

Title	一般化マッチングマッチの困難性の研究
Author(s)	飯降, 雄樹
Citation	
Issue Date	2024-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/18900">http://hdl.handle.net/10119/18900</a>
Rights	
Description	Supervisor: 上原 隆平, 先端科学技術研究科, 修士(情報科学)

修士論文

一般化マッチングマッチの困難性の研究

2230020 飯降 雄樹

主指導教員 上原 隆平

北陸先端科学技術大学院大学  
先端科学技術研究科  
(情報科学)

令和6年3月

## Abstract

We study the computational complexity of Matching-Match, which is a puzzle introduced by Daiso Publishing. The Matching-Match puzzle intended for ages 6 and up, so even children can easily understand it with simple problem settings. Moreover, we can play the Matching-Match puzzle just by looking at its input without reading the instructions. However, some instances of the Matching-Match puzzle are difficult and even adults can't solve them easily.

Study on the computational complexity of puzzles can explain the difficulty of the puzzles. Furthermore, the algorithms used to solve the puzzles can be applied to a variety of industries. While there have been studies of the computational complexity of various puzzles, the Matching-Match puzzle has not been studied. Although the Matching-Match puzzle's problem setting is very simple, it is slightly different from any other puzzle. The Matching-Match puzzle can be modeled as a variant of the graph coloring problem in a natural way. However, the corresponding graph coloring problem is new and there exists no previous work on it. Thus, it is unable to apply previous research's algorithms to the Matching-Match puzzle. The Matching-Match puzzle with quite restricted instance can be solved by using an algorithm that finds perfect matching. However, familiar matching algorithms cannot find feasible solutions almost all instances because they have more restrict than the ordinary perfect matchings. Therefore, we expect that algorithms for the Matching-Match puzzle can contribute to the design of matching algorithms.

The purpose of this research is to clarify the computational complexity of the Matching-Match puzzle. To achieve this, we conducted the following.

First, we formulate a generalized the Matching-Match puzzle. the Matching-Match puzzle's input is a problem card with a pattern drawn on it (target graph) and a set of sticks with colored ends (edge set). the Matching-Match puzzle's output is to arrange the edge set in the same pattern as the target graph. To investigate the computational complexity, we modeled the input and output of the Matching-Match puzzle as a variant of the graph coloring problem.

Second, we show the generalized the Matching-Match puzzle is NP-complete. We prove it by showing a polynomial-time reduction from the 3-Partition problem. We show that the same operation as the 3-Partition problem is required to obtain the solution by designing the instances of the Matching-Match puzzle (target graph and edge set). As a result, the Matching-Match puzzle is NP-complete even if we restrict the instance of target graph to a simple pattern such as paths.

Finally, we show the Matching-Match puzzle can be solved in polynomial time by restrict the instance of the problem. If try out all possible ways to arrange an edge set for target graph, execution time becomes exponential time. Therefore, we designed the algorithm to show, it can solve in polynomial time when appropriate restrict of target graph. If the instance of target graph is a star, a complete graph, a complete  $k$ -partite graph with a constant number of colors, or a spider with leg length 2 with a constant number of colors, which problem can be solved in polynomial time. According to these results, the radius of the graph seems to be related to the complexity of the Matching-Match puzzle.

# 目次

第 1 章 はじめに .....	1
第 2 章 一般化マッチングマッチの定式化.....	4
2.1 定義と用語.....	4
2.2 一般化マッチングマッチの入力と出力について .....	4
2.3 一般化マッチングマッチの定義.....	6
第 3 章 マッチングマッチの計算量的な困難さ .....	10
3.1 入力の頂点色付きグラフがスパイダーの場合 .....	10
3.2 入力の頂点色付きグラフがパスの場合.....	12
3.3 入力の頂点色付きグラフがサイクルの場合.....	14
第 4 章 多項式時間で解けるマッチングマッチ .....	15
4.1 入力の頂点色付きグラフがスターの場合 .....	15
4.2 入力の頂点色付きグラフが完全グラフの場合 .....	16
4.3 入力の頂点色付きグラフが完全 $k$ 部グラフの場合 .....	18
4.4 頂点色付きグラフがスパイダーグラフで,各パスの長さが 2 の場合 .....	21
第 5 章 おわりに .....	24

## 図目次

図 1.1 : マッチングマッチ(大創出版).....	1
図 1.2 : 市販のマッチングマッチの入力と出力 .....	2
図 1.3 : マッチングマッチの入力と出力の例.....	2
図 1.4 : 完全マッチングのアルゴリズムをマッチングマッチに適用した例 .....	3
図 2.1 : マッチングマッチの入力と出力の例.....	5
図 2.2 : いくつかの頂点が色付けされた無向グラフの例.....	6
図 2.3 : 両端が色付けされた辺集合の例.....	7
図 3.1 : 定理 1 の入力で用いる頂点色付きグラフ .....	11
図 3.2 : 定理 1 の入力で用いる両端色付き辺集合 .....	11
図 3.3 : 定理 1 の出力 .....	12
図 3.4 : 定理 2 の入力で用いる頂点色付きグラフ .....	13
図 3.5 : 定理 2 の入力で用いる両端色付き辺集合 .....	13
図 3.6 : 定理 2 の出力 .....	14
図 4.1 : 入力のスターの例(葉の数 5) .....	15
図 4.2 : 入力の完全グラフの例(頂点数 5) .....	16
図 4.3 : 完全グラフの両端色付き辺集合とその数え上げの結果(頂点数 5) .....	17
図 4.4 : 入力の完全 $k$ 部グラフの例(グループ数 3) .....	18
図 4.5 : 定理 6 のアルゴリズムの処理の流れ .....	20
図 4.6 : 定理 7 のアルゴリズムの処理の流れ .....	23

## 表目次

表 2.1: 変数定義一覧 .....	9
---------------------	---

## 第1章 はじめに

パズルの計算量的な困難さとその解法に関する研究は、アルゴリズムの技法の開発と密接に関連している。パズルの難易度を定量的に示すことができるだけでなく、パズルを計算機で解くために開発されたアルゴリズムは、現実社会の様々な領域に適用されている[1]。またその反対に、現実社会で使われているアルゴリズムを NP 完全問題として知られているパズルの解法に適用できるケースも多い[2]。本研究では、計算量的な困難さについて研究されていないパズルであるマッチングマッチを研究対象とする。

マッチングマッチ[3]は、パズル作者のりかちが制作したパズルで、株式会社大創出版より販売されている。マッチングマッチは、ルールが直感的で子供でも理解が容易である一方、問題によっては、大人でもすぐには解けない。子供から大人まで楽しめる難易度のパズルとなっている(図 1.1)。

市販のマッチングマッチの入力と出力の例を図 1.2 に示す。マッチングマッチは模様が描かれた問題カードと、マッチと呼ぶ両端に色がついた棒が入力として与えられる。パズルの解は、問題カードと同じ模様を、与えられたマッチで作ることである。問題カードの模様は頂点と辺から構成されるグラフであり、マッチは問題カードの模様が構成できるように、過不足なく与えられている。解は以下の 2 つのルールを満たさなければならない。問題カードの色がついている頂点には、その色と同じ色のマッチの端がこなければならない。マッチ同士はマッチの端が同じ色でなければ繋げられない。また、問題カードの模様において、白色の頂点はプレイヤーが任意に色を決めてよい。頂点の色を決定するこの操作で色を間違えると、上記のルールを満たすように問題カードと同じ模様は作れない。すべての頂点の色が決まっていれば、その色どおりにマッチを繋げていくだけだが、この操作がマッチングマッチを難しくしている。



図 1.1 : マッチングマッチ(大創出版)

