9　組み合わせ最適化問題に対する近似解法

組み合わせ最適化の問題において，これまで述べてきた手法が適用できないものが数多く存在する．また，大域的な最適解ではなく，近似解でも実用上，十分なことも多い．本講義ではこうした近似解を得るための代表的な手法について述べる．

9.1　代表的な近似解法

9.1.1　欲張り法とけちけち法

欲張り法 (greedy method) とは，目的関数への貢献度を示す局所的な評価に基づいて可能解を順次，直接構成していく方法である．通常，構成の過程には試行錯誤を含まないため，近似解はすぐに得ることができる．

p.46 の演習問題 7-1 にあげた 0-1 ナップザック問題の例題では，目的関数への貢献度を単位重量あたりの価格とおくことができる．すると，欲張り法は，重量単価の高い品物から順次，ナップザックに制約を満足する間，なるべく多く入れていくという解法となる．

品物番号	1	2	3	4
価格 (万円)	32	46	56	38
重量 (kg)	4	8	10	6
重量単価 (万円/kg)	8.00	5.75	5.60	6.33

ここでは，重量単価の高い順に商品を 1 $\rightarrow$ 4 と入れ，３番目の商品 2 を入れると総重量の制約 14kg を満足できないため，ナップザックには入れない．次に重量単価の高い（といっても最低の）商品 3 についても総重量を満足しないので，ナップザックには入らない．すべての商品についてチェックが終了したので，解は商品 1 と 4 で商品価格総額 70 万円が欲張り法の解となる．欲張り法では，解の構成に商品の数 N に比例した手順しか必要とせず，組み合わせの爆発が生じない．しかしながら，重量単価の高い順に選択したために，総重量が 10kg の構成となり，制約を 4kg 余してしまう．この問題の最適解は商品 1 と 3 の選択による 88kg であることを考えると，最適解を与えない可能性が高いことが予想される．

欲張り法では空のナップザックに順に詰めていったが，一度すべての収容して，制約条件を満足するように，目的関数への貢献度の低いものから除外していくのが，**けちけち法** (stingy method) である．この例では，商品 1,2,3,4 すべてを考えると，総重量が 28kg となるため，目的関数への貢献度の低い商品 3 を除外する．それでも総重量が 18kg で 14kg 以内という制約を満足しないため，次に貢献度の低い商品 2 を除くと総重量は 10kg となり，制約を満足するため，そのときの商品 1 と 4 で商品総額 70 万円がけちけち法の解となる．

今回の例では欲張り法とけちけち法の出力する解は同じになったが，この２つの方法の解が一致するとは限らない．

9.1.2　ランダム法

ランダム法 (random method) は，解空間から解を複数個，ランダムに選択し，その中でもっともよい目的関数の値をもつものを近似最適解とするものであり，**モンテカルロ法** (Monte Carlo method) とも呼ばれる．0-1 ナップザック問題の例題では品物 i を選択するかしないかを乱数で選択して解を生成し，重量の制約を満足しているかどうかをテストすることに該当する．ランダム法においては，解候補の点数を増やすことでよりよい近似解を得られる可能性を高めることができるが，広い解空間を均等に

カバーしようとすると計算量も同時に増加してしまう．よりよい解の存在する周辺を重点的に探索するために，他の手法と組み合わされて使われるのが通常である．

9.1.3　近傍探索法

近傍探索法 (neighborhood search) とは，なんらかの方法で得られた可能解（妥当解）x の近傍 $N(x)$ の中で，目的関数の値を改善できるものがあれば，それに置き換える方法である．近傍探索法においては，可能解の集合に適当な距離 $d(x,\ x')$ を導入して，ある解 x からある距離 λ の中に入る解候補の集合 $N_\lambda(x)$ に関して，よりよい目的関数の値を持つものを探すことを行う．

0-1 ナップザック問題においては，商品の選択の有無を表す 0-1 列のハミング距離を可能解の間の距離として近傍を定義すると，1 近傍には，ナップザックから商品を 1 つ出したり，新しく 1 つを加えることで定義される解候補が含まれる．

9.1.4　緩和法

緩和法 (relaxation method) は，制約条件を緩和することで一つの解を簡単に構成し，それに修正を加えて可能解を得る方法である．組み合わせの爆発を押さえるように，制約条件を緩和する点は，先の講義で論じた分岐限界法の緩和問題と同一である．

