集合被覆問題に対する3反転近傍を用いた局所探索法

岸田　正博
(京都大学大学院情報学研究科数理工学専攻　現所属・住友電気工業(株))
指導教官　茨木俊秀 教授

1. はじめに

集合被覆問題 (SCP) は代表的な組合せ最適化問題の一つであり, 様々な実際問題への応用がある. この問題は, 要素集合 $M=\{1,...,m\}$, Mの部分集合族 S_j, $j \in N=\{1,...,n\}$ とそれらの重み c_jが与えられたとき, Mの全ての要素をカバーするようにいくつかの集合 S_jを選び, 選んだ集合に付けられた重みの総和を最小にする問題である. 0-1 変数 $x \in \{0,1\}^n$を用いると, 以下のように定式化される:

$$\begin{array}{lll} \text{minimize} & cost(x) = \sum_{j\in N} c_j x_j & \\ \text{subject to} & \sum_{j\in N} a_{ij}x_j \geq 1, & i \in M \\ & x_j \in \{0,1\}, & j \in N. \end{array}$$

ここで,

$$a_{ij} = \begin{cases} 1, & i \in S_j\text{のとき} \\ 0, & \text{そうでないとき} \end{cases}$$

であり, 変数 x_jは集合 S_jが解に含まれるなら1, そうでなければ0をとる. 集合被覆問題はNP困難であり, これまでに様々な近似解法が提案されている.

本研究では, 3反転近傍と呼ばれる大きな近傍を用いた局所探索法に基づくアルゴリズムを提案する. 3反転近傍とは, 現在の解から同時に3つ以下の変数を反転させることによって得られる解集合であり, その大きさは $O(n^3)$ である. 大きな近傍を用いることによって解の精度の向上が期待できるが, 近傍内の全ての解を列挙して調べると計算時間が非常に大きくなる. この隘路を解決するため, 解の質を悪くすることなく効率的に近傍を探索する工夫を行っている. また, SCPでは, 最適解は実行可能領域と実行不可能領域の境界に存在するが, このような問題に対しては, 実行可能領域と実行不可能領域を交互に訪れるように探索を制御する戦略的振動と呼ばれる手法が有効である. 本アルゴリズムでは, 実行不可能解をペナルティ関数によって評価しつつ, ペナルティ係数を探索状況に応じて適応的に調節することによって戦略的振動を実現している. さらに, ラグランジュ緩和問題から得られる情報を用いて, 変数 x_jの一部を0または1に固定することにより問題サイズの縮小を行っている. なお, このとき固定する変数は状況に応じて繰り返し修正を行っている. 以下, それぞれのポイントをやや詳しく説明する.

2. アルゴリズムの詳細

局所探索法と近傍　局所探索法は, 現在の解 x の近傍 $NB(x)$ 内に x より良い解があればそれに置き換える, という操作を, 近傍内に改善解がなくなるまで反復する方法である. 近傍 $NB(x)$ は解 x に多少の変更を加えて得られる解集合である. その近傍内により良い解が存在しない解を局所最適解と呼ぶ.

近傍 $NB_r(x)$ を, x からのハミング距離が r 以下の解集合と定義する. さらに, $n' = \sum_{j\in N} x_j$, $t = \max\{\sum_{i\in M} a_{ij} \mid j \in N\}$, $l = \max\{\sum_{j\in N} a_{ij} \mid i \in M\}$ とする. これらに対し, $n' \leq n$, $t \leq m$, $l \leq n$ が常に成り立つ. 本研究では, $NB_1(x)$, $NB_2(x)$ と $NB_3(x)$ を用いるが, 探索を効率化するため, $r \geq 2$ に対しては, $NB_{r-1}(x)$ に改善解がない時に限り $NB_r(x)$ を調べる. 近傍 $NB_r(x)$ 内の解を全て列挙して調べた場合, その計算時間は $O(n^r t)$ となり, rに対し指数的に増加する. しかし, 本研究で提案するアルゴリズムでは, 改善の可能性のない解の探索を省くことによって, $NB_1(x)$ に対し $O(n+tl)$ 時間, $NB_2(x) \setminus NB_1(x)$ に対し $O(n'tl)$ 時間, $NB_3(x) \setminus NB_2(x)$ に対し $O(n'tl \min\{n, tl\})$ 時間で, 近傍内の改善解を逃すことなく探索することができる. $n' \leq n$, $l \leq n$ が成り立つので, これらの計算時間は, 解を全て列挙して調べたときの計算時間より必ず小さくなっている. 特に, $n' \ll n$, $l \ll n$ であるような問題例に対しては, かなりの高速化が期待できる.

