Alternating Path of length L in C-edges Colored Graphs

Thierry Edoh

Orsay Laboratory Research in Data processing Batiment, Universities Paris South.

Abstract

A graph $G = (V, E)$ is called C-edge colored if and only if there exists a mapping $A$ between the set $E(G)$, the edge set in the graph $G$, and a set $C$ of colors, such that $|C| = c$, and $A$ maps a color $c$ in $C$ to every edge $e \in E(G)$. A path $P$ in a C-edge colored graph is called an alternating path if two adjacent edges of $P$ differ in color.

We propose a dynamic algorithm, which resolves the following problem:

- Find an alternating path of length $L$ in a C-edge colored complete graph between two given vertices $s, t \in V(G)$.

Besides we show that the proved results for $c = 2$ in [2] for the $xy$-path is also true for $c > 2$. Thus we generalize these results. We also deal with the problem of the $(s, t)$-Cycles, for 2-edge colored presented in [7], in a C-edge colored graph with $c > 2$. We propose another algorithm that finds in polynomial time the set of all paths and cycles of length of $k$ within a C-edge colored graph, where $\log(n) < k \leq n$.

We also consider the problem to determine the length of a shortest path between two given vertices in C-edge colored complete graph. In this paper, we therefore deal with the following question:

- Under which conditions can it be possible to reduce in a graph $G$ the length of a given path between two vertices such as the path remains alternating between the two vertices?

In another paper we will deal with the minimal length of such path.

The resolution of this question obliges us to consider the different classes of graphs to find the necessary and sufficient conditions to attain such goal.

Keywords: C-edge Colored, Dynamic Algorithm, Paths and Cycles of length of $k$, Alternating Path

Introduction

Let a graph $G$ be a C-edge colored complete graph, where $c \geq 2$. We will study in this paper the necessary and sufficient conditions for the existence of alternating paths of length $k$, the possibilities to reduce the length of a given path between two given vertices in $G$.

Let be a graph $G = (V, E)$ where $E = E_1 \cup E_2 \cup \ldots \cup E_c$, represents the union of differently colored edges, let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of $G$ and let $C = \{1, 2, \ldots, c\}$ be the set of $c$ colors; $G$ is called C-edge colored complete graph, if $G$ is a complete graph and colored by $c$ different colors, for $c \geq 2$.

In this paper, we will deal with C-edge colored complete graphs where $c \geq 3$. Some interesting results for $c = 2$ are already presented in [1].

Deciding the existence of an alternating path in a C-edge colored complete graph $G$ is already known as NP-complete [2], because the number of possible paths to count

E-mail Id: oscar.edoh@mnet-online.de

can quickly explode. But also in \cite{2} it is shown that there exists an O(n^2) algorithm to find some alternating path (if any) of length of at least m, where 2 ≤ m ≤ 4, in a 2-edge colored complete graph. Still in \cite{2} it is shown that in a 2-edge colored graph G, i.e. edges are colored in blue and red, it can be decided in time O(n^5) time, whether there exists, two vertex and edge disjoint paths p1 and p2 between two given vertices x and y, where p1 or p2 has length 2.

We will generalize the results presented by Manoussakis in \cite{2} on C-edge colored complete graphs for c ≥ 2. We propose an O(n^3) dynamic algorithm that finds (if any) a path of given length k between two given vertices in a C-edge colored complete graph G. We also show that in defined conditions we can reduce the length of a given dynamic path of length k, while the reduced path keeps the two initial extremities.

**Notation and Terminology**

Throughout this paper we understand by a graph a C-edge colored complete graph, where c > 2 and by a 2-graph a 2-colored edge complete graph.

k-DAP and k-DAC represents the set of alternating paths and cycles of length k, respectively.

We denote the color of an edge by \( C(e) \) and by \( e \), the edge that connects vertices i and j. \( V(G) \) and \( E(G) \) denotes the vertex and edge set of G, respectively. The order n of a graph is the number of its vertices.

A simple path is a path, who goes through a vertex only one time.

**Alternating cycles and paths in a C-edge colored complete graph**

**Dynamic alternating Paths of length K (k-DAP)**

We consider a C-edge colored complete graph G: = (V, E), whose edges are colored by at least three different colors. E = E1 U E2 U E3 U...U Ec where c ≥ 3. Let s and t be two vertices of the graph G representing the start and destination vertices of an alternating path, of length k, existing within the graph G. A path (s, t) is called an alternating (s, t)-path of length k between s and t if and only if:

- The length of the path is exactly k,
- Two adjacent edges differ in their colors

**Problem 1**

Consider a C-edge colored complete graph G := (V, E), does there exist an polynomial time algorithm, that finds (if any) a s-t alternating path of a length exactly k within G, when k is given?

