

計算量クラス#P と n 次元多面体の体積の計算

The Computational Complexity Class #P and Computing the Volume of an n -dimensional Polytope

ネットワーク情報学部 安藤 映

School of Network and Information Ei Ando

Keywords: polytope, volume, computational complexity,

Abstract

This article aims to introduce the recent results about computing the volume of the high-dimensional polytopes. Before the results, we go through the basic definitions. Then, we introduce the recent results about the approximation algorithm for the volume of the knapsack polytope and its dual. At the end of this article, we introduce the relation between the high-dimensional volume and some combinatorial optimization problems.

1. はじめに

本稿では、必要知識を補いつつ n 次元多面体の体積に関する研究成果を紹介する。高次元での多面体の体積は、様々な理論上・実際上の問題と関連がある。問題の解決に際しては多面体の体積を計算と同等の処理が必要になる。

高次元の多面体の体積について、最も単純な例は次のような問題である。双六等で用いられるダイスには、一般的によく知られている6面のもののほかに4～100面などの様々な形状が存在している。これらの様々な n 個のダイスの面の数を a_1, a_2, \dots, a_n とする。さてこれらのダイスそれぞれは各面が同じ確率で出るとして（「一様に分布している」という）、これらのダイスを全部一度に振って、出た目の合計を計算することにすると、 n 以上 $A = a_1 + a_2 + \dots + a_n$ 以下の値が全て同じ確率で出るだろうか？答えはNoである。この違いは、一般的な6面ダイスを2つ振った場合の合計値が12である確率と、7である確率を比較すると判りやすい。では、 n 個のバラバラな面数のダイスを振った時、任意の値 b の出る確率を求めようとする、適切な問題設定の下でこれは#P-困難と呼ばれる難問になる。実はこの問題は、 n 次元の多面体の中に含まれる整数座標の点の数を数える問題であり、ダイスの面数が十分多いもの考えるときには多面体の体積を数える問題と同じといえる。

より実際的な問題では、渋滞による通過時間の変化を考慮した最短路問題が上げられる。最短路問題はカーナビなどでお馴染みの最適化問題で、与えられた地図・出発点・目的地について最短の経路を考える問題である。通常は道路区間の長さを用いて最短路を出す。しかし、場合によ

ては最短路そのものよりも目的地に到達するのにかかる時間だけが知りたいことがある。例えば消防署の立地を考える場合、出来るだけ多くの世帯に、高い確率で短い時間に到達できる経路がある立地を選ぶことが合理的である。現実には、道路の通過にかかる時間は状況に応じて様々な挙動を示すと考えられるが、そこを先程のダイスの問題と同様に一様分布に従ってランダムな値をとると仮定すると、最短路の長さがどのようなものであるかを計算する問題も一種の多面体の体積を計算する問題である。

本稿の構成は次の通りである。まず、計算量のクラスと、確率に関する定義を解説し、 n 次元の多面体の基本的な定義を示す。その後、最近の研究成果として先のダイスの問題や最短路の長さに関連のある n 次元多面体体積の計算方法を紹介する。

2. 最適化問題と計算量のクラス

最短路問題をはじめ、様々な最適化問題が存在するが、問題を解くために必要な計算機の能力に応じてクラス分けがされており、計算量クラス P および NP については[8]を読むと詳細を知ることができる。

2.1. 計算量クラスP

最もポピュラーな計算機モデルとして決定性チューリングマシンが挙げられる。これは計算機を表現する抽象モデルであり、概ね現在一般的に使用されている計算機と同等の能力があるものとして扱われる。最短路問題のように現実の計算機で素早く解決する方法が知られている問題のク

ラス, つまり決定性チューリングマシンを用いて素早く解ける問題クラスが P である.

「素早く解決できる」ということの基準は, 計算機に解かせる問題例を記述するときに必要な記号を記述する量を N とすると, 定数 c が存在して, 十分大きな N に対して N^c に比例する時間 (N の多項式時間, polynomial time) 以内に完了することを指す. P に属していない問題はどのような定数 c を用いても必要な計算時間が N^c 以下とは言えない問題である. 例えば 2^N のように N の指数に比例する時間が必要な問題があれば, その問題は P に属していないといえる.

