Vol.2013-AL-143 No.14
2013/3/1

# 木の最小コスト点彩色の列挙と一意性について

木坂 健人[1]　松井 泰子[2,a)]

**概要**：本稿では，木に対する最小コスト点彩色が一意でない場合，それらを重複無く全て列挙するアルゴリズムを提案する．一般グラフの最小コスト点彩色問題は$\mathcal{NP}$困難であるが，1997年，Kroon et. al. は木に対する線形時間アルゴリズムを提案した．しかし，最小コスト点彩色が一意で無い場合，それらを列挙するアルゴリズムはまだ提案されていない．そこで我々は初めて列挙アルゴリズムを提案する．本稿では，与えられた木$T$に対し，最小コスト点彩色を１つ当たり$O(nd^2)$時間で列挙するアルゴリズムを提案する．ただし，$n$は$T$の頂点数，$d$は$T$の最大次数である．また，木の最小コスト点彩色に対して任意の十分大きな色数が与えられた時，その色を全て使用する木の特徴付けを行う．

**キーワード**：列挙, アルゴリズム, 最小コスト点彩色, 木

# Enumeration and Uniqueness for Optimal Cost Vertex Colorings of Trees

Kizaka Kento[1]　Matsui Yasuko[2,a)]

***Abstract:*** In this paper, we propose an algorithm for enumerating all the optimal cost vertex colorings of given trees without repetitions if the optimal cost vertex coloring is not unique. The optimal cost vertex coloring problem is $\mathcal{NP}$-hard for arbitrary graphs. In 1997, Kroon et. al. showed the problem can be solved in a linear time for trees. However, there is no algorithm for enumerating all optimal cost vertex colorings of trees. We first give an enumeration algorithm for the problem. For a given tree, our algorithm can enumerate all optimal vertex colorings in $O(nd^2)$ time for each where $n$ is the number of vertices of $T$, $d$ is the number of maximum degree of $T$. Moreover, we characterize trees which use an arbitrary number of colors for optimal cost vertex coloring.

***Keywords:*** enumeration, algorithm, optimal cost vertex coloring, tree

## 1. はじめに

最小コスト点彩色問題は，1987年,SupowitによってＶＬＳＩに関連する話題の１つとして紹介された [6]．最小コスト点彩色問題は一般には$\mathcal{NP}$困難であるが，区間グラフの上では，時間制約のある機械へのジョブスケジューリングと捉えられるため，解法研究は実用の観点から注目されている．1997年にKroon et. al. は，最小コスト点彩色が木に対しては$O(n)$時間で解けることを示した [4]．しかし，実社会で解かねばならない組合せ最適化問題は得られた１つの最適解が必ずしも社会にそのまま適用できる訳では無い．定式化の際に，数式に表しにくい複雑な条件が反映されていない事が多いからである．そのため，条件を満たす解を列挙し，その中から適用可能な解を意思決定者が選択するというアプローチで，近年，列挙アルゴリズムの研究が注目されている [3], [7], [8], [9], [10].

そのような背景から，我々は木に対する最小コスト点彩色が一意でない場合，それらを重複無く全て列挙するアルゴリズムを初めて提案する．

1　東海大学大学院 理学研究科 数理科学専攻
Course of Mathematical Sciences, Graduate School of Science, Tokai University
2　東海大学 理学部 情報数理学科
Department of Mathematical Sciences, School of Science, Tokai University
a)　yasuko@tokai-u.jp

以下に，最小コスト点彩色問題の組合せ最適化問題への定式化を紹介する．グラフ $G=(V,E)$ と，色の集合 $C(|C|=|V|=n)$，コスト関数 $w:C\rightarrow R$ が与えられているとする．頂点 $v\in V$ が色 $c\in C$ に彩色されているかどうかを示す $\{0,1\}$ 変数を $x_{v,c}(v,c=1,2,\cdots,n)$ とする．目的は，彩色コストの総和が最小になるように，グラフ $G$ の全ての頂点を彩色することである．ただし，隣接頂点の色は異なるように彩色する．このとき，最小コスト点彩色問題は次のように定式化できる．本稿では一般性を失わずに色 $c<c'$ について $w(c)<w(c')$ と仮定する．