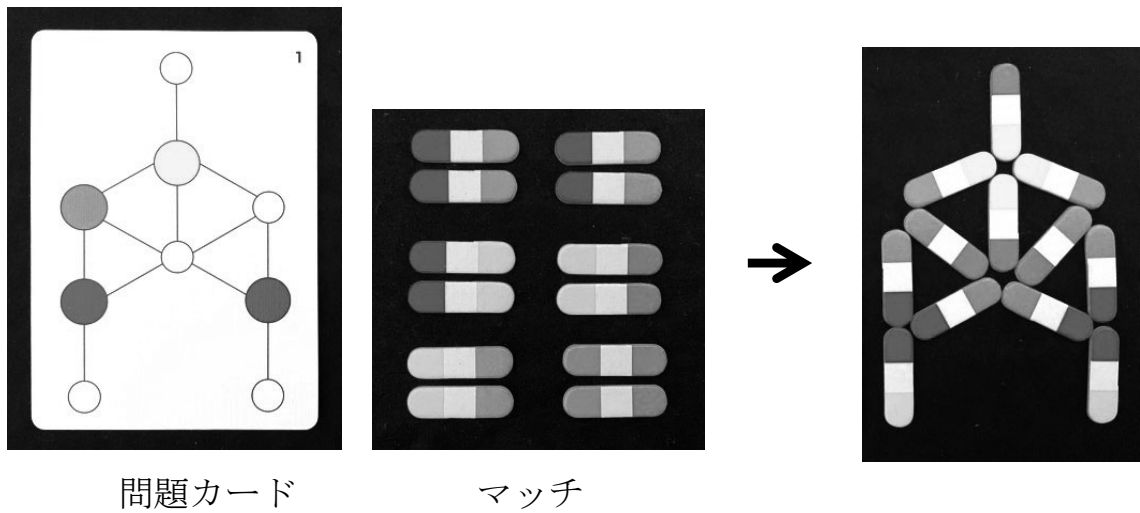


図 1.2 : 市販のマッチングマッチの入力と出力

マッチングマッチは問題カードの辺とマッチを頂点の色が一致するように 1 対 1 で対応させていくパズルと考えられる. そのため, よく知られているアルゴリズムの完全マッチングを辺同士の対応に適用すれば解けるように思われる.

しかしながら, 図 1.3 のような単純な形の問題カードの場合でも, 得られる完全マッチングが正解とは限らないことがある. 図 1.4 に示すように, マッチングマッチの入力に完全マッチングのアルゴリズムを適用する. ここでは問題を次のように 2 部グラフで表現する. 問題カードの各辺と各マッチをそれぞれの頂点集合として, マッチの頂点の色と問題カードの辺の色に同じものがあれば 2 つの頂点を辺で結ぶ. この 2 部グラフ上で完全マッチングを見つければ, 全てのマッチが問題カードに配置できる. しかし, 図 1.4 に示す完全マッチングでマッチを配置すると, マッチの端の頂点の色が異なり, 不正解であることがわかる. 図 1.3 のような簡単な入力であっても, よく知られたマッチングのアルゴリズムをそのまま適用することはできない.

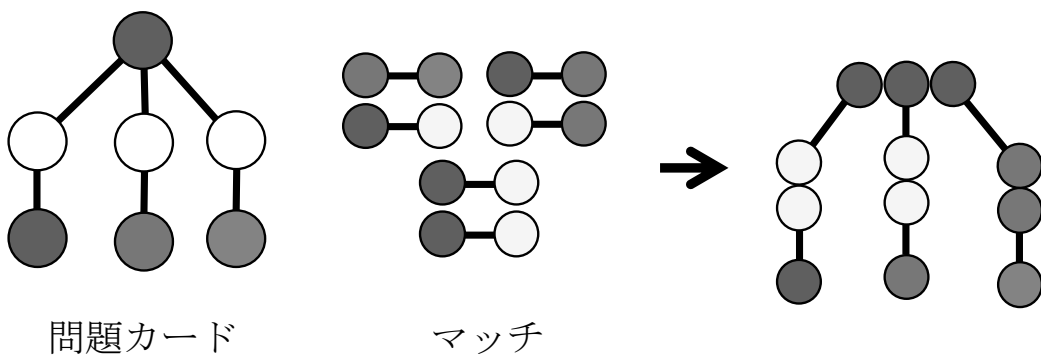


図 1.3 : マッチングマッチの入力と出力の例



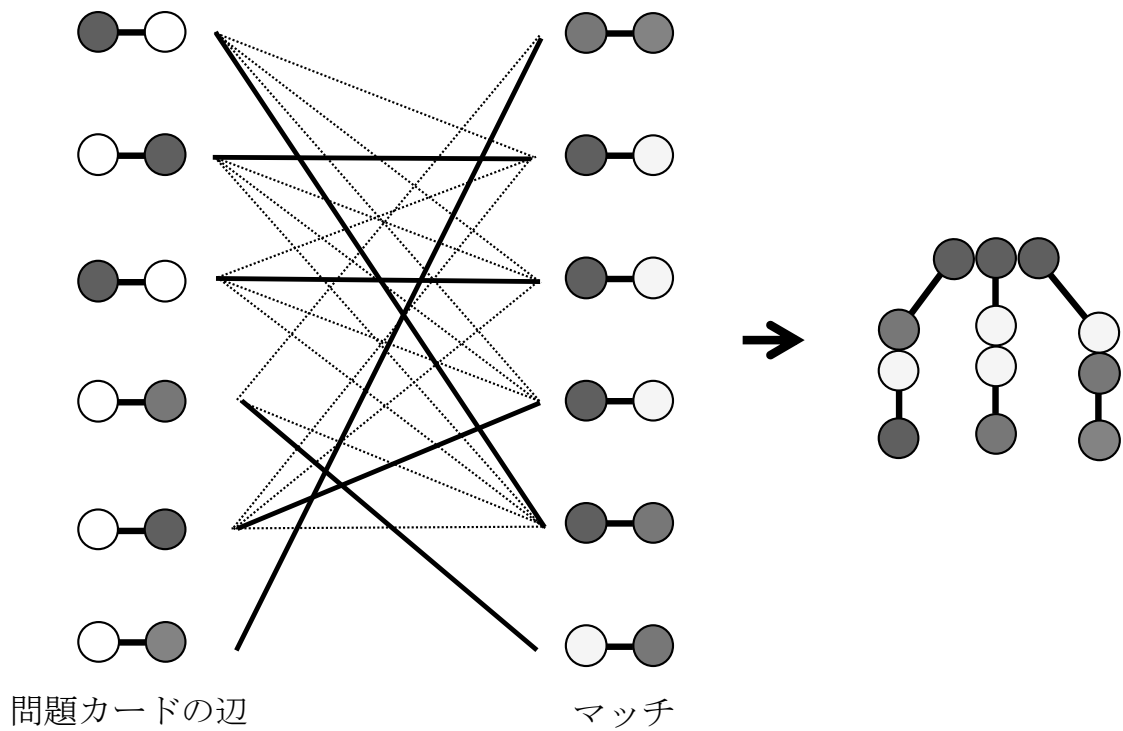


図 1.4：完全マッチングのアルゴリズムをマッチングマッチに適用した例

本研究では、マッチングマッチの計算量的な困難さを明らかにすることを目的とする。マッチングマッチは先行研究が存在しない。そのため、まず一般化マッチングマッチを定式化する。次に一般化マッチングマッチが NP 完全問題であることを証明する。最後に入力に制限を与えれば、多項式時間で解けることを示す。

## 第2章 一般化マッチングマッチの定式化

### 2.1 定義と用語

まず，一般化マッチングマッチの定式化に必要となる定義と用語を導入する。グラフは頂点と呼ばれる要素の集合と辺と呼ばれる要素の集合から相違なる2つを要素とする集合から構成される。辺が向きを持たないグラフを無向グラフ，向きを持つグラフを有向グラフという。本稿では，辺が向きを持たない無向グラフしか扱わないため，たんにグラフは無向グラフを意味している。2つの頂点の間に辺が存在する場合，2つの頂点は隣接しているという。

