局所探索法
—反復改善に基づく最適化の基本戦略—

柳浦　睦憲

1. はじめに

局所探索法というと地味な印象をもたれるかもしれない．しかし，最適化における手軽で強力なツールとして欠かせないものであり，意外と奥が深い．

簡単な例から始めよう．巡回セールスマン問題は，n個の街とそれらの間の距離が与えられたとき，すべての街をちょうど1回ずつ訪れて出発地に戻る巡回路のうち，総距離最小のものを求める問題である．街を点，距離をそれらの間のユークリッド距離，巡回路において隣り合う街の対を線で結ぶとき，図1の左の巡回路をその右のものへと少し修正すると，巡回路の長さが短くなる．適当な巡回路から始め，このような小さな修正を可能な限り繰り返せば，(最短とはいわないまでも) いい巡回路にたどり着くと予想できる．このような手法を一般に**局所探索法** (local search)，あるいは**反復改善法**，**山登り法**などと呼ぶ．

最適化問題を一般的に定義しておこう．

$$\text{最小化} \quad f(x)$$
$$\text{制約条件} \quad x \in F. \tag{1}$$

$f: F \to R$ (Rは実数の集合) を**目的関数**，Fを**実行可能領域**と呼ぶ．Fは制約条件を満たす解の全体を表し，個々の解$x \in F$を**実行可能解**と呼ぶ．$f(x)$を最小にする実行可能解を**最適解**と呼び，その1つを見つけることが最適化問題の目標である．Fが組合せ的な構造をもつ場合，(1)は**組合せ最適化問題**と呼ばれる．

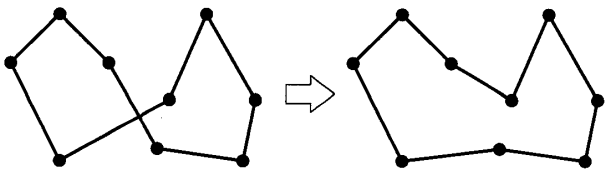


図1　近傍内の解への移動の例

やぎうら　むつのり
名古屋大学 大学院情報科学研究科
〒464-8063 名古屋市千種区不老町

NP困難性に代表されるように，多くの組合せ最適化問題に対して厳密な最適解を得ることは困難であることが知られているが，そのような問題に対して近似的によい解を短時間で得るための手軽な方法として，局所探索法は広く利用されている．

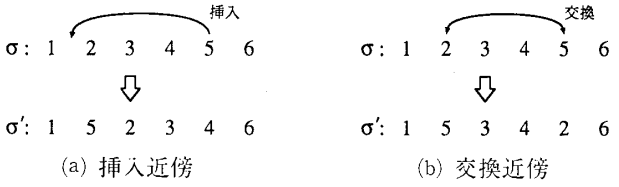
2. 局所探索法の基本要素

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，可能な限り反復する方法である．近傍内によりよい解が存在しない解を**局所最適解** (locally optimal solution) という．最適解を局所最適解と特に区別する場合には**大域最適解** (globally optimal solution) という．なお，実行可能領域を探索の対象とし (すなわち初期解$x_{\text{init}} \in F$かつ$N(x) \subseteq F (\forall x \in F)$)，目的関数値によって解を評価する方法が基本的であるが，そうでない方法が有効な場合もある．この点については節2.3で議論する．

2.1 近傍

近傍は局所探索法の設計において最も重要なポイントのひとつであり，近傍内に改善解が含まれる可能性が高まるように設計することが望ましい．代表的な例をいくつか挙げておこう．

巡回セールスマン問題は街を訪れる順序によって解を表すことができるが，このように順列σで解を表せる問題に対しては，**挿入近傍**や**交換近傍**がよく用いられる (図2)．挿入近傍は1つの街を順列の他の位置に移動することで得られる解集合，交換近傍は2つの街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．巡回路において隣り合う街の対を枝と呼