2.2. 計算量クラス NP

計算量クラス NP は非決定性チューリングマシンを用いて多項式時間で解決可能な計算量クラスである. 非決定性という言葉の意味は, 予め決められた範囲で都合よく「予想」をし, しかもそれが当たるという意味で, 現実の計算機で非決定性チューリングマシンは実現できないと考えられている. この魔法のような計算を考える意味は, 予め問題の答えが与えられていてその答えを検算するときの動作を考えることに対応する. つまり, NP は問題の検算が多項式時間で完了する問題のクラスである. NP に属する問題にはナップサック問題, 巡回セールスパーソン問題, 整数計画問題, 数独等のパズルなどがある.

なお計算機科学の分野では $P \neq NP$ 予想がよく知られている. これは計算量クラス NP に属していても P には属していないような問題が存在するかどうかという問題である. つまり, 解があれば検算は素早く実施可能であっても, その解を問題だけの情報から素早く求めることは不可能な問題が存在するに違いないと考えられている. そのような問題の候補は NP-完全問題と呼ばれている.

2.3. 計算量クラス #P

計算量クラス #P は Valiant[10]によって示された計算量のクラスである. 定義をするには, 非決定性チューリングマシンの「予想」を含めた計算を進めた先で, 得られうる正解全ての数を出力する計算機モデル(計数機械)を考え, そのモデルによって多項式時間で計算しうる問題のクラスである. この計算機モデルも実現はできないが, 実は NP と同様 #P に属する問題の中に本当に多項式時間で解けない問題が存在するかどうかは証明されていない.

興味深いのは, 二部グラフのマッチング問題など, 解を一つ見つける問題は P に属していても, 解を数える問題は #P の中で最も難しい種類の問題(#P-完全問題)であることである. つまり, 仮に $P=NP$ であっても P と #P が等しいかどうかは自明でない. なお, #P に属する問題の中で最も難しいグループが #P-完全問題と呼ばれているが, その証明をにはすでに #P-完全であると判っている問題 A と, 難しさを証明したい問題 B の二つについて「もし問題 B が決定性チューリングマシンによって多項式時間で正確に解けた

ら, その答えを使って問題 A も決定性チューリングマシンによって多項式時間で解ける」という証明を行う. この証明だけができた場合, 問題 B は少なくとも問題 A よりは難しいことになり, 問題 B は #P-困難と呼ばれる. 加えて, 計数機械によって問題 B が解けることが示されれば問題 B は #P-完全であるという. つまり, 一つの #P-完全問題が多項式時間で解けるとその解き方を使って全ての #P-完全問題が解けることになるが, そのような解き方は未知である.

2.4. その他の計算量クラス

他にも計算量のクラスは様々なものが日々提案され, そして互いの計算量クラスの包含関係を証明した論文が発表されている. 最新の情報は Complexity Zoo [1]というサイトにまとめられており, 本稿執筆時点で 535 個の計算量クラスが掲載されている.

3. 確率および確率変数, 確率分布

3.1. 確率とは

確率という言葉は様々な意味で用いられることがあるが, 本稿でいう確率は「一様な 6 面ダイスを振って 3 が出る確率」のように厳密に定義できるものだけを扱う. フェラー[7]の言葉を借りれば, このような数学上の確率とは, 物理でいうところの「質量」に相当するものである. 数学の問題として確率を求める際の計算と, 質量を求める際の計算は合計ないし積分を求める計算をするという点で似通ったものである. 質量と確率の異なる点は, 質量であれば対象領域の質量を合計したものであるのに対し, 確率は全体領域の中で対象領域の部分の割合を求める点である. 全体領域の中に占める割合が大きい場合に我々は「起こりやすい」と解釈し, 割合が小さい場合は「起こりにくい」と解釈する. 質量の大小を我々が「重い」「軽い」と解釈するのと同様, 得られた値の解釈は値を受け取った人に任される.

3.2. 確率変数と確率分布

確率変数という概念を説明する際には, 任意の面数のダイスで, 必ずしも一様でないものを考える. 他にも, 例えば同じ人物が投げたソフトボールの遠投の距離なども確率変数とみなせる. 投手が同じであれば同じような飛距離が出るであろうが, 精密に測ると風などに左右されてその時々で異なる値になると考えられる.

ダイスの目の場合と異なり, ソフトボールの飛距離のような確率変数を表現するには「飛距離が何メートルの確率」という言い方では確率 0 というつまらない結果になる. ある人物が投げたソフトボールの飛距離を便宜上 X メートルとしよう. すると, ミリメートル単位以下に十分細かく飛距離を測定すると 2 回投げて同じ飛距離が出ることはまず起こらない. このため, どのような飛距離を想定しても「飛距離 X が正確に x メートルと等しい確率」は 0 である.