$$\min \quad \sum_{v\in V}\sum_{c=1}^{n} w(c)x_{v,c}, \tag{1}$$

subject to

$$\sum_{c=1}^{n} x_{v,c}=1, \quad v=1,2,\cdots,n, \tag{2}$$

$$x_{v,c}+x_{v'c}\leq 1, \quad (v,v')\in E; \quad c=1,2,\cdots,n, \tag{3}$$

$$x_{v,c}\in\{0,1\}, \quad v,c=1,2,\cdots,n. \tag{4}$$

上記の組合せ最適化問題の目的関数 (1) は，頂点の彩色コストの総和最小化を表す．次に，式 (2) は，各頂点を丁度 1 色で彩色する事を表す．式 (3) は，各辺 $\{v,v'\}\in E$ において，両端の 2 頂点 $v$ と $v'$ は異なる色で彩色する事を表す．式 (4) の変数 $x_{v,c}$ は，頂点 $v$ が色 $c$ で彩色されない時に 0，彩色される時に 1 をとる $\{0,1\}$ 変数である．最小コスト点彩色問題の最小コスト点彩色の列挙は，上記の組合せ最適化問題の最適解を列挙すればよいが，一般にそのような列挙は容易では無い．本稿では，木の持つグラフ構造に着目し，最小コスト点彩色を列挙するアルゴリズムを提案する．

本稿の構成は以下のようになっている．2 節では，Kroon et. al. による，木の最小コスト点彩色 1 つを線形時間で求めるアルゴリズムを紹介する．我々が提案する列挙アルゴリズムでは，彼らのアルゴリズムを前処理に用いる．3 節では，提案する列挙アルゴリズムを示す．4 節では，最小コスト点彩色において，与えられた十分大きな色数を必ず使用するような木の特徴を示す．5 節では，今後の展望について述べる．

## 2. Kroon et. al. のアルゴリズム

本節では，Kroon et. al. が提案した，木の最小コスト点彩色を 1 つ求める O($n$) 時間アルゴリズムを紹介する．彼らのアルゴリズムは，動的計画法を基にしている．

アルゴリズムの概要は以下のとおりである．まず，$T$ 中の適当な頂点を根とし，各頂点 $v$ で $v$ の次数+1 色の色数を準備する．したがって，点彩色には $d+1$ 色準備すれば十分である．

木の頂点は，葉から根に向けて彩色を行う．彩色したい頂点 $v$ が葉であるならば，$v$ では，最小値の色と，第 2 最小値の色の 2 色のみの情報を持てば良い．そこで，最小値，第 2 最小値，各々の値を $K_1(v),K_2(v)$ に，最小値の色を $C^*(v)$ に格納する．

内点である頂点 $v$ に対しては，次数+1 色の各色で $v$ を彩色したとき，$v$ の各子 $v_i$ のもつ最小値 $K_1(v_i)$ と第 2 最小値 $K_2(v_i)$ の内，$v$ と色が異なり，かつ小さい方の値を選ぶことで，$v$ の最小値 $K_1(v)$ と第 2 最小値 $K_2(v)$，最小値の色 $C^*(v)$ を決定できる．この操作を $v$ が $T$ の根となるまで繰り返すことで，最小コスト点彩色を求める．

以下にアルゴリズムの詳細を示す．

**最小コスト点彩色アルゴリズム**

**入力**： 木 $T$，$T$ の根 $v$ ($T$ の任意の頂点)，色集合 $C=\{c_1,c_2,\cdots,c_n\}$，コスト関数 $w:C\rightarrow R$ $(\forall i,j(i<j);\ w(c_i)<w(c_j))$

**出力**： $T$ の最小コスト点彩色の最適値 $w^*$

1: $P_1(T,v)$ を呼ぶ．

2: $w^*:=K_1(v)$;

3: $w^*$ を出力．

**Procedure $P_1(T,v)$**

1: 頂点 $v$ が葉ならば，以下 (1-1)〜(1-3) を実行し戻る．

(1-1) $K_1(v):=$ 色 $c_1$ の値

(1-2) $C^*(v):=c_1$

(1-3) $K_2(v):=$ 色 $c_2$ の値

2: $T-v$ の各部分木 $T_i$ とその根 $v_i$ について $P_1(T_i,v_i)$ を呼ぶ．$(i=1,2,\cdots,d)$

3: $P_2(T,v,v_i,K_1,C^*,K_2)$ を呼ぶ． $(i=1,2,\cdots,d)$

**Procedure $P_2(T,v,v_i,K_1,C^*,K_2)$**

1: $S:=\sum_{i=1}^{d}K_1(v_i)$

2: $L(c):=S \quad (c=1,2,\cdots,d+1)$

3: $i=1,2,\cdots,d$ について $C(v_i)\leq d+1$ ならば
  $L(C(v_i)):=L(C(v_i))+K_2(v_i)-K_1(v_i)$

