多目的最適化問題への列挙アルゴリズム理論からのアプローチ

岡本 吉央*　　宇野 毅明†

概 要

本研究では多目的最適化問題に列挙アルゴリズム理論の手法を適用する．ケーススタディとして，多目的最小費用全域木問題を考察する．これは，連結無向グラフと複数の辺費用関数が与えられたとき，辺費用関数のある凸結合を最小化するような全域木を全て見つける問題である．この問題に対して，分割法と逆探索法という手法の適用を考える．分割法に対しては，自然な適用法から生じる部分問題が NP 完全であることを証明する．このことによって，分割法によるアプローチは困難であることが示唆される．その一方で，逆探索法に基づけば，効率的アルゴリズムが設計できることを示す．これは線型計画法の多目的版に一般化することができ，退化が無い場合に対して最初の多項式時間遅延多項式領域アルゴリズムとなる．

1 はじめに

多目的最適化は最適化理論，オペレーションズ・リサーチ，決定科学における広大な分野である．多目的最適化問題においては，大抵，パレート最適性などの特定の性質を満たす解を全て出力 (列挙) する必要がある．詳しくは Ehrgott [3] を参照のこと．

多目的最適化に対するアルゴリズム設計において主流となるアプローチは 2 つある．1 つは厳密アプローチであり，もう 1 つは近似アプローチである．厳密アプローチでは，列挙は厳密でなくてはならず，すなわち，全ての解が (重複無く) 出力されなければならない．例えば，多目的線型計画法においては，全てのパレート最適端点解や全てのパレート最適面を列挙する厳密アルゴリズムが数多く提案されて来ている (Ehrgott [3] やそこの参考文献を参照のこと)．二目的組合せ最適化問題に対しては，Ulungu & Teghem [8] が二段階法という枠組を提案しており，これは始めにパレート最適端点解全体を特定し，それから，残りのパレート最適解を列挙するという方法である．これに対して，近似アプローチでは部分的な列挙を行なう．例えば Zitzler, Laumanns & Bleuler [11] を参照のこと．少々異なる近似アプローチが Papadimitriou & Yannakakis [7] によって行なわれ，そこでは，ある種の近似保証が得られる．最近のサーベイについては Zaroliagis [10] を参照のこと．

本研究のアプローチは厳密であり，列挙アルゴリズム理論の技法を用いていく．列挙アルゴリズム理論は強力な手法を開発してきたが，それらが多目的最適化に応用されてはいないようである．そこで，ここでは例問を使って，列挙アルゴリズム理論における 2 つの主

---

*豊橋技術科学大学 情報工学系
†国立情報学研究所

要な方法を比較する．その 1 つは分割法であり，もう 1 つは逆探索法である．分割法においては，自明なインスタンスが得られるまで解空間全体を再帰的に分割することで列挙を行なう．Avis & Fukuda [1] による逆探索法では，解空間上に根付き木を陰に定義し，それを探索する．

例問として，まず，多目的最小費用全域木問題を考察する．これは，連結無向グラフと複数の辺費用関数が与えられたとき，辺費用関数のある凸結合を最小化するような全域木を全て見つける問題である．多目的最適化の用語で言うと，出力はちょうど可能な全ての加重和スカラー化に対する解全体になり，これらはパレート最適端点解全体に対応する．上で述べた 2 つの手法をこの問題に適用することを考える．まず，分割法による自然なアプローチから生じる部分問題が NP 完全であることを証明する．これによって，分割法によるアプローチは困難であることが示唆される．その一方で，逆探索法に基づいて，この問題に対する多項式時間遅延多項式領域アルゴリズムを設計する．このアルゴリズムは本問題に対してこのような理論的保証を持つ最初のアルゴリズムである．

本研究が提案するアルゴリズムはマトロイドや劣モジュラ・システムにおける最小費用基問題の多目的版にも拡張でき，さらに，同様な考察を多目的線型計画問題に対しても行なうことができる．多目的線型計画問題に対しては多くのアルゴリズムが提案されて来たが，そのどれも多項式時間遅延性と多項式時間領域性を兼ね備えていなかった (Ehrgott [3] とそこの参考文献を参照のこと)．実際，これらのアルゴリズムは重複を除くために，出力した解を全て作業領域にリストとして保持しているため，それが効率性に対するボトルネックとなっていた．小さな変更 (例えば，リストの代わりに平衡二分探索木を用いること) で，多項式時間遅延性を達成することはできるが，小さな変更で多項式領域性を達成することは困難に思える．すなわち，作業領域に対する本質的な改善はかなり難しいのである．それに対して，本研究のアルゴリズムは多項式時間遅延性も多項式領域性も達成できる．これは，列挙アルゴリズム理論の強力さを示す例であり，本研究によって多目的最適化とアルゴリズム理論の関連が成熟されていくことが期待される．

本論文の構成は以下の通りである．次節では列挙アルゴリズム理論の用語を導入し，多目的最小費用全域木問題を定義する．第 3 節では，分割法から生じる自然な部分問題が NP 完全であることを証明する．第 4 節では，逆探索法に基づく多項式時間遅延多項式領域アルゴリズムを設計する．第 5 節では，アルゴリズムの拡張を議論する．最終節では未解決問題を述べる．