(a) 挿入近傍　　(b) 交換近傍

図2　挿入近傍と交換近傍の近傍操作の例

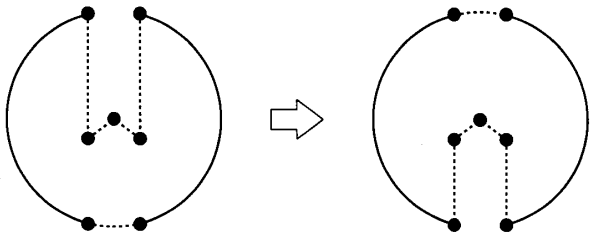


図3　Or-opt 近傍の近傍操作の例

び，枝を高々 $\lambda(\geq 2)$ 本交換することによって得られる解集合を**λ-opt 近傍**と呼ぶ．通常 $\lambda=2$ か 3 が用いられる．図 1 は 2-opt 近傍の近傍操作の例である．現在の巡回路において連続する 3 つ以下の街を他の位置に挿入することによって得られる解集合を **Or-opt 近傍**と呼ぶ．図 3 に近傍操作の例を示す．図中，点線は枝，実線は枝を複数含むパスを表す．これは挿入近傍の自然な拡張であり，また，3-opt 近傍の特別な場合になっている．交換近傍は 4-opt 近傍の特別な場合である．どの近傍が効果的であるかは問題によるが，巡回セールスマン問題に対しては，λ-opt 近傍 $(\lambda\leq 3)$ が効果的で，交換近傍はあまり有効ではない．その直感的理由は，巡回路長が枝集合によって定まるが，交換する枝数が 4 と他の有効な近傍操作よりも多く，目的関数への影響が大きいことが原因と考えられる．

解を n 次元 0-1 ベクトルで表現できる問題も多い．その場合，高々 r 箇所の 0-1 を反転することにより得られる（つまりハミング距離 r 以下の）解を近傍とする方法がしばしば採用される．これを **r 反転近傍**と呼ぶ．r は 1 か 2 程度が一般的である．この他，問題タイプに応じて様々な近傍が設計可能である．

近傍内に改善解があればその 1 つを発見し，そうでない場合には局所最適であることを結論する問題を，**改善解探索問題**と呼ぶことにする．局所探索は何度も反復してこの問題を解くので，これが高速に解けるように近傍を設計する必要がある．また，広く用いられている近傍であっても，データ構造を工夫することで改善解探索問題をより高速に解ける場合があり，局所探索の効率化に有効である．

2.2　移動戦略

近傍内の改善解は通常複数存在する．よって，近傍をどのような順序で探索し，どの改善解に移動するかによって，局所探索の動作は異なる．これを定めるルールを**移動戦略**（move strategy）という．近傍内をランダムな順序で調べていき，最初に見つかった改善解に移動する**即時移動戦略**と，近傍内の最良の解に移動する**最良移動戦略**の二つが代表的である．これらを比べると，多くの場合即時移動戦略のほうが高速であり，最終的に得られる局所最適解の精度には大きな差がない傾向にある．一方，近傍内の解の評価値を表に記憶しておき，解の移動の際にその更新を高速に行うというようなデータ構造の工夫が可能な場合，最良移動戦略のほうが有利な場合もある．また，これらの中間的な戦略もある．その一例を，挿入近傍を用いる場合について示そう．街をランダムな順序で調べていくが，各街については，挿入位置をすべて調べて最良の位置を求め，それが改善解ならば即座に移動する，というものである．これを**部分最良戦略**と呼ぶことにする．即時移動戦略よりも多少時間はかかるが，解の精度が若干改善される場合がある．

2.3　探索空間と解の評価

探索の対象となる解全体の集合を**探索空間**（search space）という．実行可能解の 1 つを見つけることと，近傍操作によって新たな実行可能解を生成することがともに容易な場合は，実行可能領域をそのまま探索空間としても困ることはない．しかし，問題によっては実行可能領域とは異なる探索空間を定義するほうが有効な場合がある．以下，そのような例を紹介しよう．

