BTST 法の 3 次元への拡張について

M2014SS007 中山莉奈
指導教員：鈴木敦夫

1 はじめに

本研究では，2 次元の Big Triangle Small Triangle method (BTST 法) [3] を 3 次元に拡張した Big Tetrahedron Small Tetrahedron method (BTeSTe 法) を新たに提案する．

2 次元の BTST 法は目的関数が非凸である配置問題の最適解を分枝限定法を用いて求める解法のことである．一般的に，目的関数が非凸である配置問題は局所最適解が複数存在する．そのため，このような配置問題は大域的最適解を見つけることが困難である．それに対して，BTST 法は大域的最適解を比較的簡単に見つけることができる．Drezner and Suzuki[3] では迷惑施設配置問題と WAR 問題 (Weber problem with Attraction and Repulsion) についてその有効性が示されていた．また，Drezner and Brimberg[2] では真円度問題についてその有効性が示されていた．しかし，従来の BTST 法では 2 次元の配置問題しか解くことができない．そこで更に多くの問題が解けるよう，BTST 法を 3 次元に拡張した BTeSTe 法を提案する．また，この解法を MATLAB 上で実現し，計算機実験を行うことでその有効性を確認する．

ここで，BTeSTe 法の概略を記す．BTeSTe 法は 2 次元の場合と同様に分枝限定法を用いる．制約条件として，最適解は需要点の凸包内にあることとし，扱う問題は非凸線形最適化問題である．BTeSTe 法の手順は，初めに制約領域内をドローネ分割を用いて需要点を重なりのない四面体に分割する．次に四面体ごとに目的関数の Lower Bounds(LB) を計算する．そして，計算した LB と実行可能解から求めた Upper Bound(UB) を比較する．この UB より値の大きい LB を持つ四面体を考慮対象から除くことで最適解の存在する領域を狭めていく．これにより，従来は厳密解を求めることが困難であった非凸線形最適化問題に対しても厳密解を求めることができる．

BTST 法より以前に，非凸線形最適化問題に対して厳密解を求めることができる解法が提案されていた．その解法は Big Square Small Square technique (BSSS 法) [4] である．BSSS 法は BTST 法の三角形の代わりに四角形を用いて解を求める解法である．BSSS 法は初めに実行可能領域である四角形を 1 つ置き，その四角形を分割していくことで最適解の存在する領域を狭めていく．この解法と比べ BTST 法の利点は，解が凸包内に存在するかどうかを調べる必要がないことである．理由はドローネ分割を用いるからである．BSSS 法は初めに全ての需要点を含むような四角形で区切るため，凸包外も解の検索範囲に含まれる．そのため，凸包内に解を限定する場合は四角形が凸包内に存在するかどうかを調べる必要がある．逆に欠点は，需要点数が多い場合 BSSS 法より計算時間が長くなることである．理由は，BTST 法は初めに需要点 n をドローネ分割することで作成した $2n-5$ の三角形の計算時間が必要だからである．BSSS 法を 3 次元に拡張した解法として Big Cube Small Cube method (BCSC 法)[7] も提案されている．

本研究の目的は，以下の 3 つの 3 次元の配置問題について BTeSTe 法を適用し，その有効性を確かめることである．

- 迷惑施設配置問題 [5]
- WAR 問題 [6]
- 真球度問題

ここで，3 つの 3 次元の配置問題について簡単に記述する．前提として，各配置問題は空間に分布した需要点とその重みを持つこととする．

迷惑施設配置問題とは，新しく施設を配置するとき危険や被害が最小となる地点を決定する配置問題のとこである．例として，原子力発電所やゴミ処理場の配置問題が挙げられる．

WAR 問題とは，Weber 問題の重みに負の値が加わった配置問題のことである．また，Weber 問題は配置場所から既存施設へのユークリッド距離の重み付きの総和が最小となる地点を決定する配置問題のことである．

真球度問題とは，部品などの表面から幾つかの点を選びその座標をデータとしたとき，球とみなしてよいかを判定する問題のことで鋼球の製品検査などに用いられている．本研究では，半径とユークリッド距離間の差の絶対値の総和を最小化することを考える．

本研究では 3 次元の BTeSTe 法を上記に挙げた 3 つの配置問題に適用し，MATLAB 上で計算機実験を行うことでその有効性を確かめる．また，BCSC 法との比較を計算機実験を用いて行う．

