

入門タブー探索法

橋本 英樹, 野々部 宏司

通信, ロジスティクス, 生産計画など社会におけるさまざまな場面で組合せ最適化問題が現れる. しかし, それらに対して厳密な最適解を効率よく求めることはしばしば困難である. そのような問題に対して, 現実的な時間で高精度の解を求めることを目的に, 近年ではメタヒューリスティクス (metaheuristics) と呼ばれる手法がよく用いられている. 本稿では, 代表的なメタヒューリスティクスの一つであるタブー探索法について基礎から簡単な例を交えながら解説した後, 高性能なタブー探索法設計のための重要な概念をいくつか紹介する.

キーワード: タブー探索法, 入門, 最適化, メタヒューリスティクス

1. はじめに

通信, ロジスティクス, 生産計画など社会におけるさまざまな場面で組合せ最適化問題が現れる. しかし, それらに対して厳密な最適解を効率よく求めることはしばしば困難である. 良質の解を少ない計算時間で探索するための基本戦略として, 局所探索法 (local search) がよく知られている. 局所探索法とは, 適当な解から始めて, 現在の解 x の近傍 $N(x)$ 内に改善解があれば移動するという操作を反復する手法である. 局所探索法の詳細については本特集に解説 [8] がある. 局所探索法は, 単純な手法であり実装が容易であるため多くの問題に自然に適用することができる基本的な最適化ツールとなっている. しかし現実には, 単純な局所探索法のみでは満足のいく結果が得られず, より高性能なアルゴリズムが必要となることも多い. この目的のために, 近年ではメタヒューリスティクス (metaheuristics) と呼ばれる手法がよく用いられている [6, 7, 9]. メタヒューリスティクスとは, 最適化問題 (特に組合せ最適化問題) に対する実用的な探索手法を設計するための一般的な枠組みを与えるものである. これには数多くの手法やアイデアが含まれるが, その中には, 局所探索法を基本として, その性能を高めるための手法と位置づけられるものが多数存在する. 代表的なものとしては, GRASP 法 (greedy randomized adaptive search procedure), 反復局所探索法 (iterated local search),

アニーリング法 (simulated annealing), タブー探索法 (tabu search) などが挙げられる.

本稿のテーマであるタブー探索法が初めて論文の中で tabu search という言葉で述べられたのは 1986 年 [1] であるが, 手法としてはそれより以前の 1970 年頃に Glover により提案されている. タブー探索法は, 局所探索法と同様に, 現在の解 x の近傍 $N(x)$ にある解 x' に移動する, という戦略を基本とする. ただし局所探索法とは違い, x' が改悪解であっても解の移動が強制的に行われるため, 局所最適解で探索が終了することなく解の移動が継続される. 一般に, 探索中で移動した解の軌跡がいくつかの解を経由して元の解に戻ってしまう現象のことをサイクリング (cycling) と呼ぶ. タブー探索法では, タブーリスト (tabu list) と呼ばれる解集合 T を定義し (例えば, 直近に訪れた解の集合など), これに含まれる解 (禁止解と呼ばれる) への移動を禁止することで元の解に戻ることを防止する.

高性能な解法を設計するには, 過去の探索で得られた情報をいろいろな形で保存しておき, 将来の探索に利用するメカニズムを組み込むことが重要である. タブー探索法では, 直近の探索履歴 (例えば直近に訪れた解の集合など) は短期メモリ (short term memory), 探索全体に渡って記録される探索履歴 (例えば統計情報など) は長期メモリ (long term memory) と呼ばれる. 短期メモリのみを用いたタブー探索法も高精度な解を生成するのに十分であるが, 一般に, 長期メモリを用いたタブー探索法のほうがさらに高性能である. 長期メモリを利用した手法は, 適応メモリ戦略 (adaptive memory programming) とも呼ばれる.

