数理解析研究所講究録
第2069巻 2018年 50-61

重心のバランスを考慮した円と長方形の詰込み問題に対する混合整数DC計画法に基づいた手法

増田 暁 (Masuda Satoru)* 奥野 貴之 (Takayuki Okuno)†
池辺 淑子 (Yoshiko Ikebe)‡

概 要

円と長方形の詰込み問題とは，与えられたいくつか円と長方形を重複なく詰込むことができる最小円を求める問題である．本研究では，この問題を詰込んだ円と長方形たちの重心と最小円の中心が一致するという条件の下で考える．本論文ではこの問題を混合整数DC計画問題として定式化を行い，混合整数DCアルゴリズムを適用した実験結果について報告する．

1 はじめに

詰込み問題とは，いくつかの図形を有限の領域に重なりなく配置する問題である．詰込み問題には，詰込む図形と有限領域の次元や形状の種類や，配置制約，目的関数などによって非常に多くのバリエーションが存在する [2]．幾何学や組合せ最適化の分野では古くから研究されており [1, 2, 4]，そのほとんどがNP困難問題であると知られている [4]．

本研究では，重心のバランスを考慮した円と長方形の詰込み問題を考える．円と長方形の詰込み問題とは，与えられたいくつかの円と長方形を互いに重なることなく詰込むことができる最小の外円を求める問題であり，重心のバランスを考慮した円と長方形の詰込み問題は，図形全体の重心と図形を囲む外円の中心が一致しなければならない条件を付加したものである．この問題は，円と円が重ならないようにする制約が非凸2次関数であるため大域的最適解を求めることが難しく，さらに外円の中心と重心を一致させる制約があるためヒューリスティックな解法の設計が難しい．

本研究では，この問題を混合整数DC計画法を用いて解いていく手法を提案する．DC計画問題とは，DC関数を最小化する最適化問題であり，DC関数 (Difference of Convex functions) とは，2つの凸関数の差で表現できる関数である．DC計画問題に

*東京理科大学大学院工学研究科経営工学専攻 4416627@ed.tus.ac.jp
†理化学研究所革新知能統合研究センター takayuki.okuno.ks@riken.jp
‡東京理科大学工学部情報工学科 yoshiko@rs.tus.ac.jp

関する研究は，連続変数の DC 計画問題に対しては，Dinh ら [10] が，Toland-Singer 双対定理を基に DC アルゴリズムを提案しており，この DC アルゴリズムは優れた収束性をもつことが知られている．離散変数のみをもつ DC 計画問題に対しては，Maehara と Murota が [6] では離散凸解析の枠組みを用いて連続 DC 計画問題の理論を離散 DC 計画問題に適用させ，[5] では離散関数の閉凸拡張を用いて整数ギャップが生じない連続緩和法と Dinh ら [10] の DC アルゴリズムに基づく手法を提案した．さらに連続変数も含めた混合整数 DC 計画問題に対しては，Okuno ら [7] が Maehara ら [5] の連続緩和法を拡張して，Dinh ら [10] の DC アルゴリズムに基づく手法を提案した．本研究では，Okuno ら [7] の混合整数 DC アルゴリズムを用いて重心のバランスを考慮した円と長方形の詰込み問題を解き，その結果を報告する．

2　重心のバランスを考慮した円と長方形の詰込み問題

2.1　考える問題

本研究では，半径 r_i, 重さ λ_i $(i \in \{1, \cdots, m\})$ の m 個の円と大きさ $a_j \times b_j$, 重さ λ_j $(j \in \{1, \cdots, l\})$ の l 個の長方形を与え，これらの図形の全てを重なりなく詰込むことのできる最小の外円を求める問題を考える．このとき，長方形は 90° 回転させて詰込むことを許し，求める最小外円の中心は詰込んだ図形全体の重心と一致するものとする (図 1 参照)．

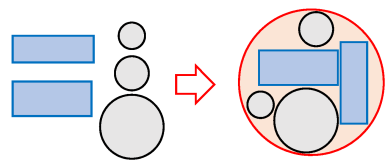

図 1: 円と長方形の詰込み問題

数理計画問題として述べると，目的関数は外円の半径を最小化することとし，制約条件は (a) 詰込んだ図形が互いに重ならない，(b) 詰込んだ図形が外円からはみ出さない，(c) 長方形は縦向きか横向きで置かれる，(d) 詰込んだ図形全体の重心が外円の中心と一致する，の 4 つとする．以下ではこの問題を解くアプローチと定式化について述べる．