そこで、少しだけ表現を変えて「飛距離 X が x メートル以下である確率」と考えると、 X が x を超える場合と越えない場合どちらも起こりうるので、確率はゼロにはならない。数式ではこの確率を $\Pr[X \leq x]$ と書く。Pr は英語の probability の先頭二文字である。特に、「確率変数 X が x 以下である確率」は確率変数の分布関数と呼ばれ、確率変数の振る舞いを表現する方法として確率分布関数を用いることが一般的に行われる。また、確率分布関数を x で微分したものは確率密度関数と呼ぶ。例えば、 X が $[0,1]$ の間のどの値も同じ確率で取りうるとは $[0,1]$ で確率密度関数が定数であることを示し、この場合 X は「区間 $[0,1]$ 上の一様分布に従う」という。これは次の式と同じ意味である。

$$\Pr[X \leq x] = \begin{cases} 0 & x \leq 0 \\ x & 0 < x \leq 1 \\ 1 & 1 < x \end{cases}$$

3.3. 確率変数の独立性

二つの確率変数 X, Y があるときに、任意の二つの実数 x, y について X, Y の分布関数が

$$\Pr[X \leq x \wedge Y \leq y] = \Pr[X \leq x] \Pr[Y \leq y]$$

を満たすとき、二つの確率変数 X, Y は互いに独立であるという。これは、一方がどのような値を取ったところで、もう一方は影響を受けないことを表す。また、 N 個の確率変数 X_1, \dots, X_N が独立とは、任意の実数 x_1, \dots, x_N について

$$\Pr[X_1 \leq x_1 \wedge \dots \wedge X_N \leq x_N] = \Pr[X_1 \leq x_1] \dots \Pr[X_N \leq x_N]$$

であることをいう。計算の容易性のため、複数の確率変数を考える際には、互いに独立なもの考えるのが最も基本的な仮定である。

4. 高次元の多面体

高次元の多面体を記述する際には、いくつかの別々の方法が考えられる。ある方法で定義した多面体が、別の方法での定義を考えるとひどく冗長な記述を要することがある。また、ここで紹介した方法を複合することによって、多様な多面体の定義を考えることができる。

4.1. 高次元空間と半空間および多面体

n 次元空間の点を n 個の数の n 項組 (ベクトル) で表現する。つまり、 x_1, x_2, \dots, x_n が全て実数であるとすると、その n 項組 (x_1, x_2, \dots, x_n) が一つの点であり、全ての実数の組み合わせについて、この点の集合を n 次元空間と定義する。例えば $n = 2$ の場合は実数 2 つ x_1, x_2 の 2 項組 (x_1, x_2) が一つの点であり、2 次元空間は全ての実数二つの組み合わせの集合である。また、この定義に従えば 3 次元や 4 次元以上の空間を自然に定義することができる。

注意する点として、我々はせいぜい 3 次元程度の空間しか体験したことがない。このため 4 次元以上の図形を無理にイメージしようとすることは著者を含め誰にとっても困難である。そこで n 次元の空間は実数が n 個並んだものと考

え、図形的なイメージは 3 次元で起こることを n 次元に拡張できるかどうかを数式で確認する。 n が大きいときのみ確認される状況を新しく見つけた場合、学術的に興味深い現象であると考えられる。

この n 次元空間を不等式によって半分に分割することを考える。実数の定数ベクトル $\mathbf{a} = (a_1, a_2, \dots, a_n) \in R^n$ と実数 $b \in R$ に について、 $\mathbf{a} \cdot \mathbf{x} \leq b$ を満たすような点 $\mathbf{x} \in R^n$ の集合、つまり $\{\mathbf{x} \in R^n \mid \mathbf{a} \cdot \mathbf{x} \leq b\}$ を n 次元の半空間と呼ぶ。 $n = 2$ の場合、平面上を直線 $\mathbf{a} \cdot \mathbf{x} = a_1 x_1 + a_2 x_2 = b$ で区切られる二つの領域のそれぞれを半空間と呼ぶ。また、半空間の境目に当たる部分 $\{\mathbf{x} \in R^n \mid \mathbf{a} \cdot \mathbf{x} = b\}$ を超平面と呼ぶ。線形代数の知識から、この超平面内の自由度は $n - 1$ であり、超平面内の点に限れば n 項組である点 \mathbf{x} を決める n 個の実数のうち、 $n - 1$ 個の実数値を決めると自動的に残りの一つの実数が決まる。

