混合多項式最適化問題のコンパクトな MIP 表現 *

檀寛成 †　　山下信雄 ‡

2023 年 8 月 10 日

概要

混合多項式最適化問題 (MIPOP) は多くの応用をもつ広いクラスの問題であるが，一般には NP 困難となる非常に難しい問題である．MIPOP を解くアプローチの一つに，MIPOP を混合整数計画問題 (MIP) に等価あるいは近似的に変換して解く手法がある．しかしながら，そのような手法の多くでは，MIPOP のクラスが限定されていたり，近似精度をあげるために数多くの変数を導入する必要があったりしていた．つい最近，Zhu らは $x, y \in [0,1]$ の積 $z = xy$ を少ない変数の導入で MIP に表現する方法を提案した．この手法を用いれば，決定変数が $[0,1]$ 区間に入る非凸な 2 次最適化問題は MIP に定式化して解くことができる．しかしながら，2 次の項の数に比例して導入する決定変数が増えるため，大規模な問題や次数が高い問題には適用できなかった．本論文では，まず，Zhu らの手法を拡張することによって，有理関数を MIP として表現する手法を提案する．さらに，2 次形式，3 次形式に対して少ない決定変数の導入で MIP に表現する方法を提案する．また，決定変数が整数などの離散変数のみの場合には，等価な MIP に変換できることをみる．

1 序論

次の混合多項式最適化問題 (Mixed Integer Polynomial Optimization, MIPOP) を考える．

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_i(x) \le 0, i = 1, \ldots, m \\ & x_I \in Z^{|I|} \end{array}$$

ここで，$f_i, i = 0, \ldots, m$ は R^n から R への多項式関数であり，$I \subseteq \{1, 2, \cdots, n\}$ である．また，x_I は $x_i, i \in I$ を並べた $|I|$ 次元ベクトルである．MIPOP は，多項式最適化問題 [1]，混合整数計画問題 (Mixed Integer Linear Optimization, MIP)[10]，Binary quadtatic optimizaiton [8] を含む広いクラスの問題で，多くの応用問題をもつ．

MIPOP は一般には NP 困難となるため，MIPOP の大域的最適解を求めることは難しい．MIPOP を解く手法としては，問題構造を利用した分枝限定法などの独自の厳密解法 (または近似解法) を開発するアプローチや，MIP など既によい汎用ソルバーが開発されている問題に変換して解くアプローチがある．本論文では後者のアプローチを考える．

混合 2 次最適化問題を MIP に定式化する方法はいくつか提案されている [4, 7]. そのような方法で重要となるポイントは，(i) 少ない変数で (コンパクトな表現で)(ii) 高精度あるいは等価に MIP に表現することであ

---

* 本研究は，京都大学とトヨタ自動車の共同研究プロジェクト「モビリティ基盤数理の研究」の支援を受けている．
† 関西大学環境都市工学部
‡ 京都大学大学院情報学研究科

る．高精度あるいは等価な MIP に変換するためには，一般には数多くの変数を導入しなければならない．その結果，MIPOP を MIP に定式化できたとしても，問題の規模が大きくなりすぎて，実用的には使えないことがある．以下では，$f(x) = x^\top Ax = \sum_{i=1}^n \sum_{j=1}^n A_{ij}x_ix_j$ を MIP で表現する既存手法を紹介し，導入する変数の数と精度の二つの観点からその特徴を議論する．

2 次形式 $f(x) = \sum_{i=1}^n \sum_{j=1}^n A_{ij}x_ix_j$ を含む最適化問題を MIP で表すためには，積 x_ix_j を線形な関数だけで表す必要がある．Binary quadtatic optimizaiton のように決定変数 x_i が 0-1 変数に限定されているときは，以下の線形な不等式を用いて $z_{ij} = x_ix_j$ を正確に表すことができる [4].

$$z_{ij} \geq x_i + x_j - 1,\ z_{ij} \leq x_i, z_{ij} \leq x_j \tag{1}$$

この不等式を満たす z_{ij} を用いれば，2 次形式は $f(x) = \sum_{i=1}^n \sum_{j=1}^n A_{ij}z_{ij}$ となり，線形な関数で表すことができる．しかしながら，2 次形式 f を表すためには $O(n^2)$ に比例した数の変数 z_{ij} と関連した不等式 (1) が必要となる．また，一般の連続変数 x_i と x_j の積については，この方法で表現することはできない．行列 A が半正定値対称行列のときは，以下のようにして，導入する変数を少なくすることができる．$A = A^{\frac{1}{2}}A^{\frac{1}{2}}$ となる半正定値対称行列 $A^{\frac{1}{2}}$ が存在するため，$w = A^{\frac{1}{2}}x$ とすると，$f(x) = \sum_{i=1}^n w_i^2$ と表すことができる．そこで，[4] にあるようなテクニックで w_i^2 を区分線形関数で近似して MIP で表現すれば，2 次形式 f を MIP で近似的に表すことができる．しかしながら，区分線形近似の精度をあげるためには多くの 0-1 変数を導入しなければならない．また，このテクニックは A が半正定値対称行列のときしか使えない．さらに A が疎な行列であっても，一般には $A^{\frac{1}{2}}$ は密な行列になるため，A の疎性を活かすことができない．