ひとつめの例は充足可能性問題である．n 個の論理変数 $x_1, \ldots, x_n$ とその否定 $\bar{x}_1, \ldots, \bar{x}_n$ $(\bar{x}_j=1-x_j)$ のそれぞれをリテラルと呼び，リテラルをいくつか選んで論理和を取ったものを節という（例えば $C_i=x_1\vee\bar{x}_3\vee x_4$）．節はその中の 1 つ以上のリテラルが値 1 を取るとき充足されるという．例えば上に例示した節 C_i は，$x_1=1, x_3=0$ あるいは $x_4=1$ のとき充足される．充足可能性問題は，m 個の節 $C_1, \ldots, C_m$ が与えられたとき，そのすべてを充足するような変数への 0-1 割当の存在を問う決定問題である．

すべての節を充足することが制約であるので，実行可能領域 F はそのような 0-1 割当全体の集合である（この場合目的関数は定数と考える）．よって，この問題は，F が空であるかどうかを判定する問題といえる．このような場合に探索空間を F とすることは現

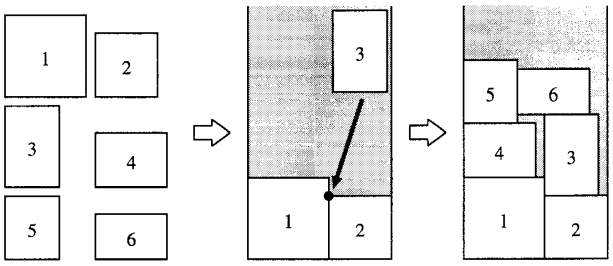
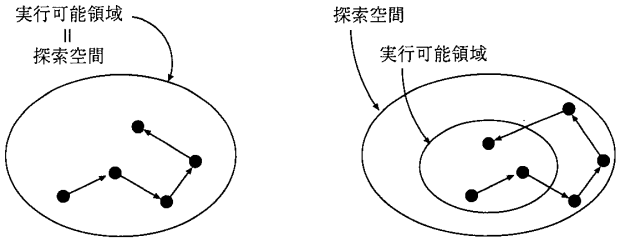
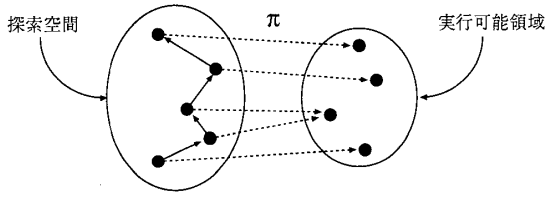


図4　BL法による配置例



(a) 実行可能領域のみの探索　　(b) 実行不可能領域も含めて探索



(c) 解空間とは異なる探索空間を用意

図5　様々な探索空間

実的でなく，以下のような方法が有効である：n次元0-1割当全体の集合を探索空間とし，充足されていない節の数を解の評価値としてその最小化を図る．このように，制約違反をペナルティとして，その総和を最小化するという方針は，様々な問題に対して有効である．

実行可能解の発見が容易でない上に，目的関数の最小化が求められる問題も多い．このような場合は，ペナルティに適当な重みをかけて目的関数に加え，解の評価関数とする方法がしばしば用いられる．ペナルティ重みを大きくすれば実行可能解が得られる可能性は上がるが，有効な探索を行うためにはペナルティは大き過ぎないほうがよい傾向にあり，この調整は難しい．そのため，制約の満たし難さや目的関数値とのバランスに応じて，ペナルティ重みを探索の進行とともに適応的に調整する方法がしばしば効果を発揮する．

もうひとつ，パッキング問題の例を紹介しよう．n個の矩形の幅と高さ，および容器の幅が与えられたとき，矩形同士を重複なく容器に配置し，高さ（矩形の上辺の位置の最大値）の最小化を図る問題である．ここでは矩形の回転を許さない場合を考える．