2.2　提案アプローチ

この問題を定式化するには，各円と各長方形に対して重心の座標を表す連続変数と，各長方形に対して回転を表す 0–1 変数を導入するのが自然である．この設定において，円同士や長方形同士の重なりは表現しやすいが円と長方形の重なりを表現することは難しい．そこで，1 つの長方形を複数の円で近似することで，円だけの詰込む問題として解く．各長方形を円で近似するとき，以下の方針に従う（図 2 参照）．

- 各長方形は半径と重さが等しい複数の円で近似する
- 近似に用いる円の数はあらかじめ定めておくが,「行」と「列」に並べて規則正しく配置する
- 長方形の 4 頂点はそれぞれ近似円に含まれるよう，外側から近似する

長方形 j を近似する円の集合を K_j とし，各 $i \in K_j$ の半径を R_i, 重さを λ_i とする（R_i, λ_i はすべての $i \in K_j$ に対して同じ値をとる）．円 $i \in K_j$ の位置を表すために，その重心と長方形の重心の差のベクトルを考え，定数 (c_i, d_i) とおく．

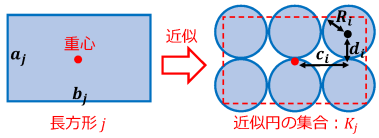

図 2: 長方形の円近似

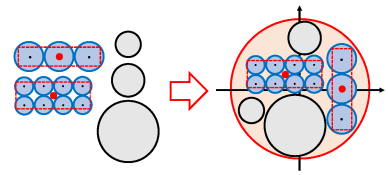

図 3: 近似した詰込み問題

本研究で考える問題は以下のように記述することができる．

目的関数: 外円の半径 → 最小化
制約条件: (1) 詰込んだ円が互いに重ならない．
(2) 詰込んだ円が外円からはみ出さない．
(3) 長方形は円の集合で近似する．
(4) 詰込んだ図形全体の重心が外円の中心と一致する．
(5) 長方形は縦置きか横置きのいずれかである．

2.3 定式化

長方形を複数の円での近似した詰込み問題を以下の記号を使って定式化する．

集合の定義
N :円の集合，$N = \{1, \cdots, n\}$
L :長方形の集合，$L = \{1, \cdots, l\}$
K_j :長方形 j の近似円の集合，$K_j := \{i \mid$ 小円 i は長方形 j の近似円 $\}$
定数の定義
R_i :各円 i の半径の大きさ $(i \in N)$
λ_i :各円 i の重さ $(i \in N)$
c_i :長方形 j の重心と近似円 i の中心との x 軸方向の距離 $(i \in K_j, j \in L)$
d_i :長方形 j の重心と近似円 i の中心との y 軸方向の距離 $(i \in K_j, j \in L)$

変数の定義

r :図形を囲む最小円の半径の大きさを表す変数 (目的関数)
x_i :各円 i の中心の x 座標を表す変数 ($i \in N$)
y_i :各円 i の中心の y 座標を表す変数 ($i \in N$)
s_j :各長方形 j の重心の x 座標を表す変数 ($j \in L$)
t_j :各長方形 j の重心の y 座標を表す変数 ($j \in L$)
z_j :各長方形 j が縦置き (1) か横置き (0) かを表す 0-1 変数 ($j \in L$)

これらの記号を使うと以下のように定式化できる：

$$
\begin{aligned}
\min \quad & r \\
\text{sub.to} \quad & (R_i + R_j)^2 \le (x_i - x_j)^2 + (y_i - y_j)^2 && (i, j \in N, i < j) && (1)\\
& x_i^2 + y_i^2 \le (r - R_i)^2 && (i \in N) && (2)\\
& \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 1 - z_j & -z_j \\ z_j & 1 - z_j \end{pmatrix} \begin{pmatrix} c_i \\ d_i \end{pmatrix} + \begin{pmatrix} s_j \\ t_j \end{pmatrix} && (i \in K_j, j \in L) && (3)\\
& \sum_{i \in N} \lambda_i x_i = 0, \quad \sum_{i \in N} \lambda_i y_i = 0 && && (4)\\
& z_i \in \{0, 1\} && (i \in L) && (5)
\end{aligned}
$$