パスは，頂点集合 $\{v_1, v_2, \dots, v_n\}$ と辺集合 $\{\{v_i, v_{i+1}\} | i = 1, 2, \dots, n-1\}$ からなるグラフとする。また，サイクルはパスに辺 $\{v_n, v_1\}$  ( $n \geq 3$ )を加えたグラフとする[4]。グラフの任意の2つの頂点の間に隣接する頂点をたどってできるパスが存在すると，グラフは連結しているという。本稿では特に記載がない限り，パスおよびサイクルはグラフの種類として定義しており，隣接する頂点をたどってできる部分グラフの種類という意味ではない。木は，グラフが連結で部分グラフとしてのサイクルを含まないグラフとする。スパイダーは，1つの頂点の次数が少なくとも3で，それ以外の頂点の次数が高々2の木とする[5]。次数が少なくとも3の頂点を他の頂点と区別するために中心と呼ぶ。スターは，中心の頂点につながる部分グラフとしての各パスの長さが1のスパイダーとする。以後，スパイダーの各パスの長さを脚の長さと呼ぶ。

### 2.2 一般化マッチングマッチの入力と出力について

一般化マッチングマッチの入力は，模様が描かれた問題カードと両端に色がついた棒の集合の2種類である。問題カードの模様は，頂点と辺から構成されるグラフと捉える事ができる。模様の頂点は，色がついている頂点もあれば，ついていない白色の頂点もある。模様の辺には，向きがない。また，長さに関しては考慮せずに，伸縮が自由な辺とする。そのため，模様の各辺がどの頂点と接続しているかが問題カードの辺に関する情報である。

問題カードの模様を，いくつかの頂点が色付けされた無向グラフと呼ぶ。

両端に色がついた棒は，問題カードの模様の辺の数と同数だけ与えられるものとする。棒は，2つの頂点とそれらを連結する辺と捉える事ができる。棒の両端の頂点には色が必ずついている。また，両端とも同じ色の棒もあれば，異なる色

がついている棒もある。棒には、問題カードの模様と同様に、向きがなく、長さも考慮せずに、伸縮が自由なものとする。両端に色がついた棒の集合を、両端が色付けされた辺集合と呼ぶ。

一般化マッチングマッチの解は、両端に色がついた棒を問題カードの模様と同じ模様となるように並べる事である。ここでいう同じ模様とは、形と色が同じとなる状態をいう。形については、棒の端同士をつなげて、棒の両端を頂点、棒を辺として、グラフとみた時に、問題カードのグラフと同型であるかをいう。つなげた棒の端は1つの頂点として考える(図 2.1)。ただし、棒の端をつけることができるのは同じ色の端同士だけである。色については、問題カードの頂点の色と、棒で作ったグラフの頂点の色が同じである事をいう。問題カードで模様の色がついていない頂点は、任意の色を取る事ができる。

問題カードの模様の頂点に、棒の端が位置するように配置して出力を作る。模様の辺の数と棒の数は同じなので、模様の1つの辺に対して、1つの棒が配置される。市販のマッチングマッチでは、入力に対して、解は必ず1つ存在するように設計されている。本研究では、解の有無やその数は、非自明な入力を与えられるとする。

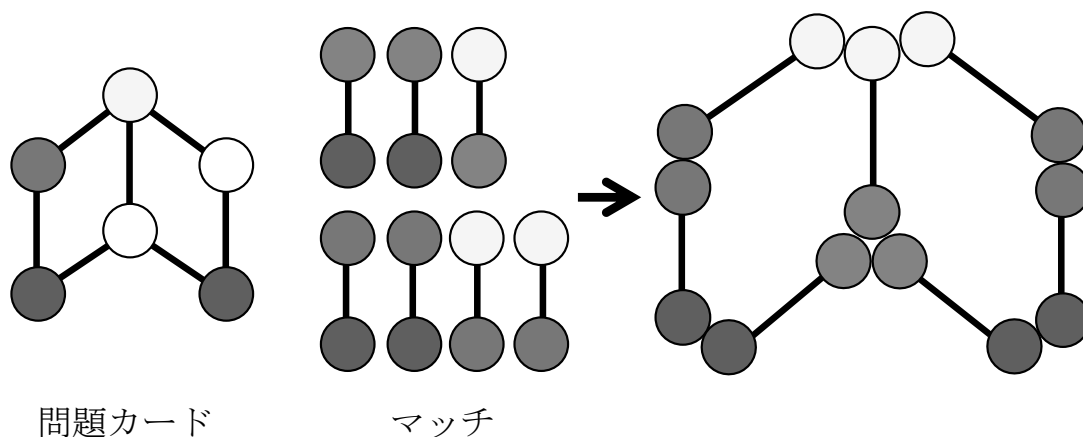


図 2.1 : マッチングマッチの入力と出力の例

## 2.3 一般化マッチングマッチの定義

一般化マッチングマッチの入力を定義する. いくつかの頂点が色付けされた無向グラフを  $G = (V_G, E_G, C^*, c)$  とする.  $V_G$  は無向グラフ  $G$  の頂点集合,  $E_G$  は無向グラフ  $G$  の辺集合  $E_G \subseteq V_G \times V_G$ ,  $C^*$  は色の集合  $C$  に任意の色とする事ができる色\* (アスタリスク記号) を加えた集合,  $c$  は頂点と色を対応づけるラベリング関数  $c: V_G \rightarrow C^*$  である. 色の数は色の集合  $C$  を用いて  $|C|$  と表記する. 本稿で\* (アスタリスク記号) は白色の頂点とする (図 2.2).

$$G = (V_G, E_G, C^*, c)$$

$$C^* = \{\text{赤, 緑, 黄, 青, *}\}$$

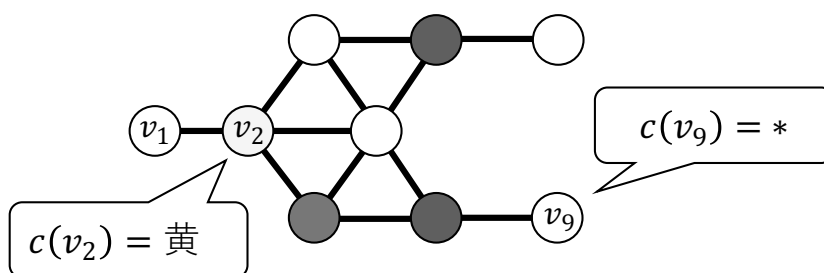


図 2.2 : いくつかの頂点が色付けされた無向グラフの例

両端が色付けされた辺集合を  $E_H = \{h_1, h_2, \dots, h_i\}$ ,  $h_i = \{u_{i,1}, u_{i,2}\}$  とする.  $V_H$  は辺集合  $E_H$  の頂点集合,  $h_i$  は辺集合  $E_H$  の  $i$  個目の辺であり,  $u_{i,1}$  と  $u_{i,2}$  は辺  $h_i$  の両端の頂点である. 色の集合  $C$  は無向グラフ  $G$  と同じである.  $c$  は無向グラフと同様に, 頂点と色を対応づけるラベリング関数  $c: V_H \rightarrow C$  である (図 2.3).

$$E_H = \{h_1, h_2, \dots, h_i\}$$

$$h_i = \{u_{i,1}, u_{i,2}\}$$

$$V_H = \{u_{1,1}, u_{1,2}, u_{2,1}, u_{2,2}, \dots, u_{i,1}, u_{i,2}\}$$

$$C = \{\text{赤}, \text{緑}, \text{黄}, \text{青}\}$$

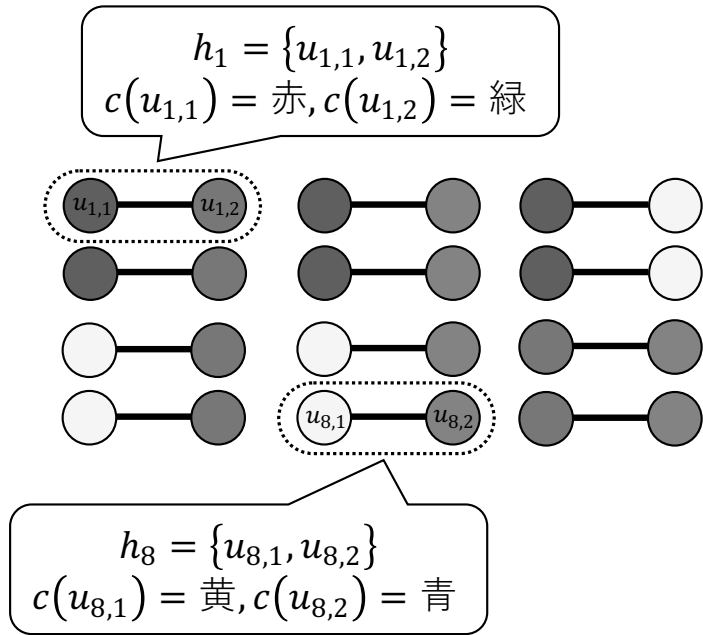


図 2.3 : 両端が色付けされた辺集合の例

以後、いくつかの頂点が色付けされた無向グラフを頂点色付きグラフ、両端が色付けされた辺集合を両端色付き辺集合と呼ぶ。

一般化マッチングマッチの解を定義する。

解は両端色付き辺集合の頂点  $V_H$  から頂点色付きグラフの頂点  $V_G$  への写像  $M: V_H \rightarrow V_G$  とする。この写像  $M$  に 4 つの制約を設けることで、一般化マッチングマッチの解の条件を満足させる。