2 BTeSTe 法のアルゴリズム

2 次元の BTST 法を 3 次元に拡張したときのアルゴリズムを説明する．本研究は最小化問題を仮定する．各需要点の座標 $X_i = (x_i, y_i, z_i)(i = 1, ..., n)$ と重み $w_i(i = 1, ..., n)$ は所与とする．そして制約条件として，最適解は需要点の凸包内にあることとし，扱う問題は非凸線形最適化問題である．目的関数の Lower Bounds の計算方法は Drezner[1] に詳しく記載されている．

1. 需要点を用いた 3 次元ドローネ分割
   需要点はドローネ分割を用いて重なりのない四面体に分割する．ドローネ分割する理由は，実行可能領域全体がカバーできるからである．ここで，ドローネ分割

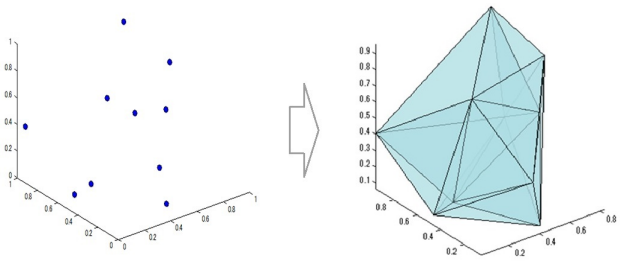

図 1　凸多角形のドローネ分割の一例

の一例を図 1 に示す．

2. Upper Bound (UB) の計算
   各四面体の重心を用いて評価を行う．そして，それらの中で最小の目的関数値を暫定解とし，UB と設定する
3. 各四面体の Lower Bounds (LB) の計算
4. $UB/(1+\epsilon)$ と LB の比較 (ϵ：十分小さい値)
   (a) $LB > UB/(1+\epsilon)$　のとき
   その LB を持つ四面体を考慮対象から除外する
   (b) $LB < UB/(1+\epsilon)$　のとき
   その四面体内に解が存在する可能性があるため，リストにその四面体の情報を格納する．ここでいうリストとは，考慮対象である四面体の情報を管理するものである．格納する情報は四面体の頂点番号と LB の値である．リストはヒープを用いて管理し，常に一番上の情報がリスト内で最小の LB（LB_{min}）を持つ四面体になるようにする．LB_{min} の値が $UB/(1+\epsilon)$ より大きいとき，アルゴリズムを終了する．このとき，最適解は UB となる．LB_{min} の値が $UB/(1+\epsilon)$ より小さいとき，その四面体を 8 つの小さな四面体に細分割する．四面体の細分割方法は，はじめに各辺の中点をとり，元の頂点 1 つと 3 つの中点から成る四面体を 4 つ作成する (図 2 の Ⅰ，Ⅱ，Ⅲ，Ⅳ)．

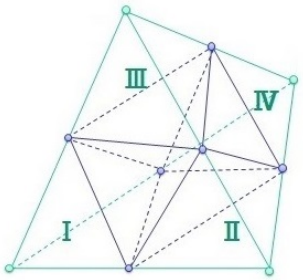

図 2　頂点と中点からなる四面体

次に，残りの中点のみで構成された八面体を 4 つの四面体に分割する．分割方法は 3 本の対角線のうち，1 番短い対角線を挿入することで 4 つの四面体に分割する (図 3 の Ⅴ，Ⅵ，Ⅶ，Ⅷ)．ここで 1 番短い対角線を用いる理由は，計算時間の短縮を計るためである．1 番短い対角線から構成される各四面体は正四面体に近い形に分割できる．そのため，極端に細長い形の四面体が減り計算時間の短縮につながると考える．このようにして 1 つの四面体を 8 つの小さな四面体に細分割する．その後，細分割した 8 つの小さな四面体の重心での目的関数値と LB をそれぞれ求める．目的関数値は UB より値が小さくなったとき UB の値をその目的関数値に更新する．LB は 8 つの小さな四面体のうち 1 つをリスト先頭の分割前の四面体と入れ替えダウンヒープを行う．残り 7 つの小さな四面体は 1 つずつリストの最後に追加しアップヒープを行うことでリストを管理する．LB_{min} の値が $UB/(1+\epsilon)$ より大きくなるまでその比較を繰り返す．

