論文

Benders の分割法と 遺伝的アルゴリズムによる混合0-1計画法†

坂和 正敏* 丹羽 啓一* 西崎 一郎*

本論文では、混合0-1計画問題に焦点をあて、Benders の分割法と遺伝的アルゴリズムを用いた近似解法の提案を試みる。混合0-1計画問題に対する Benders の分割法について概観した後、分割された問題に対する遺伝的アルゴリズムの適用方法について検討する。このような検討に基づいて、分割された問題に遺伝的アルゴリズムを適用するための適切なコード化と制約条件を満たすためのデコーディングアルゴリズムを提案する。さらに、提案手法の実行可能性と有効性を示すため、数値実験により、得られた解の精度や要した計算時間について混合0-1計画問題に対する分枝限定法との比較・検討を行う。

キーワード：混合 0-1 計画法, Benders の分割法, 遺伝的アルゴリズム

1. はじめに

現実の意思決定状況を数理計画問題として定式化する場合よりも、連続変数のみの計画問題として定式化される場合よりも、連続変数と0-1変数の混在する混合0-1計画問題として定式化される場合がしばしば見受けられる。例えば、具体的な問題の例として施設配置問題などが挙げられる[1]。

このような混合0-1計画問題に対して、これまで数理計画法的見地からの様々な解法が提案され、適用されてきたが、未だに実用的な解法が確立されているとはいえない[1]。現時点での混合0-1計画問題に対する最も実用的な解法としては分枝限定法が有力視されている。分枝限定法は原問題の制約条件を緩和することで得られる緩和問題を解き、得られた解をもとに解空間を分割する分枝過程と解の有界条件を用いて枝切りを行い探索数を減少させる限定過程に分けられる。分枝限定法の問題点としては、探索数を減少するために用いられる限定過程において良い有界条件が得られなければ組合せ爆発を起こす可能性が存在するという点が挙げられる。

混合整数計画問題をそのまま解くことを考えた場合、従来の解法では、整数変数の値を決定する際に非常に計算時間がかかっており効率的でないという問題点があった。そのような問題点に対して、Benders によって提案された分割法[2]を用いることで問題点の解決を図る。Benders の分割法を混合整数計画問題に適用

することにより、混合整数計画問題を線形計画問題と整数計画問題に分割することが可能となり、それぞれの問題を繰り返し解くことによってもとの問題の最適解を得ることができる。

つまり、Benders の分割法において繰り返し現れる整数計画問題に対する有効な解法が存在すれば、従来の解法に比べ、計算時間において効率的なアルゴリズムを得ることが可能となるかもしれない。

遺伝的アルゴリズム (Genetic Algorithm: 以下必要に応じて GA と略記する) は、Holland[3]らによって提案されたアルゴリズムで、自然界におけるシステムの適応過程を説明し、生物における進化のメカニズムを模倣している。

このアルゴリズムは、Holland らによって提案された後、しばらくの間はそれほど注目されなかったが、遺伝的アルゴリズムに関する国際会議や Goldberg の著書[4]によって脚光を浴び、近年、最適化、適応、学習の方法論として注目を集めている[5-7]。

これまでに GA は様々な組合せ最適化問題に適用され、その有効性が報告されてきている。しかし、残念ながら、混合0-1計画問題に対する GA に基づく効果的な近似解法の提案や混合0-1計画問題に対する数理計画法的立場からの実用解法として知られている分枝限定法との性能比較は、これまでほとんどなされてきていない。

このような状況の下で、本論文では、混合0-1計画問題に対する実用的な解法として、Benders の分割法と GA を効率的に用いた近似解法について考察する。混合0-1計画問題に対する Benders の分割法によって分割された問題に対して、連続変数の線形計画問題には線形計画法を適用するとともに、0-1変数を含む問題に対しては GA を適用するという考えに基づいて、Ben-

† Mixed 0-1 Programming through Benders' Partitioning Procedures and Genetic Algorithms
Masatoshi SAKAWA, Keiichi NIWA and Ichiro NISHIZAKI

* 広島大学工学部第三類(電気系)
Faculty of Engineering, Hiroshima University, Higashi-Hiroshima

ders の分割法と遺伝的アルゴリズムによる混合0-1計画法を提案する。ここで、分割された0-1計画問題にGAを適用する際に、筆者らによって提案され、その有効性が示されている2重構造文字列のGA[7-9]の適用と、問題の構造に適したデコーディングアルゴリズムを提案する。さらに、提案した手法の実行可能性と有効性を示すため、数値実験により、得られた解の精度や要した計算時間について混合0-1計画問題に対する分枝限定法との比較・検討を試みる。