つい最近，Zhu ら [13] は連続変数 $x \in [0,1]$ と $y \in [0,1]$ の積 $z = xy$ を 0-1 変数と線形不等式で高精度に近似する画期的な方法を提案した．その手法では，まず，x を p 個の 0-1 変数 $\mu_i, i = 1, \ldots, p$ を用いて近似する．

$$x \approx \sum_{i=1}^p \frac{2^{i-1}}{2^p - 1}\mu_i$$

ここで，$\sum_{i=1}^p \frac{2^{i-1}}{2^p - 1} = 1$ であることから，$\sum_{i=1}^p \frac{2^{i-1}}{2^p - 1}\mu_i \in [0,1]$ となることがわかる．その結果，x と y の積 z は以下のように近似できる．

$$z \approx \sum_{i=1}^p \frac{2^{i-1}}{2^p - 1} y\mu_i$$

ここで，積 $y\mu_i$ を λ_i とすると，λ_i は以下の線形な不等式の解として表すことができる．

$$0 \leq \lambda_i \leq \mu_i,\ y - (1 - \mu_i) \leq z_i \leq y + (1 - \mu_i)$$

実際，$\mu_i = 0$ のときは $\lambda_i = 0$, $\mu_i = 1$ のときは $\lambda_i = y$ となることが確かめられる．以上のことをまとめて，Zhu らは以下の近似的な MIP 表現[*1]を提案している．ただし，

$$\Omega^p = R \times R \times R \times \{0,1\}^p \times R^p$$

[*1] ここでは 0-1 変数と線形方程式，線形不等式で表される集合を MIP 表現とよんでいる．最適化問題を表していないが，本論文の主旨からこの用語を使うこととする．

である．

$x, y \in [0,1]$ の積 $z = xy$ の近似 MIP 表現 [13]

$$
\mathcal{M}^0(p) = \left\{ (x,y,z,\mu,\lambda) \in \Omega^p \;\middle|\; \begin{array}{l} x \geq \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1}\mu_i - \frac{1}{2^p - 1} \\ x \leq \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1}\mu_i \\ z = \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1}\lambda_i \\ 0 \leq \lambda_i \leq \mu_i,\ i = 1, \ldots, p \\ y - (1 - \mu_i) \leq \lambda_i \leq y + (1 - \mu_i),\ i = 1, \ldots, p \end{array} \right\}
$$

近似 MIP 表現 $\mathcal{M}^0$ の中の始めの二つの不等式は，x の近似度合いを表している．この近似 MIP 表現は，区分線形近似などを用いる場合と比べて，少ない 0-1 変数の導入で高精度な近似ができる．実際，$p = 7$ の場合でも，誤差は $\dfrac{1}{2^7 - 1} \approx 0.0079$ と 1% 未満となる．

Zhu ら [13] はこの近似 MIP 表現を所望の性質をもつ化学グラフの列挙に応用している．その応用では，化学グラフのデータに適当な変換を施すことによって，変数が $[0,1]$ に納まることを仮定することができた．また，その応用に現れる多項式は高々 2 次で，その数も限定されていた．そのため，Zhu ら [13] の論文では変数が一般の区間に入る場合や，積の数が多くなる一般の 2 次形式 $f(x) = x^\top Ax$ への表現については考慮されていなかった．さらに，3 次以上の多項式など，一般の有理関数に対する応用方法についても議論されていなかった．

本論文では，Zhu らの近似 MIP 表現を発展させることによって，一般の MIPOP を近似的に MIP で定式化する方法を与える．さらに，2 次関数 $f(x) = x^\top Ax$ を，変数の数を増やすことなくコンパクトに表現する方法を提案する．また，決定変数が整数である MIPOP に対しては，等価な MIP として定式化できることをみる．

本論文は以下のように構成されている．まず，2 節では，MIPOP を MIP として定式化するための基礎となる MIP 表現を与える．さらに，2 次形式や 3 次形式に対してコンパクトな MIP 表現を与えるテクニックを提案する．第 3 節では，具体的な MIPOP の応用問題に対して，その特徴を利用してコンパクトな MIP 表現を与える方法を議論する．第 4 節では本アプローチの有効性を確認するための簡単な数値例を与える．最後に，第 5 節で本論文のまとめと今後の課題について議論する．

2　基本アプローチ

この節では，MIPOP を MIP として定式化するための基本となる MIP 表現とそれを用いたいくつかのテクニックを提案する．まず，$[0,1]$ 区間の変数の積の近似 MIP 表現 $\mathcal{M}^0$ を任意の区間に一般化した近似 MIP 表現を提案する．次に，これを一般の次数の多項式や分数，平方根などの有理関数に拡張するテクニックを紹介する．さらに，2 次形式，3 次形式を少ない変数で効率的に表現する方法を提案する．連続変数の有理関数の MIP 表現は近似表現となる．そのため，変換された MIP の最適解はもとの MIPOP の近似解となる．そこで，非線形最適化ソルバーを使って解を精錬する手法について議論する．

2.1　積の MIP 表現

この副節では本論文で考えるアプローチの基礎となる積の MIP 表現を与える．まず，連続変数の積の近似 MIP 表現を与え，次に離散変数の積の (正確な)MIP 表現を提案する．

2.1.1　連続変数の積の近似 MIP 表現

ここでは，$[0,1]$ 区間の変数の積を扱った Zhu らの近似 MIP 表現 $\mathcal{M}^0$ を一般の閉区間にある連続変数の積に拡張する．以下では，$x \in [l_x, u_x]$ と $y \in [l_y, u_y]$ の積 $z = xy$ を考える．ただし，l_x, u_x, l_y, u_y は $l_x \leq u_x$，$l_y \leq u_y$ を満たす定数である．

まず，序論の $\mathcal{M}^0$ の導出と同様にして，x を p 個の 0-1 変数 μ_i を用いて以下のように近似する．

$$x \approx \bar{x} = (u_x - l_x) \sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \mu_i + l_x$$