各矩形の配置位置には無限の可能性があり，また，それらを独立に定めたのでは重複の排除が困難であることから，配置座標の組合せを直接探索の対象とすることは現実的でない．そこで，実行可能な配置を表現する様々な手法が提案されている．そのひとつに，順列を解表現とし，bottom-left法（以下BL法）と呼ばれるアルゴリズムによって順列に対応する配置を計算する方法がある．BL法は，矩形を1つずつ配置していく方法で，各反復では，次に置く矩形の位置を，配置済みの矩形に重複なく置ける最も低い位置の中の最も左に定める．図4に順列$(1, 2, ..., 6)$に対するBL法の動作例を示す．

順列が変わると対応する配置も変わるので，順列を探索の対象とし，対応する配置の目的関数値を解の評価として局所探索を行うのである[1]．探索空間はn要素の順列全体の集合であり，問題の実行可能領域とは全く別のものである．このような方法は，問題の決定変数を直接探索することが難しい問題に対してしばしば有効である．

探索空間の例をいくつか挙げたが，これらを大別すると図5の3通りが考えられる．巡回セールスマン問題の例は(a)，充足可能性問題の例は(b)，パッキング問題の例は(c)に対応（BL法が写像πを定める）する．

3.　探索空間の視覚化

探索空間に含まれるすべての解を節点集合とし，近傍Nに対して辺集合Eを$x' \in N(x) \Leftrightarrow (x, x') \in E$で定めた有向グラフを**近傍グラフ**と呼ぶ．図6に3次元0-1ベクトルの全体を探索空間として1反転近傍を用いた場合の近傍グラフを示す．$x' \in N(x) \Leftrightarrow x \in N(x')$を満たすとき近傍$N$は対称であるというが，近傍が対称なときは2本の有向辺(x, x')と(x', x)を1本の無向辺$\{x, x'\}$で表し，無向グラフで表すこともある．

近傍を設計する上で，近傍グラフをイメージすることは有用である．例えば，近傍グラフが連結である（任意の解から任意の解へ近傍操作を繰り返すことで到達できる）ように近傍を設計することが望ましい．

[1] この方法では最適配置に対応する順列が存在しない場合があるが，最適配置に対応する解表現の存在を保証する表現法もある．

図7に，充足可能性問題の $n=5$ の問題例に対する近傍グラフ[2]を示す．探索空間は任意の0-1割当とし，充足されていない節の数を評価値として，評価値の良いものほど下に配置している．二重丸は局所最適であることを表し，一番下の節点21は大域最適解である．

節点数は $2^n=32$ と小規模であるが，探索空間が結構複雑である様子が窺える．また，局所最適解の多くが大域最適解に近い評価値をもつことが観測できる．充足可能性問題では大域最適解以外は意味がないが，充足される節の数の最大化を扱う場合には（近似解が許容されるなら）これらの局所最適解は有用である．一方評価値がかなり悪い局所最適解もある（節点1）．局所探索法による解の精度に対する理論的解析には否定的な結論（例えば非常に精度の悪い局所最適解が存在するなど）もあるが，最悪ケースの解析ではこのような少数の解が足を引っ張る傾向にあるため，現実の性能とはあまり相関がないと考えたほうがよい．

