)$-Packing P_{k+1} such that for each 3-set p in P_k, $|Val(p) \cap Val(P_{k+1})| \geq 2$.

By Lemma 3.1, at least $2k$ elements of $Val(P_k)$ are contained in $Val(P_{k+1})$. Since $Val(P_{k+1})$ contains exactly $3k+3$ elements, $Val(P_{k+1}) - Val(P_k)$ contains at most $k+3$ elements, which are in $Val(S) - Val(P_k)$. The general idea using Color-Coding technique to find a P_{k+1} in S is as follows: Use $k+3$ colors to construct a $(Val(S) - Val(P_k), Val(P_{k+1}) - Val(P_k))$-coloring scheme such that $Val(P_{k+1}) - Val(P_k)$ is properly colored by at least one coloring in $(Val(S) - Val(P_k), Val(P_{k+1}) - Val(P_k))$-coloring scheme. In order to properly color $Val(P_{k+1})$, use extra $3k$ colors to color $Val(P_k)$. Therefore, a $(Val(S), 4k+3)$-coloring scheme can be constructed to color $Val(P_{k+1})$ properly .

Based on the $(Val(S), 4k+3)$-coloring scheme, dynamic programming technique is used to find a properly colored $(k+1)$-Packing, as follows. Let Q be a set to save all possible packings obtained in the process of dynamic programming, which is initialized as an empty set. For each 3-set σ_i and each packing P in Q, if the elements in σ_i have no common color with the elements in $Val(P)$, then a new packing $P' = P \cup \{\sigma_i\}$ is constructed. Moreover, if there is no packing in Q using the same colors as the elements of P', then P' is added into Q. After handling all the 3-sets in S, if there exists a $(k+1)$-Packing in S, then by searching in Q, a $(k+1)$-Packing can be returned.

3.3 Color-Coding for $n \leq 2k$

The random Color-Coding and the Color-Coding based on perfect hash function are only workable for the case when k is a small parameter, i.e., $k \ll n$. However, for many problems, parameter k is very close to n, such as Motif Finding problem, $k = 16$, $n = 20$. In this section, a Color-Coding method for $n \leq 2k$ is presented[26], which can be applied to many problems in Bioinformatics and Networks[25,28].

Assume that f is a (n,k)-coloring and $C=\{c_1,c_2,\cdots,c_k\}$ is a color set. If the elements of U are divided into k parts: $V_1,V_2,\cdots,V_k$, satisfying $V_i=\{v|f(v)=c_i\}$, then the number of k-subsets properly colored by coloring f is $\prod_{i=1}^{k}|V_i|$. What is the maximum value of $\prod_{i=1}^{k}|V_i|$? Based on the inequality $\prod_{i=1}^{k}a_i\le((\sum_{i=1}^{k}a_i)/k)^k$, when $||V_i|-|V_j||\le 1$ is true for any two sets $|V_i|$, $|V_j|$ $(i\ne j)$, the number of k-subsets properly colored by f is maximized, i.e., when the k colors of C are evenly distributed among the elements of U, (n,k)-coloring f has maximum number of k-subsets properly colored.

Assume that the number of elements of U is n. Divide set U into $\lceil n/2\rceil$ subsets $B=\{B_1,B_2,\cdots,B_{\lceil n/2\rceil}\}$ such that $U=\bigcup_{i=1}^{\lceil n/2\rceil}B_i$, and B_i, B_j $(i\ne j)$ have no common elements. Then, each subset contains at most two elements. For the subsets obtained by dividing U, each subset is called a *block*. A block with two elements is called a *double-block*, and a block with single element is called a *single-block*. It is easy to see that the number of single-block is at most one. For a coloring and a block B_i, if two elements of B_i have same color, then B_i is called 1color-block, otherwise it is called 2colors-block.

3.3.1 Algorithms for coloring scheme

The Color-Coding method for $n\le 2k$ makes full use of the idea of evenly distributing colors. For any (n,k)-coloring under the case $n\le 2k$, the number of elements with same color is at most two. In the following, we first give coloring scheme construction method for some special cases, such as $n=k$, $n=k+1$, $n=k+2$. Then, a general method of constructing coloring scheme for $n\le 2k$ is given.