本稿では, まず 2 節でタブー探索法の設計において最低限必要な基礎事項について簡単な例を交えながら

はしもと ひでき

名古屋大学大学院情報科学研究科

〒464-8601 名古屋市千種区不老町

ののべ こうじ

法政大学デザイン工学部

〒102-8160 東京都千代田区富士見 2-17-1

解説する。そして3節では高性能なタブー探索法の設計において重要となるいくつかの概念を紹介する。タブー探索法に関するより詳細な解説は文献[2~5]などがある。

2. タブー探索法の初歩

2.1 局所探索法

タブー探索法を説明する前に、まず、その基礎となる局所探索法について説明する。ある解 x に加える小さな修正を近傍操作、そのような操作により得られる解の集合 $N(x)$ を近傍 (neighborhood) と呼ぶ。局所探索法は、適当な初期解から始め、現在の解 x よりもよい解 x' が近傍 $N(x)$ 内に存在すれば $x := x'$ と置き換える操作を、可能な限り繰り返す方法である。近傍内によりよい解が存在しない解は局所最適解 (local optimum) と呼ばれ、局所最適解が得られた時点で局所探索法は終了する。局所探索法の設計において、近傍 $N(x)$ 内に改善解が複数個存在するときどの改善解に移動するかはよく検討すべき事項である。近傍内のすべての解を調べたうえで最良の解に移動する方法は最良移動戦略 (best admissible move strategy) と呼ばれている。タブー探索法では最良移動戦略に基づいた移動が行われるため、後述する局所探索法の具体例では最良移動戦略を用いる。

簡単な例を示す。巡回セールスマン問題 (traveling salesman problem, TSP) は、 n 個の都市とそれらの間の距離が与えられたとき、すべての都市をちょうど1度ずつ訪れて出発地に戻る巡回路のうち、総移動距離が最小のものを求める問題である。距離行列が非対称である場合 (すなわち、都市 i から j への距離 d_{ij} と都市 j から i への距離 d_{ji} が、必ずしも $d_{ij} = d_{ji}$ とならない場合) の TSP はとくに非対称巡回セールスマン問題 (asymmetric TSP) と呼ばれる。TSP では、都市の訪問順序で解 (巡回路) を表現できる。このように順列 σ で解を表せる問題に対しては、一般に挿入近傍や交換近傍がよく用いられる。挿入近傍 (insertion neighborhood) は一つの都市を順列のほかの位置に移動することで得られる解集合、交換近傍 (swap neighborhood) は2つの都市の順列における位置を交換することで得られる解集合である。しかし、TSP に対しては交換近傍はあまり有効ではない。TSP に対する近傍としてよく知られているものとして、Or-opt 近傍と λ -opt 近傍 ($\lambda \geq 2$) がある。Or-opt 近傍は、現在の巡回路において連続する3つ以下の都市をほかの位置に挿入することによって得られる解集合であり、挿入近

表 1 $n = 5$ の距離行列

	a	b	c	d	e
a	0	5	4	9	1
b	1	0	10	3	3
c	4	1	0	4	5
d	3	3	1	0	2
e	8	6	8	1	0

傍を含む近傍である。 λ -opt 近傍は、解をその巡回路を構成する辺の集合としてとらえ、辺に着目した近傍で、現在の解から辺を高々 λ 本交換することによって得られる解集合である。

