

# 多点探索型アルゴリズムの基礎と最前線

永田 裕一

NP 困難問題に代表されるような難しい組合せ最適化問題に対して、メタ戦略に基づく近似解法が広く利用されている。メタ戦略の枠組みには局所探索のように 1 つの探索点を動かしながら探索を行う手法（1 点探索）と遺伝的アルゴリズムに代表される複数の探索点を用いて探索を行う手法（多点探索）がある。近年、多点探索型のメタ戦略を用いて高性能な近似解法が多数考案されている。本稿では多点探索型メタ戦略の中でも遺伝的アルゴリズムとメメティックアルゴリズムに焦点を当て、これらの枠組みを用いて高性能アルゴリズムを構成するための基本的な考え方を具体例を交えて解説する。

キーワード：メタ戦略、多点探索、遺伝的アルゴリズム、メメティックアルゴリズム、パス再結合法

## 1. はじめに

一般的には最適化問題は以下のように定義される。

$$\text{最小化: } f(x)$$

$$\text{制約条件: } x \in F$$

$F$  は問題に課されている制約を満たす解集合で、 $x \in F$  は実行可能解（または解）と呼ばれる。 $f(x)$  は目的関数と呼ばれ、目的関数を最小にする解を**最適解**と呼ぶ。最適解を現実的な計算時間で求めることが困難である場合、近似解法が用いられる。近似解法ではなるべく少ない計算時間でなるべく目的関数値の小さな解（**近似解**）を求めることが目標となる。近似解法の枠組みとして探索的手法があるが、これは現在の探索点（解）を何らかのルールに従い解空間上を移動させることで、解空間の一部分を探索する方法である。探索的手法を用いて高性能な近似解法を構築するためのさまざまな枠組みが提案されているが、これらを総称して**メタ戦略** (metaheuristics) と呼ぶ。

メタ戦略の多くは 1 点の探索点を用いて探索を行う。これらの手法は基本的には局所探索をベースとして、より高性能な近似解法を構築するための工夫を導入している。代表的な手法としてはシミュレーテッド・アニーリング (simulated annealing)、タブー探索 (tabu search)、反復局所探索法 (iterated local search) などがあるが、これらの概要については本特集のそのほかの解説などを参照されたい。

メタ戦略の枠組みの中には複数の解候補を探索点として探索を進めていく手法もある。このような手法は

多点探索型のメタ戦略とも呼ばれ、代表的な手法としては**遺伝的アルゴリズム** (genetic algorithm, GA)、**メメティックアルゴリズム** (memetic algorithm, MA)、**パス再結合法** (path relinking, PR)、**アントコロニー法** (ant colony optimization, ACO)、**粒子群最適化** (particle swarm optimization, PSO)、**差分進化法** (differential evolution, DE) などが知られている。本稿では多点探索型のメタ戦略の中でも GA, MA, PR を取り上げ、基本的な枠組みの解説とこれらを用いて高性能アルゴリズムを開発するための考え方を具体例を交えて解説する。

## 2. 遺伝的アルゴリズム

遺伝的アルゴリズム (GA) は生物進化を模したアルゴリズムとして提唱された。D.E. Goldberg によって古典的な GA の枠組みが整備されて以来、多くの組合せ最適化問題に対して GA が適用されている。本節では GA の一般的な枠組みと基本的な設計指針を解説した後、巡回セールスマン問題 (traveling salesman problem, TSP) に対して提案された高性能 GA を紹介する。

TSP は最も有名な NP 困難問題の 1 つで、与えられたグラフ  $G = (V, E)$  上のすべての頂点をたどる閉路（巡回路）の中で長さが最小のものを見つける問題である。ここで、 $V = \{1, \dots, n\}$  は頂点集合、 $E = \{(i, j) \mid i, j \in V\}$  は枝（辺）集合で、各枝には長さ（重み）が定義されているものとする。