(1) $n=k$

Under this case, a coloring scheme of size one can be constructed by one-to-one mapping from U to C.

(2) $n=k+1$, $k\ge 1$

Since $n=k+1$, a coloring scheme can be constructed by using any color exactly twice, as follows.

For a block B_i, discuss the coloring on B_i by the following two cases.

(a) B_i is a double-block.

Under this case, choose an arbitrary color c_i to color the two elements of B_i. Then, get a one-to-one mapping from $U-B_i$ to $C-\{c_i\}$.

(b) B_i is a single-block.

Under this case, choose an element e from any other blocks and add e into B_i to make B_i a double-block, which can be handled by case (a).

By choosing a block B_i from B and using the above coloring process, a (n,k)-coloring can be constructed. Therefore, $\lceil n/k\rceil$ colorings for $n=k+1$ can be constructed by enumerating all possible blocks of B, denoted by F.

Now, we prove that F is a (n,k)-coloring scheme for $n=k+1$. For any k-subset $W=\{x_1,x_2,\cdots,x_k\}$, if the element of $\{y\}=U\backslash W$ is contained in a double-block B_i, then by case (a), a (n,k)-coloring can be constructed by coloring B_i with any color c_i and getting a one-to-one mapping from $U-B_i$ to $C-\{c_i\}$. On the other hand, if y is in a single-block B_i, by case (b), an element e of W can be added into B_i to make B_i a double-block, which has been handled by case (a). Therefore, for any k-subset W of U, W can be properly colored by a coloring in F. Therefore, F is (n, k)-coloring scheme for $n = k + 1$.

(3) $n = k + 2$, $k \geq 2$

Assume that Q contains all k-subsets of U, where $|Q| = \binom{n}{k} = \binom{n}{2}$. The k-subsets of Q are divided into the following two subsets to handle.

(a) $Q_1 = \{W | W \in Q$ and there exist only two blocks in B, each of which has one element not in W $\}$.

For any k-subset W of Q_1, in order to color W properly, each coloring must have two 1color-blocks. Therefore, choose any two blocks B_i, B_j from B. If $\{B_i, B_j\}$ contains single-block, choose any element from $B - \{B_i, B_j\}$ to make the single block in $\{B_i, B_j\}$ a double-block. Then, arbitrarily choose two colors c_i, c_j from C to color blocks B_i, B_j, each of which is colored by one color respectively. Finally, get a one-to-one mapping from $U - (B_i \cup B_j)$ to $C - \{c_i, c_j\}$. It is easy to see that $\binom{\lceil n/2 \rceil}{2}$ colorings are needed to properly color the k-subsets in Q_1.

(b) $Q_2 = \{W | W \in Q$ and for each block B_i of B, W either contains all elements of B_i, or contains no element of B_i. $\}$.

Since for a k-subset W of Q_2 and any block B_i of B, W either contains all elements of B_i, or contains no element of B_i, the coloring on elements of U can be transformed to the coloring on blocks of B, which is equivalent to using $k' = \lceil n/2 \rceil - 1$ colors to color $n' = \lceil n/2 \rceil$ blocks. Since $n' = k' + 1$, a set of (n', k')-coloring F' can be constructed based on the method in case (2), which is of size $\lceil n/4 \rceil$. Based on (n', k')-coloring in F', a set F of (n, k)-coloring can be obtained in the following way. For a (n', k')-coloring f', a color c_i used by f' corresponds to two colors c_{i1}, c_{i2} in a (n, k)-coloring. For each (n', k')-coloring f', and for each color c_i used by f', if a double-block B_i is colored by c_i, then use colors c_{i1}, c_{i2} to color the elements of B_i. If a single-block is colored by c_i, find a double-block B_j whose color is uniquely used by B_j under f'. Add one element of B_j to B_i to make B_i a double-block. Then, use colors c_{i1}, c_{i2} to color the elements of B_i. Therefore, a set F of (n, k)-coloring of size $\lceil n/4 \rceil$ can be constructed, which can properly color the k-subsets in Q_2.