距離行列が表1で与えられる $n = 5$ の TSP の問題例に対して 2-opt 近傍を用いた局所探索法を実行してみよう。2-opt 近傍操作は巡回路の一部を反転させることに相当する。ここで、表1の距離行列は非対称であるため、少し注意が必要である。例えば、順列 $\sigma = (a, b, c, d, e)$ に対する 2-opt 近傍操作として辺 (b, c) と (d, e) を削除する場合、辺 (b, d) と (c, e) を追加して部分パス (c, d) を反転する方法と、辺 (d, b) と (e, c) を追加して部分パス (e, a, b) を反転する方法の2つがある。前者では結果として (a, e, c, d, b) が得られ、後者では (a, b, d, c, e) が得られることになる。本稿では、説明の都合上、2-opt 近傍を都市数が少ないほうの部分パスが反転した解のみに限定する。都市数1の部分パスを反転させても現在の解になるだけで新しい近傍解は得られないため、近傍操作によって反転される部分パスの都市数は2以上である。さらに今回の問題例は $n = 5$ であるから、結局、ここで用いる 2-opt 近傍操作は、都市数が2の部分パスを反転する操作、つまり現在の巡回路で連続する2都市の位置を交換する操作ということになる。本稿では、巡回路の i 番目の都市と $i + 1$ 番目 ($i = n$ である場合は、1番目)の都市を交換する操作を f_i とおく。表2に、初期解 $\sigma = (a, b, c, d, e)$ に対して局所探索法を実行したときの様子を示す。表の各行は、探索で移動した解に対応しており探索で訪れた順番に並んでいる。表中の値は、現在の解に対して各近傍操作を適用することによって得られる解の評価値を表し、“*”は次に移動する解を示している。1番目の解である初期解 (a, b, c, d, e) を見ると、1番目と2番目の都市を交換する操作が選択されている。その結果、2番目の解 (b, a, c, d, e) が得られていることがわかる。3番目の解 (b, a, c, e, d) に対しては近傍内に改善解がなく、局所探索法はここで終了する。

表 2 局所探索法の実行例

現在の解	評価値	f_1	f_2	f_3	f_4	f_5
(a, b, c, d, e)	29	*17	18	22	24	24
(b, a, c, d, e)	17	29	31	22	*14	22
(b, a, c, e, d)	14	24	19	17	17	21

2.2 タブー探索法の基本構成

タブー探索法は、現在の解 x から、近傍 $N(x)$ に含まれるがタブーリスト T には含まれない解の中で最良の解 x' に移動する、という戦略を基本とする。タブーリスト T は、初期状態では空で、解が移動する度に探索の状況に応じて更新される。タブー探索法は、終了条件（例えば、計算時間が上限に達するなど）を満たしたとき、それまでの探索で得られている最良の解である暫定解 (incumbent solution) を出力して終了する。タブー探索法の基本的な動作を以下に示す。

タブー探索法

ステップ 1 初期解 x を生成する。タブーリスト T を初期化する。

ステップ 2 $N(x) \setminus (\{x\} \cup T)$ の中で最良の解 x' を見つけ、 $x := x'$ とする。

ステップ 3 終了条件が満たされれば暫定解を出力して探索を終了する。そうでなければ、タブーリスト T を更新した後ステップ 2 に戻る。

2.1 項で示した TSP の例と同じ設定で、タブーリスト T を探索中に訪問したすべての解の集合とした場合を見てみよう。タブー探索法の実行の途中までの様子を表 3 に示す。表中に記されている“†”はその近傍解がタブーリスト T に入っていることを示す。局所探索法の場合は、局所最適解 (b, a, c, e, d) にたどり着いて終了していた。タブー探索法では、(b, a, c, e, d) までは同じ動作となるが、そこからさらに探索が進む。解が (b, a, c, e, d) へ移動した時点を見てみよう。タブーリスト T は

$T = \{(a, b, c, d, e), (b, a, c, d, e), (b, a, c, e, d)\}$ となっている。近傍操作 f_4 に対応する解 (b, a, c, d, e) は T にあるため、表中では“†”が記されている。この反復では、タブーリストに入っていない近傍解の中で最良の評価値 17 を実現する f_3 に対応する解が選ばれる。なお、5 番目の評価値が 5 の解はこの問題例の最適解である。

