

凸でない目的関数をもつ最適配置問題に対する BTST 法

鈴木 敦夫

最適配置問題の中には凸でない目的関数をもつものが数多くあり、その厳密解を求めることは困難であった。われわれは、そのような問題の中のいくつかの問題に対して適用可能な簡単な手法を考案した。それが BTST (Big Triangle Small Triangle) 法である。本稿では南山大学で開催されてきた「都市の OR」ワークショップで、BTST 法を適用した最適配置問題について発表してきたものをまとめて紹介する。

キーワード：BTST 法、最適配置問題、ドロネ三角網

1. はじめに

最適配置問題には、凸でない目的関数をもつ問題が数多くある。われわれはそれらの問題に適用できる簡単な方法を考案して BTST (Big Triangle Small Triangle) 法と名付け、上記の問題のいくつかに適用してその有効性を確認している。

この BTST 法は、BSSS (Big Square Small Square) 法 [1] に触発されて考案した。文献 [1] では、対象領域を正方形に分割して、解が存在する領域を限定していく方法を提案している。それに対して、BTST 法は対象領域を三角形分割して解が存在する領域を限定している。BSSS 法は対象領域を正方形に分割するので対象領域が多角形の場合に、その多角形を含む正方形を分割しなくてはならず、対象領域に含まれない、もしくは一部が含まれない正方形の処理に計算時間がかかってしまう。このことは文献 [2] で 3 次元の WAR (Weber Problem with Attraction and Repulsion) の場合に示されている。

BTST 法は、筆者とカリフォルニア州立大学フラトン校のズヴィ・ドレズナー (Zvi Drezner) 教授との共同研究の成果 [3] である。以来、南山大学で開催される「都市の OR」ワークショップでここで紹介する問題を発表してきた。本稿ではこれらをまとめて紹介する。

2. BTST 法の概要

ここでは、まず 2 次元の最小化問題に対しての BTST 法の概要を紹介する。BTST 法では、解の存在する領

域を三角形分割し、各三角形の重心で目的関数を計算してその最小値を求める。三角形分割にはドロネ三角網を用いる。この値は目的関数最小値の上界であり、上限に近い値と考えられるので、以降一時的上限と呼ぶことにする。

一方、各三角形で目的関数の下界をある方法で計算する。この下界が上で求めた一時的上限より大きい三角形の中には、最適解は存在しないので、そのような三角形は以降考慮の対象からはずす。残った三角形のうち、最も小さな下界をもつ三角形を、四つの相似な三角形に分割し、それぞれの三角形の重心で目的関数を計算して、一時的上限を更新する。これを一時的上限と各三角形の下界の最小値の差が十分に小さくなるまで繰り返す。この差が十分小さくなったとき、その一時的上限を目的関数の最小値とし、その値を与える三角形の重心を最適解とする。

このようなごく簡単な方法で、今まで厳密解を求めることができなかった問題を解くことができるようになった。次節以降では、WAR を例として BTST 法を詳しく紹介する。

3. WAR に対する BTST 法の適用

ウェーバー問題は最適配置問題の古典的な問題で、多くの研究が行われている。ウェーバー問題は、与えられた需要点への距離の総和を最小にするような点を求める問題である。与えられた需要点とその座標を $P_1(x_1), \dots, P_n(x_n)$ 、それらの需要点の重みを $w_i (> 0), i = 1, \dots, n$ 、求める点とその座標を $P(x)$ とすると、ウェーバー問題は、以下のようにあらわされる。

すずき あつお

南山大学理工学部システム数理学科

〒466-8673 名古屋市昭和区山里町 18

atsuo@nanzan-u.ac.jp

$$\min \sum_{i=1}^n w_i \|\mathbf{x} - \mathbf{x}_i\| \quad (1)$$

ただし、 $\|\cdot\|$ はユークリッド距離をあらわす。ウェーバー問題の目的関数 (1) は \mathbf{x} の凸関数であり、ワイツフェルドの算法 [4] で提案された降下法で最小値を求めることができる。