n 次元空間の一部をうまく切り取った点集合を考えたとき、その要素のどの点も原点からの距離が有限であるようなものを作ることができる。そのような点集合を有界であるという。例えば、前述の半空間はベクトル \mathbf{a} をどのようにとっても有界ではないが、 $i = 1, 2, \dots, n$ について $0 \leq x_i \leq 1$ を満たすような点集合 $C = [0, 1]^n$ は有界である。なお点集合 C はハイパーキューブと呼ばれる。

n 次元空間の多面体とは、先に定義した半空間の有限個の積集合のうち、有界なものである。例えば上記のハイパーキューブ C は多面体である。

4.2. H-多面体

多面体のうち特に半空間を先に定義してその積集合を取る定義の方法を H-多面体と呼ぶ。H-多面体としてのハイパーキューブは $2n$ 個の不等式 $0 \leq x_i \leq 1$ ($i = 1, \dots, n$) によって表現される。

H-多面体で考えた H-多面体の頂点の数は、半空間の数に対して指数的に多くなることがある。例えば先のハイパーキューブ C の場合、頂点の集合は $\{0, 1\}^n$ で表され、頂点数は 2^n 個である。この性質は高次元の多面体に関する計算をする際に大きな問題になる。

4.3. V-多面体

上述とは別の方法で多面体を定義することがある。こちらの方法は V-多面体と呼ばれるものであるが、まず多面体の有限個の頂点 N 個を先に v_1, v_2, \dots, v_N と定義する。そしてそれらの頂点を含む凸な点集合のうち最小のもの (凸包) を多面体と定義する。ある点集合 S が凸であるとは、 S に含まれるどの二点 s_1, s_2 についても中間にある点が S に含まれることを言う。なお、点の集合 V の凸包のことを $\text{conv}(V)$ と書くことにする。例えば上述の V-多面体を表記する場合には $\text{conv}(\{v_1, v_2, \dots, v_N\})$ と書ける。

さて、V-多面体は H-多面体に書き直すことができる。 v_1, v_2, \dots, v_N のうち適当な n 個を選ぶと、それらを含む超平

面が一つ決まり、それら n 頂点の凸包も V -多面体に含まれる。このため、 V -多面体の面は必ず平面であることがわかる。さて、 N 個の頂点から n 個を選ぶ $\binom{N}{n}$ 通りの組み合わせによってできる超平面を全て考えて、そのうち V -多面体の内部と外部を分ける物だけを残せば V -多面体の面を与える超平面が得られる。この超平面を先に考えた場合、 V -多面体を H -多面体として書き直すことができる。

例として次の要領で与えられる多面体を考える。 \mathbf{e}_i を、 i 番目の座標だけ 1 で他は全て 0 である点とする。すると、原点 $\mathbf{0} = (0, 0, \dots, 0)$ および $\mathbf{e}_1, \dots, \mathbf{e}_n$ の $n+1$ 点の凸包を考えると V -多面体であり、このように $n+1$ 頂点で与えられる多面体をシンプレックスと呼ぶ。この V -多面体の面は、原点と $\mathbf{e}_1, \dots, \mathbf{e}_n$ から一つだけ点を除いたものの凸包と、それから $\mathbf{e}_1, \dots, \mathbf{e}_n$ の凸包である。それらの面は $i = 1, \dots, n$ について $\text{conv}(\{\mathbf{0}, \mathbf{e}_1, \dots, \mathbf{e}_n\} \setminus \{\mathbf{e}_i\})$ および $\text{conv}(\{\mathbf{e}_1, \dots, \mathbf{e}_n\})$ と書くことができる。 n 個の頂点を通る超平面は a_1, \dots, a_n および b を未知数として、超平面の方程式に各点の座標を代入した連立方程式を解くことで得られるが、 $b = 0$ の場合と $b \neq 0$ の場合に分けて考えると未知数を一つ減らして平面の方程式を得られる。 $b = 0$ の場合には求める超平面は原点を通るため、未知数は a_1, \dots, a_n の n 個である。また、 $b \neq 0$ の場合には $a_1x_1 + \dots + a_nx_n = b$ の両辺を b で割って、未知数は $a_1/b, \dots, a_n/b$ の n 個である。すると、例のシンプレックスの面を含む超平面の方程式として、 $x_i = 0$ ($i = 1, \dots, n$) および $x_1 + \dots + x_n = 1$ の $n+1$ 個の方程式が得られる。このシンプレックスはこれら $n+1$ 個の方程式によって定義される超平面の等号を不等号 \leq に変えて得られる半空間の積集合として、 H -多面体の方法でも与えることができる。