2　定式化

列挙問題では与えられた条件を満たす対象を全て出力することが要求される．出力対象は解と呼ばれる．列挙アルゴリズムが多項式時間遅延性を持つとは，任意の解に対して，その次の解を出力するまでにアルゴリズムの要する時間が入力の大きさに関する多項式で抑えられることである．列挙アルゴリズムが多項式領域性を持つとは，それが使用する作業領域の大きさが入力の大きさの多項式で抑えられることである．ここでは領域として作業領域の大きさだけを考慮し，出力に対する領域は除いているので注意する．直感的に言うと，作業領域は読み書き可能なメモリであり，出力領域は書き込みのみ可能なディスクである．

k 個の関数 $c_1, c_2, \ldots, c_k$ の凸結合とは，足し合わせると 1 となるある非負実数 $\lambda_1, \lambda_2, \ldots, \lambda_k$

を用いて $\sum_{i=1}^{k} \lambda_i c_i$ と書ける関数である．係数のベクトル $(\lambda_1, \ldots, \lambda_k)^\top$ はこの凸結合の重心座標と呼ばれる．

2 つの集合 X, Y に対して，$X \triangle Y = (X \cup Y) \setminus (X \cap Y)$ と定義し，これを X と Y の対称差と呼ぶ．

連結無向グラフ $G = (V, E)$ に対して，G の全域木とは大きさ $|V| - 1$ の辺部分集合 $T \subseteq E$ で，閉路を含まないもののことである．非負辺費用関数 $c\colon E \to \mathbb{R}_+$ に対して，c に関する G の最小費用全域木とは，G の全域木 T で費用和 $c(T) = \sum_{e \in T} c(e)$ を最小化するもののことである．本研究では主に次の問題を研究する．

問題: MC-MCST

入力: 連結無向グラフ $G = (V, E)$ と k 個の異なる辺費用関数 $c_1, \ldots, c_k\colon E \to \mathbb{R}_+$

出力: G の全域木で，$c_1, \ldots, c_k$ のある凸結合に関して最小費用となっているもの全体

G の全域木で $c_1, \ldots, c_k$ のある凸結合に関して最小費用となるもの (すなわち，MC-MCST において出力されるべきもの) を実行可能と呼ぶことにする．

3 分割法

まず，与えられた連結無向グラフ $G = (V, E)$ の全ての全域木を列挙する分割法アルゴリズムを見てみよう．始めに，任意の辺 $e_1 \in E$ を選び，2 つの部分問題を考える．1 つは e_1 を含む全域木を列挙する問題で，もう 1 つは e_1 を含まない全域木を列挙する問題である．この 2 つ問題が解ければ，元の問題も解ける．これらの部分問題を再帰的に解くことを考えると，再帰木が得られるが，部分問題が解を持たないことが簡単に判定できればそこで再帰を止めることができるので，冗長な計算をしなくて済むようになる．部分問題が解を持つかどうか判定する問題は「共通部分が空な部分集合 $E_1, E_2 \subseteq E$ が与えられたとき，G の全域木で E_1 の辺を全て含み，E_2 の辺をどれも含まないようなものは存在するだろうか？」というものになる．これは線形時間で解くことができる．

同じ方法で MC-MCST を解くためには，次の問題が部分問題として現れてくる．

問題: BinaryPartition

入力: 連結無向グラフ $G = (V, E)$，共通部分が空な 2 つの部分集合 $E_1, E_2 \subseteq E$，k 個の個となる非負辺費用関数 $c_1, \ldots, c_k\colon E \to \mathbb{R}_+$

質問: E_1 の辺を全て含み，E_2 の辺をどれも含まないような G の全域木で，$c_1, \ldots, c_k$ のある凸結合に関して最小費用になっているものが存在するか？

問題 BinaryPartition が多項式時間で解ければ，上と同じ分割法に基づくアルゴリズムによって MC-MCST が多項式時間遅延多項式領域で解けることになる．しかし，次の定理から，それはありそうもないことが分かる．

定理 3.1. 問題 BinaryPartition は NP 完全である．

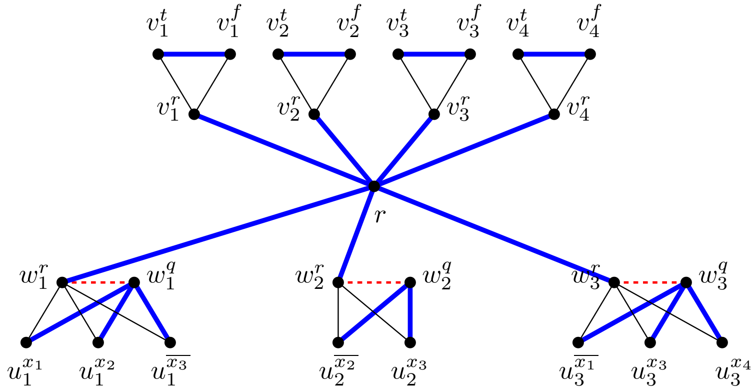

図 1: **定理** 3.1 **の証明における帰着．これは** $C_1 = x_1 \vee x_2 \vee \overline{x_3}$, $C_2 = \overline{x_2} \vee x_3$. $C_3 = \overline{x_1} \vee x_3 \vee x_4$ **としたときの** $C_1 \wedge C_2 \wedge C_3$ **というインスタンスに対する例である．黒い細線は** $E \setminus (E_1 \cup E_2)$ **に属する辺であり，青い太線は** E_1 **に属する辺，赤い破線は** E_2 **に属する辺である．**

表 1: **費用の概要．**

	$\{v_i^r, v_i^t\}$	$\{v_i^r, v_i^f\}$	$\{v_i^t, v_i^f\}$	$\{v_i^r, r\}$	$\{v_{i'}^r, v_{i'}^t\}$	$\{v_{i'}^r, v_{i'}^f\}$	$\{v_{i'}^t, v_{i'}^f\}$	$\{v_{i'}^r, r\}$
c_i	0	1	$1/n$	1	0	0	$1/n$	1
$c_{\bar{i}}$	1	0	$1/n$	1	0	0	$1/n$	1