ここで，$\mu_1 = \mu_2 = \cdots = \mu_p = 0$ のときは $\bar{x} = l_x$，$\mu_1 = \mu_2 = \cdots = \mu_p = 1$ のときは $\bar{x} = u_x$ となることから，$\bar{x} \in [l_x, u_x]$ となることがわかる．この x の近似式を用いると，$z = xy$ は以下のように近似できる．

$$z \approx y\bar{x} = (u_x - l_x) \sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} y\mu_i + l_x y$$

次に 0-1 変数 μ_i と y の積 $\lambda_i = y\mu_i$ について考える．$\mu_i \in \{0,1\}$ であることから，λ_i は以下の線形な不等式の解として表現できる．

$$l_y \mu_i \leq \lambda_i \leq u_y \mu_i,\ y - u_y(1-\mu_i) \leq \lambda_i \leq y - l_y(1-\mu_i)$$

実際，$y \in [l_y, u_y]$ であることに注意すると，$\mu_i = 0$ のときは

$$0 \leq \lambda_i \leq 0,\ y - u_y \leq \lambda_i \leq y - l_y$$

となるから，この不等式をみたす λ_i は 0 だけであり，$\mu_i = 1$ のときは

$$l_y \leq \lambda_i \leq u_y,\ y \leq \lambda_i \leq y$$

となるから，この不等式をみたす λ_i は y だけである．

以上のことをまとめると，$x \in [l_x, u_x]$ と $y \in [l_y, u_y]$ の積 $z = xy$ は以下のように近似的に表せる．

$x \in [l_x, u_x]$ と $y \in [l_y, u_y]$ の積 $z = xy$ の近似 MIP 表現 1

$$\mathcal{M}^{\mathrm{C}}(p, l_x, u_x, l_y, u_y) = \left\{ (x, y, z, \mu, \lambda) \in \Omega^p \ \middle|\ \begin{array}{l} x \geq (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \mu_i - \frac{u_x - l_x}{2^p - 1} + l_x \\ x \leq (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \mu_i + l_x \\ z = (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \lambda_i + l_x y \\ l_y \mu_i \leq \lambda_i \leq u_y \mu_i,\ i = 1, \ldots, p \\ y - u_y(1-\mu_i) \leq \lambda_i \leq y - l_y(1-\mu_i),\ i = 1, \ldots, p \end{array} \right\}$$

x の近似誤差は $\dfrac{u_x - l_x}{2^p - 1}$ であるため，$z = xy$ の最大の近似誤差は $\dfrac{\max\{|l_y|, |u_y|\}(u_x - l_x)}{2^p - 1}$ となる．

上記の近似 MIP 表現では，変数 x と μ_i の関係を不等式で表し，z と λ_i に関しては等式で表している．後で見るように変数 x と μ_i の関係を等式で表したり，z と λ_i の関係を不等式で表したほうが都合がよいときがある．そこで以下の二つの近似表現を定義する．

$x \in [l_x, u_x]$ と $y \in [l_y, u_y]$ の積 $z = xy$ の近似 MIP 表現 2

$$\mathcal{M}^{\mathrm{C}}_{\mathrm{tight}}(p, l_x, u_x, l_y, u_y) = \left\{ (x, y, z, \mu, \lambda) \in \Omega^p \;\middle|\; \begin{array}{l} x = (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \mu_i + l_x \\ z = (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \lambda_i + l_x y \\ l_y \mu_i \le \lambda_i \le u_y \mu_i, \ i = 1, \ldots, p \\ y - u_y(1 - \mu_i) \le \lambda_i \le y - l_y(1 - \mu_i), \ i = 1, \ldots, p \end{array} \right\}$$

$x \in [l_x, u_x]$ と $y \in [l_y, u_y]$ の積 $z = xy$ の近似 MIP 表現 3

$$\mathcal{M}^{\mathrm{C}}_{\mathrm{relax}}(p, l_x, u_x, l_y, u_y) = \left\{ (x, y, z, \mu, \lambda) \in \Omega^p \;\middle|\; \begin{array}{l} x \ge (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \mu_i - \frac{u_x - l_x}{2^p - 1} + l_x \\ x \le (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \mu_i + l_x \\ z \ge (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \lambda_i - \frac{M(u_x - l_x)}{2^p - 1} + l_x y \\ z \le (u_x - l_x) \displaystyle\sum_{i=1}^{p} \frac{2^{i-1}}{2^p - 1} \lambda_i + l_x y \\ l_y \mu_i \le \lambda_i \le u_y \mu_i, \ i = 1, \ldots, p \\ y - u_y(1 - \mu_i) \le \lambda_i \le y - l_y(1 - \mu_i), \ i = 1, \ldots, p \end{array} \right\}$$

2.1.2　離散変数の積の MIP 表現

連続変数の積の近似表現 $\mathcal{M}^{\mathrm{C}}$ では，連続変数 x を 0-1 変数によって近似しているため，その積も近似となっていた．一方，x が有限な離散集合の要素のときは，x を正確に 0-1 変数で表現できるため，その積も正確に与えることができる．ここでは，$x \in [l_x, u_x] \cap Z$ のときと，$x \in \{a_1, a_2, \ldots, a_p\}$ のときの積 xy の正確な MIP 表現を与える．ここで y は連続変数でもよいことに注意してほしい．

以下では $l_x, u_x \in Z$ であり，$p = \lceil \log_2(u_x - u_x) \rceil + 1$ とする．さらに $\mu_i \in \{0, 1\}, i = 1, \ldots, p$ を用いて，

$$x = \sum_{i=1}^{p} 2^{i-1} \mu_i + l_x$$

とする．このとき，x は整数であり，$\mu_1 = \mu_2 = \cdots \mu_p = 0$ のとき $x = l_x$ となり，$\mu_1 = \mu_2 = \cdots \mu_p = 1$ のとき $x \in [u_x, 2u_x)$ となる．よって，$[l_x, u_x] \cap Z$ の任意の整数を表すことができる．