ウェーバー問題では、与えられた点の重み w_i は正と仮定しているが、今この仮定をはずして、 w_i は正でも負でもよいとしたらどうなるであろうか。これが WAR である。正の重みは、施設が近くにあってほしいことを、負の重みは施設が遠くにあってほしいことをあらわしている。 $I^+ = \{i | w_i > 0\}$, $I^- = \{i | w_i < 0\}$ とすると、WAR の目的関数は、

$$\begin{aligned} f_1(\mathbf{x}) &= \sum_{i=1}^n w_i \|\mathbf{x} - \mathbf{x}_i\| \\ &= \sum_{i \in I^+} w_i \|\mathbf{x} - \mathbf{x}_i\| - \sum_{i \in I^-} (-w_i) \|\mathbf{x} - \mathbf{x}_i\| \end{aligned} \quad (2)$$

となる。(2) の最右辺の第 1 項と第 2 項は、(1) で述べたように凸関数である。すなわち、 $f(\mathbf{x})$ は凸関数の差となり、凸関数ではなくなる。WAR は凸でない目的関数をもつ問題となり、ウェーバー問題に対するような降下法では、局所最適解しか求めることはできない。ここで、WAR の実行可能領域は与えられた点の凸包内であると仮定する。さもないと、無限遠点が解になる可能性があるためである。

WAR に対して BTST 法を適用すると以下のようになる。

1. 実行可能領域の三角形分割
実行可能領域を三角形分割する。ここでは、ドロネ三角形に分割する。
2. 一時的上限の計算
1 のすべての三角形の重心で目的関数の値を計算し、その中で、最も小さなものを一時的上限 UB とする。
3. 下界の計算と初期三角形リストの作成
1 のすべての三角形について、以下のことを行う。
 - (a) 下界の計算
4 節で述べる方法の一つによって、その三角形内部の目的関数値の下界 LB を求める。
 - (b) 初期リストの作成
その三角形について、 $LB < UB/(1+\epsilon)$ ならば、解が存在する可能性のある三角形と

してリスト L に入れる。

4. 分枝限定法

以下を L が空になるまで繰り返す。算法が終了したとき、 UB を目的関数の最小値、 UB を与える三角形の重心を最適解として出力する。

(a) 三角形の再分割

L から下界の最も小さい三角形を取り出して、その三角形の 3 辺の中点を頂点として加え、相似な四つの三角形に分割する。

(b) UB の更新

上の四つの三角形の重心で目的関数値を計算し、一時的上限 UB より小さいものがあれば、 UB をその値に更新する。

(c) L の更新

上の四つの三角形の下界 LB をそれぞれ計算し、 $LB < UB/(1+\epsilon)$ ならば L にその三角形を加える。

上記の算法の中で、三角形のリスト L は、三角形の下界をキーとしたヒープを用いる。 UB の計算は、対象としている三角形の重心の座標を (2) に代入すればよい。 LB の計算は、複数の方法があるので、それらについて次節で詳しく紹介する。

4. BTST 法の WAR への適用に際する下界の計算

3 節で述べた算法の中で、 LB の計算は算法の効率にとって重要である。ここでは、二つの計算方法を紹介する。一つは、与えられた点から対象とする三角形への最短距離を用いて LB を計算する方法である。この方法は、BSSS 法でも用いられた方法で、わかりやすいがあまり良い下界は求められない。今一つは、対象とする三角形の重心での目的関数の接平面を用いる方法である。こちらの方法のほうがより良い下界を求めることができる。