タブーリスト T の実現方法としては、 t をパラメータとして、探索中に移動した解の軌跡の直近の t 個の解

表 3 タブー探索法の実行例

現在の解	評価値	f_1	f_2	f_3	f_4	f_5
(a, b, c, d, e)	29	*17	18	22	24	24
(b, a, c, d, e)	17	†29	31	22	*14	22
(b, a, c, e, d)	14	24	19	*17	†17	21
(b, a, e, c, d)	17	23	22	†14	*5	16
(b, a, e, d, c)	5	*14	22	21	†17	19

x_1, x_2, \dots, x_t を直接 $T = \{x_1, x_2, \dots, x_t\}$ と記憶するのが最も単純な方法である。生成した近傍解が T に含まれるかどうかを確認をする必要があるが、タブーリストの各解のハッシュ値を計算しておけば高速に実現することができる。この方法では、直近の t 個の解には移動しないことが保証できるので、長さ $t+1$ のサイクリングを防止することができる。しかし、 T として解集合を直接記録する方法は効果的な探索を行ううえであまり有効でないことが多い。一般には、2.3 項で解説する手法が最もよく用いられている。

2.3 短期メモリ

これまでの探索で解の移動のために実行した近傍操作の特徴（例えば、近傍操作の前後で変化する変数）を記憶しておき、記憶した特徴をもつ近傍操作を禁止することで、タブーリスト T の解集合を間接的に表現することができる。タブー探索法では、このように近傍操作の特徴を用いてタブーリストを定める方法が最もよく用いられる。なお、この方法において利用される近傍操作の特徴は属性 (attribute) と呼ばれ、記憶している属性集合が禁止解の集合 T と同じ役割を果たすことから、属性集合自体をタブーリストと呼ぶことが多い。近傍操作に対して、何を属性とすればよいかは一概には言えないが、大規模な問題例に対しては、多様化の傾向を高めるために、より多くの近傍操作を禁止する属性が有効となることがある。また、タブーとして一つの属性を禁止するのではなく、属性の組合せをタブーとする場合もある。

TSP の例では、例えば、近傍操作によって削除された辺を「禁止辺」として記憶しておき、その後の探索で、禁止辺を解に追加するような近傍操作を禁止する方法が考えられる。つまり、探索済みの解自体への移動を禁止するのではなく、過去に実行した解の変更を元に戻すような近傍操作全般を禁止するのである。近傍解が禁止辺を 1 辺でも含むとその操作を禁止するというのも一つの方法だが、近傍操作において追加する 2 辺がともに禁止辺の場合のみ禁止するという方法も考えられる。

表 4 属性を用いたタブー探索法の実行例

現在の解	評価値	f_1	f_2	f_3	f_4	f_5
(a, b, c, d, e)	29	*17	18	22	24	24
(b, a, c, d, e)	17	†29	31	22	*14	22
(b, a, c, e, d)	14	24	19	*17	†17	21
(b, a, e, c, d)	17	23	22	†14	*5	16
(b, a, e, d, c)	5	*14	22	21	†17	19
(a, b, e, d, c)	14	†5	15	22	23	*12
(c, b, e, d, a)	12	24	21	*18	†22	†14
(c, b, d, e, a)	18	†29	†22	†12	†16	†22

前述の問題例に対して属性に基づくタブーリストを用いたタブー探索法を実行した結果を表 4 に示す。ここでは新たに追加される 2 辺がともに禁止辺となる近傍操作を禁止している。例えば、7 番目の解 (c, b, e, d, a) において近傍操作 f_4 による評価値 22 の解が禁止されている。これは、近傍操作 f_4 によって得られる解 (c, b, e, a, d) が禁止辺 (d, c) と (e, a) を使うからである。8 番目の解 (c, b, d, e, a) に対する近傍操作はすべてタブーリストで禁止されているため、探索はここで終了する。

2.4 タブー期間

解の移動を行うたびにその属性をタブーリストに追加しつづけると、いずれ近傍内の解がすべて禁止解となり、探索を継続することができなくなってしまう。そこで、ある属性をタブーリストに追加したとき、解の移動を一定回数行った後に、その属性をタブーリストから取り除くことにする。属性をタブーリストに保持しておく期間はタブー期間 (tabu tenure) と呼ばれる。