一般に、 V -多面体の面の数は頂点の数に対して指数的に多くなることもある。例として、正軸体という多面体は、 $2n$ 個の点 $\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n$ の凸包つまり $\text{conv}(\{\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n\})$ である。正軸体の面を考えると、多面体の内部と外部を分ける面であるには、 $\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n$ のうち n 個の凸包であることと逆符号の頂点 \mathbf{e}_i と $-\mathbf{e}_i$ ($i = 1, \dots, n$) を同時に含まないことが必要十分条件である。すると、各 $\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n$ のうち符号の選び方の数だけ面があるので、 2^n 個の面があるとわかる。

5. 体積の計算

基本的な多面体の体積を紹介したのち、最近の結果について説明する。多くの n 次元多面体は実は体積を正確に計算することが困難である。ここではナップサック多面体、幾何双対ナップサック多面体を紹介する。なお、 n 次元凸体の体積は乱数を用いた近似アルゴリズムで比較的高速に近似でき、[5] に示されたようなアルゴリズムを疑似乱数によって実装して解くことが現実的であろう。一方で、最近のトレンドとしては「乱数なしでどこまで高速なアルゴリズムが作れるか」という点が理論的な側面から注目されている。本稿ではこちらの理論的な立場からの結果を紹介する。

5.1. 基本的な多面体の体積

ハイパーキューブの体積は何次元であろうと 1 であることは容易にわかる。また、積分を n 回繰り返すことによってシンプレックス $\text{conv}(\{\mathbf{0}, \mathbf{e}_1, \dots, \mathbf{e}_n\})$ の体積は $1/n!$ と等しいことが確認できる。また、任意の $n+1$ 点 $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n$ の凸包として与えられるシンプレックス $\text{conv}(\{\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n\})$ の体積は、 $\mathbf{a}_1 - \mathbf{a}_0, \dots, \mathbf{a}_n - \mathbf{a}_0$ を並べてできる $n \times n$ 行列 A の行列式を用いて $|A|/n!$ である。この計算には多変数の変数変換の公式を用いる。正軸体 $\text{conv}(\{\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n\})$ の体積を求めるには、シンプレックス $\text{conv}(\{\mathbf{0}, \mathbf{e}_1, \dots, \mathbf{e}_n\})$ が 2^n 個集まってこの正軸体を作っていることに気が付けば直ちに $2^n/n!$ であることがわかる。

5.2. ナップサック多面体とその体積

ナップサック多面体の体積を計算する問題は、 $\#P$ -困難である体積計算問題のうち最もシンプルなものである。 n 次元ベクトル $\mathbf{a} = (a_1, a_2, \dots, a_n) \in R^n$ と実数 $b \in R$ が与えられるとき、ナップサック多面体は $\{\mathbf{x} \in [0, 1]^n \mid \mathbf{a} \cdot \mathbf{x} \leq b\}$ である。 $\#P$ -困難性の証明は Dyer と Frieze [6] によるもので、この体積が正確に計算できると、ナップサック問題という NP-完全問題の解の数がわかるというものである。ナップサック多面体の例を図 1 に示す。2 次元や 3 次元では問題は困難ではないが、高次元ではひどく手間のかかる問題になる。

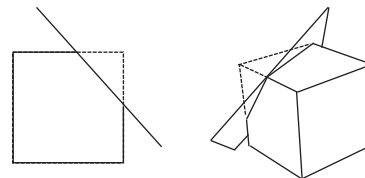


図 1. 2次元および3次元のナップサック多面体の例

ナップサック多面体の体積を計算する問題は $\#P$ -困難問題の中で良い性質を持っている。安藤と来嶋[4]は、実際の値の $1 \pm \epsilon$ 倍以内に収まる近似値であれば n^3/ϵ に比例する時間で計算を完了できることを示した。