$V_H$  の頂点は必ず、 $V_G$  の頂点のいずれかとなるので、頂点については全射の制約を数式(1)で定義する。

$$\forall v \in V_G, v = M[u] \text{ for some } u \in V_H \tag{1}$$

$E_H$ の1つの辺は $E_G$ の1つの辺と対応するため、辺については全単射の制約を数式(2)で定義する.

$$\forall (u_1, u_2) \in E_H, (M[u_1], M[u_2]) \in E_G \quad (2)$$

$V_H$ の頂点の色は $V_G$ の頂点の色と同じとなるので、ラベリング関数で出力される頂点の色が一致である制約を数式(3)で定義する.

$$\begin{aligned} \forall (u_1, u_2) \in E_H, c(u_1) = c(M[u_1]) \text{ または } c(u_1) = *, \\ c(u_2) = c(M[u_2]) \text{ または } c(u_2) = * \end{aligned} \quad (3)$$

任意の色とする事ができる色の頂点については、数式(3)に加えて数式(4)の制約を定義する. 数式(3)だけでは、両端色付き辺集合は端が同じ色でなくとも繋げることができてしまうため数式(4)の制約を定義する.

$$\begin{aligned} \forall c(v) = * \text{ for } v \in V_G, c(M[u]) = c(M[u']) \\ \text{for any pair } \{u, u'\} \text{ s.t. } M[u] = M[u'] = v \end{aligned} \quad (4)$$

一般化マッチングマッチは、頂点色付きグラフ $G$ と両端色付き辺集合 $E_H$ が与えられたとき、上記の制約を満たすマッチング $M$ を見つける問題と定義する.

表記	定義
$V_G$	グラフ $G$ の頂点集合
$E_G$	グラフ $G$ の辺集合
$C$	色の集合
$C^*$	$C$ に*を加えた集合
$c(v)$	頂点 $v$ の色
$E_H$	両端が色付けされた辺集合
$V_H$	辺集合 $E_H$ の頂点集合
$h_i$	辺集合 $E_H$ の $i$ 個目の辺
$u_{i,1}, u_{i,2}$	$h_i$ の端点
*	ワイルドカード(自由な値)

表 2.1: 変数定義一覧

## 第3章 マッチングマッチの計算量的な困難さ

本章では、一般化マッチングマッチの NP 完全性を示す。マッチングマッチは明らかに NP に属するので、NP 完全性を示すには NP 困難性を示せばよい。

還元には以下の 3-Partition 問題を用いる：

入力： $3m$ 個の正の整数集合  $A = \{a_1, a_2, \dots, a_{3m}\}$  ただしここで、

$B = (a_1 + a_2 + \dots + a_{3m})/m$  とすると、 $B/4 < a_i < B/2$  を満たす。

出力：3 組の集合を  $m$  個作り、組ごとの 3 つの要素の合計を同じ値  $B$  にできるかどうか。

3-Partition 問題は強 NP 完全問題であることが知られている [6]。

簡単な 3-Partition 問題の具体例として、 $A = \{19, 20, 22, 23, 25, 27, 40, 45, 49\}$  が与えられた時を考える。これは実現可能な入力であり、出力の 3 組は  $\{19, 22, 49\}$ ,  $\{20, 25, 45\}$ ,  $\{23, 27, 40\}$  である。そのいずれの組も和が同じ値  $B = 90$  となっていることが確認できる。

### 3.1 入力の頂点色付きグラフがスパイダーの場合

[定理 1]

一般化マッチングマッチは、 $G$  をスパイダーに限定しても NP 完全問題である。

[証明]

上記で示した 3-Partition 問題からの多項式時間還元を示すことで証明する。マッチングマッチの入力に下図に示す頂点色付きグラフを与える(図 3.1)。



$$G = (V_G, E_G, C^*, c)$$

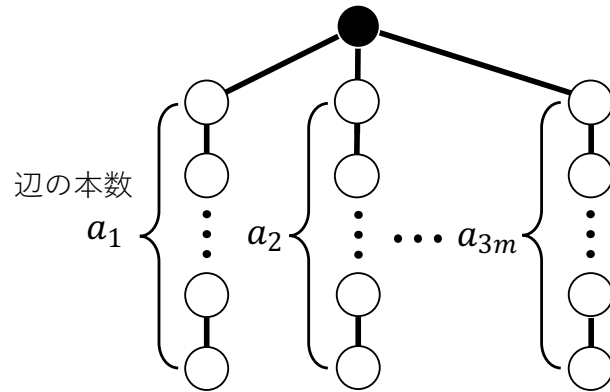


図 3.1 : 定理 1 の入力で用いる頂点色付きグラフ

入力の頂点色付きグラフは中心に次数  $3m$  の頂点があり, その周りにはパスがつながるスパイダーである. それぞれのパスの辺の本数は 3-Partition 問題の  $A = \{a_1, a_2, \dots, a_{3m}\}$  のそれぞれの  $a_i$  と対応させる. 頂点の色は中心の頂点を黒色で色付けし, 他の頂点は白色とする.

両端色付き辺集合は, 下図に示す辺を与える(図 3.2). 片端が中心の頂点と同じ色でもう一方が  $m$  色の内の 1 色である辺を各色 3 本, 両端が同じ色で  $m$  色の内の 1 色である辺を各色  $B$  本とする.