2. 混合0-1計画問題

一般に、混合0-1計画問題は次のように定式化される。

$$\left. \begin{array}{l} \text{minimize } \mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{y} \\ \text{subject to } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \geq \mathbf{b} \\ \mathbf{x} \geq 0 \\ \mathbf{y} \in \{0,1\}^{n'} \end{array} \right\} \quad (1)$$

ここで、 $\mathbf{c}$ は n 次元費用係数行ベクトル、 $\mathbf{d}$ は n' 次元費用係数行ベクトル、 $\mathbf{x}$ は連続決定変数の n 次元列ベクトル、 $\mathbf{y}$ は0-1決定変数の n' 次元列ベクトル、 $\mathbf{A}$ は $m \times n$ の係数行列、 $\mathbf{B}$ は $m \times n'$ の係数行列、 $\mathbf{b}$ は m 次元列右辺定数ベクトルを表し、 $\mathbf{A}$ 、 $\mathbf{B}$ 、 $\mathbf{b}$ の各成分は全て正であると仮定する。

本論文では、簡単化のため、 $\mathbf{c}$ 、 $\mathbf{d}$ の各成分は全て正であると仮定する。

このような混合0-1計画問題の実世界における例として、施設配置問題[1]が挙げられる。施設配置問題は、物資の輸送量を表す決定変数を連続変数で表し、物資を一時的に保管する配送センターの建設場所の決定変数を0-1変数で表すことにより混合0-1計画問題として定式化される。

3. Benders の分割法

本論文では、混合0-1計画問題に対する実用的な解法として、Benders の分割法[2]と2重構造文字列のGA[7-9]を効率的に用いた近似解法について考察する。Benders の分割法[2]は、0-1変数を固定した際に得られる線形計画問題と連続変数を固定した際に得られる0-1計画問題を繰り返し解くことで、混合0-1計画問題を解くために要求される端点や端線を生成して、有限回の繰り返しで最適解を得ることができる。

本節では、本論文で用いる Benders の分割法について考察する。問題(1)で定式化される混合0-1計画問題において、0-1変数 $\mathbf{y}$ が固定されると $\mathbf{y}$ を含む項はすべて定数となるので通常の線形計画問題に帰着されることに注目しよう。このような線形計画問題に対して、

$\mathbf{u}$ を双対変数とする双対問題を定式化すれば

$$\left. \begin{array}{l} \text{maximize } \mathbf{d}\mathbf{y} + \mathbf{u}(\mathbf{b} - \mathbf{B}\mathbf{y}) \\ \text{subject to } \mathbf{u}\mathbf{A} \leq \mathbf{c} \\ \mathbf{u} \geq 0 \end{array} \right\} \quad (2)$$

のようになる。ここで、この双対問題の制約式は、0-1変数 $\mathbf{y}$ の値に依存しない実行可能領域を決定していることに注意しよう。そこで、双対問題の実行可能領域を

$$S = \{\mathbf{u} \mid \mathbf{u}\mathbf{A} \leq \mathbf{c}, \mathbf{u} \geq 0\} \quad (3)$$

と書き表し、問題(2)で定式化される双対問題を解くことによって得られる端点や端線を有効に利用する方法を考察する。

実行可能領域 S は凸であるので、凸体の分解定理[10]を用いることにより有限個の端点 $\mathbf{u}^s, s=1, \dots, k$ と端線ベクトル $\boldsymbol{\mu}^t, t=1, \dots, l$ を用いて表現することができる。つまり、 S を実行可能領域にもつ問題(2)に対しても端点と端線で表すことができる。 S の任意の端線は、問題(2)を解くことによって得られるが、端線が得られる場合、問題(2)は上に非有界となる。つまり、 S の任意の端線ベクトル $\boldsymbol{\mu}^t$ を用いると次式が成立することになる。

$$\boldsymbol{\mu}^t(\mathbf{b} - \mathbf{B}\mathbf{y}) > 0 \quad (4)$$

そこで、問題(2)の原問題である問題(1)を実行可能にするためには、0-1変数 $\mathbf{y}$ が

$$\boldsymbol{\mu}^t(\mathbf{b} - \mathbf{B}\mathbf{y}) \leq 0 \quad (5)$$

を満たさなければならない。したがって、問題(2)を解く過程で端線が得られたとき、式(5)を制約として追加すれば、 S を満たした実行可能解を得ることができる。このようにして最適解 $\mathbf{u}^s$ が端点の一つとして表される。また、その目的関数の値は

$$\mathbf{d}\mathbf{y} + \mathbf{u}^s(\mathbf{b} - \mathbf{B}\mathbf{y}) \quad (6)$$