### 2.1 基本的枠組み

GA は**集団** (population) と呼ばれる複数の探索点を用いて探索を行う。また、個々の探索点は**個体** (individual) と呼ばれる。集団に**交叉**や**突然変異**などのオペレーターを適用することで新しい探索点を生成し、選

ながた ゆういち  
東京工業大学 情報生命博士教育院  
〒 226-8501 横浜市緑区長津田町 4259

択操作を通して次世代の集団を決定する。交叉は2つの解（親個体）を組合せて新たな解（子個体）を生成するオペレーターである。GAの一般的な枠組みを以下に示す。

#### GAの一般的枠組み：

1. 初期化：何らかの方法で  $N$  個体を生成して初期集団  $P(0)$  を生成する。 $N$  は集団サイズと呼ばれる。世代数を表すカウンターを  $t=0$  とする。各個体はランダムに生成するのが単純であるが、局所探索などを用いて生成することもある。
2. 複製選択：集団  $P(t)$  に選択操作を実行して親集団  $P'(t)$  を生成する。親集団の構成に規定はないが、 $N/2$  個の親ペアを選択するのが標準的である。
3. 交叉：親集団  $P'(t)$  に交叉を適用して子個体の集合  $P''(t)$  を生成する。交叉は事前に設定した交叉確率で適用する。1つの親ペアから生成する子個体数の数は任意に設定できる。
4. 突然変異： $P'(t)$  と（または） $P''(t)$  に突然変異を適用する。突然変異は事前に設定した突然変異確率で適用する。
5. 生存選択： $P'(t)$  と  $P''(t)$  から  $N$  個体を選択し、次世代の集団  $P(t+1)$  とする。
6. 終了判定：事前に設定した終了条件が満たされれば探索を終了、そうでなければ  $t=t+1$  としてステップ2へ。

対象とする最適化問題に対してGAを構成するためには、複製選択、生存選択、交叉、突然変異を具体的に設計する必要がある。古典的なGAでは、これらの設計は生物進化を模倣したものとして提案された。例えば、個体は0-1ビットストリング表現でコード化され、1点交叉、2点交叉や一様交叉などの交叉方法やビット反転のような突然変異が使用される。古典的なGAは（個体をうまく遺伝子にコード化する必要はあるが）汎用的である反面、問題の性質を考慮した局所探索ベースのメタ戦略を超える性能を実現することは困難であることが知られている。本稿では古典的なGAの説明は割愛する（解説は[14]などを参照されたい）。

優れた探索性能を持つGAを設計するためには生物進化の模倣にとらわれず、もっと柔軟な発想に基づいて各設計を行う必要がある。GAの多点探索という特徴を活かして効率的な探索を行うためには以下の2つの設計が特に重要である。

- (i) 集団の多様性を適切に維持するような複製/生存選択の設計。複製/生存選択を統合した枠組みを世代交代モデルと呼ぶ。

- (ii) 問題の性質を考慮した交叉および突然変異オペレーターの設計。

GAに限らずメタ戦略では「良い近似解同士には何らかの類似性が存在する」という仮定に基づいて探索の効率化を行う。この仮定は近接最適性の原理 (proximate optimality principle, POP) と呼ばれている。POPの下では、複製/生存選択において目的関数値の良い個体を優先的に選択する（選択圧をかける）ことで、集団は有望な（良い近似解が含まれると期待される）探索領域に重点的に分布することが期待される。これを探索の集中化と呼ぶ。しかし、集団サイズが有限である以上、探索の集中化のみを強調すると本来探索すべき領域を見逃してしまう危険が増加してしまう。このため、高性能なGAを設計するためには集団の多様性維持を考慮した世代交代モデルの設計が重要である。2.2項では比較的良好な実績を上げている世代交代モデルのアイデアを紹介する。

交叉オペレーターの設計はGAの性能を向上させるために特に重要である。GAを用いて優れた探索性能を実現するためには、問題の性質を考慮した交叉設計が必要となる。2.3項ではTSPを例として、そのような交叉設計の例を紹介する。