$$E_H = \{h_1, h_2, \dots, h_i\}$$

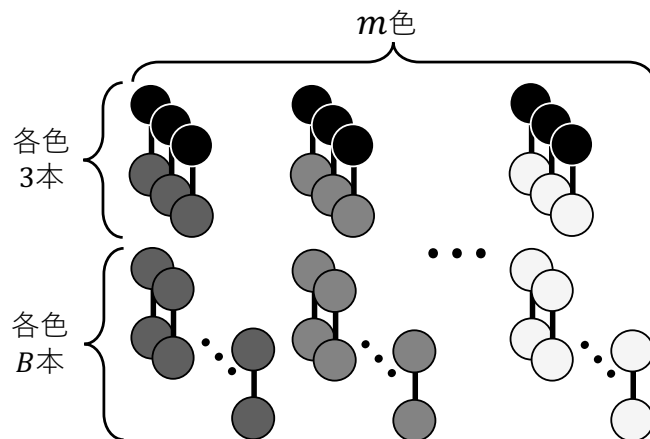


図 3.2 : 定理 1 の入力で用いる両端色付き辺集合

上記の入力を与えて、元の 3-Partition 問題が解をもつ必要十分条件は、還元で構成したマッチングマッチが解をもつことを示す。

まず、3-Partition 問題が解をもつとき、3つ組の集合を  $m$  個作り、組ごとの3つの要素の合計が同じ値  $B$  となるような分割が存在する。

この分割を用いて、3-Partition 問題の3つ組のそれぞれの要素と、マッチングマッチのパスの辺の本数が同じとなる3本のパスを同じ色で色付けする。

この操作をすべての色に対して行くと、マッチングが得られる(図 3.3)。

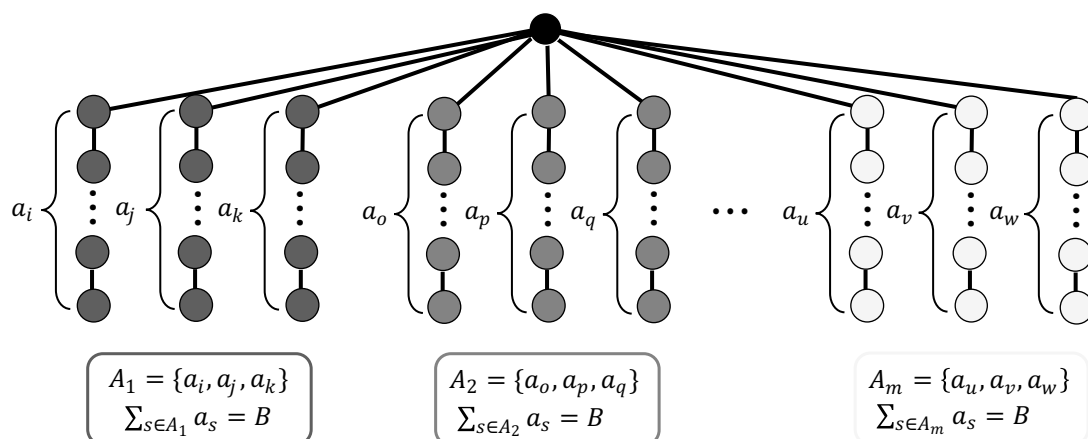


図 3.3 : 定理 1 の出力

次に、構成したマッチングマッチが解をもつとき、元の 3-Partition 問題も解をもつことを示す。マッチングマッチのマッチングが得られたとき、 $B/4 < a_i < B/2$  より、同じ色のパスが3本でき、その辺の数が 3-Partition 問題での3つ組のそれぞれの要素となる。したがって、3-Partition 問題が解をもつ必要十分条件はマッチングマッチのマッチングを見つけることである。

よって、一般化マッチングマッチは NP 完全問題である。 □

### 3.2 入力の頂点色付きグラフがパスの場合

問題の入力の頂点色付きグラフがスパイダーでも NP 完全であることがわかったが、グラフをさらに制限してもマッチングマッチは NP 完全であることを示す。

[定理 2]

マッチングマッチは入力の頂点色付きグラフをパスに制限しても NP 完全問題である。

[証明]

定理 1 の証明において、入力で用意した頂点色付きグラフと両端色付き辺集合を下図のように変更する(図 3.4). 入力に与える頂点色付きグラフはパスであり、頂点の色は色の集合のうち任意の 1 色をとる白色の頂点を黒色の頂点で挟み込むように色付けする. この際に、それぞれの挟み込む辺の本数を 3-Partition 問題の入力の  $A = \{a_1, a_2, \dots, a_{3m}\}$  のそれぞれの  $a_i$  とする.

$$G = (V_G, E_G, C^*, c)$$

辺の本数

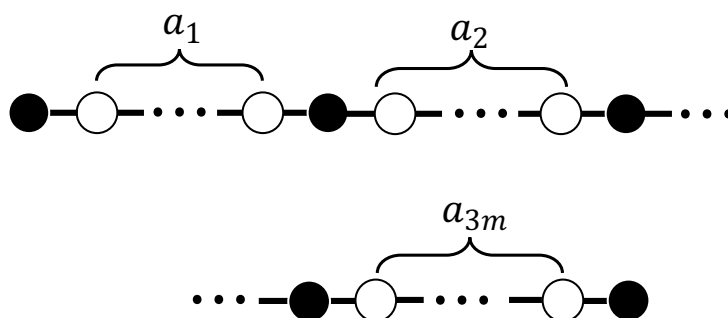


図 3.4 : 定理 2 の入力で用いる頂点色付きグラフ

両端色付き辺集合については下図のように、片端が中心の頂点と同じ色もう一方が  $m$  色の中の 1 色である辺を各色 6 本、両端が同じ色で  $m$  色の中の 1 色である辺を各色  $B$  本とする(図 3.5).

$$E_H = \{h_1, h_2, \dots, h_i\}$$

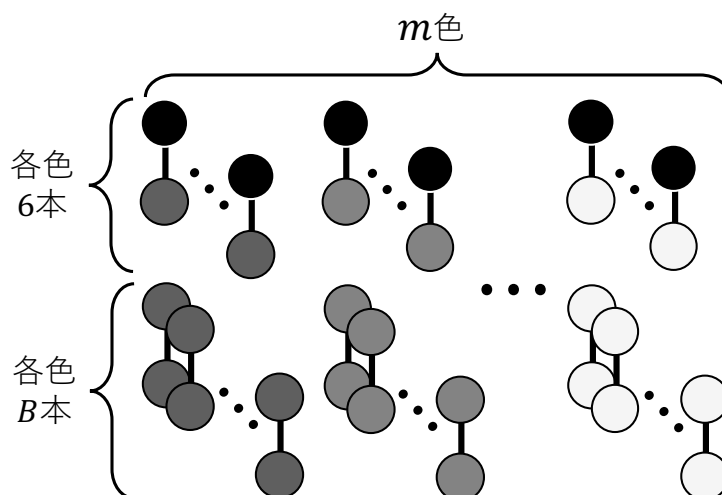


図 3.5 : 定理 2 の入力で用いる両端色付き辺集合

上記のようにマッチングマッチの入力を構成することで、入力がパスの場合でも定理 1 の証明と同様に、元の 3-Partition 問題が解をもつ必要十分条件は、還元で構成したマッチングマッチが解をもつことを示せる(図 3.6).  $\square$

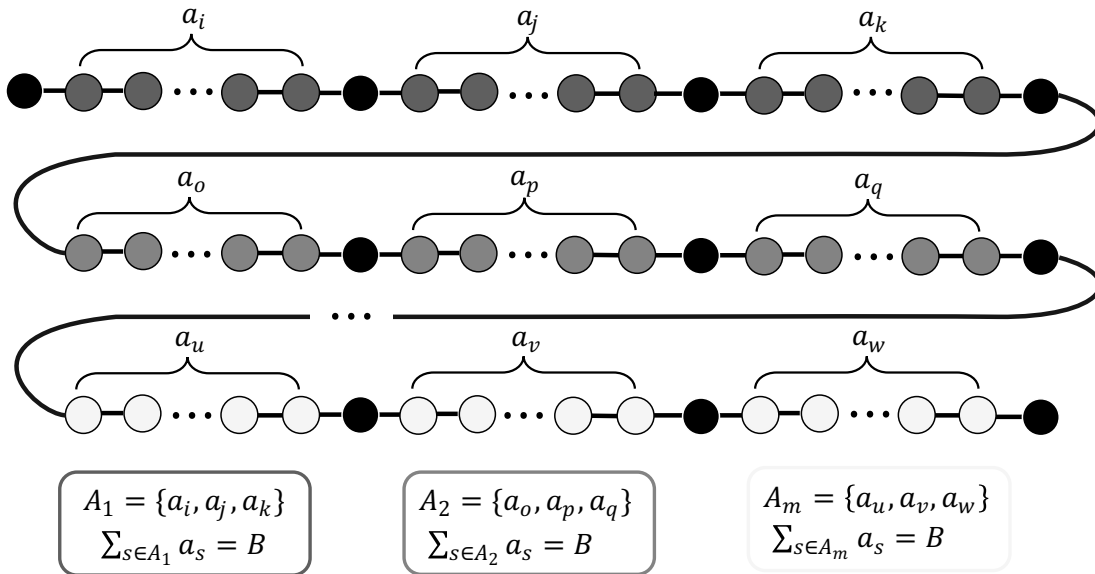


図 3.6 : 定理 2 の出力

### 3.3 入力の頂点色付きグラフがサイクルの場合

入力がサイクルの場合もパスと同様に困難さが証明できる。

[定理 3]

マッチングマッチは入力の頂点色付きグラフをサイクルに制限しても NP 完全問題である。

[証明]

定理 2 の入力において、パスの両端の黒色の頂点を同一視すれば、グラフはサイクルとなる。困難さは定理 2 の証明と同様に示せる。  $\square$

## 第4章 多項式時間で解けるマッチングマッチ

本章では，多項式時間で解けるマッチングマッチの入力とその解法を示す．

### 4.1 入力の頂点色付きグラフがスターの場合

[定理 4]