となるので、もし仮にすべての端点と端線が既知であれば、もとの問題を解くことは $\mathbf{y}$ に関する計画問題

$$\left. \begin{array}{l} \text{minimize } t_0 \\ \text{subject to } \mathbf{d}\mathbf{y} + \mathbf{u}^s(\mathbf{b} - \mathbf{B}\mathbf{y}) \leq t_0, s=1, \dots, k \\ \boldsymbol{\mu}^t(\mathbf{b} - \mathbf{B}\mathbf{y}) \leq 0, t=1, \dots, l \\ \mathbf{y} \in \{0,1\}^{n'} \end{array} \right\} \quad (7)$$

を解くことと等価であることがわかる。

ここで、問題(7)を解くときの最大の問題点は、制約式を表すために用いられている端点および端線の数である。問題の規模が比較的小さいときには、すべての端点や端線を列挙して、問題(7)の問題を解くことは比較的容易であるかも知れないが、問題の規模が大き

なればなるほど端点や端線の数が増大になってしまい、すべての端点と端線を列挙することは不能になってしまうことは明白である。

そこで、問題(2)を解くことにより、必要とされる端点や端線を生成して、得られた端点や端線に基づいて0-1計画問題(7)の制約式を作成する。さらに追加された制約式を含む0-1計画問題を解くという過程を繰り返す。つまり、必要に応じて得られた端点や端線から0-1計画問題の制約式を追加していくことで最適解を得ることができる。

これまでの議論に基づいて Benders の分割法のアルゴリズム[1,2,11]を要約すると次のようになる。

Benders の分割法のアルゴリズム

手順1 0-1変数 y の値を固定するために、適当な $\bar{y}$ を選び、問題(2)の y に代入して得られた問題を解く。そのとき、問題が実行不可能であれば手順2に行く。最適解をもてば手順3に行く。上に非有界であれば手順4に行く。

手順2 問題(2)の原問題である問題(1)は下に非有界であるから手順8に行く。

手順3 最適解 u^* をもてば、 u^* を端点の集合 V に入れて、手順5に行く。

手順4 端線ベクトル μ' を求めて端線の集合 W に入れたあと、手順5に行く。

手順5 V, W によって定められる t_0, y に関する問題(7)の緩和問題を解く。もし緩和問題が実行不可能であれば手順2に行く。そうでなければ、得られた最適解を $(\bar{t}_0, \bar{y})$ とし手順6に行く。

手順6 $\bar{y}$ により定まる u に関する問題(2)を解く。もし下に非有界であれば、端線ベクトル μ' が得られるので端線集合 W に加えて、手順5に行く。最適解 $\hat{u}$ が存在すれば手順7に行く。

手順7 得られた最適解 $\hat{u}$ に対して

$$\bar{t}_0 \geq d \bar{y} + \hat{u}(b - B \bar{y}) \quad (8)$$

が成立するかどうかを確認する。成立すれば、 $\bar{y}$ と $\hat{x}$ (手順6において解くべき問題の双対問題の最適解)は原問題の最適解であるので手順8に行く。成立しない場合、次式が成立するかどうかを確認する。

$$\bar{t}_0 < d \bar{y} + \hat{u}(b - B \bar{y}) \quad (9)$$

成立すれば $\hat{u}$ を集合 V に加えて手順5に戻る。

手順8 アルゴリズムを終了する。

このような Benders の分割法のアルゴリズムによれば、必要に応じて端点や端線を生成することにより、すべての端点と端線を列挙することなく、比較的少な

い数の端点や端線の生成により、有限回の反復で最適解に到達することができる。

上述した Benders の分割法のアルゴリズムにおいて、繰り返し現れる0-1変数を含む計画問題に対して2重構造文字列のGA[7-9]を効率的に用いた近似解法を導入するわけであるが、GAが近似解法であることから手順7の終了条件では必ずしもアルゴリズムが終了しない恐れがある。そこでGAを用いて繰り返し解くことになる0-1計画問題において得られる解の値が更新されなかった回数に応じてアルゴリズムを終了させるような条件を加えることとする。