Does there exists, for a given k, an polynomial time algorithm to find in a graph an alternating path P of length k between two given vertices s and t representing the start and the destination of P, respectively and which conditions are necessary and sufficient to assert the existence an alternating path of length k in the graph?

For a k being part of the input and k is equal the order n of the graph G, this problem is NP-hard, since we can reduce the Hamilton problem to our problem, when we find an simple alternating path of length k-1, this path surly a hamilton path. One of the big difficulties of our problem is to find vertex and edge disjoint paths. Omitted this it is very easy to find the path of length k. In this section we propose a dynamic algorithm to resolve the put problem. In \cite{2} it is shown that in special cases in polynomial time if a graph G contains a vertex and edge disjoint path of length O (log V). This helps us to develop a method with which it can be decided in polynomial time, whether a graph contains a vertex and edge disjoint path of length k, even for k = |V| while using the conjecture of Papadimitriou and Yannakakis associated with a technique that we will present in the section 2.4. Previously we show the existence of a dynamic algorithm able to resolve the problem 1 and for that we consider the following Proposition 1 and the lemma 1. In next we present our dynamic algorithm:

**Proposition 1.1**

There exists a O ((|V|2^c|E|)dynamic algorithm to find a vertex and edge disjoint alternating path of length k in a C-edge colored complete graph G, with c ≥ 2.

In the following, we try to prove the proposition 1.1 with follow lemma.

**Lemma 1.1**

Consider a C-edge colored complete graph G := (V, E), where c = 2 and an alternating path of length k between two vertices x and y. If there exists at least one vertex in this path that is visited more than two times, then there exists an other vertex disjoint path, of length k, that can be found in time O (n^3).

**Proof**

Let G = (V, E) be a graph and f a dynamic function defined as follows:

\[ F: E \times E \rightarrow \{0, 1\} \]

(\( X, y \Pfad \) f (x, y) = 1 if \( x, y \in E \) and \( C(x) \neq C(y) \) otherwise 0 \( x, y \in E \times E \)

Let P = \( s, v_1, v_2, ..., v_k, t \) be a path (s, t) starting with s and finishing by t. The function F leaves s and chooses the best neighbour \( v \), i.e. the vertex having the biggest color degree. Thus we have just found a path of length 1. We eliminate the vertex s and v from the list of the vertices to visit. From the path of length 1 we’ll find the path of length 2 such
as the path of path of length 1 will be a part of the path of length 2. Note that we assign an orientation to each edge determined by the path P, then we try to find a path starting from a given vertex s and ending by a given vertex t and exactly of a given length k.

To find the path of length 2 we consider the end of the path of length 1 as start vertex. This vertex having a degree of \( n - 1 \), \( n \) is the order of the graph; we first look to the best vertex candidate by which must pass the path. This vertex must be different of the already visited vertices. The thus formed edge must differ in color as the last edge of the path P. The operation will be repeated until to attain the given length. Note that before attaining the length \( k - 1 \) all vertices candidates must be different of the destination vertex t. Therefore we see clearly that the path that will thus be found will be vertex and edge disjoint and exactly of length \( k \) and \( k \) will never be able to be superior than \( n - 1 \).

We deal with complete graphs, so it certain, in case of existence of an alternating path within the graph this will be found. The Algorithm 1 shows how an alternating path in a graph G can be found. Thus found path is called dynamic alternating path of length \( k \) (k-DAP).

Since we can show that an \( O(n^3) \)-algorithm exists to find a K-DAP we prove the proposition 1.1, because \( n^3 \leq |V|^2 |E| \) \((n = |V| \text{ and } |V| \leq |E| \text{ for } n > 2)\)

**Algorithm 1: Construction of a dynamic alternating path**

The algorithm is based on a following dynamic function \( F \): as already above defined \( F := p(v_i, c_i, v_j, c_j, k) \in \{0, 1\} \). \( P \) is the path beginning with the vertex \( v_i \), a start color \( c_i \) and ending with vertex \( v_j \), as destination and \( c_j \) as coming color; \( k \) is the length of the path. For \( k = 1 \) \( c_i \) is null, \( v_j \) the start vertex, \( v_j \) the destination, \( c_j \) the color of the path \( v_i v_j \), an edge. Then \( c_j \) becomes the new start color and \( v_j \) the new start vertex. For a \( k = 2 \) we look for an edge that differ in color to the new start color, that mind the color of the last edge of the path. Thus, the length of the path increments itself of a edge if we find the adequate edge, that we can hang to the existing path. If no edge is found, then algorithm stops, so we can see that the graph has any path of length \( k \), but of length \( \beta \) such as: \( 1 \leq \beta < k \), then the path of length 1 always exists.