前述した 0-1 ナップザック問題のけちけち法のやり方は，総重量の制約を緩和して初期解を構成し，そこから修正（商品の除外）を加えて制約を満足する解を構成する，という点で緩和法にも該当している．

9.1.5　分割法

分割法 (partitioning method) は対象問題の定義領域をいくつかの部分領域に分割し，それぞれの解を合成してもとの問題の可能解を作る方法である．

たとえば，12 の要素の順列全体を考えると解候補は $12! = 479,001,600$ だけ存在するが，12 の要素を前，中，後の 3 つに 4 要素ずつに分割できるとすると，それぞれの分割された問題は $4! = 24$ となり，各分割された問題に対する最適解を求めることは容易となる．こうして求めた各分割問題の最適解を合成して，全体の近似解を構成するのが分割法である．

最適化問題が分割に適した構造（解空間の性質）を持っている場合には，分割法によって最適解を得られる場合もあるが，分割が不適当な場合には，分割によって最適解から離れた近似解しか得られないこともある．

9.1.6　部分列挙法

部分列挙法 (partial enumeration method) は，すべてを列挙することが困難である場合，その一部分のみを組織的に列挙し，その結果に基づいて，近似最適解を構成する方法である．

たとえば，10 個から商品からなる 0-1 ナップザック問題を考える．このままでは解候補は $2^{10} = 1024$ 存在する．たとえば，ここで 10 個の商品の中から 2 個を選ぶすべての組み合わせを列挙する．この場合，${}_{10}C_2 = 10 \times 9/2 = 45$，存在する．この中で，総重量を満足するものを選んで，適当な近似解法（たとえば，欲張り法）を用いて解を求めてやる．その中で最良のものを近似解として選択する．

このように，組織的に部分列挙することで，近似解を求める点群を解空間の中に均等に分散させることができる．

9.2 メタヒューリスティック

上で述べた近似アルゴリズムを反復したり組み合わせることで，より効果の高い近似アルゴリズムを構成するための方法を**メタヒューリスティック** (meta heuristic) とよぶ．ここでは代表的なメタヒューリスティックを列挙する．

9.2.1 局所探索法

局所探索法 (local search) は**反復改善法** (iterative improvement method) とも呼ばれる．近傍探索を改善が得られなくなるまで反復して行うもの．良い初期解から出発するとよい局所最適解が得られる傾向がある．

9.2.2 ランダム多スタート局所探索法

ランダム多スタート局所探索法 (random multi-start local search)：ランダム法と局所探索法とを組み合わせたもので，ランダムに発生させた多数の初期解それぞれに対して局所探索法を行い，その中で最良のものを近似最適解とするもの．

9.2.3 アニーリング法

アニーリング法 (simulated annealing method)：模擬アニーリング法，シミュレーティッド・アニーリング法とも呼ばれる．最小化問題の局所探索において，現在の解 x に対して任意の近傍解 y を選び，$\delta = f(y) - f(x)$ としたときに，$\delta > 0$ であっても，$e^{-\delta/T}$ の確率で y を選択するという挙動を示す．T が小さくなると選択される確率が 0 に近づく．この温度 T を反復の繰り返しに従って小さな値（低温）にしていく．

9.2.4 タブー探索法

タブー探索法 (tabu search)：局所探索において，近傍 $N(x)$ において x 以外の最良の解 y を次の解候補とする方法．この場合，目的関数の値が，悪くなる場合もあるが，それでも選択を行う．この手法を使うと一度選んだ解に戻る巡回が発生するため，前に選んだ解 x をタブーリスト (tabu list) とよばれるリストに登録して再探索しないようにする．これによって，解探索の巡回を防止でき，効率的な解探索が可能となる．適応型記憶計画 (adaptive memory programming) とも呼ばれる．

9.2.5 遺伝アルゴリズム

遺伝アルゴリズム (genetic algorithm)：遺伝子の進化からアイデアを取り入れた確率的なアルゴリズム．解の交叉 (crossover)，突然変異 (mutation)，改良，淘汰などの操作を適用する．計算の基本手順は以下の通りである．