解の評価と戦略的振動　本アルゴリズムでは, 実行不可能解も探索空間に含めるため, 目的関数 $cost(x)$ をそのまま解の評価に用いるのは不適当である. そこで, 制約条件を考慮したペナルティ関数を以下のように定義し, 解の評価に用いる. 各要素 $i \in M$に対するペナルティ係数を $p_i\ (>0)$ とし, 解 x を,

$$pcost(x) = cost(x) + \sum_{i\in M} p_i \max\left\{1 - \sum_{j\in N} a_{ij}x_j, 0\right\}$$

表 1. 3 つのアルゴリズムによる最良解の比較

問題例	最良値	CNS	CFT	3-FNLS (18,000 秒) Min	Avr.	Max	3-FNLS (3,600 秒) Min	Avr.	Max
rail2536	690	692	691	691	691.1	692	691	692.0	693
rail2586	945	951	947	945	946.9	948	946	948.2	950
rail4284	1064	1070	1065	1064	1065.7	1066	1068	1069.4	1072
rail4872	1528	1534	1534	1528	1531.8	1534	1534	1536.1	1538

によって評価する.

ところで, 局所探索法を 1 度行っただけでは未探索の領域にさらによい解が隠れているという危惧が残る. また, 本研究で提案している局所探索法は, 探索空間に実行不可能解も含めているため, ペナルティ係数 p_i の値を十分大きくしない限り, 1 度の探索で必ず実行可能解が得られるという保証はない. そこで, 局所探索法が局所最適解 $\boldsymbol{x}^*$ を求めて停止した時点で各要素のペナルティ係数 p_i を更新し, 前回の局所最適解 $\boldsymbol{x}^*$ を初期解としてさらに局所探索法を続けるという方法をとる. p_i の更新ルールはやや複雑なので詳細は省くが, $\boldsymbol{x}^*$ のコストと制約の違反度の両方を睨みつつ各 p_i の増減を決定するルールになっている. このようにペナルティ係数を適応的に制御することにより, 探索の戦略的振動を実現している.

問題サイズの縮小 大規模な問題例を解くため, 変数 x_j の一部を 0 または 1 に固定することにより問題サイズの縮小を行い, 縮小された問題に対し局所探索法を適用する. 固定する変数の選択は非常に重要であるが, 我々のアルゴリズムでは, ラグランジュ緩和問題から得られる情報に基づき, 最適解に含まれる可能性の高い変数を 1 に, 低い変数を 0 に固定する. しかし, 固定した変数が必ずしも最適解と一致するとは限らないので, 縮小した問題に対し局所探索法をしばらく適用した後, 探索状況に応じて, 固定する変数の修正を行っている. これは, 大規模な問題例に対し, 非常に効果的であることが確認された.

3. 計算実験

提案アルゴリズム (3-FNLS) の性能を他の代表的な近似アルゴリズムと比較した. 実験は, ワークステーション Sun Ultra 2 Model 2300 上で C 言語を用いて行った. 問題例は OR-Library[1] の代表的なベンチマーク問題である. 表 1 に, それらの中で最も大規模な問題例に対する結果を示す. これらはイタリアの鉄道会社のスケジューリング問題で, 最適解は未知であり, サイズは $m \simeq 2,500 \sim 5,000$, $n \simeq 1,000,000$ ときわめて大規模である. 比較のため, ラグランジュ緩和に基づく 2 つの近似アルゴリズム CFT [1] と CNS [2] による最良の結果を示す. CFT は解の質においてこれまでに提案された近似アルゴリズムの中で最良のものである. 3-FNLS は, 各問題例に対し乱数の種を変えて 10 回の試行を行った. 探索の打ち切り時間は 18,000 秒とした. 表には, 18,000 秒の結果に加えて, 他の 2 つのアルゴリズムと同等の計算時間である 3600 秒の時点の結果を記している. 表中, Min, Avr., Max は, それぞれ制限時間内に得られた暫定解の最小値, 平均値, 最大値である.

表より, 3-FNLS は, 他の 2 つのアルゴリズムと同程度の計算時間を与えた場合には, CNS と同等の性能が得られることが確認できる. また, rail2586~4872 の 3 問に関しては, さらに計算時間をかけることによって, 既知の最良値を更新することができた. その後, rail2536 に関してもパラメータを調節することによって既知の最良値を更新することができた. 表 1 に示した最良値は, いずれも 3-FNLS によって新たに求まったものである. 表に示した問題例以外に対しても, 我々のアルゴリズムは, OR-Library の全ての問題例に対し最適解もしくは既知の最良値を得ている.

4. まとめ

集合被覆問題に対し, 3 反転近傍を用いた局所探索法に基づく近似アルゴリズムを提案し, 計算実験を行った. 大きな近傍を, 解の質を悪くすることなく, 効率的に探索する工夫を行った点に大きな特徴がある. 計算実験の結果, 我々のアルゴリズムは, 既存の手法 CFT などに比べるとやや時間がかかるものの, 大規模な問題例に対しての最良解を更新するなど, 良質の解を求める高い性能を有していることが確認できた.

[1] http://mscmga.ms.ic.ac.uk/jeb/orlib/scpinfo.html

参考文献

[1] Caprara, A., Fischetti, M. and Toth, P., "A heuristic method for the set covering problem," *Operations Research*, 47 (1999) 730–743.

[2] Ceria, S., Nobili, P. and Sassano, A., "A Lagrangian-based heuristic for large-scale set covering problems," *Mathematical Programming*, 81 (1998) 215–228.