**The pseudo code of algorithm 1**

**Input:** a graph \( G := (V, E) \); \( s, t \): two given vertices (start and destination vertices), an integer \( k \), the length of the path to find.

**Output:** an alternating path of length \( k \), if any otherwise an alternating path of length \( ki \) and \( 1 \leq k < k \)

**Begin**
- Initialisation
  
  \( P= \emptyset \); Path

  int stop; // to oversee the buckle;

  Color c; // start color

  Vertex vs, vd; // start and destination vertex

  for \( k = 1 \) // length of the path to find is 1

  \( F = P(vs, c, vj, cj, 1) \)/ find the edge \( \{vs , vj\} \) and \( c \neq cj \).

  for \( k \geq 2 \) // length of the path to find is superior or equals to 2

  repeat

  if (\( vj \) not yet visited) then

  \( F = P(vs, c, vj, cj, 1) \)

  vs ← vj;

  c ← cj;

  vj ← new vertex;

  cj ← new color;

  p ← p \{vs, vj\};

  until stop = k–1

  \( F = P(vs, c, vd, cj, 1) \);

  p ← p \{vs, vd\};

  return \( P \);

**End**

**Complexity and analysis of the algorithm**

If the length of to find path is 1 the complexity is trivial \( O(|V|) \).

For a length \( k \geq 2 \), the algorithm looks among all the neighbours of the start vertex for the best candidate vertex and finds thus the path of length 1 in a time of \( O(|V|^2 |E|) \). This complexity will be explained as follows:

Our graph is complete, i.e. the vertices are pairwise neighbours. Then every vertex has a degree of \( n-1 \) with \( n \) the order of the graph. We record the set of the edge and vertices within the graph in different fields/arrays. In the third block of algorithm we try to visit only the vertices, which are not yet visited. Here we have to traverse the fields of vertices of a length equals to \( |V| \) to select the best candidate vertex To get the color of the edge, knowing its 2 vertices, i.e. extremities, we’ll traverse the fields of a length of \( |E| \), in which the edge within the graph G and their characteristics are stored. To attain the length k-1 we have to evolve step by step. Each step cost \( O(|V|^* |E|) \). For every a step we have each time to traverse the fields of the vertices. Find the path of length k-1 would cost \( O(\beta^* |V|^2 |E|) \) and \( \beta = (K-2) \). If the length k-1 is attained,
there will remain to find the last edge of the path in order to attain the length \( k \). This operation costs too \( O(|V|^2|E|) \). Altogether have a cost of \( O(|V|^2|E| + |V|^3|E|) \) noticeably equal \( O(|V|^3|E|) \). Note that \( |V|^2 < |V|^3 |E| \) then \( |E| \geq |V| \) in a complete graph, \( |E| = |V|^3(|V| - 1)/2 \), then we can say that it exists a \( O(n^3) \) algorithm with \( n^3 = |V|^3 \) to find a path of length \( k \) in a graph. □

**Theorem 1.1**

Consider a complete graph \( G \), two vertices \( s \) and \( t \) within \( G \) and an integer \( k \) such as \( k \leq n \), the order of the graph \( G \). It exists a \( O(n^3) \) dynamic algorithm to find an alternating path of length \( k \) in \( G \). 

**Proof**

Since we showed above from the lemma 1.1 that it exists a on dynamic programming based algorithm able to find an alternating path of length \( k \), the theorem is thus proved.

**x-y-alternating paths**

**Problem 2.1**

Consider a \( c \)-edge colored complete graph \( G = (V, E) \), where \( c \geq 3 \). Does there exists a polynomial time algorithm to find an alternating \( x-y \)-path? 

For \( c = 2 \) there exists a \( O(n^{2.5}) \) algorithm to find an alternating \( x-y \)-path, which was proved in \([2]\). 

Before resolving the problem 2.1, we show that there exists also another non dynamic algorithm to find a path of length \( k \) in a \( 2 \)-graph. For that we will use the lemmas below:

**Lemma 2.1**

Let \( G = (V, E) \) be a connected \( 2 \)- colored edge multi-graph and \( x, y \) two distinct vertices of \( G \). For each \( i \in \{1, 2\} \) we can find (if any) in time \( O(|E|) \) an alternating path \( P = x_1, x_2, \ldots, x_k \) with \( x_1 = x \) and \( x_k = y \), \( C(x_1, x_2) = i \) and \( C(x_{k-1}, x_k) = j \), where \( j = 3 - i \).

The “Lemma 2.1” is shown in \([3]\). Thus we can efficiently find in a \( 2 \)- colored edge multigraph two vertices which are connected by the color.

