最小矛盾の概念を用いた混合0−1整数計画問題の近似解法

5G−5

原　裕貴

(株) 富士通研究所

1．はじめに

我々は，組合せ最適化に対して最小矛盾の概念を用いた近似解法の研究を行っており，すでにジョブショップスケジューリング問題および全0-1整数計画問題に対して適用した結果を報告をしている[1][2]。本稿では，我々の方式を混合0-1整数計画問題に適用した結果について報告する。

混合0-1整数計画問題は以下のように定式化される。

$$
\begin{aligned}
z = \max \quad & cx + hy \\
\text{subject to} \quad & Ax + Gy \leq b \\
& x \in \{0, 1\}^n \\
& 0 \leq y
\end{aligned}
\qquad (1)
$$

ここで、A は $m \times n$ 行列、G は $m \times l$ 行列、c は $1 \times n$ 行列、h は $1 \times l$ 行列、b は $m \times 1$ 行列である。

混合0-1整数計画問題は，生産計画問題、配置問題、スケジューリング問題など多くの応用があり、重要な問題である。混合0-1整数計画問題に対する厳密な解法としては分枝限定法、切除平面法などがあるが、一般に混合0-1整数計画問題はNP-completeであり、問題の規模が大きくなると厳密に解くことは極めて難しくなる。このため、個別の問題ごとに各種の近似手法が開発されている。本論文では混合0-1整数計画問題一般に適用可能な汎用的な近似解法を提案する。

2．最小矛盾を用いた最適化方式

我々の方式は，近傍探索法の一種である。すなわち，全ての0-1変数に0か1を割り当てて初期解を生成し，ある0-1変数の値を0から1（または1から0）へと変更していくことを繰り返していく。我々のアイデアは最小矛盾の概念を用いることである。以下では，いくつかの用語を定義する。

仮定(assumption)は，0-1変数に対する0-1値の代入，すなわち $x_i = v_i$ ($v_i \in \{0,1\}$)である。

割当(assignment)は，仮定の集合 $\{x_i = v_i : i \in N^* \subset \{0,1,\ldots,n\},\ v_i \in \{0,1\}\}$ である。

解(solution)は，すべての0-1変数に対する仮定の集合 $\{x_i = v_i : i \in \{0,1,\ldots,n\},\ v_i \in \{0,1\}\}$ であり，解S に対して $x^s = (v_1, v_2, \ldots, v_n)$を解ベクトルと呼ぶ。

解S に対して、LP(x^s)を以下のように定義する。

$$
\begin{aligned}
z_{LP}(x^s) = \max \quad & hy \\
\text{subject to} \quad & Gy \leq b - Ax^s \\
& 0 \leq y
\end{aligned}
\qquad (2)
$$

LP(x^s)は整数変数を含まない線形計画問題であり、良く知られたSimplex法を用いることによって効率良く解くことができる。LP(x^s)が解をもつとき、解S を制約充足解と呼び、そうでないとき制約非充足解と呼ぶ。

問題(1)の最適解を (x_0, y_0)，その時の目的関数値を z_0 とすると、LP(x_0)の最適解が y_0 で、$z_0 = cx_0 + z_{LP}(x_0) = cx_0 + hy_0$ であることは明らかである。

不等式 $ax \leq (<)\ b$ に対して，$ax^s > (\geq)\ b$ であるとき，この不等式を解S に対する違反制約と呼ぶ。制約違反を引き起こすような最小の割当($\subset S$)を最小矛盾(minimal conflict)と呼ぶ。

LP(x^s)が解をもつとき、双対問題の最適解を u とすると、双対定理から、$z_{LP}(x^s) = u\,(b - Ax^s)$ と表すことができる。元の問題に対して、それまでに求められた最大の目的関数値を *BestValue* とすると、不等式 $cx + u\,(b - Ax) > BestValue$ は解S に対する違反制約になる。

LP(x^s)が解をもたないとき、$v\,(b - Ax^s) < 0$ を満たすような双対レイ(dual ray) v が存在する。このとき、不等式 $v\,(b - Ax) \geq 0$ は解S に対する違反制約になる。

本手法のアルゴリズムを以下に示す。

アルゴリズム

(1) 初期解を生成し，S とする。
    MinimalConflictSet = {}
    BestValue = $-\infty$

(2) 停止条件が満足されたら、*BestSolution* と *BestValue* を出力して停止する。

(3) LP(x^s)を解く。
    (a) LP(x^s)が解をもたないならば、$v\,(b - Ax^s) < 0$ を満たすような双対レイを求め、v とする。
    (b) LP(x^s)が解をもつならばその双対解を u とする。

(4) LP(x^s)が解をもち、$BestValue < cx^s + z_{LP}(x^s)$ ならば、$BestSolution = S$，$BestValue = cx^s + z_{LP}(x^s)$ とする。

(5) 違反制約を発見する。
    (a) LP(x^s)が解をもたない場合
        $v\,(b - Ax) \geq 0$ を違反制約とする。
    (b) LP(x^s)が解をもつ場合
        $cx + u\,(b - Ax) > BestValue$ を違反制約とする。

(6) (5)で求めた違反制約の最小矛盾を求め，MC とする。

(7) *MinimalConflictSet* にMC を追加する。

An Approximate Method for Mixed 0-1 Integer Programming
Hirotaka Hara,
FUJITSU LABORATORIES LTD.