In conclusion, for the case $n = k + 2$, a coloring scheme of size $\binom{\lceil n/2 \rceil}{2} + \lceil n/4 \rceil$ can be constructed.

Before presenting the idea for constructing coloring scheme for $n \leq 2k$, we first give a method to adjust a single-block to a double-block, as follows. Assume that P contains all the blocks to be adjusted. For any block B_i of P, arbitrarily choose a block B_j from $B - P$, and add one element of B_j into B_i to make B_i a double-block.

The process of constructing (n, k)-coloring scheme for $n = k$, $n = k+1$, $n = k+2$ gives a basic idea how to get (n, k)-coloring scheme for $n \leq 2k$, which is specifically given in the following.

(1) Divide the elements of U into $\lceil n/2 \rceil$ blocks, and get a set B of blocks.

(2) Enumerate all possible 1color-blocks from B.

(3) For each enumeration on the 1color-blocks, let B' be the set of 1color-blocks obtained. Then, all blocks in $B - B'$ are 2colors-block. The coloring on all 2colors-blocks can be transformed to a (n', k')-coloring, which can be recursively solved, where n' is the number of 2colors-blocks, $k' = \lceil (k - |B'|)/2 \rceil$.

(4) Based on the enumeration on 1color-blocks and the (n', k')-coloring on 2colors-blocks, a (n, k)-coloring scheme can be obtained by the relationship between colors

used by (n', k')-coloring and colors used by (n, k)-coloring. .

For $n \leq 2k$, by using the above process, a (n, k)-coloring scheme can be constructed.

Theorem 3.2.[26] Given any two integers n, k such that $n \leq 2k$, a (n, k)-coloring scheme of size $O(e^{m(n-k)})$ can be constructed, where m is the maximum root of $e^x - e^{(3-2\beta)x} + 1 = 0$, $0.5 \leq \beta \leq k/n < 1$.

3.3.2 Application on motif finding

Motif Finding problem is an important problem in Bioinformatics, which is to identify motif model and motif instance in DNA sequence. We first give related definition.

Definition.[10,25] (l, d)-k Motif Finding: Given a set $S = \{s_1, s_2, \cdots, s_k\}$ of K strings, where $|s_i| = L$ $(1 \leq i \leq k)$, construct a string x of length l, satisfying that there exists a subset $S' \subseteq S$, $|S'| \geq k$, such that for any string s_i in S', a substring y_i of length l in s_i having d different positions with string x can be found.

For (l, d)-16 Motif Finding problem with $K = 20$, a (n, k)-coloring scheme of size 403 can be constructed with $n = 20$ and $k = 16$, which greatly improves the enumeration number $\binom{20}{16} = 4845$. Based on the Color-Coding method, the (l, d)-16 Motif Finding problem with $K = 20$ can be transformed to (l, d)-16 Motif Finding problem with $K = 16$, which can be solved using branch-and-bound technique.

4 Applications of Color-Coding

As an efficient way solving subset selection problem, Color-Coding technique has great applications in many fields, such as Bioinformatics, Networks, Model Checking[8,12], Counting[3], etc. In this section, we give brief introduction on applications of Color-Coding technique, especially in solving problems related to k-Path problem, Subgraph Isomorphism problem, Matching and Packing problems, (t, n)-Ring Signature problem, and Worm Signature problem.

4.1 Problems related to k-Path

In section 3.1, we have shown that random Color-Coding method can be used to solve k-Path problem efficiently.

Recently, lots of attention has been focused on using Color-Coding to solve path finding problems in Bioinformatics. Scott *et al.*[23] applied Color-Coding method to find protein path in protein interaction networks. Based on real biological data, the algorithm in Ref. [23] can find a 8-path in 1 minutes and 10-path in 2 hours. By using Color-Coding on path finding, Shlomi *et al.*[24] designed a tool called Q-Path to find paths in biology data.