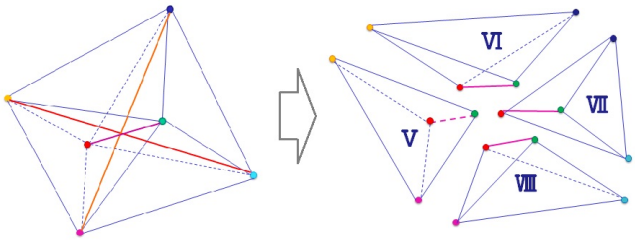

図 3　八面体の分割方法

3　配置問題の定式化と計算機実験の結果

ここでは 3 つの配置問題について説明し，これらの配置問題に BTeSTe 法，BCSC 法を適用し計算機実験を行なった結果を記す．使用する計算機は Windows7 Intel(R)Core(TM) 2 Duo CPU P8700 @2.53GHz 2.53GHz，ソフトウェアは MATLAB R2013a である．また，計算機実験は $\epsilon = 10^{-6}$ で行い，需要点の座標はランダムに与える．そして，10 個の問題を作成し計算した結果の平均を記す．2 つの解法は制約条件として，最適解は需要点の凸包内にあることと仮定する．本研究は最小化問題を取り扱う．

次に，表記方法を定義しておく．$\{X_1, ..., X_n\}$ は需要点 n の集合で，それらの需要点 i の座標は $X_i = (x_i, y_i, z_i)$ と示し，配置場所 X の座標は $X = (x, y, z)$ と示す．このとき配置場所 X と需要点 i 間のユークリッド距離を $d_i(X)$ とする．図形 s の頂点 T_j の座標は X_{sT_j} で示す．X_{sc} は図形 s の重心座標を示す．j は図形の頂点番号を示し，BTeSTe 法のとき $j = 1, ..., 4$，BCSC 法のとき $j = 1, ..., 8$ となる．

3.1　迷惑施設配置問題

迷惑施設配置問題は各需要点の迷惑度の大きさを $w_i \geq 0 (i = 1, ..., n)$ とする．この迷惑度は距離の 2 乗に反比例して減少すると仮定する．このとき，迷惑度の総和が最小となる点が解となる．そのため，目的関数は式 (1) のようになる [3]．

$$F(X) = \sum_{i=1}^{n} \frac{w_i}{d_i^2(X)} \tag{1}$$

迷惑施設配置問題は配置場所が需要点と等しい ($X = X_i$) とき，ユークリッド距離が $d_i(X) = 0$ となる．そのため，迷惑度が ∞ となり，目的関数の一部が ∞ になる．したがって，迷惑施設配置問題の目的関数は非凸である．

3.1.1 Lower Bounds の計算方法

迷惑施設配置問題の LB の計算方法は Drezner and Suzuki[3] の考え方を用いる．$d_i^2(X) = x$，$d_i^2(X_{sc}) = a$ とし，関数 $y = 1/x$ の $(a, 1/a)$ における接線の方程式を利用する．つまり，式 (2) のように下から値をおさえる．

$$\frac{1}{x} \geq \frac{1}{a} - \frac{1}{a^2}(x - a) = \frac{1}{a}(2 - \frac{x}{a}) \tag{2}$$

式 (2) を利用して図形 s の重心での目的関数値を式 (3) のように下からおさえる．

$$F(X_{sc}) \geq \min_j L_{sT_j} \tag{3}$$
$$L_{sT_j} = \sum_{i=1}^{n} \frac{w_i}{d_i^2(X_{sc})} \left(2 - \frac{d_i(X_{sT_j})}{d_i^2(X_{sc})}\right)$$

このとき，各図形 s の LB は下記のようになる．

$$LB_s = \min_j L_{sT_j}, \quad (s = 1, ..., S)$$

3.1.2 計算機実験の結果

MATLAB を用いて算法を実現し，計算した迷惑施設配置問題の結果を表 1 に記す．迷惑度はランダム ($w_i \leq 1$) に与えた．

表 1 迷惑施設配置問題の計算結果

需要	反復回数		計算時間 (秒)	
点数	BTeSTe	BCSC	BTeSTe	BCSC
10	2210.6	3969.5	3.73	219.70
20	2179.1	7710.6	5.05	733.59
50	1994.4	8783.8	8.44	1499.32
100	5441.7	7803.1	15.35	1485.72

3.2 WAR 問題

WAR 問題は Weber 問題の各需要点の重み w_i が正負の値をとる配置問題のことである．Weber 問題の目的関数は既存施設へのユークリッド距離の重み付きの総和の最小化のため，目的関数は式 (4) のようになる [3].