対象とする三角形を T とし、その三つの頂点とその座標を $T_1(\mathbf{t}_1), T_2(\mathbf{t}_2), T_3(\mathbf{t}_3)$ 、点 $P_i(\mathbf{x}_i), i = 1, \dots, n$ から T への最短距離を $d(T, \mathbf{x}_i), i = 1, \dots, n$ とする。このとき、(2) の右辺第 1 項の $\|\mathbf{x} - \mathbf{x}_i\|$ を $d(T, \mathbf{x}_i)$ で置き換えると、

$$\begin{aligned} f_1(\mathbf{x}) &= \sum_{i=1}^n w_i \|\mathbf{x} - \mathbf{x}_i\| \\ &\geq \sum_{i \in I^+} w_i d(T, \mathbf{x}_i) - \sum_{i \in I^-} (-w_i) \|\mathbf{x} - \mathbf{x}_i\| \end{aligned} \quad (3)$$

と評価できる。(2)の右辺第2項は凸関数であり、それに負の符号を付しているので凹関数となる。 T の中で、この関数の最小値は、 T の頂点のいずれかで達成されるので、下界として、

$$LB_1 = \sum_{i \in I^+} w_i d(T, \mathbf{x}_i) + \min_{k=1,2,3} \left\{ - \sum_{i \in I^-} (-w_i) \|\mathbf{t}_k - \mathbf{x}_i\| \right\} \quad (4)$$

が得られる。

次の下界の計算方法は、(2)の第1項の接平面を利用する方法である。ここで、

$$f_1^+(\mathbf{x}) = \sum_{i \in I^+} w_i \|\mathbf{x} - \mathbf{x}_i\| \quad (5)$$

とする。 $f_1^+(\mathbf{x})$ の三角形 T の重心 $\mathbf{x}_g = (a, b)$ における接平面の方程式は、 $\mathbf{x}_i = (x_i, y_i)$ として、

$$z = f_1^+(\mathbf{x}_g) + \sum_{i \in I^+} w_i \left\{ \frac{(x_i - a)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (x - a) + \frac{(y_i - b)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (y - b) \right\} \quad (6)$$

となる。 $f_1(\mathbf{x})$ は凸関数だから、 $\mathbf{x} = (x, y) \in T$ に対して図1でわかるように、

$$f_1^+(\mathbf{x}) \geq f_1(\mathbf{x}_g) + \sum_{i \in I^+} w_i \left\{ \frac{(x_i - a)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (x - a) + \frac{(y_i - b)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (y - b) \right\} \quad (7)$$

すなわち、

$$f_1(\mathbf{x}) \geq f_1^+(\mathbf{x}_g) + \sum_{i \in I^+} w_i \left\{ \frac{(x_i - a)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (x - a) + \frac{(y_i - b)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (y - b) \right\} - \sum_{i \in I^-} (-w_i) \|\mathbf{x} - \mathbf{x}_i\| \quad (8)$$

上式の右辺は凹関数なので、 T の三つの頂点のいずれかで最小値をとる。したがって、 $\mathbf{t}_k = (u_k, v_k), k = 1, 2, 3$ として、

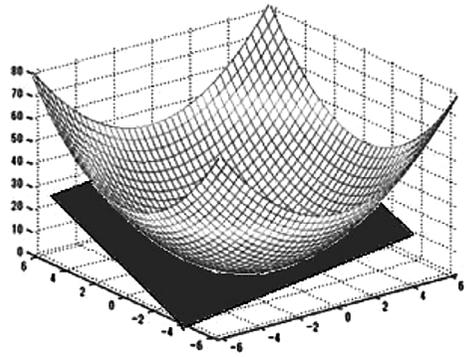


図1 凸関数の接平面
凸関数の接平面は常に関数の下側にある

$$LB_2 = \min_{k=1,2,3} \left[f_1^+(\mathbf{x}_g) + \sum_{i \in I^+} w_i \left\{ \frac{(x_i - a)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (u_k - a) + \frac{(y_i - b)}{\|\mathbf{x}_g - \mathbf{x}_i\|} (v_k - b) \right\} - \sum_{i \in I^-} (-w_i) \|\mathbf{t}_k - \mathbf{x}_i\| \right] \quad (9)$$