連続変数の積の近似表現 $\mathcal{M}^{\mathrm{C}}$ と同様の議論をすることよって，$x \in [l_x, u_x] \cap Z$ と $y \in [l_y, u_y]$ の積を以下のように表すことができる．

$x \in [l_x, u_x] \cap Z$ と $y \in [l_y, u_y]$ の積 $z = xy$ の MIP 表現

$$\mathcal{M}^{\mathrm{I}}(l_x, u_x, l_y, u_y) = \left\{ (x, y, z, \mu, \lambda) \in \Omega^p \;\middle|\; \begin{array}{l} p = \lceil \log_2(u_x - u_x) \rceil + 1 \\ x = \displaystyle\sum_{i=1}^{p} 2^{i-1}\mu_i + l_x \\ z = \displaystyle\sum_{i=1}^{p} 2^{i-1}\lambda_i + l_x y \\ l_y\mu_i \le \lambda_i \le u_y\mu_i, \ i = 1, \ldots, p \\ y - u_y(1-\mu_i) \le \lambda_i \le y - l_y(1-\mu_i), \ i = 1, \ldots, p \end{array} \right\}$$

ここで，x と μ_i の関係が等号で表されていることに注意してほしい．等号であることから，連続変数の積の近似 MIP 表現 $\mathcal{M}^{\mathrm{C}}$ とは異なり，$z = xy$ が厳密に成り立っている．

次に，$x \in \{a_1, a_2, \ldots, a_p\}$ のときの MIP 表現を与える．このような x は $\mu_i \in \{0, 1\}, i = 1, \ldots, p$ を用いて，

$$x = \sum_{i=1}^{p} a_i\mu_i, \ \sum_{i=1}^{p} \mu_i = 1$$

と表せる．よって，これまでの議論と同様にして，$x \in \{a_1, a_2, \ldots, a_p\}$ と $y \in [l_y, u_y]$ の積は以下のように表すことができる．ただし，a は $a = (a_1, a_2, \cdots, a_p)$ で定義されるベクトルである．

$x \in \{a_1, a_2, \ldots, a_p\}$ と $y \in [l_y, u_y]$ の積 $z = xy$ の MIP 表現

$$\mathcal{M}^{\mathrm{D}}(a, l_y, u_y) = \left\{ (x, y, z, \mu, \lambda) \in \Omega^p \;\middle|\; \begin{array}{l} x = \displaystyle\sum_{i=1}^{p} a_i\mu_i, \ \sum_{i=1}^{p} \mu_i = 1 \\ z = \displaystyle\sum_{i=1}^{p} a_i\lambda_i \\ l_y\mu_i \le \lambda_i \le u_y\mu_i, \ i = 1, \ldots, p \\ y - u_y(1-\mu_i) \le \lambda_i \le y - l_y(1-\mu_i), \ i = 1, \ldots, p \end{array} \right\}$$

2.2 有理関数の近似 MIP 表現

前副節では，積の MIP 表現を与えた．本副節ではその表現を用いて，いくつかの有理関数を表す方法を紹介する．さらに，2 次形式 $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}x_i x_j$ および 3 次形式 $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} T_{ijk}x_i x_j x_k$ のコンパクトな表現方法を提案する．

2.2.1 有理単項式の近似 MIP 表現

任意の単項式は積を繰り返しことによって与えられる．そのため，前副節で与えた近似 MIP 表現を繰り返すことによって，有限区間にある変数の任意の多項式は近似的に MIP 表現することができる．例えば，$w = x^2y$ は $z = xy$，$w = zx$ として表現できる．

さらに，平方根 $\sqrt{x}$ や分数 $\dfrac{x}{y}$ などの有理関数も，積の近似 MIP 表現を用いることで，近似的に MIP として表現することができる．実際， $x \in [0, u_x]$ の平方根 $z = \sqrt{x}$ は $x = z^2$ であることを利用すれば，以下のように表現できる．

$x \in [0, u_x]$ の平方根 $z = \sqrt{x}$ の近似 MIP 表現

$$\mathcal{M}^{\mathrm{sqrt}}(p, u_x) = \left\{ (x, z, \mu, \lambda) \in R \times R \times \{0,1\}^p \times R^p \,\middle|\, (z, z, x, \mu, \lambda) \in \mathcal{M}^{\mathrm{C}}(p, 0, \sqrt{u_x}, 0, \sqrt{u_x}) \right\}$$

さらに，分数 $z = \dfrac{x}{y}$ についても $x = yz$ であることから，以下のように表せる．ただし，$l_x, l_y > 0$ とする．

$x \in [l_x, u_x]$ と $y \in [l_y, u_y]$ の分数 $z = \dfrac{x}{y}$ の近似 MIP 表現

$$\mathcal{M}^{\mathrm{frac}}(p, l_x, u_x, l_y, u_y) = \left\{ (x, y, z, \mu, \lambda) \in \Omega^p \,\middle|\, (y, z, x, \mu, \lambda) \in \mathcal{M}^{\mathrm{C}} \left(p, l_y, u_y, \frac{l_x}{u_y}, \frac{u_x}{l_y} \right) \right\}$$