近似アルゴリズムのアイデアは、まず区間 $[0, 1]$ 上の一様分布に従う互いに独立な確率変数 X_1, \dots, X_n を用いて、ナップサック多面体の体積が確率 $\Pr[a_1X_1 + \dots + a_nX_n \leq b]$ と等しいことを利用する。そこで不等式の左辺にて 1 番目から i 番目までの和が b 以下である確率を表す関数 $\Phi_i(b)$ を $\Phi_i(b) = \Pr[a_1X_1 + \dots + a_iX_i \leq b]$ と定義すると、確率変数の和の分布関数を求める際の公式を用いて、

$$\begin{aligned} \Phi_i(b) &= \Pr[a_1X_1 + \dots + a_{i+1}X_{i+1} \leq b] \\ &= \int \Pr[a_1X_1 + \dots + a_iX_i \leq b - a_{i+1}s] f(s) ds \\ &= \int \Phi_i(b - a_{i+1}s) f(s) ds \end{aligned}$$

と書ける。ただし、 $f(s)$ は一様分布の確率密度関数である。

この $\Phi_i(b)$ が $i = 1, \dots, n$ について n の多項式時間で計算で

きるとナップサック多面体の体積は $\Phi_n(b)$ であるが、 $\Phi_i(b)$ を計算することは容易ではない。その理由は、 $\Phi_i(b)$ は、 i が一つ増えるたびに区切りが倍々に増えてしまうからである。この計算の困難さは、ハイパーキューブの頂点数が 2^n 個あることに由来する。

そこで今回のアイデアは上述の積分をする際に $\Phi_i(b)$ を階段の形をした関数で近似する。この階段近似を行うことで i について指数的に増える関数の区切りのうち、不要な区切りをうまく飛ばして計算できる。アルゴリズムはまず $\Phi_1(b)$ を階段近似で置き換えた後、積分の結果をまた階段近似する。これを n 回繰り返すことで $\Phi_n(b)$ の近似値を得るが、正確な値とは少し異なる値になるものの、階段近似の階段の段数を $2n^2/\epsilon$ 個程度に増やすことで正確な値の $1 \pm \epsilon$ 倍以内になる。

5.3. 幾何双対ナップサック多面体とその体積

V -多面体の体積を求める問題の中で、#P-困難なものうち最もシンプルなもの次は幾何双対ナップサック多面体の体積を求める問題である。 n 次元整数ベクトル $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{Z}^n$ が入力として与えられるとき、幾何双対ナップサック多面体は $\text{conv}(\{\mathbf{a}, \pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n\})$ つまり一つの点と正軸体の凸包である。#P-困難性の証明は Khachiyan [9] によるもので、この体積が正確に計算できてしまうと、SUBSETSUM 問題という NP-完全問題の解の数がわかるというものである。なお、Khachiyan は線形計画問題に対する解法である楕円体法を考案したのと同人物である。

この多面体を「幾何双対ナップサック多面体」と呼ぶのは次の幾何双対の定義による。原点を含む点集合 $K \subseteq \mathbb{R}^n$ に対して、点集合 K^* が K の幾何双対であるとは、

$$K^* = \{y \in \mathbb{R}^n \mid \forall x \in K, x \cdot y \leq 1\}$$

ということである。例えば、ハイパーキューブ $[-1, 1]^n$ の幾何双対は正軸体 $\text{conv}(\{\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n\})$ であり、ハイパーキューブ $[-1, 1]^n$ を $\mathbf{a} \cdot \mathbf{x} \leq 1$ という不等式で切ったナップサック多面体の幾何双対が幾何双対ナップサック多面体である。

幾何双対ナップサック多面体の体積を多項式時間で近似するアルゴリズム[3]を紹介する。こちらは n^{10}/ϵ^6 に比例する時間で正確な値の $1 \pm \epsilon$ 倍以内の値を出力する。幾何双対ナップサック多面体の体積に関する近似アルゴリズムを設計するにあたり、次の手順で問題を変形する。

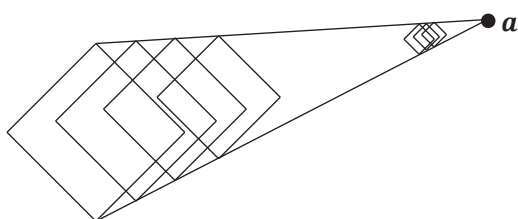


図 2 幾何双対多面体(2次元)と正軸体の無限列

まず、図 2 のように正軸体を少しずつ縮小しながら点 a

β を用意して、最初の正軸体は $\text{conv}(\{\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_n\})$ で、 i 番目の正軸体の中心座標を $(1 - \beta^i)\mathbf{a}$ とし、大きさは元の正軸体を β^i 倍したものにする。この正軸体の列を無限に続けると、点 \mathbf{a} に収束する正軸体の列ができる。このような列を考えて、無限にある正軸体の列の和集合を取ると、幾何双対ナップサック多面体を近似する多面体になる。この近似は β を 1 より僅かに小さな値にすると、元の幾何双対ナップサック多面体に近い体積になる。