4: $K(v,c):=w(c)+L(c) \quad (c=1,2,\cdots,d+1)$

5: $K_1(v):=\min\{K(v,c)\mid c=1,2,\cdots,d+1\}$
  $C^*(v):=\{\mathrm{argmin}\{K(v,c)\mid c=1,2,\cdots,d+1\}$
  $K_2(v):=\min\{K(v,c)\mid c=1,2,\cdots,d+1; c\neq C(v)\}$

上記のアルゴリズムを実行すると，木 $T$ の最小コスト点彩色の最適値が求まる．最適値を用いて，根から葉に向けて頂点をたどりながら，$K_1(v),K_2(v),C^*(v)$ の値を基に，各頂点に割り当てられる色を構成することが出来る．

以下に，Kroon et. al. のアルゴリズムの計算量を示す．

**定理 1 (L.G. Kroon, A. Sen, H. Deng, and A. Roy [4])** 木 $T = (V, E)$ と，色の集合 $C(|C| = n)$，コスト関数 $w : C \to R$ が与えられている．この時，$T$ の最小コスト点彩色は $O(n)$ 時間で求められる．ただし，$n$ は $T$ の頂点数である． □

彼らは論文で明記してないが，必要とする記憶容量は $O(n)$ である．ただし，$d$ は頂点 $v$ の子の個数である．各頂点では，2 つの値と色のみ保持すれば良い．そのため，アルゴリズムの計算量は容易に算定できる．

図 1 の木に対して Kroon et. al. のアルゴリズムを適用すると，図 2 のように葉から色の候補を探し，図 3 の最小コスト点彩色を得ることができる．

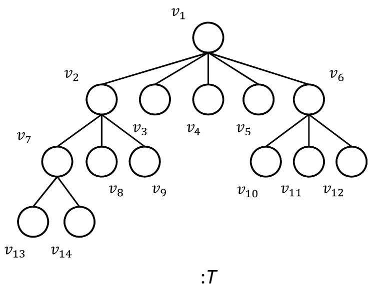


図 1 $T$ の例
**Fig. 1** Example of $T$

## 3. 列挙アルゴリズム

本節では，与えられた木の最小コスト点彩色が一意でない場合，それらを重複なく，全て列挙するアルゴリズムを提案する．我々の列挙アルゴリズムは，Kroon et. al. のアルゴリズムを前処理に使用する．また，最小コスト点彩色問題を整数計画問題に対する解法を適用せず，問題特有のグラフ構造に着目し，アルゴリズミックに解を列挙する．

列挙アルゴリズムを詳細に記述する前に，以下に概要を説明する．我々のアルゴリズムでは，まず前処理として，Kroon et. al. のアルゴリズムの実行する．彼らのアルゴリズムでは，$T$ 中の各頂点では，最小値と第 2 最小値，そして，最小値の色のみを保持していた．しかし，我々のアルゴリズムでは，各頂点で得られた，次数＋ 1 色分の値全てを保持する．最小コスト点彩色が 1 つ得られたら，各頂点での次数＋ 1 色の値を昇順にソートする．そして，ソートされた値の最小値，もしくは第 2 最小値を持つ色集合を用いて，条件を分けをして各頂点での色の候補を絞り，最小コスト点彩色を列挙する．