式 (1) は詰込んだ円が互いに重ならないことを表し，式 (2) は詰込んだ円が外円からはみ出さないことを表し，式 (3) は長方形が円の集合で近似されることを表し，式 (4) は詰込んだ図形全体の重心が外円の中心と一致することを表し，式 (5) は長方形が縦置きか横置きのいずれかであることを表す．

3 DC計画法

3.1 連続DC計画問題

この節では，連続変数の DC 計画問題について説明する．DC 計画問題とは，DC 関数を最小化する最適化問題であり，DC 関数 (Difference of Convex function) とは，2 つの凸関数の差で表現できる関数である．DC 計画問題は以下のように表される最適化問題である：

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) = g(\boldsymbol{x}) - h(\boldsymbol{x}) \\
\text{sub.to} \quad & \boldsymbol{x} \in S,\ \boldsymbol{x} \in \mathbb{R}^n
\end{aligned}
\tag{3.1}
$$

ただし，関数 $g, h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ は閉真凸関数であり，S は閉凸集合である．また，$\emptyset = \mathrm{dom}\, g \subseteq \mathrm{dom}\, h$ であり，$\infty - \infty = \infty$ と便宜上定める．

DC 計画問題は，理論的にも実用的にも非常に重要な非凸最適化のクラスであり，高い表現力をもつ．また，Toland-Singer 双対定理 [10] などの数学的理論の研究や，効率的に局所最適解を求める DC アルゴリズム [10] の提案がされている．以下では DC 計画問題において知られている主な結果を記述する．ただし，$\partial g(\boldsymbol{x})$ は g の $\boldsymbol{x}$ における劣微分であり，g^* は g の共役関数である．

定義 3.1. 以下の式で定義される集合を，g の $\boldsymbol{x}$ における劣微分と呼ぶ．

$$\partial g(\boldsymbol{x}) = \left\{\boldsymbol{p} \in \mathbb{R}^n \,|\, g(\boldsymbol{y}) - g(\boldsymbol{x}) \geq \langle \boldsymbol{p}, \boldsymbol{y} - \boldsymbol{x} \rangle \ (\forall \boldsymbol{y} \in \mathbb{R}^n)\right\} \tag{3.2}$$

ただし，$\langle \boldsymbol{p}, \boldsymbol{x} \rangle = \boldsymbol{p}^T \boldsymbol{x}$ である．また，$\partial g(\boldsymbol{x})$ の要素 $\boldsymbol{p}$ は劣勾配と呼ばれる．

定義 3.2. 以下の式で定義される g^* は g の共役関数である．

$$g^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} \left\{\langle \boldsymbol{y}, \boldsymbol{x} \rangle - g(\boldsymbol{x})\right\} \tag{3.3}$$

命題 3.3 ([10])**.** DC 計画問題 (3.1) の最適解を $\boldsymbol{x}^*$ としたとき以下が成り立つ．

1. $\partial g(\boldsymbol{x}^*) \supseteq \partial h(\boldsymbol{x}^*)$
2. $\bar{\boldsymbol{y}} \in \partial h(\boldsymbol{x}^*) \Leftrightarrow \boldsymbol{x}^* \in \partial h^*(\bar{\boldsymbol{y}})$
3. $\bar{\boldsymbol{y}} \in \partial h(\boldsymbol{x}^*) \Rightarrow \bar{\boldsymbol{y}}$ は $\inf_{\boldsymbol{y} \in \mathbb{R}^n} \left\{h^*(\boldsymbol{y}) - g^*(\boldsymbol{y})\right\}$ の最適解である．

定理 3.4. (Toland-Singer 双対定理) DC 関数 $f = g - h$ において以下が成立する．

$$\inf_{\boldsymbol{x} \in \mathbb{R}^n} \left\{g(\boldsymbol{x}) - h(\boldsymbol{x})\right\} = \inf_{\boldsymbol{y} \in \mathbb{R}^n} \left\{h^*(\boldsymbol{y}) - g^*(\boldsymbol{y})\right\} \tag{3.4}$$