2.2.2　2 次形式，3 次形式のコンパクトな MIP 表現

まず，2 次形式 $x^\top Ax = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j$ の MIP 表現を考える．すべての積 $x_i x_j$ に対して近似 MIP 表現 $\mathcal{M}^C$ を使うと，$O(pn^2)$ 個の変数と不等式の追加が必要となる．そのため n が大きいときには現実的でない．以下のように変数 $w \in R^n$ を導入することによって，追加する変数と方程式の数を $O(np)$ に抑えることができる．

$$w = Ax, z = \sum_{i=1}^{n} x_i w_i$$

近似 MIP 表現を用いて $x_i w_i$ を表すために追加される不等式と変数の数は $O(p)$ である．よって，全体で追加される変数の数は $O(np)$ となる．ただし，w_i の上限と下限をあらかじめ見積もる必要がある．

3 次形式 $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} T_{ijk} x_i x_j x_k$ に関しても同様にして少ない変数で MIP 表現することができる．

まず，$T^{(k)}$ を第 (i, j) 成分が T_{ijk} となる $n \times n$ 行列とする．そのとき

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} T_{ijk} x_i x_j x_k = \sum_{k=1}^{n} x_k \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ijk} x_i x_j = \sum_{k=1}^{n} x_k x^\top T^{(k)} x$$

となる．ここで，v_k を

$$v_k = x^\top T^{(k)} x$$

と定義し，上記の 2 次形式の近似 MIP 表現を用いて v_k を表す．このとき，追加される変数は $O(n^2 p)$ である．この v_k を用いると，

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} T_{ijk} x_i x_j x_k = \sum_{k=1}^{n} v_k x_k$$

となるから，積 $v_k x_k$ を近似 MIP 表現すれば，3 次形式を近似 MIP 表現できたことになる．なお，追加される変数の数は $O(n)$ である．ここで，x_k を表現するために用いられる 0-1 変数 μ_i は，$v_k = x^\top T^{(k)} x$ でも使われていることに注意して欲しい．そのため，3 次形式を表すために必要となる 0-1 変数の数は np である．

2.3　近似計算による下界値と上界値

近似 MIP 表現の近似度合を測る上で，元の MIPOP の上界値と下界値がわかると都合がよい．

いま MIPOP を MIP に変換する際に，x においても z においても不等式で近似した表現 $\mathcal{M}^{\mathrm{C}}_{\mathrm{relax}}$ を用いていたとしよう．このとき，z は真の積 xy を含む範囲のなかで値をとることができる．そのため，$\mathcal{M}^{\mathrm{C}}_{\mathrm{relax}}$ のみで変換された MIP の最適値は元の MIPOP の下界値となる．

一方で，どちらも等式で表した $\mathcal{M}^{\mathrm{C}}_{\mathrm{tight}}$ のみを用いた場合は，厳密に $z = xy$ が成り立つが，x の取りうる値は 0-1 変数 μ_i で表せる値に制限されている．これは実行可能領域がせばまることを意味している．よって，$\mathcal{M}^{\mathrm{C}}_{\mathrm{tight}}$ のみで変換された MIP の最適値は元の MIPOP の上界値となる．なお，元の MIPOP の制約条件によっては実行可能解が求まらないことに注意が必要である．

いまある p において計算した MIPOP の上界値と下界値の差が十分小さいとき，精度のよい最適解が求まっていると考えることができる．一方，上界値と下界値の差が大きいときは，p が小さすぎると想定されるので，より大きい p で解きなおす必要がある．

2.4 連続最適化による精錬と上界値の計算

近似 MIP 表現を用いた MIP の最適解は元の MIPOP の厳密解ではない．そこで，連続最適化に対する局所的最適化手法を用いて，より精度の高い解を求める，つまり解の精錬をすることを考える．ただし，連続最適化の解法では整数変数を扱うことができないので，元の MIPOP に整数変数を含む場合は，整数変数は MIP で求めた解に固定する．

MIP で求めた解は元の MIPOP の良い近似解であるので，連続最適化の解法の初期点とし利用する．そうすることによって，適切な降下法を用いれば，よりよい最適解を得ることが保証できる．このアイデアを実現するためには，精錬に利用する連続最適化の解法は warm start できる必要がある．そのような手法には逐次 2 次計画法がある．

MIPOP が整数変数を含まないとき，つまり連続最適化問題であったとき，局所的最適化手法で局所的最適解を求めることができる．この局所的最適解やその目的関数値は変換された MIP の暫定解や上界値として利用することができる．

3 応用

この節では MIPOP として表される応用問題をいくつか取り上げ，その特徴を利用した MIP への定式化手法を考える．

3.1 Binary quadtatic optimizaiton

決定変数がすべて整数変数となる MIPOP は，2.1.2 節で提案した MIP 表現 $\mathcal{M}^{\mathrm{integer}}$ と 2.2.2 節で与えたテクニックを利用することによって，等価でコンパクトな MIP へと変換できる．

ここでは，その具体的な例として，次の Binary unconstrained quadtatic optimizaiton (BUQO) の変換方法を紹介する．

$$\begin{array}{ll} \min & x^\top A x + b^\top x \\ \text{s.t.} & x \in \{0,1\}^n \end{array}$$

この問題は 2.1.2 節の $\mathcal{M}^{\mathrm{I}}$ と 2.2.2 節で与えたテクニックを使うと，以下のように変換できる．

$$\begin{array}{ll}
\min & \displaystyle\sum_{i=1}^{m} z_i + b^\top x \\
\text{s.t.} & w = Ax \\
& -M_i x_i \le z_i \le M_i x_i,\ i = 1,\ldots,n \\
& w_i - M_i(1-x_i) \le z_i \le w_i + M_i(1-x_i),\ i = 1,\ldots,n \\
& x \in \{0,1\}^n
\end{array}$$

ただし，$M_i = \displaystyle\sum_{j=1}^{n} |A_{i,j}|$ である．

この問題では元の問題の変数 x がそのまま 0-1 変数として使われていることに注意して欲しい．この MIP は BUQP と等価な問題である．さらに 0-1 変数は BUQO の x だけであり，追加された連続変数，等式制約，不等式制約の数はそれぞれ n，n，$4n$ である．