まず、 $\bar{t}_0$ の値を保存するための変数として $\hat{t}_0$ を、 $\bar{t}_0$ が更新されなかった回数を調べるために $count_t$ を導入し、アルゴリズムを終了させるためのパラメータとして c_{end} を導入する。これらを用いることによってアルゴリズムの終了条件を加えることとし、手順5および手順7を改訂する。

GA の終了条件を加えた Benders の分割法のアルゴリズム

手順5' V, W によって定められる t_0, y に関する問題(7)の緩和問題をGAにより解く。もし緩和問題が実行不可能であれば手順2に行く。そうでなければ、得られた近似解を $(\bar{t}_0, \bar{y})$ とする。アルゴリズムの反復回数が1回目であれば、 $\hat{t}_0 = \bar{t}_0$, $count_t = 0$ とし、手順6に行く。そうでなければ、 $\hat{t}_0$ と $\bar{t}_0$ の比較を行う。 $\hat{t}_0 > \bar{t}_0$ ならば、 $\hat{t}_0 = \bar{t}_0$, $count_t = 0$ とし手順6に行く。そうでなければ、 $count_t = count_t + 1$ とし手順6に行く。

手順7' 得られた最適解 $\hat{u}$ に対して

$$\bar{t}_0 \geq d \bar{y} + \hat{u}(b - B \bar{y})$$

が成立するかどうかを確認する。成立すれば、 $\bar{y}$ と $\hat{x}$ (手順6において解くべき問題の双対問題の最適解)は原問題の近似解であるので手順8に行く。成立しない場合、次式が成立するかどうかを確認する。

$$\bar{t}_0 < d \bar{y} + \hat{u}(b - B \bar{y})$$

成立すれば $\hat{u}$ を集合 V に加える。 $count_t > c_{end}$ ならば手順8に行く。そうでなければ手順5'に戻る。

このように改訂した終了条件を含む Benders の分割法を用いることでGAを混合0-1計画問題に対して適用する。

4. GA の適用

前節で考察した Benders の分割法のアルゴリズムにおいて繰り返し現れる0-1変数を含む計画問題(7)に対してGAを効率的に適用する方法を考察してみよ

う、その際に問題となるのが、問題の解集合の空間をGAの個体表現の空間に変換するコーディングの手法と、GAの個体表現の空間を問題の解集合の空間に変換するデコーディングの手法が挙げられる。

そこで本節では、問題に適した個体表現を導入するためにGAのコーディングおよびデコーディングを考察する。さらに、その個体表現に応じた遺伝的オペレータの設計を行い、その詳細について述べる。

4.1 コーディングとデコーディング

本論文においてGAを適用する問題は0-1変数を含む計画問題であり、単純GA(SGA)に用いられる0-1コードの遺伝子をそのまま導入することができれば、遺伝的オペレータの設計という点から考えても最も適している。つまり、GAの個体表現として $(y_1, \dots, y_n)$ の0-1コードとして表すことができる。しかしながら、0-1コードをそのまま用いると制約を満たさない個体(致死遺伝子を含む個体)を生成する可能性がある。

致死遺伝子を含む個体の発生を抑制する方法として次の二つが提案されている。一つは、致死遺伝子を含む個体が発生した場合、その個体の適合度を下げて自然に淘汰するペナルティ法[12]が挙げられる。もう一つは、GAの個体表現の空間から問題の解集合の空間にデコードする際、致死遺伝子を含む個体を発生しないようにデコーディングアルゴリズムを工夫する方法[7-9]が挙げられる。

前者を0-1計画問題に用いる利点は、オペレータの設計時に致死遺伝子を含む個体への対応を考える必要がないことから0-1コードをそのまま用いることができるという点が挙げられる。また、0-1コードに対応した遺伝的オペレータをそのまま導入することが可能であるとともに解の評価が容易に行えることも挙げられる。しかしながら、ペナルティ法の問題点は、致死遺伝子を含む個体が大量に発生する場合、探索の初期において個体群の多様性が失われ、GAの性能が低下することが考えられる。