定義 3.5. DC 関数 $f = g - h$ において以下が成り立つとき，$\boldsymbol{x}^*$ は停留点である．

$$\partial g(\boldsymbol{x}^*) \cap \partial h(\boldsymbol{x}^*) \neq \emptyset \tag{3.5}$$

DC 計画問題 (3.1) を解くアルゴリズムとして以下の DC アルゴリズム [10] が知られている．

DC アルゴリズム

Step0: 初期解 $\boldsymbol{x}^0$ を選ぶ．$k = 0$ とする．

Step1: $\boldsymbol{y}^k \in \partial h(\boldsymbol{x}^k)$ を選び，$\boldsymbol{x}^{k+1} \in \partial g^*(\boldsymbol{y}^k)$ を得る．

Step2: 停止条件の式 (3.5) を満たしていれば終了し，そうでなければ $k = k + 1$ として **Step 1** に戻る．

このアルゴリズムが停止したとき停留点 $\boldsymbol{x}^*$ が得られる．g, h が微分可能であった場合には式 (3.5) は $\nabla g(\boldsymbol{x}^*) = \nabla h(\boldsymbol{x}^*)$ を意味し，$\nabla f(\boldsymbol{x}^*) = \nabla g(\boldsymbol{x}^*) - \nabla h(\boldsymbol{x}^*) = 0$ を満たす $\boldsymbol{x}^*$ が得られる．

3.2 離散 DC 計画問題に対する連続緩和

この節では，離散変数の DC 計画問題と，離散 DC 計画問題の連続緩和法 [5] について説明する．離散 DC 計画問題は以下のように表される最適化問題である：

$$\begin{array}{ll} \min & f(\boldsymbol{x}) = g(\boldsymbol{x}) - h(\boldsymbol{x}) \\ \text{sub.to} & \boldsymbol{x} \in S,\ \boldsymbol{x} \in \mathbb{Z}^n \end{array} \tag{3.6}$$

ただし，関数 $g, h : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は閉真凸関数であり，S は閉凸集合である．また，$\emptyset = \mathrm{dom}\, g \subseteq \mathrm{dom}\, h$ であり，$\infty - \infty = \infty$ と便宜上定める．

離散DC計画問題に対しては，連続緩和法 [5] が提案されている．この連続緩和法では，閉凸拡張と閉凸包を用いることで整数性ギャップが生じないことが知られている．

定義 3.6. 連続な閉凸関数 $\hat{g} : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が以下の式を満たすとき，$\hat{g}$ を離散関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の閉凸拡張という．

$$\hat{g}(\boldsymbol{x}) = g(\boldsymbol{x}) \quad (\boldsymbol{x} \in \mathbb{Z}^n) \tag{3.7}$$

定義 3.7. 連続な閉凸関数 $g^{\mathrm{cl}} : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が以下の式を満たすとき，g^{cl} を離散関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の閉凸包という．

$$g^{\mathrm{cl}}(\boldsymbol{x}) = \sup\bigl\{\alpha + \langle \boldsymbol{p}, \boldsymbol{x}\rangle \,|\, \alpha + \langle \boldsymbol{p}, \boldsymbol{y}\rangle \leq g(\boldsymbol{y}) \quad (\boldsymbol{y} \in \mathbb{Z}^n)\bigr\} \tag{3.8}$$

定理 3.8. (整数性ギャップの非存在) 離散関数 $g, h : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を閉凸拡張可能な関数とし，$\mathrm{dom}\, g$ は有界かつ $\mathrm{dom}\, g \subseteq \mathrm{dom}\, h$ としたとき以下の等式が成立する．

$$\inf_{\boldsymbol{x} \in \mathbb{Z}^n} \bigl\{g(\boldsymbol{x}) - h(\boldsymbol{x})\bigr\} = \inf_{\boldsymbol{x} \in \mathbb{R}^n} \bigl\{g^{\mathrm{cl}}(\boldsymbol{x}) - \hat{h}(\boldsymbol{x})\bigr\} \tag{3.9}$$

この定理により，整数性ギャップが生じない連続緩和が可能となる．

Maeharaら [5] は，この連続緩和法とDCアルゴリズム [10] を用いて，離散DC計画問題に対するDCアルゴリズムを提案した．

3.3 混合整数DC計画問題

この節では，混合整数DC計画問題と，混合整数DCアルゴリズムについて説明する．混合整数DC計画問題とは，整数変数と連続変数が混在しているDC計画問題であり，以下のように表される最適化問題である：