入力の頂点色付きグラフをスターに制限した場合，マッチングマッチは線形時間で解ける(図 4.1)．

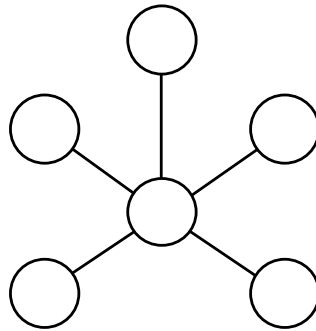


図 4.1：入力のスターの例(葉の数 5)

[証明]

入力のスターの頂点数を $n$ とする．入力をスターに制限した場合は，両端色付き辺集合のどの辺をスターのどこの葉に配置するか，また，どの向きで配置するかを適切に選択しないといけない．スターのいずれの辺も中心の頂点と接続しているため，両端色付き辺集合の辺の少なくとも片端は中心の頂点と同じ色となっていることに着目して，アルゴリズムを考える．

まず，中心の頂点の色が決まっている場合を考える．中心の頂点の色が既知のため，両端色付き辺集合の辺を配置する向きは，中心の頂点と同じ色の方端が中心を向くように配置すれば良い．また，どの葉にどの辺を配置するかは，色のついている葉から順次配置していき，色のついていない葉を最後に選ぶようにする．これは，色のついていない葉を先に選ぶと，色の付いてある葉に配置する配色の辺がなくなる可能性が生じるためである．両端色付き辺集合に，両端の色となる辺がなければ，マッチングがなく，解は存在しないとわかる．中心の頂点の

色が決まっている場合は、色の付いているスターの葉から順次、両端の色が同じとなるように、両端色付き辺集合から辺を選んでいけばよく、実行時間は $O(n)$ 時間となる。

次に、中心の頂点の色が決まっていない場合を考える。中心の頂点の色が決まっていない場合では、初めに中心の頂点の色を仮定する。入力がスターなので中心の頂点の色は、両端色付き辺集合の中から任意の 1 辺を選び、その辺の両端のどちらかの色となる。両端色付き辺集合の辺の少なくとも片端は中心の頂点と同じ色となっているため、中心の頂点の色の候補は 1 つの辺の両端の色の 2 色となる。中心の頂点の色を仮定した後の処理は、上記の色が決まっている場合と同じ操作を行えばよい。最初の仮定が間違いで頂点の色が異なる場合は、頂点の色の不一致で辺を配置できなくなる。その場合は、中心の頂点の色の候補のもう一方で同じ操作を行う。中心の頂点の色が決まっていない場合のチェックの回数は中心の頂点の色の仮定が誤りの場合でも高々 $2n$ 回でよい。入力の頂点色付きグラフをスターに制限した場合、線形時間 $O(n)$ で解ける。 □

## 4.2 入力の頂点色付きグラフが完全グラフの場合

[定理 5]

入力の頂点色付きグラフを頂点数 $n$ の完全グラフに制限した場合、マッチングマッチは多項式時間 $O(n^2)$ で解ける(図 4.2)。

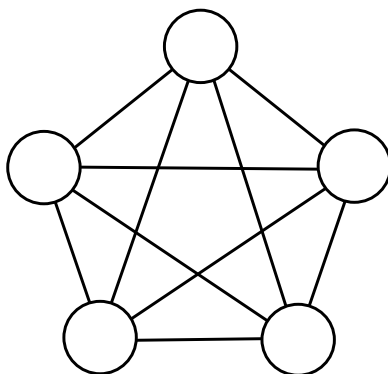


図 4.2 : 入力の完全グラフの例(頂点数 5)

[証明]

完全グラフは、全ての頂点の次数が同じで、どの2頂点も隣接している性質があるため、両端色付き辺集合の両端の頂点の色を数え上げると、完全グラフの頂点の色が一意に決定できる。これは両端色付き辺集合の両端の色を数えて、同じ色で集めると頂点の次数個の組が、頂点の数だけできるからである(図4.3)。

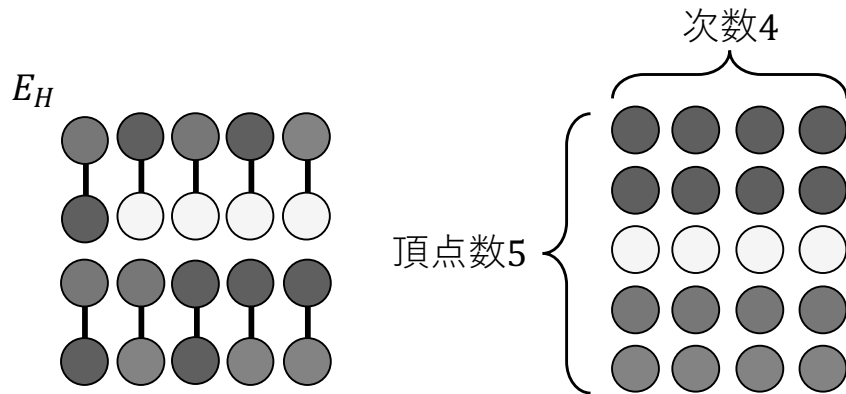


図 4.3 : 完全グラフの両端色付き辺集合とその数え上げの結果(頂点数 5)

両端色付き辺集合の両端の頂点の色を数え上げて上記のようにならなければ、与えられた両端色付き辺集合から完全グラフにはできないことがわかる。数え上げを行い、得られた頂点の色で完全グラフの頂点を色付けする。頂点の色が全て分かれば、あとはその色通りに辺を選び、向きを合わせて配置していただくとなる。実行時間は頂点の色の数え上げにかかる処理が主であり、完全グラフの辺の数は $n(n-1)/2$ である。したがって、入力の頂点色付きグラフを完全グラフに制限した場合、多項式時間 $O(n^2)$ で解ける。□

### 4.3 入力の頂点色付きグラフが完全 $k$ 部グラフの場合

次に、定理 5 の多項式時間で解くことができた完全グラフを完全 $k$ 部グラフにした場合を考える(図 4.4). ここで、完全 $k$ 部グラフの部分集合 $\{1, 2, \dots, k\}$ をグループと呼ぶ. 完全 $k$ 部グラフの場合も、各グループの頂点の次数が同じであれば、両端色付き辺集合の頂点の色の数え上げで、使用する頂点の色をある程度絞り込むことができる.

しかし、得られた頂点の色をどのように各グループに振り分ければ、与えられた辺集合でマッチングが得られるかという問いが残る. そのため、完全グラフの解法で行ったような、頂点の色を数え上げただけでは、頂点の色を一意に決められない.

[定理 6]

入力の頂点色付きグラフが完全 $k$ 部グラフの場合、一般の $n, |C|$ に対して  $O\left(\frac{|C|n^{k|C|}}{|C|!^k}\right)$  時間で解ける. 色の数 $|C|$ が定数のときは、これは $n$ の多項式時間である.

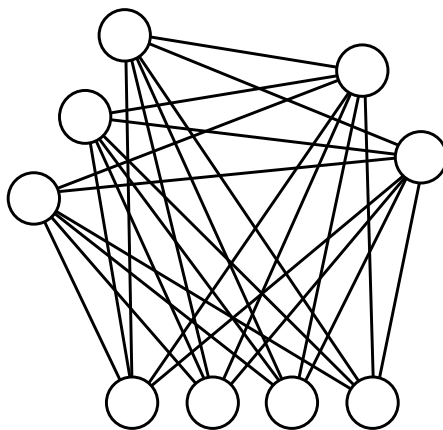


図 4.4 : 入力の完全 $k$ 部グラフの例(グループ数 3)

[証明]

各グループの頂点の数を $n_1, n_2, \dots, n_k$ とする. 完全 $k$ 部グラフの頂点の色の塗り分けを全通り試す方針でマッチングを見つける. 塗り分けは次の 8 つの Step の処理を行い、全通り試す.

図 4.5 に完全 3 部グラフを例に各 Step で塗り分ける頂点を示す. 各 Step の点線の丸で囲った箇所がその Step で塗り分けの対象とする頂点である.