が得られる。

この二つの下界を用いた場合の計算時間の違いは大きく、BTST法における LB の重要性がわかる。詳しくは、文献[3]を参照されたい。

文献[3]では、本節で紹介した算法をFORTRANで実現して計算機実験を行っている。そこでは、文献[5]のポロノイ図構成算法を利用してドロネ三角網を作っている。現在では、Pythonにもドロネ三角網を構成するライブラリーがあり、またMatlabにもそのようなツールがある。プログラムの作成の容易さでは、これらの言語で実現するのが良いだろう。ただし、計算の実行時間はFORTRANのほうが短い。

次節以降では、BTST法を適用して最適解を求めた問題の例として、3次元WAR[2]、2施設のウェーバー問題[6]、球面上のウェーバー問題[7]を紹介する。

5. 3次元WAR

3次元WARは前節で紹介したWARの3次元への拡張である。すなわち、3次元空間の需要点と重みを、 $P_1(\mathbf{x}_1), \dots, P_n(\mathbf{x}_n)$, $w_i, i = 1, \dots, n$ とし、求める点とその座標を $P(\mathbf{x})$ としたときに、目的関数

$$f_2(\mathbf{x}) = \sum_{i=1}^n w_i \|\mathbf{x} - \mathbf{x}_i\| \quad (10)$$

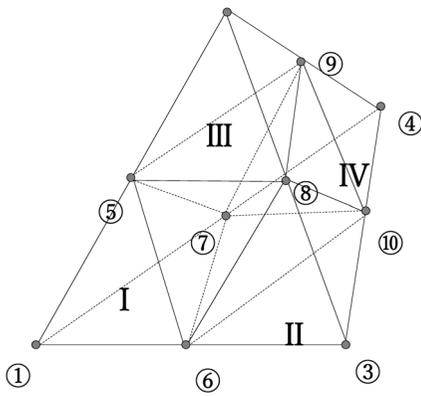


図2 四面体の分割
四面体から四つの頂点を含む四面体 I, II, III, IV を分割

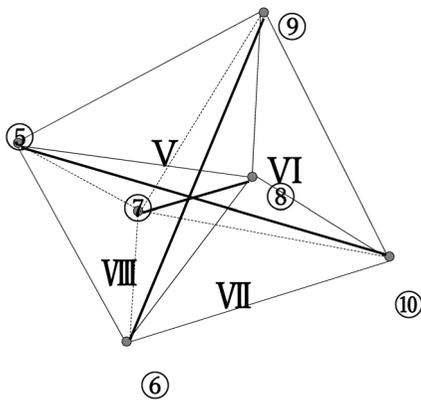


図3 八面体の分割
残りの八面体を四面体 V, VI, VII, VIII に分割

を最小にする問題である。3次元のドロネー網で与えられた点の凸包を分割し、2次元の場合と同様の算法で解を求めることができる。

ここで3次元の場合は、ドロネー網は四面体から構成される。2次元の場合は、三角形を四つの相似な三角形に分割したが、3次元の場合は、四面体を八つの小さな四面体に分割しなくてはならない。図2にあるように、まず、もとの四面体の4頂点から四面体を作成し、次に図3のように残った八面体を四つに分割する。

各四面体の下界の計算は、WARの場合の UB_2 と同様に目的関数の超接平面を利用すればよい。

文献[2]では、MatlabでBTST法の算法を実現し、数値実験を行っている。そこでは、3次元のBSSS法との比較も行われており、BSSS法では、凸包に含まれない立方体のための例外処理に計算時間がかかることが示されている。文献[2]の要旨は文献[8]にも掲載されている。

6. 2施設ウェーバー問題

本節では、2施設を設置する場合のウェーバー問題にBTST法を適用した例を紹介する。2施設のウェーバー問題の目的関数は、与えられた需要点の重みがかすべて正であっても凸ではない。施設の配置によって、与えられた需要点がどちらの施設を利用するかどうかが変わるからである。この問題に対して、与えられた点の凸包をドロネー三角形分割し、三角形の対に対して、目的関数の下界を計算しなくてはならない。