この正軸体の列がキーアイデアである。第一の正軸体から第二の正軸体と重なる部分を取り除いてできる多面体を考えると、実は第 i 番目の正軸体から第 $i + 1$ 番目の正軸体と重なる部分を取り除いてできる多面体はすべて相似形になる。そこで問題は十分小さな β について、第一の正軸体から第二の正軸体が重なる部分を取り除いてできる多面体の体積を求めれば公比 β の無限級数和の公式を用いて正軸体の無限列の体積が得られる。この計算のために、第一と第二の正軸体が重なる部分の体積を近似することで、幾何双対ナップサック問題の体積を近似できる。

さて、二つの正軸体が重なる部分の体積を求める問題も実は #P-困難であるが、近似アルゴリズムはナップサック多面体の体積を近似するアルゴリズムを自然に拡張した形で得られる。詳しく言うと、中心座標 $\mathbf{c} = (c_1, \dots, c_n) \in \mathbb{R}^n$ で半径 r の正軸体は

$$\left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^n |x_i - c_i| \leq r \right\}$$

という形で表現できる。絶対値をうまく取り扱えば、正軸体はたった一つの不等式で表現できる多面体であり、二つの正軸体が重なる部分は二つの不等式で表現できる多面体である。そこで先述のナップサック多面体の体積を近似するアルゴリズムを少しだけ拡張して、ハイパーキューブおよび二つの半空間の積集合の体積を近似できる。そして同様に、二つの正軸体が重なる部分の体積を近似できる。得られた体積を第一の正軸体の体積 $2^n/n!$ から差し引いたものを初項とし、公比 β の等比級数の無限和を公式で求めると幾何双対ナップサック多面体体積の近似値を得る。

アイデアは以上であるが、このようなアルゴリズムが狙い通りの近似値を出すことを証明することが最も重要な点である。今回は特に β の値の取り方のバランスが重要である。 β が 1 から離れていると正軸体の無限列にある凸凹の影響が無視できなくなる。 β が 1 に近いと、第一と第二の正軸体の差分の体積が小さく(薄く)なり、そのため二つの正軸体の重なり部分の体積を精密に計算しなくてはならない。 β が大きすぎても小さすぎても目的の近似アルゴリズムが得られないが、うまいバランスをとることで n^{10}/ϵ^6 に比例するという(比較的大きな計算時間ではあるが)多項式時間で完了するアルゴリズムを示すことができる。アルゴリズムの実行時間のほとんどは正軸体の重なり部分の体積

5.4. 最短路問題との関連

冒頭で述べた最短路の問題はどのような多面体の体積だろうか. 通常, 最短路問題を考える際には頂点の集合 V と辺の集合 E を持つ離散グラフ $G=(V,E)$ および, 辺の長さ $\ell:E\rightarrow R$, 出発地点 s , 目的地 t が問題として与えられる($s,t\in V$). 各辺 $e\in E$ の長さ $\ell(e)$ が確定的な値であればダイクストラ法を用いて, 適切な実装の下で高速に最短路問題を解くことができる.

グラフ G が道路地図を表している, 辺 $e\in E$ が道路のひと区間を表している場合, より現実的に近いモデルを考えるため, 各辺を通過するのにかかる時間が渋滞等の理由で変化することを表現する. そこで, 各辺 e の長さ $\ell(e)$ は確率変数 X_e であると考え. 現実をうまくモデル化するには信頼できる統計データを元にして確率変数 X_e の確率分布関数を適切に設定する必要があるが, ここでは0から各辺に指定する値 a_e までの実数を一様な確率でとる場合を考える. この場合, X_e は区間 $[0,1]$ で一様に分布していて, グラフ G の中で始点 s から終点 t までの道全体の集合を Π とすると, 最短路の長さが実数 b 以下である確率は確率の定義から

$$\Pr\left[\bigvee_{\pi\in\Pi}\sum_{e\in\pi}a_eX_e\leq b\right]=1-\Pr\left[\bigwedge_{\pi\in\Pi}\sum_{e\in\pi}a_eX_e>b\right]$$

と書くことができる. すると, 右辺に現れる確率は全ての辺 $e\in E$ について $0\leq x_e\leq 1$, 始点 s から終点 t までの全ての道 $\pi\in\Pi$ について $\sum_{e\in\pi}a_ex_e>b$ という不等式で表される半空間すべての積集合(多面体)の体積が, $|E|$ 次元の領域 $[0,1]^{|E|}$ に占める割合と一致する.