定理 6 のアルゴリズムの 8 つの Step を下記に示す.

- Step1 : 白色の頂点を 1 つ選択し, その頂点の色を仮定する.
- Step2 : Step1 で選択した頂点から出ている辺をグループごとに  
色が塗られている頂点  $S0_1, \dots, S0_{k-1}$  と  
色が塗られていない頂点  $S1_1, \dots, S1_{k-1}$  の 2 種類に分割する.
- Step3 :  $S1$  の頂点をすべて塗り分ける.
- Step4 : Step1 で選択した頂点が属していないグループから  
頂点を 1 つ選択する.
- Step5 : Step4 で選択した頂点から出ている辺の白色の頂点が  
残っているグループを, 色が塗られている頂点  $S2$  と  
色が塗られていない頂点  $S3$  の 2 種類に分割する.
- Step6 :  $S3$  の頂点をすべて塗り分ける.
- Step7 : 1 辺ずつ辺集合と完全  $k$  部グラフの辺をマッチングする
- Step8 : マッチングする辺が辺集合にない場合, Step 0 に戻り,  
試していない塗り分け方で再度実行する.

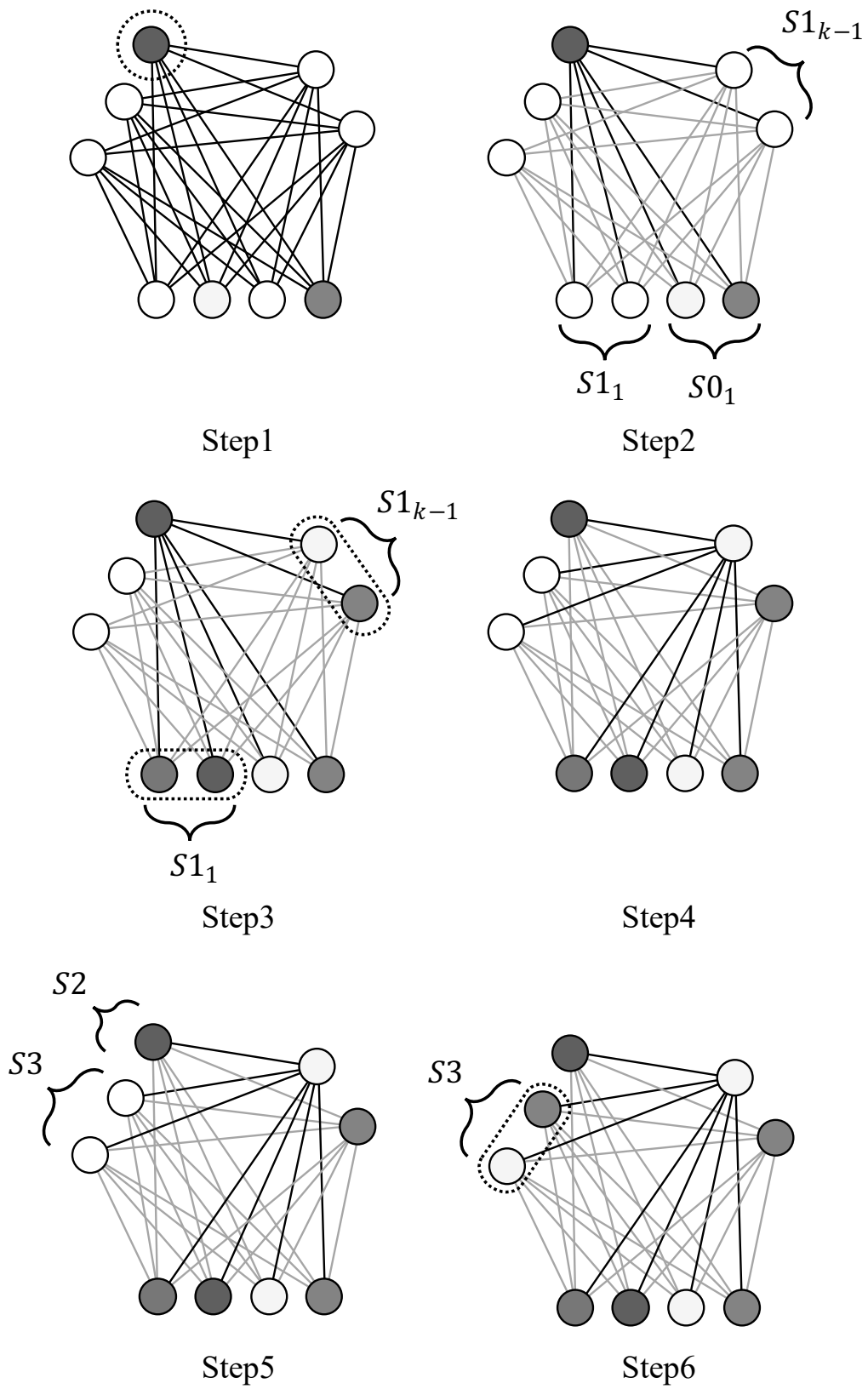


図 4.5 : 定理 6 のアルゴリズムの処理の流れ

各ステップの頂点の塗り分け方の場合分けの数は、  
Step1 で $|C|$ 通り、

Step3 で $\frac{(|C|-1+|S_{11}|)!}{(|C|-1)!|S_{11}|!} \times \dots \times \frac{(|C|-1+|S_{1k-1}|)!}{(|C|-1)!|S_{1k-1}|!}$ 通り、

Step6 で $\frac{(|C|-1+|S_3|)!}{(|C|-1)!|S_3|!}$ 通りの塗り分け方ができる。

各ステップの積が塗り分け方の全通りとなるので、実行時間は $O\left(\frac{|C|(|C|+n)^{k(|C|-1)}}{(|C|!)^k}\right)$ である。よって、入力の頂点色付きグラフを完全 $k$ 部グラフに制限した場合、 $O\left(\frac{|C|n^{k|C|}}{|C|!^k}\right)$ 時間で解ける。  $\square$

#### 4.4 頂点色付きグラフがスパイダーグラフで、各パスの長さが2の場合

[定理 7]

入力の頂点色付きグラフをスパイダーグラフで各パスの長さが高々2 に制限した場合、一般の $n, |C|$ に対して $O\left(\frac{|C|(|C|+n)^{|C|^2}}{|C|!^{|C|}}\right)$ 時間で解ける。色の数 $|C|$ が定数のときは、これは $n$ の多項式時間である。

[証明]

パスの数を $n$ とする。スパイダーグラフの頂点の色の塗り分けを全通り試す方針でマッチングを見つける。そのために、次の8つの Step をおこなう。

定理 7 のアルゴリズムの 8 つの Step を下記に示す.

Step1 : 中央の頂点の色を仮定する.

Step2 : Step1 で選択した頂点から出ている辺を残った 2 頂点とも色が塗られている頂点  $S_0$ ,

中央の頂点に隣接する頂点の色が塗られている頂点を色ごとに  $S_{1_1}, \dots, S_{1_{|C|}}$ ,

中央の頂点に隣接しない頂点の色が塗られている頂点を色ごとに  $S_{2_1}, \dots, S_{2_{|C|}}$ ,

残った 2 頂点とも色が塗られていない頂点  $S_3$  の 4 種類に分割する.

Step3 :  $S_{1_1}, \dots, S_{1_{|C|}}$  の頂点をすべて塗り分ける.

Step4 :  $S_{2_1}, \dots, S_{2_{|C|}}$  の頂点をすべて塗り分ける.

Step5 :  $S_3$  の中央の頂点に隣接する頂点を塗り分ける.

Step6 :  $S_3$  の中央の頂点に隣接しない頂点を Step5 で

塗り分けた色ごとに  $S_{3_1}, \dots, S_{3_{|C|}}$  分割し, 頂点を塗り分ける.

Step7 : 1 辺ずつ辺集合とスパイダーグラフの辺をマッチングする

Step8 : マッチングする辺が辺集合にない場合, Step 0 に戻り,

試していない塗り分け方で再度実行する.