	$\{w_j^r, w_j^q\}$	$\{w_j^r, r\}$	$\{w_j^q, u_j^\ell\}$	$\{w_j^r, u_j^\ell\}$
c_i	$2 - 1/(2n)$	1	1	1 if $\ell = x_i$, 2 otherwise
$c_{\bar{i}}$	$2 - 1/(2n)$	1	1	1 if $\ell = \overline{x_i}$, 2 otherwise

証明. **この問題が** NP **に属することは直ちに分かるので，**NP **困難性を示す．そのために，**SAT (**充足可能性問題**) **を** BinaryPartition **に帰着する．**SAT **のインスタンスは論理変数** $x_1, \ldots, x_n$ **の集合と節** $C_1, \ldots, C_m$ **の集合で与えられる．各節は** 1 **つ以上のリテラルから成り，ここで，リテラルとは論理変数かその否定である．**

与えられた SAT **のインスタンスから連結グラフ** $G = (V, E)$ **を構成する．各変数** x_i **に対して** 3 **つの頂点** v_i^r, v_i^t, v_i^f **を用意する．各節** C_j **に対して** 2 **つの頂点** w_j^r, w_j^q **を用意する．また，各節** C_j **の各リテラル** ℓ **に対して** 1 **つの頂点** u_j^ℓ **を用意する．最後にもう** 1 **つ頂点** r **を用意する．これら全体が** G **の頂点集合である．**

次に，G **の辺を準備する．各変数** x_i **に対して** $\{v_i^t, v_i^f\} \in E_1$, $\{v_i^r, v_i^t\} \in E \setminus (E_1 \cup E_2)$, $\{v_i^r, v_i^f\} \in E \setminus (E_1 \cup E_2)$, $\{v_i^r, r\} \in E_1$ **という** 4 **つの辺を用意する．各節** C_j **に対して** $\{w_j^r, w_j^q\} \in E_2$, $\{w_j^r, r\} \in E_1$ **という** 2 **つの辺を用意し，**C_j **の各リテラル** ℓ **に対して** $\{w_j^r, u_j^\ell\} \in E \setminus (E_1 \cup E_2)$, $\{w_j^q, u_j^\ell\} \in E_1$ **という** 2 **つの辺を用意する．これで，**G **と** E_1, E_2 **の構成は終わった．図** 1 **に例が示してある．**

次に，$2n$ **個の辺費用関数を定義する．各費用関数は変数かその否定** (**すなわちリテラル**) **に対応している．すなわち，各正リテラル** x_i **に対して費用関数** c_i **を定義し，各負リテラル** $\overline{x_i}$ **に対して費用関数** $c_{\bar{i}}$ **を以下のように定義する．定義は表** 1 **にまとめられている．まず，**$c_i(\{v_i^r, v_i^t\}) = 0$, $c_i(\{v_i^r, v_i^f\}) = 1$, $c_i(\{v_i^t, v_i^f\}) = 1/n$, $c_i(\{v_i^r, r\}) = 1$ **とし，各** $i' \in \{1, \ldots, n\} \setminus \{i\}$ **に対して** $c_i(\{v_{i'}^r, v_{i'}^t\}) = 0$, $c_i(\{v_{i'}^r, v_{i'}^f\}) = 0$, $c_i(\{v_{i'}^t, v_{i'}^f\}) = 1/n$, $c_i(\{v_{i'}^r, r\}) = 1$ **とする．各** $j \in \{1, \ldots, m\}$ **と節** C_j **の各リテラル** ℓ **に対して** $c_i(\{w_j^r, w_j^q\}) = 2 - 1/(2n)$, $c_i(\{w_j^r, r\}) = 1$,

$c_i(\{w_j^q, u_j^\ell\}) = 1$ とし，

$$c_i(\{w_j^r, u_j^\ell\}) = \begin{cases} 1 & (\ell = x_i) \\ 2 & (\text{その他}) \end{cases}$$

とする．同様に，$c_{\bar{i}}(\{v_i^r, v_i^t\}) = 1$, $c_{\bar{i}}(\{v_i^r, v_i^f\}) = 0$, $c_{\bar{i}}(\{v_i^t, v_i^f\}) = 1/n$, $c_{\bar{i}}(\{v_i^r, r\}) = 1$ とし，各 $i' \in \{1, \ldots, n\} \setminus \{i\}$ に対して $c_{\bar{i}}(\{v_{i'}^r, v_{i'}^t\}) = 0$, $c_{\bar{i}}(\{v_{i'}^r, v_{i'}^f\}) = 0$, $c_{\bar{i}}(\{v_{i'}^t, v_{i'}^f\}) = 1/n$, $c_{\bar{i}}(\{v_{i'}^r, r\}) = 1$ とする．各 $j \in \{1, \ldots, m\}$ と節 C_j の各リテラル ℓ に対して $c_{\bar{i}}(\{w_j^r, w_j^q\}) = 2 - 1/(2n)$, $c_{\bar{i}}(\{w_j^r, r\}) = 1$, $c_{\bar{i}}(\{w_j^q, u_j^\ell\}) = 1$ とし，

$$c_{\bar{i}}(\{w_j^r, u_j^\ell\}) = \begin{cases} 1 & (\ell = \overline{x_i}) \\ 2 & (\text{その他}) \end{cases}$$