**Lemma 2.2**

Let \( G = (V, E) \) be a \( 2 \)-graph and \( x, y \) two distinct vertices in \( G \) representing the start and the destination, respectively. We can find, if any, an alternating path \( P := x_1, x_2, \ldots, x_k \) of length \( k \) for a given \( k \), where \( x_1 = x \) and \( x_k = y \), \( C(x_1, x_2) \neq C(x_{k-1}, x_k) \) in time \( O(|E|) \).

**Proof**

Let \( G = (V, E) \) be a \( 2 \)-graph. We transform \( G \) into a multigraph \( G' \), where we substitute each colored edge by a colored multi-edge of same color and extremities. It exists an algorithm that finds in \( G' \) an alternating path of length \( k \), proved in \([3]\). It is clear to see that while reducing the multi-edge of \( G' \) in simple edge the obtained path remains unchanged and we obtain again the initial graph \( G \) that becomes a minor of \( G' \). Thus it is showed that one can find in a complete 2-graph a path of length \( k \) in \( O(|E|) \). □

**Theorem 2.1**

An alternating path of length \( k \) can be found in a 2-graph \( G := (V, E) \) in time \( O(|E|) \), if a minor of \( G \) contains a path of length \( k \).

**Proof**

Since we showed above from lemma 2.2 that it exists an \( O(|E|) \)-algorithm to find an alternating path in 2-edge colored graph, the theorem 2.1 is thus proved.

**Corollary 2.1.1**

If a graph \( G \) contains a path of length \( k \), then it exists one of its minor that contains also an alternating path of length \( k \).

**Proof**

A minor \( H \) of a graph \( G \) can be his sub-graph, if and only if we just delete some edges from \( G \) to obtain \( H \). We can also contract vertices and delete edges to get a minor, but her we consider that only some edges are deleting. Such minor is too a sub-graph. Then all that contains \( H \), \( G \) contains it also. And \( H \) can contain also things of \( G \) if they are located in \( H \). □

We showed above that an alternating path of length \( k \) can be found in a complete 2-graph in \( O(|E|) \) and it’s proved in \([3]\) that it exists a \( O(n^{2.5}) \) algorithm to find an alternating path in a complete graph.

In Figure 1 we can observe that an alternating \( v_s-v_e \) path \( P \) in \( G \) corresponds to a perfect matching in \( G \). Reciprocally

**Figure 1.** An alternating vs- ve path

it is easy to check that perfect matching in \( G \) correspond to an alternating \( v_y-v_x \) path in \( G \). This is done by taking the first edge \( v_i \) in the matching set, \( M \), adjacent to \( v_i \) and the red/blue edge \( v_j \) in \( G \). We repeat this until we obtain the destination.

**Theorem 2.2**

Consider a complete graph \( G \), two vertices \( s, t \) in \( G \) and
an integer \( k \) such as \( k < n \), \( n \) the order of the graph \( G \). \( G \) is \( C \)-edge colored and contains an alternating vertex and edge disjoint path of length \( k \), \( k\text{-DAP} \), if and only if \( G^{\text{c}} = (V', E') \) contains a perfect matching.

**Proof**

In[1] it’s showed and proved that a graph \( G \) contains an alternating path if \( G^{\text{c}} = (V', E') \) contains a perfect matching. So we can see that this path can have the length \( k \) and then no vertex will be visited more than two times, this here guaranteed that the path will be vertex and edge disjoint and the length \( k \) would be inferior to \( n \). □

Now we will try to generalize the obtained results in[2], that proved that it exists a \( O(n^{2,5}) \) algorithm to find a path in a 2-graph. To show that this here is too valid for \( c > 2 \) (the number of used colors), we’ll differentiate two groups of numbers representing \( c \).

- \( c = 2^k \) with \( k \in \{1, 2, 3, 4, 5, \ldots, +\infty\} \)
- \( c = 2K + 1 \) (odd) and \( c \neq 2^k \)

\( C = 2^k \)

Here we will use a technique that reduces the number \( c \) to 2, \( c > 2 \), as follows:

We have even \( c \) that divided by 2 yields an even number so that again divided by two yields an even number and so further until to attain the number 2 in the last. For such a group of number \( c \) we put together two different colors in a color. For example blue and red in black or yellow and gray in red, etc. This operation will be followed until at end we obtain a 2-graph. The thus obtained 2-graph satisfies the Lemma (J. Edmonds) proved in.[2] So for \( c = 2^k \) it exists a \( O(n^{1.5}) \) algorithm pour to find a path \( x-y \) path in a \( C \)-edge colored graph.

**Theorem 2.3**

For each \( c \)-colored edge graph \( G \), \( c = 3 \), that possesses two sub-graphs, whose one is monochromatic with a perfect matching and the other is \( (c-1) \)-edge colored with a simple matching, it exists a \( O(n^{2,5}) \) algorithm to find a \( x-y \)-path within \( G \).