後者に関しては、デコーディングの際に、GAの個体表現から問題の解集合の空間に対応付けを行うため、個体表現は0-1コード以外のコードを用いてもよい。そのためコーディングを行う際にGAのオペレータにおいて取り扱いやすいこと、また、デコーディングを行う際に複雑な操作をできるだけ用いないことを併せて考えておく効果的である。

この手法を用いる利点は、個体群に致死遺伝子を含む個体が存在しないことから、個体群の多様性を保つことができることが挙げられる。しかしながら、個体ごとにデコーディングアルゴリズムを用いることから

0-1コードを用いたGAに比べ、解の評価に時間がかかるという欠点が同時に現れる。

本論文においては、初期における個体の多様性を保つという観点からデコーディングアルゴリズムを工夫することで致死遺伝子への対応を行う。

まず、導入した文字列について述べる。本論文において、2重構造文字列を用いる。この文字列は、筆者ら[7-9]によって提案され、扱う問題によって異なるデコーディングアルゴリズムと組み合わせることで様々な離散最適化問題において良好な結果が得られている。2重構造文字列は、上段が変数の添字番号を表すとともに文字列全体における変数の優先度を表す。下段は変数の値を表す列であり、上下の要素をセットとして考える。(図1)

添字 $s(1)$	添字 $s(2)$	...	添字 $s(n)$
変数 $y_{s(1)}$	変数 $y_{s(2)}$	...	変数 $y_{s(n)}$

図1 2重構造文字列を用いた個体表現

しかしながら、この文字列を用いただけでは制約条件を満たした解を得ることができるとは限らない。そこで筆者らによって提案されている2重構造文字列の特徴を活かしたデコーディングアルゴリズムを拡張することで制約を満たした解の生成を行う。

2重構造の文字列の特徴を活かしたデコーディングアルゴリズムを考えるために、問題(7)に含まれる制約

$$dy + u^s(b - By) \leq t_0 \quad (10)$$

$$\mu^t(b - By) \leq 0 \quad (11)$$

について考察する。端点が得られた場合に生じる式(10)の右辺 t_0 が問題(7)の目的関数となっていることから、実際には式(11)を満足するようにデコードされた y に対して、式(10)の左辺を $s=1, \dots, k$ に対して計算し、その最小値を目的関数値 t_0 とすればよい。

式(11)に関しては、 $\mu^t B$ 、 $\mu^t b$ の要素がすべて正なので、添字列と0-1変数列の値に従って、制約を満たすまで変数の値を1に設定し、残りの変数を0とする。つまり、文字列の左から順に制約を満足する限り1をとる変数に対して、1としてデコードし、そうでなくなった変数からはすべて0とデコードするので、左側の変数が優先されていることがわかる。

このようなアルゴリズムを用いることで式(11)を満たした実行可能解を得ることが可能となり、デコードされた y の値を用いて、式(10)より目的関数値 t_0 を求めることができる。

2重構造文字列と文字列の左側を優先するデコーデ

ィングアルゴリズムを併せて用いることにより、変数の添字番号の位置によるデコードへの影響を均等にすることができる。次に、本論文で用いたGAの遺伝的オペレータを示す。

4.2 遺伝的オペレータ

本論文で扱うGAに関して4.1節において個体表現を示した。本節では、2重構造文字列に即した遺伝的オペレータの設計について考察する。

1) 初期集団の発生

2重構造文字列を用いることから、変数の添字列における添字番号、及び0-1変数の値{0,1}を乱数を用いて発生させ、各染色体の遺伝子とする。また、設定された個体群サイズを満たすまで個体を発生させる。

2) 評価

各世代において、問題の解空間における各個体の適合の度合を評価し、その評価に基づいて個体を淘汰、増殖する必要がある。本論文において、個体の評価値を算出するために4.1節のデコーディングアルゴリズムを用いる。各個体の評価値は t_0 で表される。

デコーディング手順と評価値の算出方法のアルゴリズムを要約すると次のようになる。

デコーディングアルゴリズム

手順1 現在までに端線ベクトル μ が得られているかを確認して、得られていれば手順2に、得られていなければ、手順3に行く。

手順2 4.1節のアルゴリズムを適用し、制約を満たした個体を作成する。その後、手順3に行く。

手順3 式(10)に y の値を代入することで最小の t_0 を求め、その値を各個体の評価値とし終了する。

このようなデコーディングアルゴリズムの具体例を示すと図2のようになる。

3) 選択

本論文においてGAを適用する問題が最小化問題であることから2)で示した個体の評価値をそのまま適合度の値として用いることができない。そこで、ランキング選択を用いることで、評価値(関数値)が低い個体から順に大きな適合度の値を与え、淘汰および増殖を行う。また、各世代において最も適合度が優れた個体を次世代に残すエリート保存戦略も併用する。