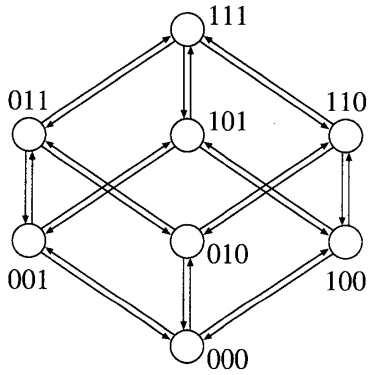


図6　近傍グラフの例

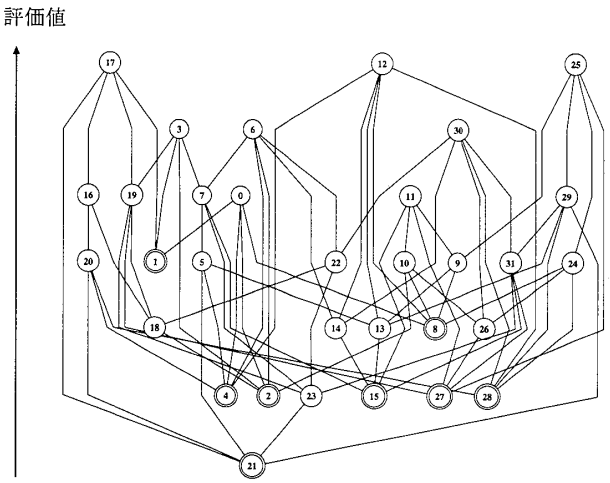


図7　充足可能性問題に対する近傍グラフ

[2] 番号は解に機械的につけたものである．作図の都合上，局所最適性に影響しない一部の辺（水平方向の辺など）を省略した．

4．反復局所探索法

近傍グラフで観測したように，局所最適解の中には精度の低いものも存在するため，局所探索法を1度適用しただけでは，そのような解を出力して終了してしまう心配が残る．また，多くの問題において，局所最適解の周囲によりよい解が潜んでいる可能性が高いことが経験的に知られている．そこで，局所探索を復数回行う，改悪解への移動を許すことで局所最適解でも探索が停止しないようにするなど，局所探索法を発展させるための種々のアイデアが提案されており，総称して**メタ戦略**（metaheuristics）と呼んでいる（そのような拡張も含めて局所探索法と呼ぶ場合もある）．

メタ戦略の様々なアイデアを紹介するのは，紙面の都合上困難であるが，単純でしかも比較的高い性能が期待できる方法の一つとして，**反復局所探索法**（iterated local search）を紹介しておく．これは，過去の探索で得られた最良解 x_{best} にランダムな変形を加えたものを初期解として局所探索を行う操作を反復する方法である．

初期解生成に用いるランダムな変形としては，適当な近傍の中からランダムに解を選ぶ方法が自然であるが，このための近傍に局所探索と同じ近傍を用いると，局所探索の1回の移動によってすぐに x_{best} に戻ってしまう可能性がある．これを避けるため，局所探索に利用する近傍より多少大きな近傍や，少し構造の異なる近傍を利用することが推奨される．例えば，巡回セールスマン問題で局所探索法に2-opt近傍を用いたときには，図8に示すようなdouble bridge近傍をランダムな変形に利用するのが効果的であるといわれている．交換する枝は4本であるが，2-opt近傍の操作を2回繰り返しても到達できない解が生成されるため，x_{best} にすぐに逆戻りする現象を防ぐ効果がある．