This theorem allows resolving the problem for odd \( c \) while using the technique of reduction colors to obtain at end \( c = 3 \).

In conclusion we just show that for all \( c \in \mathbb{N} \), it exists a \( O(n^{2,5}) \) algorithm to find a \( x-y \)-path in a \( C \)-edge colored graph. Such an algorithm for \( c = 2 \) is described in[1]. We can generalize algorithm 1 in[1] for \( c > 2 \) while using in one hand the theory of Ramsey to decompose the graph \( G \) in monochromatic and polychromatic sub-graphs. And creating the graph \( H \) as above. These modifications will allow generalizing the algorithm 1 described in[1] and to extend it to \( c > 2 \).

**Alternating cycles of length \( K \) (k-DAC)**

In the section 2.1 we already show that an alternating path of length \( k \) can, in a complete graph, be found in \( O(n^3) \) through a dynamic algorithm. Such a path, we call it a dynamic alternating path of length \( k \) and denote it \( k\text{-DAP} \). In this section we’ll deal with 2 Propositions to find an alternating cycle of length \( k \) in a complete graph \( G \).

**Proposition 3.1**

Based on Lemma 1.1 if a complete 2-graph \( G \) contains...
an alternating path of length k, this can also contain an
alternating cycle and it exists a O(|E|) and/or O(n^{2.5})
algorithm to find it.

**Proposition 3.2**

Consider a complete graph G, a vertex s within G and, a
given integer k. If it exists in G a k-DAP with start vertex s,
so it exists in G a k-DAC with start vertex s too if and only if:
- the adjacent vertex, s_a, to the destination vertex within
  the DAP has color degree at least 3 and
- if it exists an edge (s, s_a) and C(s, s_a) and C(x, s_a)
differ in color. x is the neighbour of
s_a and x is not the destination's vertex.

**Proposition 3.3**

As corollary to the Proposition 3.2, it exists a O(n^3)
dynamic algorithm to find a k-DAC in a complete graph.

**Proof**

We prove above that it exists a dynamic algorithm to find
a k-DAP. To find a k-DAC, it suffices to find an alternating
path of length k between two non distinct vertices in a
graph. The start and the destination are same.□

**Finding alternating paths and cycles of length K, K = log(n)**

In this session we’ll present one of matrix operations based
algorithm, which is able to enumerate all the paths indeed
cycle of length k, for k = log(n), n the order of the graph.
We prove that finding all alternating paths existing in a
graph for k >log(n) and big n can explode. The complexity
will be O(n^\delta*(n-2)), where \delta < 2,376 and \delta = (n-3)*(n-4)*
... (n-k). This problem is NP-complete and since it can be
reduced to the hamiltonian cycle and/or to the travelling
salesman problem. In\cite{3,4} it is proved that for k = log(n) this
problem has a polynomial time solution. We propose an
algorithm near to the method of “color-coding” used in\cite{5}
without going through the “Color Coding”, then our graph
is already colored and the “Color –coding” is an exponential
operation, see.\cite{6} Our algorithm works on the basis of the
matrix operations and respecting the principle of “color-
coding” in the path or cycle. Successive edges are differ in
color and using the boolean operators (V, Λ) instead of (*, +).

A problem remains open: for which maximum k superior
than log(n) the algorithm remains polynomial and the path
remains vertex and edge disjoint, i.e. the vertices remain
disjoint, knowing that the smallest value of k is log(n)/
log(log(n))?

For an integer k and log(n)/log(log(n)) ≤ k ≤ log(n) we know
that the problem remains polynomial, but for k>log(n) the
problem will be in NP-hard, then we have a complexity of
O(n^{\delta}*8^{(n-2)}), where \delta < 2,376 and \delta = (n-3)*(n-4)*
... (n-k). And k is part of the input. In next we’ll present first our on

matrix operations based algorithm and after a technique,
that allows finding all the alternating hamiltonian paths
existing in the graph. Note that finding hamiltonian paths
and/or cycles in a graph is NP-complete.