とする．これで，MC-MCST のインスタンスの構成が完了した．

G の全域木 T が許容的であることを，それが $E_1 \subseteq T$ と $T \cap E_2 = \emptyset$ の両方を満たすこととして定義する．ここで観察を行なう．最初に，各 $i \in \{1, \ldots, n\}$ に対して G の許容的全域木は $\{v_i^r, v_i^t\}$ か $\{v_i^r, v_i^f\}$ のどちらかを含むが，両方を含むことはない．これは $\{v_i^t, v_i^f\} \in E_1$ であるからである．同様な理由から，各 $j \in \{1, \ldots, m\}$ に対して，G の許容的全域木は C_j の全てのリテラル ℓ の中からちょうど 1 つの $\{w_j^r, u_j^\ell\}$ を含む．

辺費用関数 $c_1, \ldots, c_n, c_{\bar{1}}, \ldots, c_{\bar{n}}$ の凸結合 c を考える．関数 c における c_i の係数を λ_i で表し，$c_{\bar{i}}$ の係数を $\lambda_{\bar{i}}$ で表す．このとき，T が c に関する許容的最小費用全域木であり，ある $i \in \{1, \ldots, n\}$ に対する辺 $\{v_i^r, v_i^t\}$ を含んでいるならば，$\lambda_i \geq 1/n$ が成り立つことを主張する．この主張を示すために，T から $\{v_i^t, v_i^f\}$ を除き，$\{v_i^r, v_i^f\}$ を含めることで得られる G の全域木を T' とする．これは $\{v_i^t, v_i^f\} \in E_1$ を含まないので，許容的ではない．計算をすると，$c(T') - c(T) = \sum_{i'=1}^n (\lambda_{i'} c_{i'}(\{v_{i'}^r, v_{i'}^f\}) + \lambda_{\bar{i'}} c_{\bar{i'}}(\{v_{i'}^r, v_{i'}^f\})) - \sum_{i'=1}^n (\lambda_{i'} c_{i'}(\{v_{i'}^t, v_{i'}^f\}) + \lambda_{\bar{i'}} c_{\bar{i'}}(\{v_{i'}^t, v_{i'}^f\})) = (\lambda_i + 0) - \sum_{i'=1}^n (\lambda_{i'}/n + \lambda_{\bar{i'}}/n) = \lambda_i - 1/n$ となり，T は c に関する最小費用全域木であるから $c(T') \geq c(T)$ も成り立つ．したがって，$\lambda_i \geq 1/n$ となり，主張の成立が分かる．同様に，T が c に関する許容的最小費用全域木で，ある $i \in \{1, \ldots, n\}$ に対する辺 $\{v_i^r, v_i^f\}$ を含むならば，$\lambda_{\bar{i}} \geq 1/n$ が成り立つ．ここで，$\sum_{i'=1}^n (\lambda_{i'} + \lambda_{\bar{i'}}) = 1$ であるから，各 $i \in \{1, \ldots, n\}$ に対して T が $\{v_i^r, v_i^t\}$ を含むとき $\lambda_i = 1/n$ かつ $\lambda_{\bar{i}} = 0$ であり，T が $\{v_i^r, v_i^f\}$ を含むとき $\lambda_i = 0$ かつ $\lambda_{\bar{i}} = 1/n$ となることが分かる．

では，c_i 全体と $c_{\bar{i}}$ 全体 $(i \in \{1, \ldots, n\})$ のある凸結合に関する G の許容的最小費用全域木が存在することと与えられた SAT のインスタンスが充足可能であることが同値であることを示す．最初に，凸結合 $c = \sum_{i=1}^n (\lambda_i c_i + \lambda_{\bar{i}} c_{\bar{i}})$ に関する許容的最小費用全域木 T が存在すると仮定する．上で述べた主張より，各 $i \in \{1, \ldots, n\}$ に対して木 T は $\{v_i^r, v_i^t\}$ と $\{v_i^r, v_i^f\}$ のどちらか一方を含む．その選択に応じて，与えられた SAT のインスタンスに対する真理値割当を構成する．すなわち，T が $\{v_i^r, v_i^t\}$ を含む場合 x_i を真にし，T が $\{v_i^r, v_i^f\}$ を含む場合 x_i を偽にする．これによって，真理値割当が確かに与えられ，後はこの割当が与えられたインスタンスを充足することを確認すればよい．

T が G の許容的全域木であることから，各 $j \in \{1, \ldots, m\}$ に対して節 C_j の全てのリテラル ℓ の中からちょうど 1 つの $\{w_j^r, u_j^\ell\}$ が T に含まれる．このとき，T が更に c に関する最小費用全域木であり，ある $j \in \{1, \ldots, m\}$ に対する C_j のあるリテラル ℓ に対して $\{w_j^r, u_j^\ell\}$ が T に含まれるならば，ℓ が上で構成した割当によって真になっていなければならないことを主張する．この

主張が成り立てば，与えられた SAT のインスタンスが充足可能であることが分かる．主張を証明するために，ℓ が偽であると仮定する．そのとき，T から $\{w_j^r, u_j^\ell\}$ を除き $\{w_j^r, w_j^q\}$ を加えることで出来上がる G の (許容的ではない) 全域木 T' を考える．前の主張と上の構成法から $\lambda_\ell = 0$ が得られる．したがって，計算すると $c(T') - c(T) = \sum_{i=1}^n (\lambda_i c_i(\{w_j^r, w_j^q\}) + \lambda_{\bar{i}} c_{\bar{i}}(\{w_j^r, w_j^q\})) - \sum_{i=1}^n (\lambda_i c_i(\{w_j^r, u_j^\ell\}) + \lambda_{\bar{i}} c_{\bar{i}}(\{w_j^r, u_j^\ell\})) = \sum_{i=1}^n (\lambda_i + \lambda_{\bar{i}})(2 - 1/(2n)) - (\sum_{i=1}^n (2(\lambda_i + \lambda_{\bar{i}})) - \lambda_\ell) = (2 - 1/(2n)) - (2 - 0) = -1/(2n) < 0$ が得られる．ゆえに，T は c に関する G の最小費用全域木ではないことになり，これは矛盾である．