4) 交叉

SGAにおける0-1文字列に対して用いられる交叉方法を2重構造文字列に対してそのまま適用すると、同一個体内に同じ添字番号が現れるような個体、つまり致死遺伝子を含む個体を生成する可能性がある。このよ

個体の例

7	5	4	1	2	8	6	3
0	1	1	0	1	0	1	1

$$\begin{array}{l}
 y_5, y_4, y_2, y_6, y_3 \text{ が } 1 \\
 \Downarrow \\
 6y_1 + 5y_2 + 6y_3 + 9y_4 + y_5 + 8y_6 + 2y_7 + 3y_8 \leq 18 \\
 \Downarrow \\
 y_3, y_6 \text{ を } 0 \text{ に変換} \\
 \Downarrow \\
 3y_1 + 6y_2 + 4y_3 + 2y_4 + y_5 + 7y_6 + 9y_7 + 5y_8 \leq t_0 \\
 \Downarrow \\
 t_0 = 9
 \end{array}$$

図2 デコーディング

うな問題は、巡回セールスマン問題やスケジューリング問題を解く際にも起こりうる[13-15]。そこで、このような問題に対処するために順序交叉[7]を用いる。

5) 突然変異

遺伝的アルゴリズムにおいて突然変異は局所的なランダムサーチの働きをしていると考えられる。本論文においては、4.1節で示したデコーディングアルゴリズムからわかるように端線が得られている場合とそうでない場合において個体の評価値の算出方法が異なることからそれぞれの場合について考察する。まず、端線ベクトル μ が得られていない場合には、0-1文字列の値をそのまま式(10)に代入することで評価値を得ることができる。つまり、0-1変数列が評価値の算出において重要な役割を果たしている。それに対して、端線ベクトル μ が得られた場合には、0-1変数列の値が1であり添字列の左側にある変数ほどデコードの際に優先的に扱われることから、変数の添字列が個体の評価値に対して重要な役割を担うことがわかる。このような観点から2重構造文字列のGAに対して突然変異を行うとき、添字列および0-1変数列のそれぞれに対して突然変異を行う必要がある。

そこで、本論文においては、突然変異オペレータとして、添字列に対して逆位を行い、0-1変数列に対してビット反転の手法を適用することとする。このような突然変異のアルゴリズムは次のように要約される。

突然変異のアルゴリズム

手順1 あらかじめ設定された突然変異率 p_m に従って突然変異を行う個体を決定し、手順2に進む。

手順2 [0,1]の乱数 ra を発生させ、あらかじめ設定してある突然変異選択用定数 M_{pm} 以下ならば逆位を行い、そうでなければ、ビット反転を行う。

手順3 個体群サイズの大きさだけ反復を行えば終了する。

突然変異のアルゴリズムの具体例を例示すると図3のようになる。

5. 数値実験

本論文において提案した手法の有効性を調べるために次のような数値実験を行った。数値例として作成した混合0-1計画問題の規模を表1に示す。0-1変数の規模が計算時間や精度に影響すると考え、実数変数は10変数として固定し、0-1変数を100, 200, 300変数の場合に分けて実験を行った。

目的関数および制約の係数値は、[100,999]の乱数を発生させることで決定し、右辺定数は、左辺の係数値の総和に対して[0.75,0.85]の乱数を掛け合わせることで決定する。また、提案手法におけるGAのパラメータ設定は、交叉率が0.70、突然変異率が0.10、突然変異選択用定数0.50とし、GAで得られた解を近似解として採用するためのパラメータを $c_{end}=5$ とし、GAの世代数を0-1変数を100変数である問題に対しては100とし、それ以外を400とした。このような設定に基づいて各問題に対して10回ずつのシミュレーションを行った。数値実験において、計算機にはSUN SPARK STATION 10を用い、プログラミング言語にはgnu-Cを用いた。

表2では比較の対象として分枝限定法に基づく混合整数計画問題に対するBerkelaarによるフリーソフトウェア[16]を用いた計算結果が示されている。0-1変数の規模が増大するにつれて計算時間が増加し、0-1変数が300変数となった問題については解が得られなかった。