**Principle and functioning of algorithm 2**

**Construction of a (n x n)-matrix associated to the graph G**

Let G = (V, E) be a graph. We construct the associated
matrix as follows:

**Finding paths and/or cycles**

Finding a paths and/or a cycle of length k is done using a
similar method to the matrix multiplication. Here we will
use the operations of Boole as follows:

Let R be a boolean operation and C a (n x n)-matrix, M the
(n x n)-associated matrix to our graph. To begin we initialize
the matrix C as follows:

\[ C_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } C_{ii} = 1 \\ 0 & \text{otherwise} \end{cases} \]

R is defined as follows: C_{ij} = V_{k=1}^{n} M_{kj}

Note that this operation is dynamic recursive, i. e. that an
obtained result will be used to find a next result in reusing
the function itself. The operation OR is OR-operation and \Lambda
a concatenation defined as follows:

\[ 0 \Lambda 0 = 0 \]

\[ 0 \Lambda x = 0 \]

\[ x \Lambda 0 = x \]

\[ x \Lambda y = P_{xy} \text{ is a path with } xy \text{ as last edge.} \]

\[ C(x) \neq C(y) \text{ if by joining } y \text{ to the path and}
\]

the path won’t be more alternating.

**Remarks**

The matrix M^k represents the set of the paths and/or cycles
of length k. This operation is expected in best case in O(n^3)
and in worst case in O(n^{n}*[(n-1) ![l/(n-1-k)!]). We notice that
more when k approaches (n-1) the operation explodes. And for k = log(n) the complexity remains around O(n^3).

**Method to obtain the list of all the paths and cycles of length k, with log(n) ≤ k ≤ n**

We noticed that for a k ≤ log(n) our problem remains in
polynomial time. How the problem behaves when k>log(n)?
It’s too remaining in polynomial time?
Here we present a method that being able helps us to resolve the put problem in polynomial time, but costly.

Let $G = (V, E)$ be a graph and $k$ an integer. Find the set of all the alternating paths and cycles of length exactly $k$ in $G$ and so that $k$ is far near $n$ or equal to $n$ brings us previously to find the set of the minors of $G$ (this operation is expected in $O(n^3)$, Robertson and Seymour proved it in [10]). We look only for those minors where we eliminate some edges from $G$ and no the vertices, those will be concatenated. It is so important to note that each minor does not have more than $\log(n)$ vertices. The minors are linked up between them by the initial eliminated edges of $G$. Now that we have our sub-graphs of $G$ we’ll apply our matrix algorithm to find in each of the sub-graphs the set of the alternating paths and cycles of length exactly $k$ in $G$ and so that $k$ is far near $n$ or equal to $n$ brings us previously to find the set of the minors of $G$ (this operation is expected in $O(n^3)$, Robertson and Seymour proved it in [10]).

We look only for those minors where we eliminate some edges from $G$ and no the vertices, those will be concatenated. It is so important to note that each minor does not have more than $\log(n)$ vertices. The minors are linked up between them by the initial eliminated edges of $G$. Now that we have our sub-graphs of $G$ we’ll apply our matrix algorithm to find in each of the sub-graphs the set of the alternating paths and cycles of length exactly $k$ in $G$ and so that $k$ is far near $n$ or equal to $n$ brings us previously to find the set of the minors of $G$ (this operation is expected in $O(n^3)$, Robertson and Seymour proved it in [10]).

We look only for those minors where we eliminate some edges from $G$ and no the vertices, those will be concatenated. It is so important to note that each minor does not have more than $\log(n)$ vertices. The minors are linked up between them by the initial eliminated edges of $G$. Now that we have our sub-graphs of $G$ we’ll apply our matrix algorithm to find in each of the sub-graphs the set of the alternating paths and cycles of length exactly $k$ in $G$ and so that $k$ is far near $n$ or equal to $n$ brings us previously to find the set of the minors of $G$ (this operation is expected in $O(n^3)$, Robertson and Seymour proved it in [10]).

**Algorithm 2**

Enumeration of the alternating paths and cycles of length $k \leq \log(n)$

**Input:** $(n \times n)$-associated Matrix, $M$, of the graph $G$; an integer $k$

**Output:** $(n \times n)$-result matrix that contains the paths and cycles

**Begin**

1. **Initialization**
   
   $P = \emptyset$; //empty path
   
   $C[a(n \times n)]$-Matrix;
   
   int $Stop1$; // loop counter
   
   int $Stop2$; // loop counter
   
   int $Stop3$; // loop counter
   
   int $Stop4$; // loop counter

2. **for** $k \geq 2$
   
   repeat // k-1 times
   
   repeat
   
   if($C_{ij}$ contains more than one Path)
   
   then
   
   repeat
   
   $P = C[i][k]*M[k][j]$;
   
   $C_{ij} = C_{ij} U P$;
   
   Until $stop1 \leftarrow l$; // $l$= length of $C_{ij}$
   
   else

**Complexity and analysis of the algorithm**

Our algorithm behaves as a classical multiplication of $(n \times n)$-matrix by itself. The result of a first multiplication delivers us all path of lengths 2. The diagonal of the result matrix is filled with 0 if it only looks for alternating paths. Otherwise the diagonal contains the cycles of length $k$ within the graph. The Algorithm stops before the length $k$ will be attained if to a given moment a 0 matrix is obtained.