逆に，与えられた SAT のインスタンスが充足可能であると仮定する．では，充足可能真理値割当を固定する．この割当が x_i を真にするとき，$\lambda_i = 0, \lambda_{\bar{i}} = 1/n$ と置く．そうでないとき (すなわち，x_i が偽となるとき)，$\lambda_i = 1/n, \lambda_{\bar{i}} = 0$ とする．ここで，$c = \sum_{i=1}^n (\lambda_i c_i + \lambda_{\bar{i}} c_{\bar{i}})$ とする．計算をすると，各 $i \in \{1, \ldots, n\}$ に対して $c(\{r, v_i^r\}) = 1$ かつ $c(\{v_i^t, v_i^f\}) = 1/n$ となり，x_i が真であるとき，$c(\{v_i^r, v_i^t\}) = 0$ かつ $c(\{v_i^r, v_i^f\}) = 1/n$ となり，x_i が偽であるとき，$c(\{v_i^r, v_i^t\}) = 1/n$ かつ $c(\{v_i^r, v_i^f\}) = 0$ となることが分かる．また，各 $j \in \{1, \ldots, m\}$ と節 C_j の各リテラル ℓ に対して $c(\{r, w_j^r\}) = 1$, $c(\{w_j^r, w_j^q\}) = 2 - 1/(2n)$, $c(\{w_j^q, u_j^\ell\}) = 1$ となり，ℓ が真のとき，$c(\{w_j^r, u_j^\ell\}) = 2 - 1/n$ となり，ℓ が偽のとき，$c(\{w_j^r, u_j^\ell\}) = 2$ となることも分かる．

各 $j \in \{1, \ldots, m\}$ に対して，節 C_j のリテラル ℓ で，上で固定した真理値割当によって真となるものを 1 つ選んで来る．仮定より，そのようなリテラルは常に存在する．このとき，$T = E_1 \cup \{\{v_i^r, v_i^t\} \mid x_i \text{ は真}\} \cup \{\{v_i^r, v_i^f\} \mid x_i \text{ は偽}\} \cup \{\{w_j^r, u_j^{\ell_j}\} \mid j \in \{1, \ldots, m\}\}$ を考えると，これは G の全域木である．また，T は許容的であり，c に関する最小費用全域木であることも分かる．これによって，帰着全体が完了した． □

4 逆探索法

本節では，MC-MCST に逆探索法をどのように適用するか述べる．そのために，列挙される G の全域木全体の上に根付き木 $\mathcal{R}$ を用意する．すなわち，$\mathcal{R}$ の各ノードは G の全域木である．列挙は $\mathcal{R}$ を深さ優先探索することによって行なわれるが，$\mathcal{R}$ 自体を完全に保持して探索するのではなく，$\mathcal{R}$ を陰に定義する親子関係だけから探索を行なう．したがって，効率的な逆探索アルゴリズムを設計するためには親や子を効率良く発見できるような親子関係を与えれば十分である．これによって，列挙する全域木全体を格納しなくても探索ができるようになり，全体としてアルゴリズムが多項式時間遅延性と多項式領域性を持つようになる．

まず，G の全域木全体の上の隣接関係を定義する．G の 2 つの異なる全域木 T と T' が隣接するとは，T と T' の対称差の大きさが 2 であることである．この隣接関係から，G の全域木全体をノード集合とする無向グラフ $\mathcal{G}(G)$ が自然に定義できる．更に，実行可能な全域木 (すなわち，MC-MCST において列挙されるべき全域木) 全体が誘導する $\mathcal{G}(G)$ の部分グラフを $\mathcal{G}_M(G)$ で表す．グラフ $\mathcal{G}_M(G)$ は辺費用関数に依存するが，簡便さのためにそれらは固定されていると見なし，記法には含めないことにする．

命題 4.1. 任意の連結無向グラフ G と任意の辺費用関数に対して，$\mathcal{G}(G)$ と $\mathcal{G}_M(G)$ はともに連結である．

$\mathcal{G}(G)$ の連結性はよく知られたもので [9, Exercise 2.1.62]，$\mathcal{G}_M(G)$ の連結性は Ehrgott [2] による．

$\mathcal{G}_M(G)$ 上に約束した根付き木 $\mathcal{R}$ を定義する．そのため，各実行可能全域木に次のような列を割り当てる．まず，G の辺全体がある固定された全順序 $\prec$ に従って $e_1 \prec e_2 \prec \cdots \prec e_m$ とラベル付けされていると仮定する．この順序 $\prec$ はタイブレイクに用いられる．G の実行可能全域木 T に対して $\boldsymbol{\lambda}_T \in \mathbb{R}^k$ を T が最小化する $c_1, \ldots, c_k$ の凸結合の重心座標の中で辞書式最大のものとする．次の補題で $\boldsymbol{\lambda}_T$ が多項式時間計算可能であることを示す．

補題 4.2. G の任意の実行可能全域木 T に対して，$\boldsymbol{\lambda}_T$ は多項式時間で見つけられる．

証明. ベクトル $\boldsymbol{\lambda}_T$ を求める問題を次の形で表現する．

$$
\begin{aligned}
&\text{辞書式最大化} \quad \boldsymbol{\lambda} \\
&\qquad\text{条件} \quad \sum_{e \in T} \sum_{i=1}^{k} \lambda_i c_i(e) \leq \sum_{e \in T'} \sum_{i=1}^{k} \lambda_i c_i(e) \quad (T \text{ に隣接する実行可能な } T'), \\
&\qquad\qquad\quad \sum_{i=1}^{k} \lambda_i = 1, \\
&\qquad\qquad\quad \lambda_i \geq 0 \qquad\qquad\qquad\qquad\qquad\quad (i \in \{1, \ldots, k\}).
\end{aligned}
$$