初期設定： 局所最適解を N 個用意し，それを解の初期集合とする．

進化： 解の集合に関して，以下の操作を行う．

- **交叉：** 2 個以上の局所最適解を選び，それらの交叉によって新しい解を作る．これを適当な回数実施する．

突然変異： 局所最適解のいくつかにランダムに変更を加えて新しい解を作る．

改良： 交叉や突然変異で得られた新しい解を改良して局所最適解を得る．

淘汰： 進化によって増加した解集合の中から N 個の解を残す．また，最良の局所最適解を暫定解とする．

こうした操作を，決められた進化の世代数や暫定解の変化を見て，適当なところで停止させ，近似解を得る．

遺伝アルゴリズムを行うに際して，解候補をどのように遺伝子操作を行う１次元のコードで記述するか，ということがポイントとなる．たとえば，巡回セールスマン問題のような順列の問題を例に挙げて考えてみる．今，６つの都市，A〜F があって，A から出発して A に戻るとすると，巡回順はたとえば，$BDEFC$ のような A を除く５つの都市の列として表現できる．しかしながら，ここで２つの解の交叉を行ったり，突然変異を単一の記号に対して行ってしまうと，巡回セールスマン問題の解として不適切なもの（２回，訪問する都市があったり，１度も訪問しない都市があったりする）が出来てしまう．そのため，問題ごとにコード化するかが，遺伝アルゴリズムを行う際の最初の重要なポイントとなる．

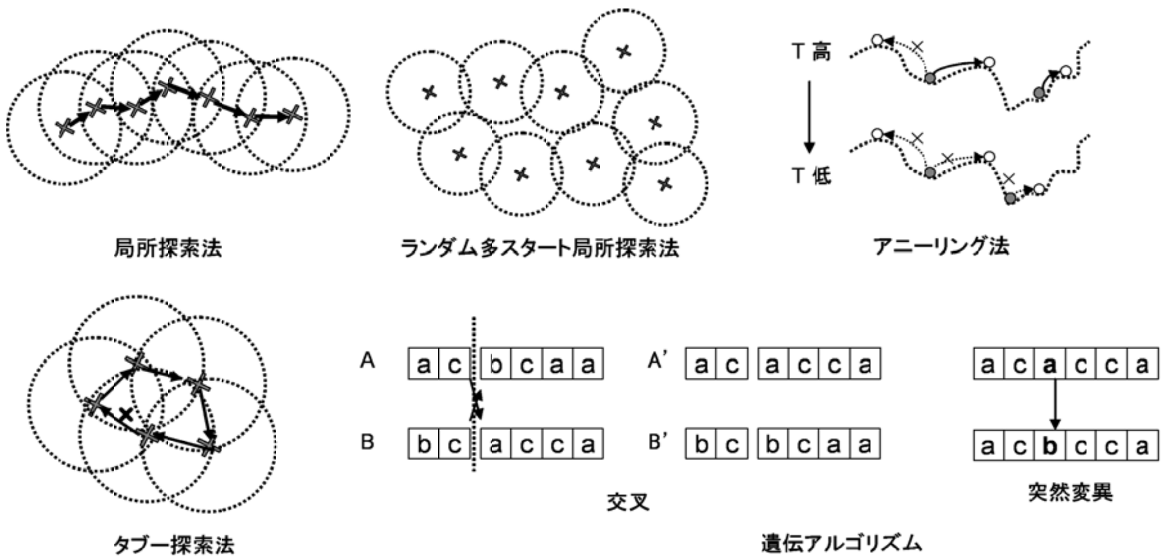

図 9.1: 組合せ最適化に関するメタヒューリスティック

9.3 離散最適化の近似解法のまとめ

こうしたさまざまな方法が提案されているが，対象とする問題の性質をよく検討し，それに適合した手法を選択して，手法に固有の各種パラメータの設定を適切に行うことが，最適化問題のよい近似最適解を得る上では不可欠である．すなわち，万能の最適化手法は現在のところ存在せず，問題の分析と各手法の適用範囲と注意事項をよく理解していることが利用者に求められている．

【参考文献】

- 茨木俊秀著：「離散最適化法とアルゴリズム」，岩波講座応用数学「方法８」，岩波書店
- 今野浩著：「数理決定法入門／キャンパスの OR」，朝倉書店