In worst case we’ll have inside each box within the result matrix $(n-1)^{k-1}$ elements $k$ is the current length of the found paths or $k$ is the exponent of the matrix. This here confirms that for $k \leq \log(n)$ the operation remains polynomial. Then for bigger $k$ we have a complexity of $O(n^{3*[(n-1)^{k-1}]})$.

$(s, s_2, \ldots, s_k, t)$-Cycle

In [7] it was proved that under certain necessary and sufficient conditions it exists in a 2-graph $G$ a $(s, t)$-path. A $(s, t)$-cycle, $s, t \in \mathbb{N}$ and $s \leq t$, is a cycle of length $s + t$, composed from $s$ consecutive edges in one color and the remaining $t$ consecutive edges in other color, see [7]. It clear to see that for a graph filling the same sufficient conditions as a 2-graph such a cycle would exist within.

Let $G = (V, E)$ be a $C$-edge colored graph, $C > 2$. Contains $G$ a $(s, s_2, \ldots, s_k, t)$-cycle such as $k$ the number of colors used to color the graph $G$. To which condition does it exist such a cycle in $G$?

A $(s, s_2, \ldots, s_{k+1}, t)$-cycle, $s, s_2, \ldots, s_k, t \in \mathbb{N}$ and $s \leq s_2 \leq \ldots \leq s_{k+1} \leq t$, is a cycle of length $s + s_2 + \ldots + s_{k+1} + t$, composed from $s$ consecutive edges in one color, $s_2$ consecutive edges in another color, such as $s_2, \ldots, s_{k+1}$ and the remaining $t$ consecutive edges in the remaining color.

To prove the existence or non-existence of such cycle we deal with following theorem:

**Theorem (Ramsey)**

For a given integer $C$ and given numbers $n_1, \ldots, n_C$ it exists a number, noted by $R(n_1, \ldots, n_C; C)$, such as if the edges of a complete graph $G$ of order $R(n_1, \ldots, n_C; C)$ are colored with $C$ different colors, then for every integer $i$ between 1 and $C$, $G$ must contain a complete sub-graph of order $n_i$.
whose edges are all colored with the colori. The special case corresponds to $c = 2$ and $n_1 = n_2 = 3$.

Theorem 5.1

$s, s_2, ..., s_k, t \in \mathbb{N}$ and $s \leq s_2 \leq ... \leq s_k \leq t$, if $n$ rather big such as $n \geq f(s)$ a function $f(s)$, then a graph $G$ contains a $(s, s_2, ..., s_k, t)$-cycle, if and only if:

- $G$ contains a monochromatic sub-graph, having in his breast a path of length $t$ in every color,

- if $s, s_2, ..., s_k, t$ are even then $G$ has a bipartite complete coloring

Proof

While applying the theorem of Ramsey, we assume that for a $n$ rather big, $G$ contains a monochromatic sub-graph. Let us suppose this color be $k$. If $s = s_2 = ... = s_k = t = 1$ then we are in the case of an alternating path of length $k-1$ and an alternating cycle of length $k$. This was already shown above.

If $t = 1$ then the condition (i) always is verified. Let $k$ be integer representing the norm of $(s, s_2, ..., s_k, t)$, for odd $k$ let we eliminate the sub-graph $G'$ of $G$ with the path of length $t$. We transform the remaining sub-graph $G''$ of $G$ in complete sub-graphs $G_j$ colored with $s_j$ on $s_j + 1$ vertices and $G_j$ contains a path in color $s_j + 1$. The sub-graphs $G_j$ are 2-graphs with $n'$ vertices and $n' = s_j + 1$. If $n'$ is rather big such as the theorem 2.1 in [7] be applicable then every $G_j$ contains a $(s_j, s_j + 1)$-cycle. These cycles will be after broken at specific points in order to join them one to another, thus the graph $G$ contains a $(s_1, s_2, ..., s_k, t)$-cycle.

Shortest alternating path between two vertices

Problem 3

How to reduce the length of a $s$-$t$ alternating path (k-DAP) instead of a cycle (k-DAC) found in a complete graph $G$ such as they remain alternating path indeed cycle with initial extremities?

In this part of this paper we’ll present necessary and sufficient conditions that a path, within a graph and the graph must fill, so that the initial length, of the path to a minimal length with same extremities, can be reduce and the path remains alternating.