第 1 の制約で，G の全ての全域木を考慮する必要はなく，T と隣接する全域木のみを考えればよいことに注意する．これは最小費用全域木問題の凸性 (すなわちマトロイド性) による．全域木 T に隣接する全域木の数は $\mathrm{O}(m^2)$ であるので，制約の数は入力の大きさに関する多項式である．この辞書式最大化問題を解くためには，$i \in \{1, \ldots, k\}$ の小さい方から順に λ_i を最大化すればよく，各最大化問題は線型計画問題となる．したがって，線型計画法に対する任意の多項式時間アルゴリズムを用いることで，この問題も多項式時間で解けることが分かる． □

グラフ G の実行可能全域木 T の中で，$\boldsymbol{\lambda}_T$ を辞書式最大にするもの R が $\mathcal{R}$ の根として選ばれる．そのような R に対して $\boldsymbol{\lambda}_R$ は $(\lambda_R)_1 = 1$，かつ，全ての $i \in \{2, \ldots, k\}$ に対して $(\lambda_R)_i = 0$ を満たさなければならないので，すなわち，R は c_1 に関する最小費用全域木である．関数 c_1 に関する最小費用全域木が複数ある場合は，その中で $\prec$ に関して最大のものが R として選ばれる．そのような全域木 R は唯一存在し，任意の多項式時間最小費用全域木アルゴリズムによって多項式時間で見つけられる．

グラフ G の根でない実行可能全域木 T の親を指定するために，2 つの場合を分けて考える．まず始めの場合では，$\boldsymbol{\lambda}_T = (1, 0, 0, \ldots, 0)^\top$ を仮定する．このとき，T と R はどちらも c_1 を最小化する．したがって，次の補題が示すように，T から辺を 1 つ取り除き，R の辺を 1 つ付け加えることで，c_1 に関する別の最小費用全域木を得ることができる．

補題 4.3. 連結無向グラフ $G = (V, E)$，非負辺費用関数 $c\colon E \to \mathbb{R}_+$ に対して，$T_1, T_2 \subseteq E$ を c に関する G の異なる最小費用全域木とする．このとき，ある 2 辺 $e_1 \in T_1 \setminus T_2$ と $e_2 \in T_2 \setminus T_1$ が存在して，$(T_2 \cup \{e_1\}) \setminus \{e_2\}$ も c に関する G の最小費用全域木である．

証明. 辺 e_1, e_2 として補題が要求する性質を満たすものが存在することを言えばよい．まず，$e_1 \in T_1 \setminus T_2$ として，$T_1 \setminus T_2$ の辺 e の中で $c(e)$ が最小のものを任意に選ぶ．ここで，$T_2 \cup \{e_1\}$ は唯一の閉路を含むが，$e_2 \in T_2 \setminus T_1$ として，その閉路の中の e_1 以外の辺 e で $c(e)$ が最大のも

のを任意に選ぶ．そのような辺 e_2 が必ず存在することに注意すれば，$T = (T_1 \cup \{e_1\}) \setminus \{e_2\}$ は G の全域木である．

もし $c(e_1) < c(e_2)$ であるならば，$c(T) = c(T_2) + c(e_1) - c(e_2) < c(T_2)$ となり，T_2 が c を最小化することに矛盾する．また，$c(e_1) > c(e_2)$ であるならば，e_1 の選び方より，任意の $e \in T_1 \setminus T_2$ に対して $c(e) > c(e_2)$ が成り立つ．したがって，$T_1 \cup \{e_2\}$ を考えたときにできる閉路から $T_1 \setminus T_2$ の辺を任意に除去してできる全域木の費用は T_1 よりも小さくなり，これは T_1 が最小全域木であったことに矛盾する．ゆえに，$c(e_1) = c(e_2)$ であり，$c(T) = c(T_1) = c(T_2)$ が結論できる． □

全域木 T の親は次のように構成される．まず，$R \setminus T$ の辺 e で $c_1(e)$ が最小のもの e_R を選ぶ．そのようなものが複数ある場合，その中で $\prec$ において最大の辺を選ぶ．このとき，$T \cup \{e_R\}$ は閉路を含むが，その閉路から $T \setminus R$ の辺 e で $c_1(e)$ が最大のもの e_T を選ぶ．複数ある場合は，その中で $\prec$ において最小の辺を選ぶ．これによって，$T' = (T \cup \{e_R\}) \setminus \{e_T\}$ とする．上の議論より，T' は実行可能全域木であり，$|R \triangle T'| < |R \triangle T|$ である．ここで，T' を T の親とする．多項式時間で T' が見つかることも分かる．