$$F(X) = \sum_{i=1}^{n} w_i d_i(X) \tag{4}$$

式 (4) は式 (5) のように重みの正負で分けることができる．需要点が正の重みを持つときを I^+，負の重みを持つときを I^- と仮定する．

$$F^+(X) = \sum_{i \in I^+} w_i d_i(X), \quad F^-(X) = \sum_{i \in I^-} \{-w_i d_i(X)\} \tag{5}$$

このとき $F^+(X)$，$F^-(X)$ はどちらも凸になる．WAR 問題の目的関数は $F(X) = F^+(X) - F^-(X)$ の最小化となる．しかし，この 2 つの凸関数間の差は一般的に局所最適解を含む．そのため，WAR 問題の目的関数は凸になるとは限らない．

3.2.1 Lower Bounds の計算方法

WAR 問題の LB の計算方法は Drezner and Suzuki[3] の考え方を用いる．式 (6) のように $F^+(X)$ の接平面を用いて下から値をおさえる．

$$\begin{aligned} F^+(X_{sT_j}) &\geq \overline{F}(X_{sT_j}) \\ &= F^+(X_{sc}) + \sum_{i \in I^+} w_i \Big\{ \frac{x_{sc} - x_i}{d_i(X_{sT_j})}[x_{sT_j} - x_{sc}] \\ &+ \frac{y_{sc} - y_i}{d_i(X_{sT_j})}[y_{sT_j} - y_{sc}] + \frac{z_{sc} - z_i}{d_i(X_{sT_j})s}[z_{sT_j} - z_{sc}] \Big\} \end{aligned} \tag{6}$$

よって，図形 s の重心での目的関数値は式 (7) のように下からおさえることができる．

$$F(X_{sc}) = F^+(X_{sc}) - F^-(X_{sc}) \geq \min_j L_{sT_j} \tag{7}$$
$$L_{sT_j} = \overline{F}(X_{sT_j}) - F^-(X_{sT_j})$$

このとき，各図形 s の LB は下記のようになる．

$$LB_s = \min_j L_{sT_j}, \quad (s = 1, ..., S)$$

3.2.2 計算機実験の結果

MATLAB を用いて算法を実現し，計算した WAR 問題の結果を表 2 に記す．重みはランダム ($-1 \leq w_i \leq 1$) に与えた．

表 2 WAR 問題の計算結果

需要	反復回数		計算時間 (秒)	
点数	BTeSTe	BCSC	BTeSTe	BCSC
10	327.6	114.3	0.65	4.93
20	708.5	197.0	1.27	15.39
50	1401.0	283.5	2.52	28.42
100	2368.9	480.5	4.54	87.89

3.3 真球度問題

真球度問題は球と見なしてよいかどうかを判定する問題のことである．本研究は式 (8) のように半径 r とユークリッド距離 $d_i(X)$ の差の絶対値の総和が最小となる点を求める．

$$F(X) = \sum_{i=1}^{n} |d_i(X) - r| \tag{8}$$

しかし，この問題は $d_i(X) - r$ の計算より正負の値が混在するため解くことが困難である．そこで，$d_i(X)$ のメディアン距離を半径 r と仮定し，Drezner and Brimberg[2] で提案された順序メディアン問題を用いることで真球度問題を式 (9) のように定式化する．

$$F(X)=\sum_{i=1}^{n}|d_i(X)-\text{median }\{d_i(X)|i\in n\}| \qquad (9)$$

$a_{(q)}(X)$ は $d_i(X)$ の距離を短い順に並び替えたもの $(a_{(1)}(X)\le ...\le a_{(n)}(X))$ とする．このとき，式 (9) は式 (10) のように置き換えることができる．

$$\begin{aligned}F(X)&=\sum_{i=1}^{n}|d_i(X)-\text{median}\{d_i(X)\}|\\&=\sum_{q>n/2}\Big(a_{(q)}(X)-\text{median}\{d_i(X)\}\Big)\\&\quad-\sum_{q\le n/2}\Big(a_{(q)}(X)-\text{median}\{d_i(X)\}\Big)\\&=2\sum_{q>n/2}a_{(q)}(X)-\sum_{i\in n}d_i(X)\end{aligned} \qquad (10)$$