表3では提案したGAに基づくアルゴリズムによって計算された解の精度が示される。0-1変数が100変数と200変数である問題Aと問題Bに関しては最適解(表2)には到達していないものの、かなり精度の良い解が得られていると考えられる。0-1変数が300変数の問題Cに関しては最適解が未知なので、精度についての検討が困難であるが、10回の試行から得られた解はある程度収束していると考えられる。

表4では、GAに基づくアルゴリズムによる計算時間が示されている。分枝限定法の処理時間と比べると問題A, Bにおいて約半分の時間で解を得ることが可能であることがわかる。また、問題Cは、分枝限定法

個体の例

図3 突然変異

表1 問題の規模

問題の種類	連続変数	0-1 変数
A	10	100
B	10	200
C	10	300

表2 分枝限定法による結果

問題の種類	最適解	処理時間 (秒)
A	39019.64	398.45
B	73113.51	2736.76
C	—	—

表3 提案法による解の精度

問題の種類	最小値	最大値	平均値	分散
A	40793.16	41810.96	41225.46	146699.06
B	77702.91	78952.19	78357.21	301349.36
C	112299.84	119503.14	116918.36	4582973.50

表4 提案法の処理時間(秒)

問題の種類	最小値	最大値	平均値	分散
A	147.11	275.07	197.94	3228.28
B	1703.70	2094.76	1786.52	26279.32
C	3822.00	9029.00	4974.50	2783113.17

において解くことが不可能であった問題であり、この問題に対しても提案手法は、処理時間においてばらつきが見られるが、得られた解のばらつきが少ないことから良好な結果が得られたと考えられる。

各問題において Benders の分割法で端線・端点が生
成された回数が表5に示されており,その回数分0-1計
画問題(7)がGAで解かれている。

6. おわりに

本論文において, Benders の分割法とGAを用いた
混合0-1計画問題に対する近似解法を提案した。分割法
を適用することで得られた0-1計画問題に対して2重構
造文字列を用いたGAを適用し,問題の特徴を活かし
たデコーディングアルゴリズムを提案した。提案手法
の有効性を調べるために数値実験によって分枝限定法
との比較を行った。提案手法は,分枝限定法に比べて,
約半分の処理時間で近似解を得ることが可能であり,
近似精度も良好であることがいえる。また,分枝限定
法において解を得ることができない問題に対しても良
好な結果が得られた。今後,本論文で提案した手法の
現実問題への適用などにより,さらに提案した手法の
改良や拡張が望まれる。

参 考 文 献

- [1] 今野浩, 鈴木久敏(編), “整数計画法と組合せ最適化,”
日科技連, 1982.
- [2] J.F. Benders, “Partitioning procedures for solving
mixed-variables programming problems,” *Numerische
Mathematik*, vol.4, pp.238-252, 1962.
- [3] J.H. Holland, “Adaptation in Natural and Artificial
Systems,” University of Michigan Press, 1975,
MIT Press, Cambridge, 1992.
- [4] D.E. Goldberg, “Genetic Algorithms in Search,
Optimization, and Machine Learning,” Addison
Wesley, 1989.
- [5] Z. Michalewicz, “Genetic Algorithms+Data Struc-
tures=Evolution Programs,” Springer-Verlag,
1992, Second, extended edition, 1994, Third,
revised and extended edition, 1996.
- [6] M. Gen and R. Cheng, “Genetic Algorithms and
Engineering Design,” John Wiley & Sons, 1996.
- [7] 坂和正敏, 田中雅博, “遺伝的アルゴリズム,” 朝倉書
店, 1995.
- [8] 坂和正敏, 乾口雅弘, 砂田英昭, 澤田一哉, “改良型遺
伝的アルゴリズムによるファジィ多目的組合せ最適
化,” 日本ファジィ学会誌, vol.6, no.1, pp.177-185,
1994.