タブー期間の設定には自由度があり、属性ごとに異なるタブー期間を設定したり、探索の段階によってタブー期間を変化させたりすることが考えられる。一般に、制限の強い禁止ルールは短いタブー期間に設定するとよい。通常、タブー期間が短すぎる場合はサイクリングが発生し、長過ぎる場合は解の精度が悪化するため、効果的なタブー期間のおおよその区間は、簡単な予備実験により比較的容易に推定することができる。タブー期間の決め方の一つとしては、予備実験により推定された区間から最も良いと思われる値の一つ選ぶ方法がある。また、もう一つの方法として、属性をタブーリストに追加する際にこの推定された区間から毎回異なる値をランダムに一つ選ぶという方法もある。後者の方法のように動的なタブー期間を用いると問題の構造に捕われないロバスタな探索が行われる傾向がある。

2.5 特別選択基準

属性を用いてタブーリストを定義する場合、よい解が近傍内にありながら、それを禁止解として探索対象から除外してしまう可能性がある。そこで、ある近傍解 x' が禁止解であっても、 x' が十分によい解であり、 x' への移動によってサイクリングが発生しないと考えられる場合には、例外的に x' への移動を許可する場合が多い。禁止解への移動を許可するかどうかの判断基準は特別選択基準 (aspiration criterion) と呼ばれ、「過去の探索で得られた最良解よりもよい解が得られる」などが代表的である。

3. 高性能なタブー探索法の開発に向けて

この節では、高性能なタブー探索法の開発に役立ついくつかの重要な概念を紹介する。

3.1 集中化と多様化

一般に探索型のアルゴリズムにおいては、似通った構造をもつ解を集中的に探索すること (探索の集中化) と、ときどきは、一時的に解の構造を大きく崩して、未探索の領域に探索を移すこと (探索の多様化) の相反する 2 つの動作をバランスよく行うことが重要である。しかし、タブー探索法では、禁止解以外の最良解へ移動することが基本戦略であるため、大幅な改悪を伴う解の移動があまり行われず、集中化のみが強調される傾向にある。探索済みの解を直接タブーリストとする方法は、この傾向が強くなりやすい戦略といえる。一方、属性に基づく方法は、解の移動を強く制限することも可能であり、結果として多様化の実現にも利用できる。

3.2 候補リスト戦略

近傍探索において、現在の解を改善する可能性がない近傍操作を探索の候補から除外したり、改善の見込みのありそうなものに限定したりすることで、探索の効率を高めることができる。これは、局所探索法の高速度化に欠かせないテクニックであるが、タブー探索法においては単なる高速化以上の意味をもつ重要な考え方であり、候補リスト戦略 (candidate list strategy) などとも呼ばれる。タブー探索法では集中化が強調されやすく、この傾向は属性やタブー期間の調整によってある程度解消できるが、候補リスト戦略を積極的に取り入れることも有用である。

3.2.1 限定選択戦略

問題構造に依存しない汎用的な候補リスト戦略である限定選択戦略 (aspiration plus strategy) を紹介す

る。まず、評価値についての閾値を設定しておき、それよりもよい評価値をもつ近傍解の発見を目標とする。(現在の解の評価値を閾値に設定すれば改善解を見つめることが目標となるが、この閾値を探索履歴に基づいて適応的に設定することも可能である)。そして、近傍解をランダムな順序で調べていき、閾値を満たす解が見つかった時点から、ある一定個数の近傍解を探索し、その中の最良の解に移動するのである。ただし、調べる近傍解の個数が多すぎる、もしくは少なすぎるのではないように、あらかじめ個数に関する上下限を設定しておく。そして、上の方法で探索した結果、調べた解の個数が下限に満たない場合には下限に達するまで探索を続け、逆に、探索の途中で上限に達した場合にはその時点で探索を打ち切ることとする。