For k-Path problem, Hüffner *et al.*[13] gave that by using $1.3k$ colors, k-Path problem can be solved in time $O(|\ln\varepsilon|(4.32)^k m)$ with probability ε, where m is the number of edges of given graph. The implemented algorithm of Ref. [13] can find 13-path in a few seconds.

Line Planning is an important problem in public transport system, which is closely related to maximum weighted k-Path problem[6]. For the maximum weighted k-Path problem, Color-Coding technique can be used to give an efficient algorithm[6].

4.2 Subgraph Isomorphism problem

Subgraph Isomorphism problem is an important model matching problem, which has great applications in Bioinformatics, VLSI, etc.

Definition.[1] Subgraph Isomorphism: Given two graphs G and Q, does there exist a subgraph W of G which is isomorphic to Q.

When Q is a forest, by using Color-Coding technique, Alon *et al.*[1] gave algorithms of expected time complexity $O(2^{O(k)}|E|)$ and $O(2^{O(k)}|V|)$ for directed and undirected graphs respectively.

In order to solve Steiner tree problem in biology networks analysis, Betzler[5] used Color-Coding and dynamic programming to solve tree isomorphism problem, and obtained an algorithm of running time $O(2^{O(k)}\log|V|\cdot|E|\cdot k)$, where $|V|$, $|E|$ are the number of vertices and edges of given graph respectively.

4.3 Matching and packing problems

Matching and Packing problems form an important class of NP-hard problems, which have wide applications in the fields of scheduling[4] and code optimization[21]. In section 3.2, we have shown that Color-Coding can be successfully used to solve 3-Set Packing problem. In the following, we give some other results using Color-Coding to solve Matching and Packing problems.

Fellows *et al.*[11] gave a systematic study on rD-Matching, r-Set Packing, Graph Packing and Graph Edge Packing problems. By using Color-Coding and dynamic programming technique, an algorithm of time $O(n+2^{O(k)})$ was given in Ref. [11].

By using Color-Coding technique, Koutis[17] proposed an algorithm of time $O(2^{O(t)}nN\log N)$ for r-Set Packing problem, where n is the number of sets in given instance, and N is the number of elements in given instance.

For 3D-Matching problem, Chen *et al.*[9] pointed out that for a given instance of 3D-Matching problem (S,k), where S is a collection of n triples, if S contains a matching S_k of size k, by using $3k$ colors, S_k can be properly colored, and for each coloring, dynamic programming can return a matching of size k in time $O(2^{3k}n)$ if such matching exists. Finally, an algorithm of time $O^*(12.8^{3k}n^2)$ was presented in Ref. [9].

For the weighted mD-Matching and weighted m-Set Packing problems, Wang and Liu[27] gave parameterized algorithms of time $O^*(12.8^{(m-1)k})$ and $O^*(12.8^{mk})$ respectively by using Color-Coding and dynamic programming technique.

For Edge Disjoint Triangle Packing problem, by using Color-Coding method, an algorithm of time $O(2^{(9k/2)logk+(9k/2)})$ was presented in Ref. [18].

4.4 (t,n)-ring signature problem

(t,n)-ring signature is a popular encryption technique, which has been used in electronic voting, digital lottery, electronic credit card, etc. For (t,n)-ring signature technique, assume that there are n users, each of which has a public key and a private key. If an information is delivered, in order to guarantee the correctness of delivered information, the information must contain the public keys of n users and the private keys of t users, i.e., the correctness of information is guaranteed by t users whose private keys are contained in the information, and it can be said that the t users sign on the information.