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) = g(\boldsymbol{x}) - h(\boldsymbol{x}) \\ \text{sub.to} \quad & \boldsymbol{x} = (\boldsymbol{x}_M, \boldsymbol{x}_N) \in S \\ & \boldsymbol{x}_M \in \mathbb{R}^M, \boldsymbol{x}_N \in \mathbb{Z}^N \end{aligned} \tag{3.10}$$

ただし，関数 $g, h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ は閉真凸関数であり，S は閉凸集合である．また，$\emptyset = \mathrm{dom}\, g \subseteq \mathrm{dom}\, h$ であり，$\infty - \infty = \infty$ と便宜上定める．

混合整数DC計画問題に対しては，Okunoら [7] によって，DCアルゴリズム [10] を基にした混合整数DCアルゴリズムが提案されている．このアルゴリズムはMaeharaら [5] の離散DC計画問題に対する連続緩和法を混合整数DC計画問題に拡張したものである．

混合整数DCアルゴリズム

Step0: 初期解 $\boldsymbol{x}^0$ を選ぶ．$k=0$ とする．

Step1: $\boldsymbol{y}^k \in \partial h(\boldsymbol{x}^k)$ を選び，以下の混合整数凸計画問題を解いて $\boldsymbol{x}^{k+1}$ を得る．

$$\begin{aligned} \min \quad & g(\boldsymbol{x}) - \langle \boldsymbol{y}^k, \boldsymbol{x} \rangle \\ \text{sub.to} \quad & \boldsymbol{x} = (\boldsymbol{x}_M, \boldsymbol{x}_N) \in S \\ & \boldsymbol{x}_M \in \mathbb{R}^M, \boldsymbol{x}_N \in \mathbb{Z}^N \end{aligned}$$

Step2: $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|$ が十分小さければ終了し，そうでなければ $k = k+1$ として **Step 1** に戻る．

各反復では，混合整数DC計画問題 (3.10) を $\boldsymbol{x}^k$ における混合整数凸計画近似した子問題を解き $\boldsymbol{x}^{k+1}$ を得る．得られた $\boldsymbol{x}^{k+1}$ は元の問題 (3.10) の実行可能解であり，アルゴリズムが停止したとき，式 (3.5) を満たす停留点 $\boldsymbol{x}^*$ が得られる．

4 混合整数DC計画問題へ再定式化

この節では，2.3節の定式化を混合整数DC計画問題へ再定式化していく．2.3節の定式化を以下に再掲する．この問題は (1) 式が非凸2次な制約であるため，このままの形では解くことが難しい．

$$\begin{aligned} \min \quad & r \\ \text{sub.to} \quad & (R_i + R_j)^2 \le (x_i - x_j)^2 + (y_i - y_j)^2 && (i,j \in N, i<j) \quad (1) \\ & x_i^2 + y_i^2 \le (r - R_i)^2 && (i \in N) \\ & \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 1-z_j & -z_j \\ z_j & 1-z_j \end{pmatrix} \begin{pmatrix} c_i \\ d_i \end{pmatrix} + \begin{pmatrix} s_j \\ t_j \end{pmatrix} && (i \in K_j, j \in L) \\ & \sum_{i \in N} \lambda_i x_i = 0, \quad \sum_{i \in N} \lambda_i y_i = 0 \\ & z_i \in \{0,1\} && (i \in L) \end{aligned}$$

そこで (1) 式をペナルティをつけて目的関数に加えた $f_0(\boldsymbol{v})$ を定義する．このとき $\boldsymbol{v} := (r, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ である：

$$f_0(\boldsymbol{v}) := r + \sum_{\substack{i,j \in N \\ i<j}} P_{ij} \max\left\{0, (R_i + R_j)^2 - (x_i - x_j)^2 - (y_i - y_j)^2\right\}$$