列挙アルゴリズムは以下のとおりである．アルゴリズムで呼び出す **Procedure** $P_1, P_2$ は 2 節と同じであるため割愛する．

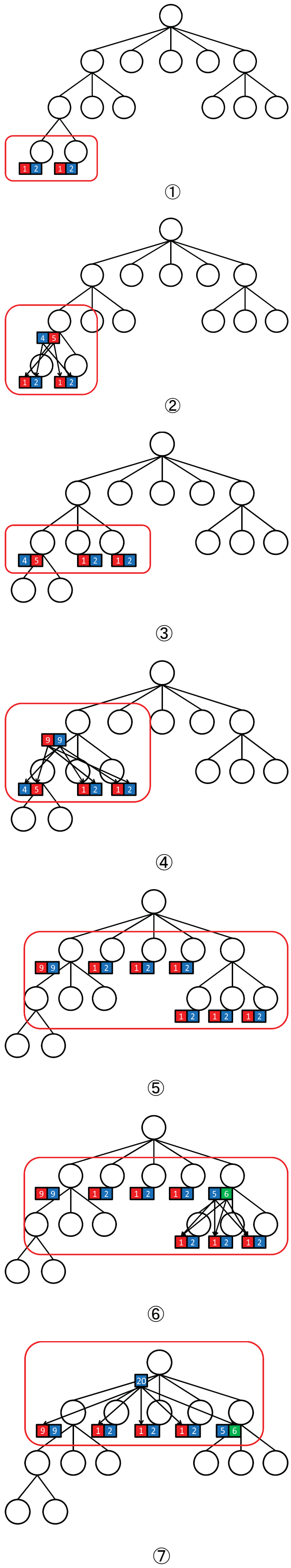

図 2 最小コスト点彩色アルゴリズムの実行例
**Fig. 2** Example of optimal cost vertex coloring algorithm

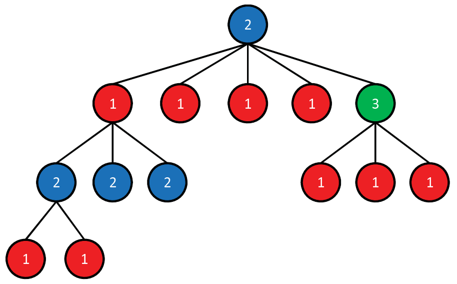


**図 3** $T$ の最小コスト点彩色（最適コスト：20）
**Fig. 3** Optimal cost vertex coloring of $T$ (The optimal value: 20)

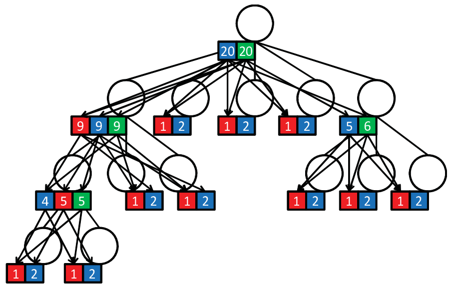


**図 4** $T$ の最小コスト点彩色を列挙するための木構造
**Fig. 4** Tree structure for enumerating all optimal cost vertex colorings of $T$

**最小コスト点彩色列挙アルゴリズム**

**入力：** 木 $T$，$T$ の根 $v$ ($T$ 中の最大次数の頂点)，色集合 $C = \{c_1, c_2, \cdots, c_n\}$，コスト関数 $w : C \to R$

**出力：** $T$ の最小コスト点彩色を重複無くすべて

1: $P_1(T, v)$ を呼ぶ．
2: $T$ 中の各頂点 $v$ に対し，以下を実行する．
(2-1) $CS_1(v) := \{\}$，$CS_2(v) := \{\}$.
(2-2) $K(v, c)$ を昇順に整列する．
(2-3) $CS_1(v) :=$ 頂点 $v$ で最小値をもつ色集合．
(2-4) 頂点 $v$ が内点で，かつ $|CS_1(v)| = 1$ ならば $CS_2(v) :=$ 頂点 $v$ で第 2 最小値をもつ色集合．
3: 根 $v$ に対し，$CL(v) := CS_1(v)$.
4: $P_3(T, v, CS_1, CS_2, CL(v))$ を呼ぶ．
5: 4 で構築した木構造に対し，根 $v$ の $CL(v)$ 中の各色から有向木を列挙する．そして，有向木を列挙する度に，各頂点の色を出力する．

**Procedure** $P_3(T, v, CS_1, CS_2, CL(v))$

1: 頂点 $v$ が葉ならば戻る．
2: 各色 $c' \in CL(v)$ について，頂点 $v$ の各子 $v_i$ に対して以下を実行する．
(2-1) $CL(v_i) := \{\}$
(2-2)Case 1) $|CS_1(v_i)| = 1$
$CS_1(v_i) = c'$ ならば $CL(v_i) := CS_2(v_i)$,
それ以外は $CL(v_i) := CS_1(v_i)$.
Case 2) $|CS_1(v_i)| > 1$
$CL(v_i) := CS_1(v_i) \setminus c'$
(2-3)$CL(v)$ 中の色 $c'$ から $CL(v_i)$ 中の各色を，有向辺で結ぶ．
(2-4) $P_3(T, v_i, CS_1, CS_2, CL(v_i))$ を呼ぶ．