Reductions of the length of an alternating path between two vertices

Consider a complete graph $G = (V, E)$ and an alternating path of length $k$ found in a complete graph $G$. Which the necessary and sufficient conditions will provide to reduce the length of the path? To answer this question we propose the following lemma:

Lemma 6

Consider a dynamic alternating path $p$ of length $k$ (k-DAP) between two vertices $s$ and $t$ in a graph $G = (V, E)$. We suppose that it exists at least two vertices $x$ and $y$ of $p$, that have at least a “color degree” $> 2$ and are not neighbours in $p$. It exists then another edge and vertex disjoint path $p'$ of length $k_i, k_i < k$, between $s$ and $t$, such as all vertex belonging to the new path $p'$ are included as a whole vertices of $p$.

Theorem 6.1

Consider a complete graph $G = (V, E)$ and a dynamic alternating path $p$ of length $k$ (k-DAP) between two vertices $s$ and $t$ and $p \in G$. Between $s$ and $t$ it exists another edge and vertex disjoint path $p'$ such as $p' \subseteq p$ and $|p'| < |p|$, if and only if it exists

- in $p$ at least two vertices $x$, $y$, who are not neighbours and each has at least a “color degree” $≥ 2$ and a degree $> 2$ (number of neighbours must be superior to 2) if $s$, $t \in \{x, y\}$ it suffices that the “color degree” and the degree be equal to 2

- $x$ and $y$ are neighbours in the graph $G$

- $C(x, y) \neq C(.)$ and $C(x, y) \neq C(.)$ the color of the edge $\{x, y\}$ differ of the one of the edge of $p$, whose an extremity is $x$ and different also of the edge, whose an extremity is $y$.

Proof

$x$ and $y$ are neighbours in $G$ and they belong to the path $p$. If $p$ is edge and vertex disjoint, so it is clear to see that $p'$ is also edge and vertex disjoint. $x$ and $y$ belong to $p$ and have consequently each a “color-degree” $cd \geq 2$, then to belong to the path $p$, a vertex must have at least color degree 2. One edge in color 1 and another in color 2. Let suppose that $x$ and $y$ are neighbours in $G$ and not in $p$. Then between $x$ and $y$ they exist other vertices belonging to $p$. If it exists then a legal edge/path between $x$ and $y$ without going through the vertices of $p$ between $x$ and $y$ therefore $|p|$ will be reduced. A path is legal only if the successive edges differ in colors. This here is only possible if the condition (i) is filled then we abandon one of the two edges of $x$ on $p$ as one of $y$ linking up to the other vertices of $p$, therefore a third edge in different color to the color of the non eliminated edge is necessary to join $x$ to $y$. To pass directly from $x$ to $y$, the condition (ii) must be filled. At end the edge $(x, y)$ belongs to $p'$ if the condition (iii) is filled.

Theorem 6.2

- Consider a complete graph $G = (V, E)$ and $C$ a dynamic alternating cycle of length $k$ (k-DAC) and $C \in G$. It exists another edge and vertex disjoint cycle $C'$ such as $C' \subset C$ and $|C'| < |C|$ if and only if it exists

- In $C$ at least two vertices $x$, $y$ are not neighbours and each has at least a “color degree” $≥ 2$ and a degree $> 2$ (number of neighbours must be superior to 2)

- $x$ and $y$ are neighbours in the graph $G$

- $C(x, y) \neq C(.)$ and $C(x', y) \neq C(y, .)$ the color of the edge
\{x, y\} different of the one of the edge of p of which an extremity is x and different also of the edge of which an extremity is y.

**Proof**

The proof is similar to the proof of the theorem 6.1.

**Conclusion**

In this paper we deal with alternating paths and cycles of length k between two given vertices s, t in complete C-edge colored graphs. We showed that the k-DAP and k-DAC problems are polynomial. We presented a dynamic algorithm, that finds a k-DAP or k-DAC in $O(n^3)$ time. We also presented a technique to find all the k-DAP and k-DAC in a graph such as $k \leq |V|$ and showed as they can be found in polynomial time. Note that for $k = \log(|v|)$ the problem is polynomial. We generalize for C-edge colored complete graphs, $c > 2$, the $(x-y)$-path and of $(s, t)$-cycles problems proved by Y. Manoussakis for 2-edge colored complete graphs. At end we presented necessary and sufficient conditions to reduce the size of k-DAP and of a k-DAC. The problem below remains open.

**Open problems**

To which minimal length can be reduced a dynamic alternating path of length k (k-DAP), between two given vertices s, t, within a C-edge colored complete graph?

**Acknowledgement**

I thank Pr. Y. Manoussakis of the university of Paris South Orsay (France) and Pr. V. Heun of the university Ludwig Maximilian of Munich (Germany) for their contributions and corrections of this paper. I thank Dr. Martinhon C. and Abouelaouali for their help. Thanks to Prof. Scheideler C. of technical university of Munich (Germany) for its constructive discussion.

**References**


Date of Submission: 2017-04-23
Date of Acceptance: 2017-05-08