2 つ目の場合では，$\boldsymbol{\lambda}_T \neq (1, 0, 0, \ldots, 0)^\top$ を仮定する．このとき，対応する $\boldsymbol{\lambda}_T$ を考え，$j \in \{2, \ldots, k\}$ を $(\lambda_T)_j \neq 0$ となる最小添字とする．ここで，$\boldsymbol{\lambda}_T$ の第 1 成分を十分小さな $\varepsilon > 0$ だけ増加させて，第 j 成分を ε だけ減少させることでベクトル $\boldsymbol{\mu} \in \mathbb{R}^k$ を構成する．そのような ε が存在して，$\boldsymbol{\mu}$ を重心座標とできることは 2 つ目の場合に対する仮定より分かる．関数 $\sum_{i=1}^k \mu_i c_i$ に関する G の最小費用全域木を S とする．そのようなものが複数ある場合は，その中から $\prec$ に関して最大のものを S とする．ベクトル $\boldsymbol{\lambda}_T$ の辞書式最大性と $\boldsymbol{\mu}$ が $\boldsymbol{\lambda}_T$ よりも辞書的に大きいことから，S と T は異なる全域木であることが分かる．パラメータ ε は十分小さく選んだので，S は $c = \sum_{i=1}^k (\lambda_T)_i c_i$ に関する最小費用全域木でもある．したがって，第 1 の場合と同様に補題 4.3 を用いた議論ができる．今回は，$S \setminus T$ の辺 e で，$c(e)$ が最小のものを e_S とし (複数ある場合は，その中から $\prec$ に関して最大のものを選ぶ)，$T \cup \{e_S\}$ に存在する唯一の閉路の辺 $e \in T \setminus T'$ の中で $c(e)$ が最大のもの e_T を選ぶ (複数ある場合は，その中から $\prec$ に関して最小のものを選ぶ)．このとき，補題 4.3 (の証明) から，$T' = (T \cup \{e_S\}) \setminus \{e_T\}$ は c に関する最小費用全域木であり，$|S \triangle T'| < |S \triangle T|$ が成り立つ．この T' を T の親とする．この場合も多項式時間で T' を見つけられる．これで，根でない実行可能全域木に対する親の定義が終了した．構成法より，T の親は $\mathcal{G}_M(G)$ において T と隣接しており，親は一意である．更に，次の補題は重要である．

補題 4.4. 連結無向グラフ $G = (V, E)$，非負辺費用関数 $c_1, \ldots, c_k \colon E \to \mathbb{R}_+$ に対して上で定義した親子関係は well-defined である．すなわち，実行可能な根でない全域木 $T \subseteq E$ からその親を順に辿っていくと根 R に行き着く．

証明. まず，根でない実行可能な全域木 T とその親 T' を考える．このとき，先ほどの場合分けの流れに従って考察する．まず，上記の場合 1 が適用される場合を考える．このときは $\boldsymbol{\lambda}_{T'} = \boldsymbol{\lambda}_T = (1, 0, 0, \ldots, 0)^\top$ が成立し，$|R \triangle T'| < |R \triangle T|$ となる．よって，親を順に辿って行くことで R に行き着くことができる．

次に，上記の場合 2 が適用される場合を考える．このとき，$T = T_0, T' = T_1$ として，一般的に，T_j の親を T_{j+1} とする記法を用いる．上で T に対して S を構成したときと同じように，T_j からは S_j という全域木が得られるものとする．この構成は $\boldsymbol{\lambda}_{T_j} \neq (1, 0, 0, \ldots, 0)^\top$ である

限り続けることができる．よって，任意の j に対して，ある $j' > j$ が存在して，$\boldsymbol{\lambda}_{T_{j'}}$ が $\boldsymbol{\lambda}_{T_j}$ よりも辞書式順序で大きいことを証明すればよい．そうすれば，いずれ $\boldsymbol{\lambda}_{T_j} = (1,0,0,\ldots,0)^\top$ となるときが訪れて，上記の場合 1 に帰着されるので，証明全体が終了する．

では，任意の j を固定する．$\boldsymbol{\lambda}_{T_{j+1}}$ が $\boldsymbol{\lambda}_{T_j}$ よりも辞書式順序で大きければ証明終了となるので，$\boldsymbol{\lambda}_{T_{j+1}} = \boldsymbol{\lambda}_{T_j}$ と仮定する．このとき，S_j と S_{j+1} の構成はそれぞれ $\boldsymbol{\lambda}_{T_{j+1}}$ と $\boldsymbol{\lambda}_{T_j}$ のみに依存していたので，$S_j = S_{j+1}$ となることが分かる．しかし，任意の j に対して，$|S_j \triangle T_{j+1}| < |S_j \triangle T_j|$ が成り立つので，無限に $S_j = S_{j+1} = S_{j+2} = \cdots$ と続くことはあり得ない．したがって，ある $j' > j$ に対して $\boldsymbol{\lambda}_{T_{j'}}$ が $\boldsymbol{\lambda}_{T_j}$ よりも辞書式順序で大きくなくてはならない． □

以上の議論より次の定理が得られることが分かった．

定理 4.5. 上で記述した逆探索法に基づくアルゴリズムによって，MC-MCST は多項式時間遅延多項式領域で解くことができる．

実行可能全域木 T の親を $\mathcal{G}_M(G)$ における T の近傍の中で $\prec$ に関して最大のものとして定義したらどうなるのだろうか，と思うかもしれない．しかし，この定義ではうまく行かない例を構成することができる．つまり，そのようなナイーブなアプローチでは MC-MCST を解くことはできず，そのためここでは λ というベクトルを親子関係の定義に用いたのである．

5 一般化

前節の逆探索アルゴリズムはいくつかの方向に一般化できる．議論の詳細を見てみると，アルゴリズムにおいて用いている最小費用全域木問題の特性はマトロイド性のみであることが分かる．したがって，マトロイドにおける最小費用基問題の多目的版も多項式時間遅延多項式領域で解けることが分かる．より一般的に，劣モジュラ・システムにおける最小費用基問題も多項式時間遅延多項式領域で解けることが分かる．そのためには，劣モジュラ・システムにおいて与えられた基の隣接基を特定する必要があるが，これは劣モジュラ関数最小化問題のインスタンスであり，多項式時間で解くことができる [4]．