上記のアルゴリズム中では，親子の頂点間の彩色可能なパターン同士を有向辺で結び，図 4 に示すような木構造を構築した後，根から有向木を深さ優先探索を施すことで，最小コスト点彩色を列挙する．

以下に，提案する列挙アルゴリズムの計算量を示す．

**定理 2** 与えられた木 $T = (V, E)$，色の集合 $C = \{c_1, c_2, \cdots, c_n\}$，コスト関数 $w : C \to R$ に対し，提案する列挙アルゴリズムは，前処理に $\mathrm{O}(nd \log d)$ 時間を要した後，$T$ の全ての最小コスト点彩色を 1 つ当たり $\mathrm{O}(nd^2)$ 時間で列挙する．総計算時間量は $\mathrm{O}(nd \log d + Knd^2)$ である．また，必要とする記憶容量は $\mathrm{O}(nd^2)$ である．ただし，$n$ は $T$ の頂点数，$d$ は頂点 $v$ の子の個数，$K$ は列挙される最小コスト点彩色の総数である．

**証明**上記のアルゴリズムの時間計算量を算定する．前処理には Kroon et. al. のアルゴリズムを用いており，各頂点の彩色コストの整列には $\mathrm{O}(d \log d)$ 時間，木全体では $\mathrm{O}(nd \log d)$ 時間を必要とする．内点とその子の頂点間の色の組合せは高々 $(d+1)^2$ 通りある．ゆえに，木構造から有向木 1 つを出力するのに $\mathrm{O}(nd^2)$ 時間を必要とする．したがって，総時間計算量は $\mathrm{O}(nd \log d + Knd^2)$ となる．

次に，記憶容量を算定する．前処理には Kroon et. al. のアルゴリズムを用いるため，木全体では $\mathrm{O}(n)$ を必要とする．列挙には木の各頂点とその子に対して全ての最小値と第 2 最小値の色集合を保持する必要があるため，高々 $(d+1)^2$ を必要とする．ゆえに，各頂点に対しては，$\mathrm{O}(d^2)$，木全体では $\mathrm{O}(nd^2)$ を必要とする．したがって，木全体の記憶容量は $\mathrm{O}(nd^2)$ で抑えられる． □