Combining Color-Coding technique with (t, n)-ring signature technique, Bresson *et al.*[7] proposed a technique called Ad-Hoc ring signature. Different from (t, n)-ring signature, for Ad-Hoc ring signature, there exists an Ad-Hoc group, i.e., a list of subsets of users, each of which is called an *acceptable-subset*. Moreover, Ad-Hoc ring signature requires that all users signed belong to at least one acceptable-subset. In order to satisfy the above requirement that all signed users are in at least one acceptable-subset, the user ring is divided into sub-rings such that each sub-ring contains exactly one user signed, which is called *Fair Partition*. For achieving Fair Partition, Color-Coding technique can be used to color the ring, i.e., divide the ring into several sub-rings such that the users in each sub-ring is colored by the same color. Based on the coloring on the ring, the signature process can be achieved by using sub-ring to sign on the information.

Isshiki and Tanaka[14] applied Color-Coding technique to solve the $(n-t)$-out-of-n signature problem, where t is the number of users not signing.

4.5 Worm signature

In order to prevent worms from propagating rapidly, worm signature should be generated quickly and accurately. Wang *et al.*[28] applied Color-Coding technique to generate worm signature. Firstly, the given sequences can be divided into groups such that each group contains 20 sequences. In each group, worm signatures can be generated by using Color-Coding for $n \le 2k$. Experiment results in Ref. [28] show that worm signatures generated by Color-Coding have obvious advantages over other approaches. Table 1 gives a comparison between the number of colorings used and the corresponding enumeration number.

Table 1 Comparison between $(20, u)$-coloring and $\binom{20}{u}$

	$(20, u)$-coloring	$\binom{20}{u}$
$u = 19$	10	20
$u = 18$	50	190
$u = 17$	170	1140
$u = 16$	403	4845
$u = 15$	862	15504
$u = 14$	1220	38760
$u = 13$	2036	77520
$u = 12$	2085	125970
$u = 11$	3250	167960

5 Conclusions and Further Research

In this paper, we give brief introduction on Color-Coding technique, mainly focusing on three Color-Coding methods: random Color-Coding, Color-Coding based on perfect hash function, and Color-Coding for $n \le 2k$. Moreover, applications of Color-Coding technique are presented.

Although Color-Coding technique is well-studied, there still exist some interesting and challenging problems.

(1) Practical software of Color-Coding.

The involved problems include: How to construct coloring scheme database? How to save coloring in an efficient way? How to avoid repeated coloring (A subset is properly colored by many colorings)?

(2) Extend applications of Color-Coding.

How to apply Color-Coding technique to solve problems in Database System, Artificial Intelligence, Social Science, etc? On the other hand, for some problems, based on real data set, how to design Color-Coding for special application cases?

References

[1] Alon N, Yuster R, Zwick U. Color-coding. Journal of the ACM, 1995, 42: 844–856.

[2] Alon N, Gutner S. Balanced hashing, color coding and approximate counting. Proc. of the 4th International Workshop on Parameterized and Exact Computation (IWPEC 2009). Lecture Notes in Computer Science 5917. 2009. 1–16.

[3] Arvind V, Raman V. Approximation algorithms for some parameterized counting problems. Proc. of the 13th International Symposium on Algorithms and Computation (ISAAC 2002). Lecture Notes in Computer Science 2518. 2002. 453–464.

[4] Bar-Yehuda R, Halldórsson M, Naor J, Shachnai H, Shapira I. Scheduling split intervals. Proc. of the 13th Annual ACM-SIAM symposium on Discrete Algorithms (SODA 2002). 2002. 732–741.

[5] Betzler N. Steiner Tree Problems in the Analysis of Biological Networks. Diplomarbeit, Wilhelm-Schickard-Institut für Informatik. Universität Tübingen. 2006.

[6] Borndörfer R, Grötschel M, Pfetsch ME. A path-based model for line planning in public transport. Konrad-Zuse-Zentrum für Informationstechnik Berlin. 2005.

[7] Bresson E, Stern J, Szydlo M. Threshold ring signatures and applications to ad-hoc groups. Proc. of the Advances in CryptologyCrypto. Lecture Notes in Computer Science 2442. 2002. 465–480.

[8] Chandra AK, Merlin PM. Optimal implementation of conjunctive queries in relational data bases. Proc. of the 9th Annual ACM Symposium on Theory of Computing (STOC 1977). 1977. 77–90.