二つの三角形 T^1, T^2 を考え、各需要点 \mathbf{x}_i がその三角形のどちらに近いかで分類する。 $\mathbf{x}^l \in T^l, l = 1, 2$ として、

$$I_1 = \{i \mid \max \|\mathbf{x}_i - \mathbf{x}^1\| < \min \|\mathbf{x}_i - \mathbf{x}^2\|\}$$

$$I_2 = \{i \mid \max \|\mathbf{x}_i - \mathbf{x}^2\| < \min \|\mathbf{x}_i - \mathbf{x}^1\|\}$$

を考える。2施設ウェーバー問題の目的関数は、

$$f_3(\mathbf{x}^1, \mathbf{x}^2) = \sum_{i=1}^n w_i \min\{\|\mathbf{x}^1 - \mathbf{x}_i\|, \|\mathbf{x}^2 - \mathbf{x}_i\|\}$$

$$= \sum_{i \in I^1} w_i \|\mathbf{x}^1 - \mathbf{x}_i\| + \sum_{i \in I^2} w_i \|\mathbf{x}^2 - \mathbf{x}_i\|$$

$$+ \sum_{i \notin (I^1 \cup I^2)} w_i \min\{\|\mathbf{x}^1 - \mathbf{x}_i\|, \|\mathbf{x}^2 - \mathbf{x}_i\|\}$$
(11)

のように三つの項に分けることができる。この式の第1項と第2項は、それぞれ $\mathbf{x}^1, \mathbf{x}^2$ の凸関数なので、WARの節で紹介した接平面を利用した下界を求めることができる。第3項はWARの節で紹介した最初の下界を利用することができる。文献[6]では、この問題に対する算法をFORTRANで実現し、計算機実験を行っている。

7. 球面上のウェーバー問題

球面上のウェーバー問題を考える際には、需要点から施設までの距離として大円距離を考える。球面上の座標として、球面座標を考えると、重み $w_i, i = 1, \dots, n$ をもつ需要点 $P_i(\phi_i, \theta_i)$ から最適点 $P(\phi, \theta)$ への距離は、

$$-\frac{\pi}{2} \leq \phi_i \leq \frac{\pi}{2}, -\pi \leq \theta_i \leq \pi, i = 1, \dots, n.$$

として、

$$d_g(P, P_i) = \arccos\{\cos \phi \cos \phi_i \cos(\theta - \theta_i) + \sin \phi \sin \phi_i\}$$

で与えられる。したがって、球面上のウェーバー問題

の目的関数は,

$$f_4(\phi, \theta) = \sum_{i=1}^n w_i d_g(P, P_i) \quad (12)$$

となる. 球面上の三角形分割は, 球面上のドローネ三角形網を用いる [9]. BTST 法を球面上のウェーバー問題に適用するために, 次の性質を利用する. すなわち, 球面上の点 P を頂点とした半球上, すなわち P からの距離が $\pi/2$ 以下の点では大円距離は凸関数である. さらに, この性質を用いると, P の対蹠点を頂点とした半球上では大円距離は凹関数であることがわかる.

P を頂点とした半円を $SS(P)$ であらわすことにする. また, P の対蹠点を \bar{P} であらわすと, 目的関数は,

$$f_4(\phi, \theta) = \sum_{P_i \in SS(P)} w_i d_g(P, P_i) + \sum_{P_i \in SS(\bar{P})} w_i d_g(P, P_i) \quad (13)$$

のように凸関数と凹関数の和であらわされる.

球面三角形 T 内のすべての点から $\pi/2$ 以内にある需要点の集合を I_P , T 内のすべての点の対蹠点から $\pi/2$ 以内にある需要点の集合を $I_{\bar{P}}$ とする. すると, T 内の点での目的関数は, 以下のように, 需要点が I_P に属する項, $I_{\bar{P}}$ に属する項, どちらでもない項に分けることができる.

$$f_4(\phi, \theta) = \sum_{i \in I_P} w_i d_g(P, P_i) + \sum_{i \in I_{\bar{P}}} w_i d_g(P, P_i) + \sum_{i \notin I_P \cup I_{\bar{P}}} w_i d_g(P, P_i) \quad (14)$$

この式の第 1 項は凸関数, 第 2 項は凹関数, 第 3 項はどちらでもない項になる. 第 1 項については, 目的関数の接平面を用いて下界を求めることができる. 第 2 項は凹関数なので, 球面三角形の頂点の一つで最小値をとる. 第 3 項については, WAR の最初の下界のように, 需要点から球面三角形の最短距離を用いて下界を求めることができる.