この 2 つの凸関数間の差は一般的に局所最適解を含む．そのため，真球度問題の目的関数は凸になるとは限らない．

3.3.1 Lower Bounds の計算方法

真球度問題の LB の計算方法は Drezner and Brimberg[2] の考え方を用いる．式 (11) のように $2\sum_{q>n/2}a_{(q)}(X)$ の接平面を用いて下から値をおさえる．$(x_{(q)},y_{(q)},z_{(q)})$ は $a_{(q)}(X_{sc})$ に対応する需要点 (q) の座標を示す．

$$\begin{aligned}U(X_{sT_j})=2\Big\{\sum_{q>\frac{n}{2}}\Big(a_{(q)}(X_{sc})+\frac{(x_{sc}-x_{(q)})(x_{sT_j}-x_{sc})}{a_{(q)}(X_{sc})}\\+\frac{(y_{sc}-y_{(q)})(y_{sT_j}-y_{sc})}{a_{(q)}(X_{sc})}+\frac{(z_{sc}-z_{(q)})(z_{sT_j}-z_{sc})}{a_{(q)}(X_{sc})}\Big)\Big\}\end{aligned} \qquad (11)$$

したがって，図形 s の重心での目的関数値は式 (12) のように下からおさえることができる．

$$F(X_{sc})\ge\min_j\Big\{U(X_{sT_j})-\sum_{i=1}^{n}d_i(X_{sT_j})\Big\} \qquad (12)$$

このとき，各図形 s の LB は下記のようになる．

$$LB_s=\min_j\Big\{U(X_{sT_j})-\sum_{i=1}^{n}d_i(X_{sT_j})\Big\},\quad (s=1,...,S)$$

3.3.2 計算機実験の結果

MATLAB を用いて算法を実現し，計算した真球度問題の結果を表 3 に記す．半径 $(r=20)$ から $p^{1/5}$（但し，p は 0 から 1 の一様分布に従う乱数）を引いた結果の座標データを用いている．

表 3 真球度問題の計算結果

需要点数	反復回数	計算時間 (秒)
10	1970.5	4.61
20	1200.7	3.20
50	894.6	3.41
100	789.8	4.23

次に球面内に予め四面体または立方体を入れ，その図形から解法を用いて計算を行った結果を表 4 に記す．

表 4 球面内に予め図形を入れた時の計算結果

需要	反復回数		計算時間 (秒)	
点数	BTeSTe	BCSC	BTeSTe	BCSC
10	1706.7	782.0	3.95	3.29
20	1639.3	567.5	4.28	2.67
50	714.1	275.1	2.56	1.80
100	517.8	237.3	2.58	2.19

3.4 考察

BTeSTe 法，BCSC 法を用いた計算機実験の結果より，解が凸包上になる可能性が高い配置問題は BTeSTe 法の方が有効であることがわかった．また，解の大まかな場所が分かる問題は，その部分を含むような図形を初めに入れ，そこから計算を行った方が計算時間が早くなることがわかった．その場合，BTeSTe 法と BCSC 法では大きな差はなかった．

4 おわりに

本研究は BTST 法を 3 次元に拡張した BTeSTe 法を新たに提案した．そして，この解法を 3 つの 3 次元の配置問題に適用し，計算機実験を行うことで実用的な時間内で問題が解けることを確認した．

参考文献

[1] Z. Drezner, A general global optimization approach for solving location problems in the plane, *Journal of Global Optimization*, **37**, 305–319, 2007.

[2] Z. Drezner, J. Brimberg, Fitting concentric circles to measurements, *Mathematical Methods of Operations Research*, **79**, 119–133, 2014.

[3] Z. Drezner, A. Suzuki, The big triangle small triangle method for the solution of non-convex facility location problems, *Operations Research*, **52**, 128–135, 2004.

[4] P. Hansen, D. Peeters, D. Richard, J.F. Thisse, The minisum and minimax location problems revisited,*Operations Research*, **33**, 1251–1265, 1985.

[5] 中山莉奈, 鈴木敦夫, BTST 法の 3 次元への拡張について, 日本 OR 学会 2015 年春季研究発表会, 1-D-4, 2015.

[6] R. Nakayama, A. Suzuki, Z. Drezner, T. Drezner, The Big Tetrahedron Small Tetrahedron Method for Three Dimensional Location Problems, INFORMS. Philadelphia, 2015-11.

[7] A. Schobel, D. Scholz, The big cube small cube solution method for multidimensional facility location problems ,*Computers & Operations Research*, **37**, 115–122, 2010.