(8) MC の中からひとつの仮説 $x_r = v_r$ を選ぶ。この仮説の値を反転して新しい解 S' を生成する。すなわち、$S' = \{x_i = v'_i : i \in \{1,2,...,n\},\ v'_r = {}^{-}v_r,\ v'_i = v_i\ (i \neq r)\}$。ただし，$v=1$ ならば $^{-}v=0$，$v=0$ ならば $^{-}v=1$。

(9) S' が *MinimalConflictSet* に含まれる違反制約の一つでも含んでいたら，(8)に戻り別の仮説を選ぶ。

(10) S' を新しい解として S と置き換え，(2)に戻る。

アルゴリズムの特徴

(1) 最小矛盾の意味

解が制約非充足の場合の最小矛盾は，解が制約非充足のための十分条件を示している。したがって，最小矛盾に含まれる仮説の少なくとも一つを変更することが，解が制約充足するための必要条件となる。解が制約充足の場合の最小矛盾は，解の目的関数値が *BestValue* より悪いための十分条件になっている。したがって，最小矛盾に含まれる仮説の少なくとも一つを変更することが，解の目的関数を *BestValue* 以上に改善するための必要条件となる。

(2) 山登り法との比較

近傍探索として良く用いられる山登り法の場合，制約充足解のみを対象として，目的関数が改善される方向に変更を繰り返す。このため，制約が強く制約充足解が少なくなると，局所最適解に陥って停止する問題点が生じる。本方式は，目的関数が悪くなる方向への変更も受理されるので、局所最適解で停止することはない。また、制約非充足解も探索空間中にあるため、制約が強い問題でも効率良く動作する。

(3) ループの回避

本方式では，生成された解に対して必ず最小矛盾が生成，保存され，その後新しく生成された解に対してはこの最小矛盾とのチェックが行われるため，再び同じ解が生成されることはない。

(4) 探索空間の枝刈り

最小矛盾は，生成された解以外にも，その最小矛盾を含む解が生成されることを防ぐ。その意味において，無駄な探索空間を枝刈りする効果がある。

ヒューリスティクスの利用

上のアルゴリズム中，二つの部分でヒューリスティクスを利用することができる。一つは初期解の生成であり，もう一つはステップ8で最小矛盾の中から変更する仮説を選ぶ部分である。

我々は，元の0-1整数計画問題から整数制約を除いたLP緩和問題の解を利用している。LP緩和問題は，シンプレックス法を用いて効率的に解くことができる。

初期解としては，LP緩和解を丸めたものを使う(この解は制約充足解とは限らないことに注意)。ステップ8では，LP緩和解に最も近くなるような反転から優先的に試す。

3．実験結果

我々の手法を評価するために，分枝限定法との比較を行った。測定は Sun SPARC ELC(16Mbyte)によって行なわれ、分枝限定法の実行は商用のパッケージCPLEXを利用した。

測定用の問題としては、典型的な混合０－１整数計画問題である多品種生産計画問題をとりあげた。この問題はｐ品種の生産物のｔ期間の生産計画を立てる問題で、各期間に各生産物を生産するかどうかが０－１変数に対応する。各期間毎に資源に関する制約があり、目的関数は、セットアップコスト、生産コスト、在庫コストの和の最小化である。

二つの規模の異なる問題を作成した。問題１は５品種、８期間（４０整数変数）、問題２は５０品種、１６期間（８００整数変数）である。それぞれの問題に対して、最初に発見された制約充足解と最良の解のコストと発見にかかったＣＰＵ時間（秒）を以下の表に示す。問題１では、CPLEXの 100倍の早さで最適解を発見することができた（この解が最適であることは、CPLEXをさらに約15分実行することで証明された）。問題２では、CPLEXは３時間以上実行を行なったが、表に示した以上の解を発見することはできなかった。我々の手法は、CPLEXより約14%良い解を10分以内で発見することができた。

		本手法	分枝限定法
問題1	最初の解	162572 (0.67)	192881 (1.08)
	最良解	158912 (6.78)	158912 (675.35)
問題2	最初の解	3356921 (531.35)	3931493 (475.38)
	最良解	3319902 (3149.95)	3893975 (1346.92)

謝辞

MIT留学中、本研究に対して貴重なアドバイスをいただいたShapiro教授に深謝致します。

参考文献

[1] Hara, Yugami, and Yoshida : An Assumption-based Combinatorial Optimization System, in Proceedings of the 8th Conference on AI for Applications, Monterey (1992).

[2] Hara : Solving the Large-scale 0-1 Integer Programming Problem Using an Assumption-based Method, in Proceedings of the 10th European Conference on AI , Vienna (1992).