さらに，$f_0(\boldsymbol{v})$ をDC関数にするために以下のように等価に書き換える：

$$\begin{aligned} f_0(\boldsymbol{v}) = & \left(r + \sum_{\substack{i,j \in N \\ i<j}} P_{ij} \max\left\{(x_i - x_j)^2 + (y_i - y_j)^2, (R_i + R_j)^2\right\} \right) \\ & - \left(\sum_{\substack{i,j \in N \\ i<j}} P_{ij} \left\{(x_i - x_j)^2 + (y_i - y_j)^2\right\} \right) \end{aligned}$$

これで $f_0(\boldsymbol{v})$ はDC関数として表現できたが，max関数を含むため微分可能でない．そのため，さらに補助変数 u_{ij} を導入して，$f_0(\boldsymbol{v})$ を以下のように等価に書き換える：

$$\begin{aligned}
\min \quad & \left(r + \sum_{\substack{i,j\in N\\ i<j}} P_{ij}u_{ij}\right) - \left(\sum_{\substack{i,j\in N\\ i<j}} P_{ij}\left\{(x_i-x_j)^2+(y_i-y_j)^2\right\}\right) \\
\text{sub.to} \quad & (x_i-x_j)^2+(y_i-y_j)^2 \le u_{ij} \qquad (i,j\in N, i<j) \\
& (R_i+R_j)^2 \le u_{ij} \qquad (i,j\in N, i<j)
\end{aligned}$$

この目的関数を $f(\boldsymbol{v})$ と定義すると，$f(\boldsymbol{v}) = g(\boldsymbol{v}) - h(\boldsymbol{v})$ となる2つの凸関数 $g(\boldsymbol{v}), h(\boldsymbol{v})$ が以下のように定義できる．このとき，$g(\boldsymbol{v}), h(\boldsymbol{v})$ にはパラメーター $\rho\,(\rho>0)$ を用いた強凸項を加えて強凸関数とした．

$$\begin{aligned}
f(\boldsymbol{v}) &:= \left(r + \sum_{\substack{i,j\in N\\ i<j}} P_{ij}u_{ij}\right) - \left(\sum_{\substack{i,j\in N\\ i<j}} P_{ij}\left\{(x_i-x_j)^2+(y_i-y_j)^2\right\}\right) \\
g(\boldsymbol{v}) &:= r + \sum_{\substack{i,j\in N\\ i<j}} P_{ij}u_{ij} + \rho\frac{\|\boldsymbol{v}\|^2}{2} \\
h(\boldsymbol{v}) &:= \sum_{\substack{i,j\in N\\ i<j}} P_{ij}\left\{(x_i-x_j)^2+(y_i-y_j)^2\right\} + \rho\frac{\|\boldsymbol{v}\|^2}{2}
\end{aligned}$$

これによって，元の問題は以下の混合整数DC計画問題に再定式化できる：

$$\begin{aligned}
\min \quad & f(\boldsymbol{v}) = g(\boldsymbol{v}) - h(\boldsymbol{v}) \\
\text{sub.to} \quad & (x_i-x_j)^2+(y_i-y_j)^2 \le u_{ij} && (i,j\in N, i<j) \\
& (R_i+R_j)^2 \le u_{ij} && (i,j\in N, i<j) \\
& x_i^2+y_i^2 \le (r-R_i)^2 && (i\in N) \\
& \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 1-z_j & -z_j \\ z_j & 1-z_j \end{pmatrix}\begin{pmatrix} c_i \\ d_i \end{pmatrix} + \begin{pmatrix} s_j \\ t_j \end{pmatrix} && (i\in K_j, j\in L) \\
& \sum_{i\in N}\lambda_i x_i = 0, \quad \sum_{i\in N}\lambda_i y_i = 0 \\
& z_i \in \{0,1\} && (i\in L)
\end{aligned} \tag{4.1}$$

5 計算機実験

この節では，定式化した混合整数DC計画問題 (4.1) に対して混合整数DCアルゴリズムを適用した結果を報告する．混合整数DCアルゴリズムの **Step1** では汎用ソルバー Gurobi 7.5.1 を用いて子問題の混合整数凸計画問題を解き，反復点の生成を行った．