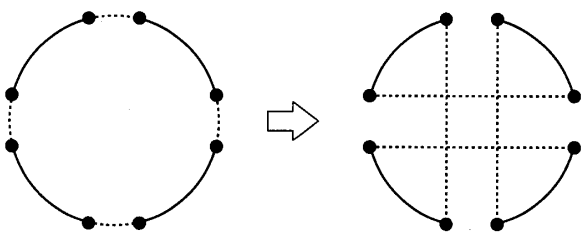


図8　double bridge近傍の近傍操作の例

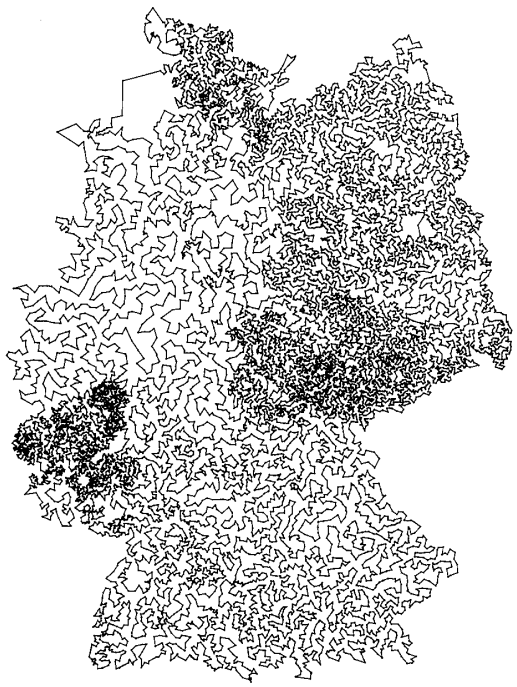


図9　15,112都市を回る最適巡回路（距離1,573,084）

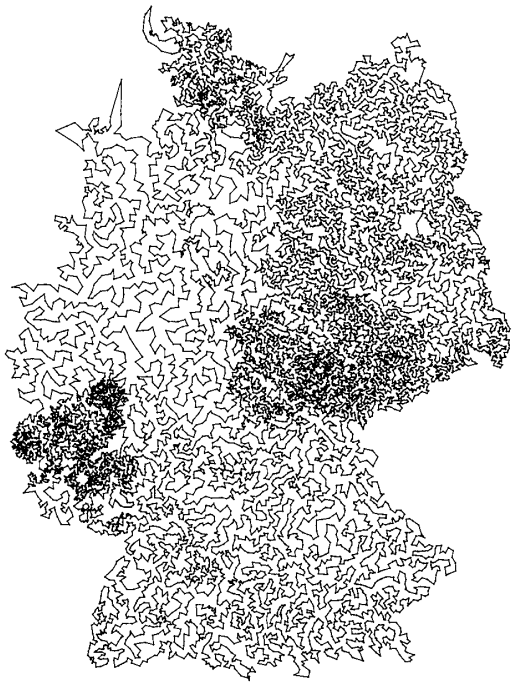


図10　反復局所探索法による巡回路（距離1,576,015）

5. トピックス

局所探索法の成功例として，巡回セールスマン問題への適用例を紹介しておく．ドイツの15,112都市を回る問題例（d15112）に対する最適解を図9に，反復局所探索法による解を図10に示す．最適解は，分枝カット法を用いて110個のCPUを用いた分散計算によって長時間（Compaq EV 6 Alpha（500 MHz）1台の逐次計算に換算すると22.6年）かけて得られた．一方，反復局所探索法は，Lin-Kernighan法と呼ばれる精巧な近傍をベースにして効率的なデータ構造を利用して実装されたもので，図10の解はPentium M（1.6 GHz）のノートPCで約38秒で得られた．最適解の距離は1,573,084，反復局所探索法による解の距離は1,576,015であり，その差は0.2%弱と極めて小さい．局所探索が大きな成功を収めた一例である．なお，2つの巡回路を見比べると，部分的に似通ったところが極めて多いことが観測でき，興味深い．

本節の話題に関しては，[4]に詳しい情報がある．また，本節の実験に用いた反復局所探索法は，同サイトにて公開されているConcordeソルバの一部である．

6. おわりに

局所探索法は，一見極めて単純であるが，近傍，移動戦略，探索空間，解の評価法と，設計における自由度が大きく，様々な工夫を行うことが可能な奥の深い枠組みであることを述べてきた．理論的には，最適性あるいは解の精度保証，多項式時間停止性などの良い性質を保証できないことが多く，その良さを証明しにくい手法である．しかし，それでも実用上は成功例が多く，非常に有用である．本稿が局所探索法に対する理解を深める一助となれば幸いである．局所探索法やその拡張について，より詳しくは文献[1]～[3]などを参照いただきたい．

謝辞　本稿の執筆にあたって貴重なコメントをいただいた今堀慎治，梅谷俊治，橋本英樹の諸氏に感謝します．

参考文献

[1] E. H. L. Aarts and J. K. Lenstra, eds.: *Local Search in Combinatorial Optimization*, John Wiley & Sons, Chichester, 1997.

[2] H. H. Hoos and T. Stützle: *Stochastic Local Search—Foundations and Applications*, Morgan Kaufmann/Elsevier, San Francisco, 2005.

[3] 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.

[4] http://www.tsp.gatech.edu/