例えば図3のようなグラフで s から t までの最短路長さについて, 最短路の長さが b 以下である確率を考える.

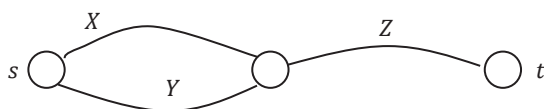


図3 最短路長さの確率分布を求める例

各辺の長さは確率変数 X,Y,Z で, それぞれ区間 $[0,1]$ 上の一様分布に従い, 互いに独立とする. すると最短路の長さの分布関数は $1-\Pr[X+Z>b\wedge Y+Z>b]$ であるが, これはちょうど立方体 $[0,1]^3$ の体積1から $x+z>b$ かつ $y+z>b$ という条件を満たす部分の体積を差し引いた値と等しい.

ナップサック多面体の体積を近似するアルゴリズムを拡張すると, 閉路のない有向グラフにおいて, パス幅というパラメータが小さい場合に最長路の長さの確率分布の計算について高速な近似アルゴリズムを示せる[3]. 「パス幅が小さい」とは, 直観的には細長い形状のグラフといえる. このアルゴリズムは同様のグラフについて最短路についても適用できる. なお, 閉路のある無向グラフの最短路に対して利用できるアルゴリズムを得るには煩雑な拡張が必要である.

6. おわりに

本稿では, ナップサック多面体および幾何双対ナップサック多面体の体積を近似計算する方法について説明をし, 拡張した最短路問題との関連を示した. 現在の所, 本稿の著者である安藤は本学ネットワーク情報学部の土屋翔一准教授と共同研究として, n 次元多面体の幾何双対性と体積の多項式時間計算可能性についての研究を進めている. 本稿執筆時点ではさらに, 一つの点と一つのハイパーキューブの凸包, 正軸体を一つの半空間で切り取ってできる多面体の体積の計算について発表予定である. 現在の所, 互いに幾何双対の関係にある多面体の体積を計算する問題を考える場合, その問題の計算複雑さのクラスが両方で一致するかどうかについて研究を進める方針である.

参考文献

- [1] Complexity Zoo, <https://complexityzoo.uwaterloo.ca/Complexity_Zoo>, (参照 2019-1-7)
- [2] E. Ando, S. Kijima (2017) An FPTAS for the Volume of Some V -polytopes - It is Hard to Compute the Volume of the Intersection of Two Cross-Polytopes. *Proc. of COCOON 2017*, LNCS 10392, pp. 13-24.
- [3] E. Ando (2017) An FPTAS for Computing the Distribution Function of the Longest Path Length in DAGs with Uniformly Distributed Edge Lengths, *Proc. of WALCOM 2017*, LNCS 10167, pp. 421-432.
- [4] E. Ando and S. Kijima (2016) An FPTAS for the Volume Computation of 0-1 Knapsack Polytopes Based on Approximate Convolution, *Algorithmica*, Vol. 76, Issue 4, pp 1245-1263.
- [5] B. Cousins, S. Vempala (2015) Bypassing, K.L.S.: Gaussian cooling and an $O^*(n^3)$ volume algorithm, *Proc. of STOC 2015*, pp. 539-548.
- [6] M. Dyer and A. Frieze (1988) On the complexity of computing the volume of a polyhedron, *SIAM Journal on Computing*, 17(5), 967-974.
- [7] W. Feller (1968) An Introduction to Probability Theory and Its Applications, Vol. 1, Wiley.
- [8] M. Garey and D. Johnson (1979) Computers and Intractability: A Guide to the Theory of NP Completeness, W. H. Freeman and Company.
- [9] L. Khachiyan (1989) The problem of computing the volume of polytopes is #P-hard, *Uspekhi Mat. Nauk.*, 44, pp. 199-200.
- [10] L.G.Valiant (1979) The complexity of computing the permanent, *Theoretical Computer Science*, 8, pp. 189-201.