実験データとして円と長方形の個数は表1のように与えた．円の半径は [1,2] の範囲でランダムに生成し，長方形は2つの等円で近似できるものとして，近似円の半

径は [0.5,1.5] の範囲でランダムに生成した．また，混合整数 DC アルゴリズムの初期解はランダムに生成した．生成した実験データに対して，以下の 2 通りの実験を行った．このとき，ペナルティ値 $P = 1$，強凸パラメーター $\rho = 10^{-3}$ と設定し，終了条件は $\|\boldsymbol{v}^{k+1} - \boldsymbol{v}^{k}\| < 10^{-3}$ または $|f(\boldsymbol{v}^{k+1}) - f(\boldsymbol{v}^{k})| < 10^{-6}$ と設定した．

1. ランダムに生成した初期解から混合整数 DC アルゴリズムを実行
2. 0–1 変数制約 $z_i \in \{0, 1\}$ $(i \in L)$ を $0 \leq z_i \leq 0$ $(i \in L)$ に緩和した問題を連続 DC アルゴリズムで解き，得られた緩和解を初期解として混合整数 DC アルゴリズムを実行

まず，実験 1 ではランダムな初期解から混合整数 DC アルゴリズムを実行し，以下の表 1 の結果が得られた．表 1 は，3 種類の問題をそれぞれ 5 回解き，その 5 回を平均した反復回数と計算時間をまとめたものである．反復回数は混合整数 DC アルゴリズム内で部分問題を解いた回数である．また，図 4 は円 10 個と長方形を近似した円集合を 10 個のデータに対してアルゴリズムを実行した結果である．図 5 は，縦軸に目的関数値 $f(\boldsymbol{v})$，横軸に反復回数をとり，目的関数値の変化の様子を示している．図 6 は，縦軸に目的関数値の減少量 $f(\boldsymbol{v}^{k}) - f(\boldsymbol{v}^{k+1})$，横軸に反復回数をとったものを示している．

表 1: ランダムな初期解から混合整数 DC アルゴリズムを実行した結果 (5 回平均)

問題	円 (個)	長方形 (個)	反復回数	時間 $[s]$
1	10	5	2444.2	315.5
2	10	10	2002.2	1304.6
3	10	15	1945.0	8289.8

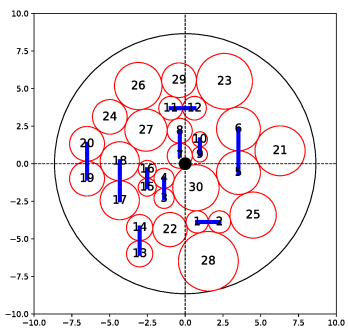

図 4: 実行結果

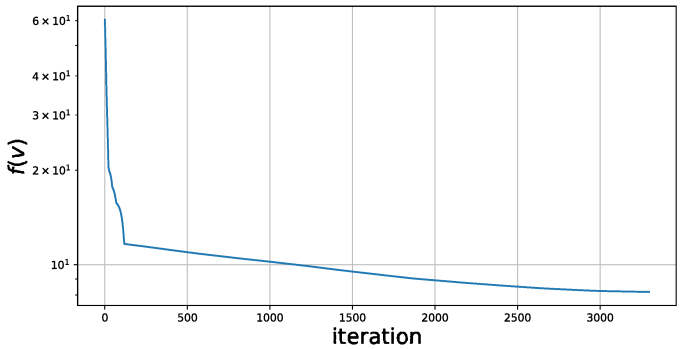

図 5: 実行結果

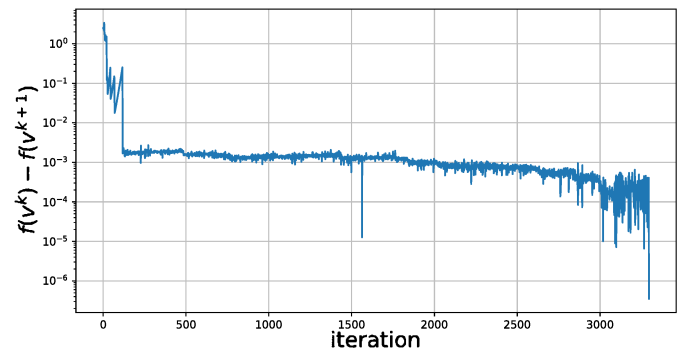

図 6: 実行結果

表 1 から反復回数が 2000 回程度と非常に多く，そのため計算時間が長くなっていることが分かる．また，問題のサイズは大きくなるほど計算時間が長くなるが，反復回数には影響しないと分かる．そして，図 5,6 からは反復回数 100 回程度を境目に目的関数値が減少しにくくなっていることが分かる．これは，まず円と円の重なりのペナルティ関数の値を減少させて実行可能性を満たし，その後に半径を減少させていくというアルゴリズムの動きになっているからである．特に実行可能解が得られて以降の反復回数が多くなっているため，できるだけ良い実行可能解を初期解として与えて計算時間を短縮させる必要がある．