図 4.6 にスパイダーグラフで各パスの長さが高々 2 かつパスの数が 7 を例に各 Step で塗り分ける頂点を示す.

各ステップの頂点の塗り分け方の場合分けの数は,

Step1 で  $|C|$  通り,

Step3 で  $\frac{(|C|-1+|S_{1_1}|)!}{(|C|-1)!|S_{1_1}|!} \times \dots \times \frac{(|C|-1+|S_{1_{|C|}}|)!}{(|C|-1)!|S_{1_{|C|}}|!}$  通り,

Step4 で  $\frac{(|C|-1+|S_{2_1}|)!}{(|C|-1)!|S_{2_1}|!} \times \dots \times \frac{(|C|-1+|S_{2_{|C|}}|)!}{(|C|-1)!|S_{2_{|C|}}|!}$  通り,

Step5 で  $\frac{(|C|-1+|S_3|)!}{(|C|-1)!|S_3|!}$  通り,

Step6 で  $\frac{(|C|-1+|S_{3_1}|)!}{(|C|-1)!|S_{3_1}|!} \times \dots \times \frac{(|C|-1+|S_{3_{|C|}}|)!}{(|C|-1)!|S_{3_{|C|}}|!}$  通りの塗り分け方ができる. 各ステップ

の積が塗り分け方の全通りとなるので, 実行時間は  $O\left(\frac{|C|( |C|+n)^{3|C|( |C|-1)+|C|-1}}{(|C|!)^{3|C|+1}}\right)$  であ

る. よって, 入力の頂点色付きグラフをスパイダーグラフで各パスの長さが高々

2 に制限した場合,  $O\left(\frac{|C|( |C|+n)^{|C|^2}}{|C|!^{|C|}}\right)$  時間で解ける.  $\square$

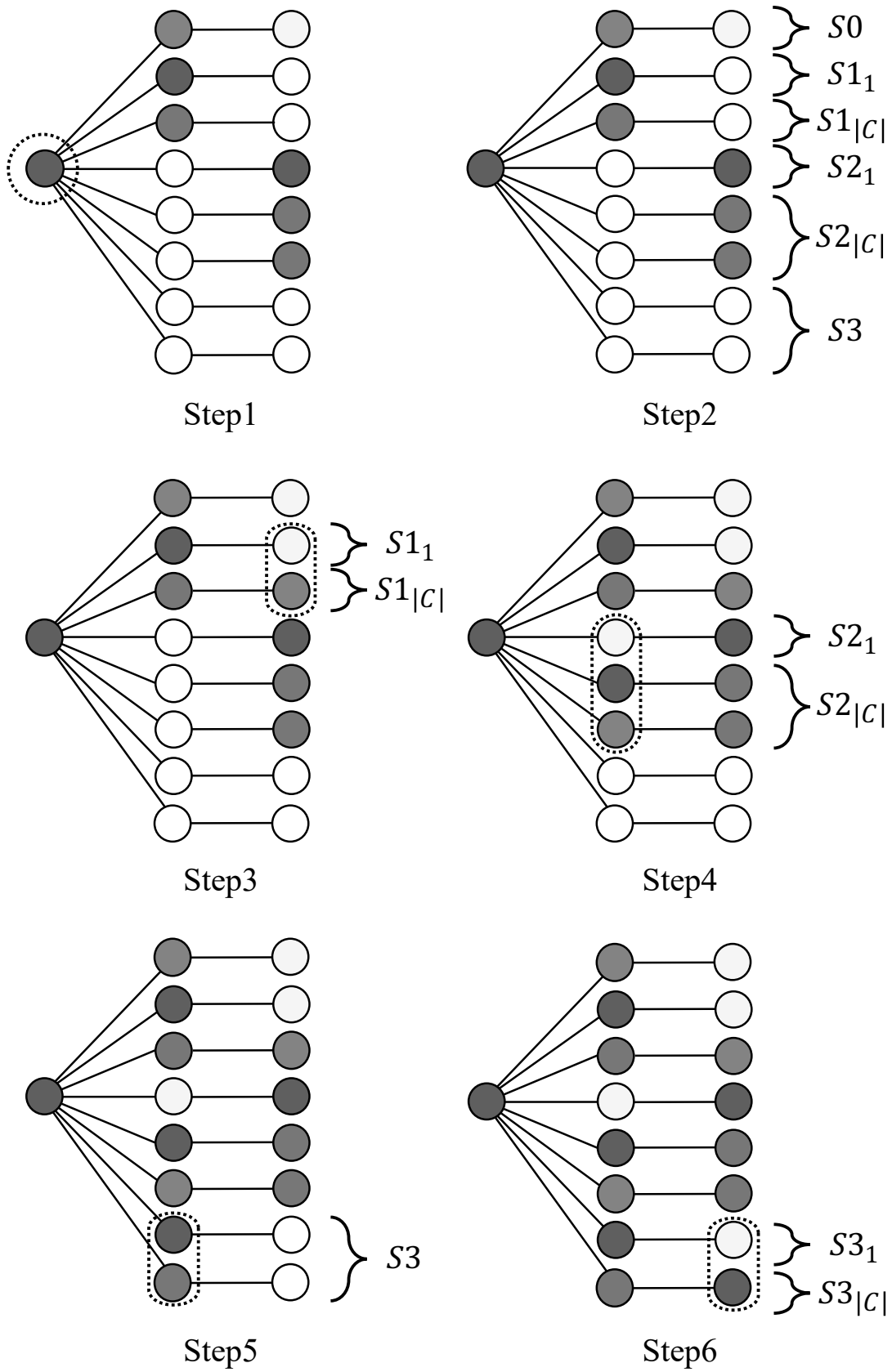


図 4.6 : 定理 7 のアルゴリズムの処理の流れ

## 第5章 おわりに

本研究では、一般化マッチングマッチを定義し、計算量的な困難性について考えた。問題を入力の色付きグラフで分類し、以下を示した。

- ・パスに制限しても NP 完全である。
- ・完全グラフのときは、多項式時間で解ける。
- ・完全 $k$ 部グラフで色の数が定数のときは、多項式時間で解ける。
- ・スパイダーで脚の長さが一般の $B$ のときは NP 完全である。
- ・スパイダーで脚の長さが 1 のスターのときは、多項式時間で解ける。
- ・スパイダーで脚の長さが 2 で色の数が定数のときは、多項式時間で解ける。

結果から入力の色付きグラフの半径が、問題の困難性に関連があると考えられる。グラフの半径は、中心の頂点から任意の頂点への最短距離の最大値のことをいう。スパイダーの脚の長さ $l$ とグラフの半径は同値となる。グラフの半径はそれぞれ、NP 完全であったスパイダーで脚の長さが一般の $B$ のグラフの半径は $B$ 、多項式時間で解ける完全グラフとスターは 1、色の数を定数とすれば、多項式時間で解ける完全 $k$ 部グラフとスパイダーで脚の長さが 2 は 2 である。グラフの半径が小さいと多項式時間で解け、大きいと NP 完全となる傾向がみられた。この傾向を定量的に示すために、以下の未解決問題が提案でき、今後明らかにしていきたい。

### 未解決問題

- ・スパイダーで脚の長さが $k$ 未満のときは多項式時間で解けるが、 $k$ 以上のときは NP 完全になるという性質を持つ定数 $k$ の値はいくつか。

## 参考文献

- [1] Ryuhei Uehara. Computational Complexity of Puzzles and Related Topics, *Interdisciplinary Information Sciences*, Vol. 29, No. 2, pp. 119-140, December, 2023.
- [2] Kendall, G., Parkes, A., and Spoerer, K., A survey of NP-complete puzzles, *ICGA Journal*, 31(1): pp13-34 2008.
- [3] Wolfram MathWorld, Spider Graph,  
<https://mathworld.wolfram.com/SpiderGraph.html>
- [4] 惠羅博, 土屋守正. グラフ理論. 産業図書, 1997.
- [5] 大創出版, 「脳トレチャレンジパズル (マッピングマッチ)」,  
<https://www.daiso-syuppan.com/noutore/>
- [6] Michael R. Garey and David S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness*, W. H. FREEMAN AND Co., 1979.