この変換は基本的には Kaufman と Broeckx [7] による 2 次割当問題の MIP 表現と同じである．そのため，連続緩和した問題*2はもとの BUQO を十分反映していない弱い緩和問題となる可能性がある．2 次割当問題については，より強い緩和問題を構成できる MIP 表現がいろいろと提案されている [12]．MIP 表現 $\mathcal{M}^{\mathrm{integer}}$ においてもそのような表現を考えることが今後の課題となる．

3.2 ベイズ最適化

ベイズ最適化は目的関数 f の計算に時間がかかりその勾配が計算できないような最適化問題に対する解法の一つある [11]．ベイズ最適化はシミュレーション最適化や機械学習などに現れるハイパーパラメータのチューニングなどに利用されている．

ベイズ最適化は，これまでに評価された点 x^t と目的関数値 $f(x^t)$ の集合 (以下データとよぶ) をもとにガウス過程回帰によって目的関数を推定し，その推定された関数の期待値と分散を利用して最適解の候補を探していくという手法である．その最適解の候補の探索は推定された期待値と分散によって構成された獲得関数とよばれる関数を最小化することによって行われる．

よく用いられる獲得関数に次のものがある．

$$g(x) = m(x) - \alpha\rho(x)$$

ここで，$m(x)$ および $\rho(x)$ はガウス過程回帰によって推定された目的関数の期待値 (関数) と分散 (関数) である，期待値 $m(x)$ を最小化することで，目的関数 f の最小化を近似的に行うことができる．しかしながら，少ないデータによって推定された $m(x)$ は正しくない可能性がある．そこで，その推定誤差を考慮しているのが，分散によって構成された第 2 項 $-\alpha\rho(x)$ である．この α は最小化と誤差のバランスを調整するパラメータである．

ガウス過程回帰では，カーネル関数 $\kappa : R^n \times R^n \to R$ と T 個のデータ $\{(x^t, f(x^t))\}$ を用いて

$$m(x) = \sum_{t=1}^{T} \beta_t \kappa(x, x^t)$$

---

*2 整数制約をなくした問題を連続緩和問題とよぶ．連続緩和問題の最適値はもとの問題の最適値の下界値となる．ここではよりよい下界値を与える緩和問題を「強い」と表現している．

と

$$\rho(x) = \kappa(x,x) + \sum_{s=1}^{T}\sum_{t=1}^{T} G_{st}\kappa(x,x^s)\kappa(x,x^t)$$

と表される (定数項等は省略している)．ここで β_t および G_{st} はガウス過程回帰で求められる定数である．その詳細については [11] やその参考文献をみてほしい．

もともとの目的関数 f が凸関数であったとしても，獲得関数 g は一般には非凸な関数になるため，その大域的最小解を求めることは難しい．そのことがベイズ最適化の応用先を限定する理由の一つになっている．

ここではカーネル関数 κ が多項式カーネルのときに，前節で与えた MIP 表現を用いて，獲得関数の最小化問題を MIP に変換することを考える．多項式カーネルは

$$\kappa(x,y) = (x^\top y + c)^r$$

として定義される関数である．ただし c および r は正の定数である．ここでは簡単のため，$c = 1, r = 4$ とする．さらにデータ x は適当な正規化によって，$x \in [0,1]^n$ であるとする．

ここで

$$w_t = x^\top x^t + 1$$

とすると，多項式カーネルは

$$\kappa(x,x^t) = (w_t)^4$$

となる，ここで，$w_t \in [1, n+1]$ であり，$(w_t)^\ell \in [1, (n+1)^\ell]$ となる．

多項式カーネルは，$\gamma_1 = w_t w_t$，$\gamma_2 = \gamma_1 w_t$，$\kappa(x,x^t) = \gamma_2 w_t$ と表せるため，3 回 $\mathcal{M}^{\mathrm{C}}$ を使うことで MIP として表現できる．このとき必要となる 0-1 変数は w_t を表す p 個だけであることに注意してほしい．もちろん，$\kappa(x,x^t) = \gamma_1\gamma_1$ とすれば 2 回の積ですむ．しかしながら，この場合は γ_1 を 0-1 変数で表す必要があるため，必要となる 0-1 変数は 2 倍となる．

カーネル $\kappa(x,x^t)$ の関数値を MIP 表現できたら，その線形和である $m(x)$ は線形な式で表せる．一方，分散 $\rho(x)$ の第 2 項の計算には 2.2.2 節のテクニックを使って MIP で表現する必要がある．そのとき，$\kappa(x,x^t), t = 1,\ldots,T$ を 0-1 変数で表現する必要があり，そのときに必要となる 0-1 変数の数は Tp である．また，分散の第 1 項 $\kappa(x,x) = (x^\top x + c)^r$ の MIP 表現には，まず $z_i = x_i^2$ を MIP 表現し，さらに $\sum_{i=1}^{n} z_i + c$ の r 乗の MIP 表現を行う．そのためには，$np + p$ 個の 0-1 変数が必要となる．

獲得関数 g を MIP で表現するために必要な 0-1 変数の数は，$\{w_t\}$ を表すために Tp 個，$\kappa(x,x^t)$ を表すために Tp 個，$\kappa(x,x)$ を表すために $(n+1)p$ 個となるから，トータルで $(n+2T+1)p$ である．なお，$m(x)$ のみの最小化を行う場合は，Tp だけでよいことに注意してほしい．