[9] Chen J, Lu S, Sze SH, Zhang F. Improved algorithms for path, matching, and packing problems. Proc. of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 07). 2007. 298–307.

[10] Chin F, Leung H. An efficient algorithm for the extended (l, d)-motif problem with unknown number of binding sites. Proc. of IEEE 5th Symp. on Bioinformatics and Bioengineering (BIBE 2005). 2005. 11–18.

[11] Fellows M, Knauer C, Nishimura N, Ragde P, Rosamond F, Stege U, Thilikos D, Whitesides S. Faster fixed-parameter tractable algorithms for matching and packing problems. Proc. of the 12th Annual European Symposium on Algorithms (ESA 2004). Lecture Notes in Computer Science 3221. 2004. 311–322.

[12] Grohe M. Descriptive and parameterized complexity. Proc. of the 13th International Workshop and 8th Annual Conference of the EACSL on Computer Science Logic (CSL 1999). Lecture Notes in Computer Science 1683. 1999. 14–31.

[13] Hüffner F, Wernicke S, Zichner T. Algorithm engineering for color-coding with applications to signaling pathway detection. Algorithmica, 2008, 52(2): 114–132.

[14] Isshiki T, Tanaka K. An $(n-t)$-out-of-n threshold ring signature scheme. Proc. of the 10th 10th Australasian Conference on Information Security and Privacy (ACISP 2005). Lecture Notes in Computer Science 3574. 2005. 4–6.

[15] Liu Y, Lu S, Chen J, Sze SH. Greedy localization and color-coding: improved matching and packing algorithms. Proc. of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC 2006). Lecture Notes in Computer Science 4169. 2006. 84–95.

[16] Michael L, Komlós J, Szemerédi E. Storing a sparse table with O(1) worst case access time. Journal of the ACM, 1984, 31(3): 538–544.

[17] Koutis I. A faster parameterized algorithm for set packing. Information Processing Letters 94. 2005. 7–9.

[18] Mathieson L, Prieto E, Shaw P. Packing edge disjoint triangles: a parameterized view. Proc. of the 1st International Workshop on Parameterized and Exact Computation (IWPEC 2004). Lecture Notes in Computer Science 3162. 2004. 127–137.

[19] Naor M, Schulman L, Srinivasan A. Splitters and near-optimal derandomization. Proc. of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95). 1995. 182–190.

[20] Naor J, Naor M. Small-bias probability spaces: efficient constructions and applications. SIAM J. Comput., 1993, 22: 838–856.

[21] Hell P, Kirkpatrick D. On the complexity of a generalized matching problem. Proc. of the 10th Annual ACM Symposium on Theory of Computing (STOC 1978). 1978. 240–245.

[22] Schmidt JP, Siegel A. The spatial complexity of oblivious k-probe hash functions. SIAM J. Comput., 1990, 19(5): 775–786.

[23] Scott J, Ideker T, Karp RM, Sharan R. Efficient algorithms for detecting signaling pathways in protein interaction networks. Proc. of the 9th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2005). Lecture Notes in Computer Science 3500. 2005. 1–13.

[24] Shlomi T, Segal D, Ruppin E, Sharan R. QPath: a method for querying pathways in a protein-protein interaction network. BMC Bioinformatics, 2006, 7: 199.

[25] Wang J, Huang Y, Chen J. A motif finding algorithm based on color coding technology. Journal of Software, 2007, 18(6): 1298–1307 (in Chinese with English abstract).

[26] Wang J, Liu Y, Chen J. Effective coloring algorithm for close relationship between the scales of element set and color set and its application. Chinese Journal of Computers, 2008, 31(1): 32–42 (in Chinese with English abstract).

[27] Wang J, Liu Y. Parameterized algorithms for weighted matching and packing problems. Discrete Optimization, 2008, 5(4): 748–754.

[28] Wang J, Wang J, Chen J. An Automated Signature Generation Approach for Polymorphic Worm Based on Color Coding. Proc. of IEEE International Conference on Communications. 2009. 1–6.