より一般的に，多目的線型計画問題を考えることもできる．線型計画問題においては，不等式系 $Ax \geq \boldsymbol{b}, x \geq \boldsymbol{0}$ が与えられ (ただし，$A \in \mathbb{R}^{m\times n}$ は行列で，$\boldsymbol{b} \in \mathbb{R}^m$ はベクトルである)，与えられた $\boldsymbol{c} \in \mathbb{R}^n$ に対して，不等式系の解 x の中で $\boldsymbol{c}^\top x$ を最小化するものを見つけなくてはならない．

上の不等式系は少なくとも 1 つ端点を持つ凸多面体を定義し，問題に対する最適解が存在するとき，端点最適解が存在する．多目的線型計画問題においては，与えられた k 個の費用ベクトル $\boldsymbol{c}^1,\ldots,\boldsymbol{c}^k$ のある凸結合を最小化する端点最適解全体を列挙したい．

問題: MC-LP
入力: 行列 $A \in \mathbb{R}^{m\times n}$, ベクトル $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{c}^1,\ldots,\boldsymbol{c}^k \in \mathbb{R}^n$ **出力**: $\boldsymbol{c}^1,\ldots,\boldsymbol{c}^k$ のある凸結合を最小化する不等式系 $Ax \geq \boldsymbol{b}, x \geq \boldsymbol{0}$ の端点解 x 全体．

与えられた不等式系が定める凸多面体において，各端点解は辺を通じて他の端点解と隣接関係を持っている．この隣接関係から自然に無向グラフを定義することができ，MC-MCST に対して行なったのと同様にこのグラフ上の根付き木を陰に導入できる．

MC-LP のインスタンスに対して，与えられた不等式系が定める多面体の任意の端点解がちょうど n 個のファセット上にあるとき，それを退化していないインスタンスと呼ぶ．そのような場合，各端点解は高々n 個の端点解と隣接していて，隣接端点解はピボット操作によって多項式時間で発見できる．このようにして，補題 4.2，補題 4.3，補題 4.4 と同様の補題を確立することができて，効率的な列挙アルゴリズムを構築できる．

定理 5.1. 退化していない MC-LP は多項式時間遅延多項式領域で解くことができる．

もう一度強調しておきたいが，今まで MC-LP に対して多くのアルゴリズムが提案されてきたが，そのどれも重複を取り除くために列挙した解全体を作業領域に保持する必要があった．しかし，本研究の手法ではそうする必要が無く，それが効率性を導いている．

6 終わりに

本稿では列挙アルゴリズム理論の視点から多目的最適化問題を見てきた．アルゴリズム理論が貢献できそうな多目的最適化における問題はまだまだありそうである．

退化していてもよいような多目的線型計画問題はとても扱いにくそうである．一目的線型計画問題は凸多面体の頂点列挙に対応するが，それを多項式時間で解けるかどうか分かっていない [5].

MC-MCST に対して提案した逆探索アルゴリズムでは，1 つの全域木に隣接する全域木の数が高々入力の大きさに関する多項式であったことが鍵となっている．これは，例えば二部グラフにおけるマッチング問題を考えた場合，成り立つ性質ではない．今のところ，多目的割当問題 (すなわち，多目的最大二部マッチング問題) に対しては多目的最小費用全域木問題のように自然な分割法ではうまく行かないことは証明できる (証明は省略) が，どのようにすれば多項式時間遅延多項式領域で解けるのかは未解決である．

他の問題として，補題 4.2 に関連するものがある．そこでは，λ_T が多項式時間で計算できることを観察したが，そのために線型計画法に対する多項式時間アルゴリズムを用いている．よって，強多項式時間アルゴリズムではない．これが強多項式時間で計算できるかどうかよく分からない．

謝辞

本研究は両著者がスイス連邦工科大学チューリッヒ校理論情報科学科を訪問した際に始まった．Emo Welzl と彼の研究室メンバーに感謝したい．本研究の一部は文部科学省と日本学術振興会による科学研究補助金の援助を受けて行なわれた．

参考文献

[1] D. Avis and K. Fukuda, Reverse search for enumeration. Discrete Applied Mathematics **65** (1996) 21–46.

[2] M. Ehrgott, On matroids with multiple objectives. Optimization **38** (1996) 73–84.

[3] M. Ehrgott, Multicriteria Optimization (Second Edition). Springer, Berlin Heidelberg, 2005.

[4] M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization (2nd Corrected Edition). Springer-Verlag, Berlin New York, 1993.

[5] V. Kaibel and M. Pfetsch, Some algorithmic problems in polytope theory. In: Algebra, Geometry, and Software Systems, M. Joswig and N. Takayama, eds., Springer-Verlag, 2003, 23-47. http://www.zib.de/pfetsch/apropo/

[6] B. Korte and J. Vygen, Combinatorial Optimization: Theory and Algorithms, (3rd Edition). Springer, Berlin, 2005.

[7] C. Papadimitriou and M. Yannakakis, On the approximability of trade-offs and optimal access of web sources. Proc. 41st FOCS (2000) 86–92.

[8] E.L. Ulungu and J. Teghem, The two phase method: An efficient procedure to solve bi-objective combinatorial optimization problems. Foundations of Computing and Decision Sciences **20** (1995) 149–165.

[9] D.B. West, Introduction to Graph Theory (Second Edition). Prentice Hall, Upper Saddle River, 2001.

[10] C. Zaroliagis: Recent advances in multiobjective optimization. Proc. 3rd SAGA (2005) 45–47.

[11] E. Zitzler, M. Laumanns, and S. Bleuler, A tutorial on evolutionary multiobjective optimization. In: X. Gandibleux, M. Sevaux, K. Sörensen, V. T'kindt, eds., Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems **535** (2004) pp. 3–38.