3.2.2 エリート候補戦略

エリート候補戦略 (elite candidate list) では、まず最初に現在の解に対してすべての (もしくは非常に多くの) 近傍操作を評価し、その上位 k (k はパラメータ) 個を記録したマスターリスト (master list) を作成する。その後の探索では、マスターリストの中で最良の近傍操作に対応する解へ移動するという操作を反復する。マスターリストは、解の評価値が閾値以下になるか、解が一定回数移動すると再構成される。この手法の発想は、現在の解に対してよい近傍操作は、解が少し変形したとしても、依然としてよい近傍操作であるという考えである。解が移動していくと、マスターリストに記録されている近傍操作の評価値と現在の解にその操作を適用して得られる解の評価値に差異が生じる。そのためマスターリストに記録されている近傍操作の評価を定期的に更新するなどの方法がとられる。また、エリート候補戦略の拡張として、解が移動する際に、いくつかの近傍操作を新たにマスターリストに追加するなどの方法も考えられる。

3.3 頻度メモリ

長期メモリを活用する代表的なものとして、頻度メモリ (frequency-based approach) を紹介する。これは、ある変数が解の移動によって変更された頻度や、ある変数が特定の値をとっていた頻度を探索の全期間にわたって記録しておき、これを探索の集中化や多様化の制御に活用するものである。例えば、ある一部の変数のみが高い頻度で変更されている場合には、探索が一部の領域に偏っていると判断できる。変更頻度の高い変数の値を変更しにくくすることで探索の多様化を促すことが有効であり、これは、近傍探索において解の評価値を計算する際、頻度に応じたペナルティを加

えることで実現できる。

3.4 戦略的振動

制約のある問題では、実行可能領域と実行不可能領域の境界付近により解が存在するケースが多い。戦略的振動 (strategic oscillation) は、そのような境界周辺の集中的探索を実現するため、実行不可能領域も探索の対象に含め、境界の内側と外側を交互に行き来するようにタブー探索法における移動を全体的な視点から制御する方法である。実行不可能領域も探索の対象とすることに伴い、実行不可能解の評価方法を定める必要がある。代表的な方法としては、実行不可能解がどの程度実行可能な状態から離れているかを表す制約違反度を定義し (例えば、制約が不等式で $\sum_{i=1}^n a_i x_i \leq b$ のように表現されているならば、制約違反度を $\max\{\sum_{i=1}^n a_i x_i - b, 0\}$ のように定義するのが自然である)、この制約違反度にパラメータ α をかけて (元の問題が最小化問題の場合) 目的関数に足し合わせたものを評価関数とすることが考えられる。このとき、探索している解が実行可能解ばかりであれば α を小さくし、実行不可能解ばかりであれば α を大きくすることで、実行可能領域と実行不可能領域の境界付近の探索を重点的に行うことができる。

4. まとめ

タブー探索法は代表的なメタヒューリスティクスの一つである。本稿では、タブー探索法の開発において最低限必要な基礎的な事柄について簡単な例を交えながら解説した。そして、高性能なタブー探索法の設計において重要となるいくつかの概念を紹介した。

参考文献

- [1] F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research*, **13**, 533–549, 1986.
- [2] F. Glover, Tabu Search—Part I, *ORSA Journal on Computing*, **1**, 190–206, 1989.
- [3] F. Glover, Tabu Search—Part II, *ORSA Journal on Computing*, **2**, 4–32, 1990.
- [4] F. Glover and M. Laguna, Tabu Search, Kluwer Academic Publishers, Boston, 1997.
- [5] F. Glover, E. Taillard, and D. de Werra, A user's guide to tabu search, *Annals of Operations Research*, **41**, 3–28, 1993.
- [6] 久保幹雄, J. P. ペドロソ, メタヒューリスティクスの数理, 共立出版, 2009.
- [7] 野々部宏司, 柳浦睦憲, 局所探索法とその拡張—タブー探索法を中心として—, 計測と制御, **47**, 493–499, 2008.
- [8] 梅谷俊治, 柳浦睦憲, メタヒューリスティクス事始め—まずは局所探索法から—, オペレーションズ・リサーチ, **58**, 689–694, 2013.
- [9] 柳浦睦憲, 茨木俊秀, 組合せ最適化—メタ戦略を中心として—, 朝倉書店, 2001.