文献 [7] では, この算法を FORTRAN を用いて実現して, BSSS 法を球面上のウェーバー問題に対して適用した結果 [10] と計算時間の比較を行っている.

8. おわりに

BTST 法はここで紹介した以外にも多くの問題に適用されている. 文献 [11] にはそれらの問題の一部が紹介されているので, こちらも参照していただきたい.

また, 現在, われわれは, BTST 法を活用して多くの問題を解こうとしており, たとえば, 文献 [12] では競争立地の問題について BTST 法を適用している.

BTST 法は簡単にプログラムを作成できるが, 三角形を相似な小さな三角形に分割するので, 三角形の形が, 鈍角三角形であると, 数値的に不安定になりやすい. たとえば WAR の場合, 解が凸包の境界近くになる場合もあるので, ドローネ三角形が鈍角三角形になることも多い. その際には, 数値誤差を避けるために倍精度で関数値を計算することが必要になる. また, ϵ の値は, 解にさほど影響を与えないことが文献 [7] で示されているので, 10^{-3} の精度でも十分な場合がある. これらは実際に問題を解いて調整する必要がある.

参考文献

- [1] P. Hansen, D. Peeters, D. Richard and J.-F. Thisse, "The minisum and minimax location problems revisited," *Operations Research*, **33**, pp. 1251–1265, 1985.
- [2] 中山莉奈, "BTST 法の 3 次元への拡張について," 南山大学大学院理工学研究科 2015 年度修士論文, 2016.
- [3] Z. Drezner and A. Suzuki, "The big triangle small triangle method for the solution of non-convex facility location problems," *Operations Research*, **52**, pp. 128–135, 2004.
- [4] E. Weiszfeld, "Sur le point pour lequel la somme des distances de n points donnees est minimum," *Tohoku Mathematical Journal*, **43**, pp. 355–386, 1936.
- [5] T. Ohya, M. Iri and K. Murota, "Improvements of the incremental method of the Voronoi diagram with computational comparison of various algorithms," *Journal of the Operations Research Society of Japan*, **27**, pp. 306–337, 1984.
- [6] A. Suzuki and Z. Drezner, "Solving constrained two-facility location problems," *The Journal of the Operations Research Society of Japan*, **56**, pp. 157–165, 2013.
- [7] A. Suzuki, "Big triangle small triangle method for the Weber problem on the sphere," *Contributions to Location Analysis:—In Honor of Zvi Drezner's 75th Birthday*, Chapter 4, H. A. Eiselt and V. Marianov (eds.), Springer International, pp. 109–123, 2019.
- [8] 南山大学大学院理工学研究科修士論文要旨集, <http://www.st.nanzan-u.jp/info/ma-thesis/> (2020 年 7 月 23 日閲覧)
- [9] K. Sugihara, "Laguerre Voronoi diagram on the sphere," *Journal for Geometry and Graphics*, **6**, pp. 69–81, 2002.
- [10] P. Hansen, B. Jaumard and S. Krau, "An algorithm for Weber's problem on the sphere," *Location Science*, **3**, pp. 217–237, 1995.
- [11] Z. Drezner, "A general global optimization approach for solving location problems in the plane," *Journal of Global Optimization*, **37**, pp. 305–319, 2007.
- [12] T. Drezner, Z. Drezner and A. Suzuki, "A cover based competitive facility location model with continuous demand," *Naval Research Logistics*, **66**, pp. 565–581, 2019.