## 4. 最小コスト点彩色の一意性

本節では，木の最小コスト点彩色で，与えられた十分大きな任意の色数を必ず使用し，最小コスト点彩色が一意とある木の特徴付けを示す．

以下に，木の最小コスト彩色が一意である木の特徴を示す．

**定理 3** 木 $T = (V, E)$ と，色集合 $C = \{c_1, c_2, \cdots, c_n\}$，コ

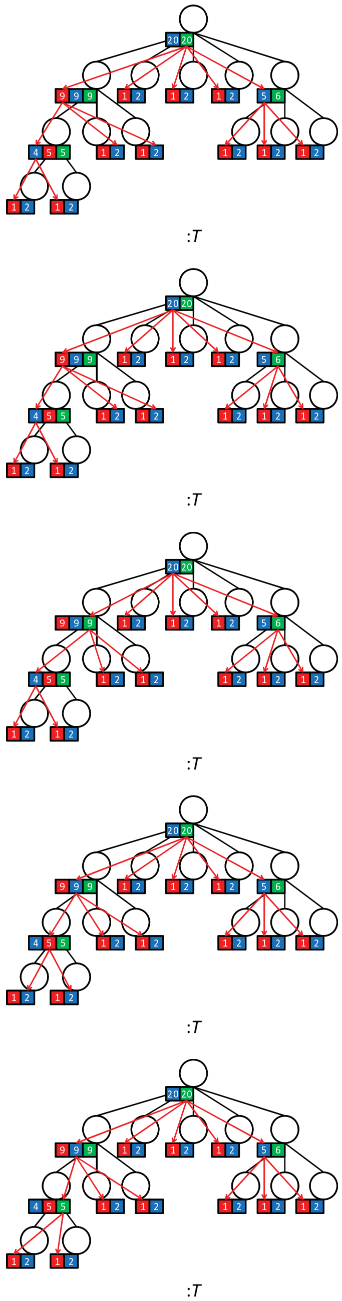


**図 5**　$T$ に対する全ての最小コスト点彩色
**Fig. 5**　All optimal cost vertex colorings of $T$

スト関数 $w(c_i) = i$ $(i = 1, 2, \cdots, n)$ が与えられている．$T$ の最小コスト点彩色を $T^*$,$T^*$ の根 $r$ には，$T^*$ で使用されている色の中で，最もコストの高い色 $c_k$ が割り当たられているとする．また，$T^* - r$ の部分木の根において，第 2 最小コスト点彩色は一意でないとする．この時，最小コスト点彩色 $T^*$ が一意ならば，$T^*$ は以下の 2 つの条件をすべて満たす．

*i)* $T^* - r$ の各部分木 $T_1, T_2, \cdots, T_l$ は最小コスト点彩色である．

*ii)* $T^* - r$ の各部分木 $T_1, T_2, \cdots, T_l$ の根には，色 $c_1, c_2, \cdots, c_{k-1}$ が全て現れ，色 $c_{k-i}$ が割り当てられている部分木の根は，少なくとも $i + 1$ 個存在する．□

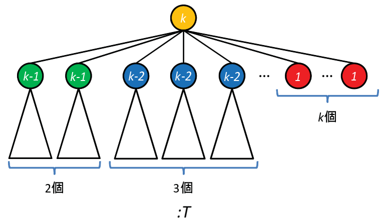


**図 6**　最小コスト点彩色が一意となる木
**Fig. 6**　Tree has the unique optimal cost vertex coloring

## 5. 結論と今後の課題

本稿では，木に対する最小コスト点彩色を重複無く全て列挙アルゴリズムを初めて提案した．提案したアルゴリズムは最小コスト点彩色を 1 つ当たり $O(nd^2)$ 時間で求めることが出来る．

今後の課題として，提案した列挙アルゴリズムの高速化や，他のグラフに対する最小コスト点彩色の列挙アルゴリズムの提案が挙げられる．

## 参考文献

[1] Isobe, S., Zhou, X. and Nishizeki, T.: *Cost total colorings of trees,* IEICE TRANSACTIONS on Information and Systems, E87-D, **2**, 337–342 (2004).

[2] Ito, T., Kaminski, M. and Demaine, E. D.: *Reconfiguration of List Edge-Colorings in a Graph,* Lecture Notes in Computer Science, **5664**, 375–386 (2009).

[3] Ito, T., Sakamoto, N., Zhou, X. and Nishizeki, T. : *Minimum Cost Edge-Colorings of Trees can be reduced to Matchings,* IEICE TRANSACTIONS on Information and Systems, E94-D, **2**, 190–195 (2011).

[4] Kroon, L.G., Sen, A., Deng, H. and Roy, A. : *The optimal cost chromatic partition problem for trees and interval graphs, Proc. Workshop on Graph-Theoretic Concepts in Computer Science (WG '96),* Lecture Notes in Computer Science, Springer, **1197**, 279–292 (1997).

[5] Sen, A., Deng, H. and Guha, S. : *On a graph partition problem with an application to VLSI layout,* Information Processing Letters, **43**, 87–94 (1991).

[6] Supowit, K.J. : *Finding a maximum planar subset of a set of nets in a channel,* IEEE Transactions on Computer Aided Design, CAD-6, **1**, 93–94 (1987).

[7] Yamanaka, K. and Nakano, S.: *Listing All Plane Graphs,* Journal of Graph Algorithms and Its Applications, **13**, 1, 5–18 (2009).

[8] Yamanaka, K., Nakano, S., Matsui, Y., Uehara, R. and Nakada, K.: *Efficient Enumeration of All Ladder Lotteries and Its Application,* Theoretical Computer Science, **411**,1714–1722 (2010).

[9] Yamanaka, K., Otachi, Y., and Nakano, S.: *Efficient Enumeration of Ordered Trees with k Leaves,* Theoret-

ical Computer Science, **442**, 22–27 (2012).
[10] Zhou, X. and Nishizeki, T.: *Algorithm for the cost edge-coloring of trees, Proc. 7th Annual International Computing and Combinatorics Conference (COCOON '01),* Lecture Notes in Computer Science, **2108**, 288–297 (2001).