表5 端線、端点の生成回数

問題の種類	最小値	最大値	平均値	分散
A	7	12	9.00	5.14
B	7	8	7.20	0.18
C	7	16	8.80	8.84

- [9] 坂和正敏, 加藤浩介, 砂田英昭, 園田慶史, “多目的0-
1計画問題に対する改良型遺伝的アルゴリズムによる
対話型ファジィ満足化手法,” 日本ファジィ学会誌,
vol.7, no.2, pp.361-370, 1995.
- [10] 坂和正敏, “線形システムの最適化,” 森北出版, 1984.
- [11] 刀根薫, “数理計画,” 朝倉書店, 1978.
- [12] T. Yokota, M. Gen and Y. Li, “Genetic algorithms
for non-linear mixed integer programming problems
and its applications,” *Computers & Industrial Engi-
neering*, vol.30, no.4, pp.905-917, 1996.
- [13] T. Starkweather, S. McDaniel, K. Mathias and D.
Whitley, “A comparison of genetic sequencing
operators,” *Proceedings of Fourth International
Conference on Genetic Algorithms and their Appli-
cations*, (ed. R.K. Belew), pp.69-76. Kaufmann,
1991.
- [14] B.R. Fox and M. B. McMahon, “Genetic operators
for sequencing problems,” In *Foundations of
Genetic Algorithms*, (ed. G. J. E. Rawlins), pp.284-
300. Kaufmann, 1991.
- [15] P.W. Poon and J.N. Carter, “Genetic algorithm
crossover operators for ordering applications,”
Computers & Operations Research, vol.22, pp.135-
147, 1995.
- [16] M. Berkelaar, “lp_solve 2.0,” ftp://ftp.es.ele.tue.nl/pub/lp_solve

(1997年9月4日 受付)

(1998年3月5日 再受付)

[問い合わせ先]

〒739-8527

東広島市鏡山1-4-1

広島大学工学部第二類(電気系)

計数管理工学講座

坂和 正敏

TEL : 0824(24)7694, (22)7159

FAX : 0824(24)7694, (22)7195

E-mail : sakawa@msl.sys.hiroshima-u.ac.jp

著者紹介

坂和 正敏 (さかわ まさとし)

広島大学工学部第二類(電気系)計数管理工学講座

昭45 京大・工・数理卒, 昭50 同大学院博士課程了, 昭50 神大・工・システム助手, 昭56 同助教授, 昭62 岩手大・工・教授を経て, 平2 広島大・工・第二類(電気系)計数管理工学講座教授となり現在に至る。大規模システム, 多目的システム, ファジィシステムにおける意思決定手法とその応用に関する研究に従事。工博。著書線形システムの最適化, 非線形システムの最適化, ファジィ理論の基礎と応用, 経営数理システムの基礎(森北出版), ソフト最適化, 遺伝的アルゴリズム(朝倉書店), Fuzzy Sets and Interactive Multiobjective Optimization(Plenum Press)等。

丹羽 啓一 (にわ けいいち)

広島大学工学部第二類(電気系)計数管理工学講座

平7 関西大・工・管理工卒。平9 広島大学大学院博士課程前期了。平9 同後期入学。現在に至る。遺伝的アルゴリズムによる混合整数計画法や多重レベル計画法に関する研究に従事。

西崎 一郎 (にしざき いちろう)

広島大学工学部第二類(電気系)計数管理工学講座

昭57 神大・工・システム卒。昭59 同大学大学院修士課程了。同年 新日本製鐵(株)入社。平2 京大・経研・助手。平5 摂南大・経営情報・助教授。平9 広島大・工・第二類(電気系)計数管理工学講座助教授となり現在に至る。

Mixed 0-1 Programming through Benders' Partitioning Procedures and Genetic Algorithms

by

Masatoshi SAKAWA, Keiichi NIWA and Ichiro NISHIZAKI

Abstract :

In this paper, we focus on mixed 0-1 programming problems and propose an approximate solution method by combined use of the Benders' partitioning procedures and genetic algorithms. After overviewing the Benders' partitioning procedures, we examine how to apply genetic algorithms to the subproblems with 0-1 variables. Then we propose a coding method for applying genetic algorithms to the subproblems and a new decoding method for satisfying the constraints. Through computational experiments, the feasibility and efficiency of the proposed method are examined and compared with the branch and bound method for mixed 0-1 programming problems with respect to CPU time and performance.

Keywords : Mixed 0-1 Programming Problem, Benders' Partitioning Procedures, Genetic Algorithms

Contact Address : **Masatoshi SAKAWA**

Faculty of Engineering, Hiroshima University

1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 Japan

TEL : 0824-24-7694/22-7159

FAX : 0824-24-7694/22-7195

E-mail : sakawa@msl.sys.hiroshima-u.ac.jp