この副節の初めにも述べたように，ベイズ最適化は目的関数 f の評価に時間がかかる問題の解法である．そのような問題に対して，ベイズ最適化で目的関数 f を評価する回数は高々数十になることが多い．また，獲得関数の大域最適化には目的関数 f の評価時間程度の時間をかけても問題ない．そのため，目的関数の評価に時間がかかる問題であれば，ここで考えた手法を使う価値が十分にある．

3.3 非線形相補性問題

次の条件をみたす $x \in R^n$ を求める問題を非線形相補性問題 (Nonlinear Complemetarity Problem, NCP) とよぶ [3].

$$x_i \geq 0, F_i(x) \geq 0, x_i F_i(x) = 0,\ i = 1,\ldots,n$$

ここで，F_i は R^n から R への関数である．NCP はゲーム理論や交通流均衡問題を定式化するときに現れる基本的な問題である．関数 F_i がよい性質をもつときは効率のよい解法が存在するが，一般の関数に対しては NP 困難な問題であることが知られている．

NCP は以下の二つの最適化問題と等価である．

$$\begin{array}{ll} \min & \displaystyle\sum_{i=1}^{n} x_i F_i(x) \\ \text{s.t.} & x_i \geq 0, F_i(x) \geq 0,\ i = 1, \ldots, n \end{array}$$

$$\begin{array}{ll} \min & 0 \\ \text{s.t.} & 0 \leq x_i \leq M\mu_i \\ & 0 \leq F_i(x) \leq M(1-\mu_i) \\ & \mu \in \{0,1\}^n \end{array}$$

ここで M は十分大きい正の定数である．関数 F_i が線形な関数のとき，下の問題は MIP になる．一方，上の問題は一般には非凸な連続最適化問題となる．どちらも F_i が 2 次以上の多項式のときには難しい最適化問題である．

交通流均衡問題では F_i が多項式関数として与えられる．関数 F_i が多項式関数であるときには，本論文の手法を用いれば，どちらの最適化問題も MIP として表現することができる．

3.4 DC 最適化

汎用 MIP ソルバーである Gurobi[5] や CPLEX[6] の最新版では混合 2 次最適化問題を解くことができる．混合 2 次最適化問題を連続緩和した問題は一般には非凸の 2 次最適化問題となり，それ自体も解くことが難しい．連続緩和した問題が凸最適化問題であれば，その大域的最適解を求めることが容易になり，分枝限定法などのための下界値も効率よく計算できる．そのため，上記の汎用ソルバーで混合 2 次最適化問題を解く場合には，連続緩和した問題が凸最適化となることが望ましい．

また，3.1 節末でも議論したように，MIP 表現の連続緩和問題が弱すぎると，汎用ソルバーで最適解を求めるのに時間がかかることがある．そのようなときは，凸 2 次関数の部分を残し，非凸な関数の部分に対してのみ MIP 表現を行うと，より強い連続緩和問題となることが期待できる．混合 2 次最適化問題を扱える汎用ソルバーを使う際は，MIP として定式化するよりも，連続緩和問題が凸となる混合 2 次最適化問題で定式化した方がよい場合がありうる．

そこで，2 次形式 $x^\top Ax$ が非凸な関数であるとき，二つの凸関数の差 (diffrence of convex functions, DC) で表し，凸関数の部分はそのままに残し，残りの部分に対して MIP 表現を使うことを考える．以下では簡単な例を使ってこのアイデアを説明する．

いま行列 A を対称行列とし，その最小固有値を γ とする．さらに，$\bar{\gamma} = \max\{0, -\gamma\}$ とする．このとき，$\bar{\gamma} \geq 0$ であり $A + \bar{\gamma}I$ は半正定値対称行列になる．ここで，I は単位行列である．2 次形式 $x^\top Ax$ は

$$x^\top Ax = x^\top (A + \gamma I)x - \gamma x^\top x$$

として二つの凸関数 $x^\top (A + \gamma I)x$ と $\gamma x^\top x$ の差で表せる．ここで，$-\gamma x^\top x$ を MIP 表現すれば，得られた問題の連続緩和問題は凸最適化問題となる．

ここで，3.1 節の BUQP に対して，そのアプローチを取ると

$$\begin{array}{ll}\min & x^\top (A+\bar{\gamma} I)x + \sum_{i=1}^{n}(b_i - \bar{\gamma})x_i \\ \text{s.t.} & x \in \{0,1\}^n\end{array}$$

と表すことができる [2]．ここで，$x_i \in [0,1]$ のときは $x_i^2 = x_i$ となることを用いている．この問題の連続緩和問題は凸 2 次最適化問題となる．

4　数値計算例

この節では非凸な多項式最適化問題を MIP にして定式化して解いた数値実験結果を報告する．MIP への変換には $\mathcal{M}^C$ を用い，MIP ソルバーとしては Gurobi 9.1.1 を用いた．実験は Mac mini (CPU: Apple M1 チップ, 2020, Memory: 16GB) 上で行った．

テスト問題には [9] に掲載されていた以下の Example 2 と Example 3 を用いた．ただし，$\mathcal{M}^C$ で変換する際には，変数の上下限が必要となるため，どちらのテスト問題においても制約 $-2 \le x_1 \le 2,\ -2 \le x_2 \le 2$ を加えている．

Example 2

$$\begin{array}{ll}\min & (x_1^2+1)^2 + (x_2^2+1)^2 - 2(x_1+x_2+1)^2 \\ \text{s.t.} & -2 \le x_1 \le 2,\ -2 \le x_2 \le 2\end{array}$$

この問題の大域的最適解は $(x_1^*, x_2^*) \simeq (1.3247, 1.3247)$ である．また，大域的最適解における最適値 f^* は $f^* \simeq -11.4581$ である．