次に，実験 2 では連続緩和した問題を連続 DC アルゴリズムを実行し，得られた緩和解を初期解として混合整数 DC アルゴリズムを実行すると以下の表 2,3 の結果が得られた．表 2 はランダム生成した初期解から連続緩和した問題を連続 DC アルゴリズムで解いた結果であり，表 3 は緩和解を初期解にして混合整数 DC アルゴリズムを実行した結果である．

表 2: 緩和問題を連続 DC アルゴリズムで解いた結果 (5 回平均)

問題	円 (個)	長方形 (個)	反復回数	時間 [s]
1	10	5	2158.2	265.5
2	10	10	1475.4	586.6
3	10	15	1490.6	5163.5

表 3: 緩和解から混合 DC アルゴリズムを実行した結果 (5 回平均)

問題	円 (個)	長方形 (個)	反復回数	時間 [s]
1	10	5	321.8	39.5
2	10	10	1132.0	488.2
3	10	15	964.0	3317.6

表 2 では緩和問題を連続 DC アルゴリズムで解いているため，表 1 の結果と比べて計算時間が短いことが分かる．そして，表 3 では初期解が緩和解になったことで反復回数が減り，計算時間が短くなったことが分かる．しかし，表 2,3 での計算時間を合計すると表 1 での計算時間と変わらないため，効果的な高速化はできなかった．

6 まとめ

本研究では，重心のバランスを考慮した円と長方形の詰込み問題の混合整数 2 次計画問題への定式化と混合整数 DC 計画問題への再定式化を行い，混合整数 DC アルゴリズムの適応を提案した．計算機実験の結果から反復回数と計算時間がかかるものの妥当な解が得られることが分かった．しかし，得られた解に改善の余地があり，停止条件やペナルティ値などの調整が必要である．今後は，初期解の工夫や局所探索などによって計算時間と解の精度の両面での向上を目指していきたい．

参考文献

[1] Castillo, Ignacio, Frank J. Kampas, and Jnos D. Pintr. “Solving circle packing problems by global optimization: numerical results and industrial applications.” European Journal of Operational Research 191.3 (2008): 786-802.

[2] Dyckhoff, Harald. “A typology of cutting and packing problems.” European Journal of Operational Research 44.2 (1990): 145-159.

[3] Fasano, Giorgio, and János D. Pintér, eds. “Modeling and optimization in space engineering.” Vol. 73. Springer Science & Business Media (2012).

[4] Leung, Joseph YT, et al. “Packing squares into a square.” Journal of Parallel and Distributed Computing 10.3 (1990): 271-275.

[5] Maehara, Takanori, Naoki Marumo, and Kazuo Murota. "Continuous relaxation for discrete DC programming." Modelling, Computation and Optimization in Information Systems and Management Sciences (2015): 181-190.

[6] Maehara, Takanori, and Kazuo Murota. "A framework of discrete DC programming by discrete convex analysis." Mathematical Programming 152.1-2 (2015): 435-466.

[7] Okuno, Takayuki, and Yoshiko T. Ikebe. "A new approach for solving mixed integer DC programs using a continuous relaxation with no integrality gap and smoothing techniques." arXiv preprint arXiv:1702.00553 (2017).

[8] Shor, Naum Zuselevich. "Nondifferentiable optimization and polynomial problems." Vol. 24. Springer Science & Business Media (2013).

[9] Stetsyuk, Petro I., Tatiana E. Romanova, and Guntram Scheithauer. "On the global minimum in a balanced circular packing problem." Optimization Letters 10.6 (2016): 1347-1360.

[10] Tao, Pham Dinh, and Le Thi Hoai An. "Convex analysis approach to dc programming: Theory, algorithms and applications." Acta Mathematica Vietnamica 22.1 (1997): 289-355.