Example 3

$$\begin{array}{ll}\min & x_1^2 x_2^2 (x_1^2 + x_2^2 - 1) \\ \text{s.t.} & -2 \le x_1 \le 2,\ -2 \le x_2 \le 2\end{array}$$

この問題の大域的最適解は $(x_1^*, x_2^*) = (\pm(1/\sqrt{3}), \pm(1/\sqrt{3}))$（複号順不同）であり，最適値は $-1/27 \simeq -0.0370370$ である．

これらのテスト問題に対して，様々な p で近似した MIP を解いた結果を，表 1 と 2 にまとめる．それぞれの表において，「連続変数」と「0-1 変数」は変換された MIP に含まれる連続変数と 0-1 変数の数である．また，「目的関数値」は MIP で求めた最適解を元の問題の目的関数に入れた値である．

表 1 と 2 より，連続変数と 0-1 変数の数は p の増加に対して線形に変化することがわかる．さらに，p の増加に伴い，MIP の最適解による目的関数値が 元の問題の最適値に近づくことがわかる．比較的小さな p でも，精度のよい目的関数値が得られていることがわかる．

5　まとめ

本論文では，Zhu ら [13] が提案した $x, y \in [0,1]$ の積を近似的にあらわす MIP 表現を一般化した MIP 表現を与えた．さらにその MIP 表現を利用することによって，2 次形式や 3 次形式をコンパクトに表す手法を提案した．また，決定変数が離散変数であるときは，等価な MIP に表現可能であることを示した．

表 1　Example 2 に対する数値実験結果

	MIP の規模			
p	連続変数	0-1 変数	計算時間 (s)	目的関数値
4	34	20	0.06	-1.00000000e+01
6	44	30	0.08	-1.14315962e+01
8	54	40	0.19	-1.14541620e+01
10	64	50	0.27	-1.14576048e+01
12	74	60	0.61	-1.14578855e+01
14	84	70	1.84	-1.14580399e+01
16	94	80	119.36	-1.14580608e+01
18	104	90	677.64	-1.14580606e+01
20	114	100	270.42	-1.14580580e+01

表 2　Example 3 に対する数値実験結果

	MIP の規模			
p	連続変数	0-1 変数	計算時間 (s)	目的関数値
4	24	16	0.08	0.00000000+e00
6	32	24	0.11	1.17225647e-02
8	40	32	0.22	-3.64965251e-02
10	48	40	0.46	-3.67421211e-02
12	56	48	0.72	-3.70229807e-02
14	64	56	1.18	-3.70363397e-02
16	72	64	27.94	-3.70368800e-02
18	80	72	1681.51	-3.70367940e-02
20	88	80	1112.48	-3.70370195e-02

本論文のアイデアはシンプルで，簡単に理解できるものである．一方で，その実装はかなりの手間がかかる．特に，2.3 節で議論したような下界値と上界値から p を適応的に変化させて解くような手法や 3.2 節で考えたベイズ最適化を実現するためには，自動で MIP に定式化するツールが必要である．現在，著者らは与えられた MIPOP に対して本論文で提案した MIP 表現を用いて自動で MIP に定式化するツールの開発に取り組んでいる．また，3.1 節でも議論したように，MIPOP によっては，本論文で提案している MIP 表現ではよい連続緩和問題を与えないこともありうる．多項式の性質などを利用してより強い緩和問題を与えるような MIP 表現を構築することは今後の重要な課題の一つである．

参考文献

[1] M. F. Anjos and J. B. Lasserre, eds., Handbook on Semidefinite, Conic and Polynomial Optimization. Vol. 166. Springer Science & Business Media, 2011.

[2] C. Bliek, P. Bonami, and A. Lodi, "Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report," the proceedings of the twenty-sixth RAMP Symposium, the Research Association of Mathematical Programming in the Operations Research Society of Japan, 2014.

[3] F. Facchinei and J.-S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Springer New York, NY, 2003.

[4] 藤江哲也，整数計画法による定式化入門， オペレーションズ・リサーチ，Vol. 57，pp. 190-197, 2012.

[5] Gurobi Optimization: Gurobi Optimizer Reference Manual Version 10.0, Gurobi Optimization, 2022.

[6] IBM ILOG: IBM ILOG CPLEX Optimization Studio 22.1.0 documentation, IBM ILOG, 2022.

[7] L. Kaufman and F. Broeckx. "An algorithm for the quadratic assignment problem using benders' decomposition," European Journal of Operational Research, Vol. 2, pp. 204211, 1978.

[8] G. Kochenberger, J. K. Hao, F. Glover, M. Lewis, Z. L, H. Wang and Y. Wang, "The unconstrained binary quadratic programming problem: a survey," Journal of combinatorial optimization, Vol. 28, pp. 58-81, 2014.

[9] J. B. Lasserre, "Global optimization with polynomials and the problem of moment," SIAM Journal on Optimization, Vol. 11, pp. 796-817, 2001.

[10] L. A. Wolsey, Integer Programming. John Wiley & Sons, 2020.

[11] 松井孝太, 金森研太, 豊浦和明, 竹内一郎, ベイズ最適化の基礎と材料工学への応用, まてりあ, Vol. 58, pp. 12-16, 2019.

[12] H. Zhang, C. Beltran-Royo and L. Ma, "Solving the quadratic assignment problem by means of general purpose mixed integer linear programming solvers," Annals of Operations Research, Vol. 207, pp. 261278, 2013.

[13] J. Zhu, N. A. Azam, S. Cao, R. Ido, K. Haraguchi, L. Zhao, H. Nagamochi, T. Akutsu, "Molecular design based on integer programming and quadratic descriptors in a two layered model," 21st international Conference on Bioinformatics (InCoB), pp. 21-23, November, 2022.