#### 2.2 世代交代モデル

複製選択で選択圧をかける方法として、古典的なGAではルーレット選択やランキング選択などが提案されている。例えばランキング選択では集団中の個体の適応度に比例した確率で親集団を選択する。ただし、生存選択で選択圧をかける場合には複製選択では選択圧をかけないほうが良いようである。この場合は現集団をそのまま親集団とすれば良い（例えば、 $N/2$  個の親ペアを個体の重複を許さずにランダムに構成）。

生存選択で最も選択圧の高い方法は  $P'(t) \cup P''(t)$  の中から目的関数が良い順に集団サイズ分の個体を選択することであるが、この方法は多様性維持の点で問題がある。簡単かつ効果的に多様性維持を促進する工夫として親子間で生存選択を行う方法がある（家族内淘汰）。各親ペアから交叉で生成される子個体は親ペアに類似する傾向があるため、家族内選択を行うことで、類似個体が複数選択されることを制限することができる。家族内淘汰を採用した世代交代モデルとして、**minimal generation gap (MGG)** [15] はロバストに良い結果が得られると報告されている。MGGでは複製選択はランダムに行い、生存選択は親ペアと生成子個体の中から最良1個とランダム1個体を選択する。家族内淘汰の例として、個体間の類似度を利用する

方法も提案されている。一例として **deterministic crowding (DC)** という方法を説明する。DC では複製選択はランダムに行う。生存選択では各親ペア ( $p_1, p_2$ ) から交叉で子個体を生成し、生成された子個体を  $p_1$  に類似した子個体群  $C_1$  と  $p_2$  に類似した子個体群  $C_2$  に分ける。次に  $C_1$  中の最良子個体が  $p_1$  を改善していれば、その子個体で集団中の個体  $p_1$  を置き換える ( $C_2$  と  $p_2$  についても同様)。仮に、生存選択においてある親ペア ( $p_1, p_2$ ) から  $C_1$  内の最良子個体と  $p_1$  が選択された場合は類似した 2 個体が選択されるが、DC ではそのような状況を禁止していることがわかる。

より高度な生存選択として、集団の多様性を明示的に評価して選択に利用する方法も提案されている。そのような方法の 1 つに熱力学的遺伝的アルゴリズム (thermodynamic GA, TDGA) [17] がある。TDGA では物理学における自由エネルギーの最小化に着想を得た方法で生存選択を行う。自由エネルギーは  $F = E - TH$  で表され、 $E$  を集団の目的関数値の平均、 $H$  を集団の近似的なエントロピー値、 $T$  は温度パラメーターである。TDGA では  $F$  を最小化するように  $P'(t) \cup P''(t)$  から次世代の集団を選択する。例えば、TSP に対して集団のエントロピーは次のように定義される。

$$H_i = - \sum_{j=1}^n r_{ij} \log(r_{ij}) \quad (i = 1, \dots, n),$$

$$H = \sum_{i=1}^n H_i = - \sum_{i=1}^n \sum_{j=1}^n r_{ij} \log(r_{ij}).$$

ここで、各  $i (= 1, \dots, n)$  に対して、 $r_{ij} (j = 1, \dots, n)$  は集団中の個体 (巡回路) において頂点  $i$  が頂点  $j$  と接続している割合を表す。すなわち、 $H_i$  は頂点  $i$  と隣り合う頂点の確率分布のエントロピーを表す。

### 2.3 交叉設計

交叉は親として選ばれた 2 つの解を組み合わせる新しい子個体を生成するオペレーターである。POP の考えに従えば、精度の高い近似解に共通する解の部品 (形質) を組み合わせる子個体を生成するように交叉設計するのが望ましい。優れた交叉を設計するためには、何を形質として選び、どのように親の形質を組み合わせる子個体を生成するかを考える必要がある。このような交叉の設計方針を形質遺伝を考慮した交叉設計という [16]。

TSP では巡回路を構成する枝を形質と考えるのが自然である。このことは直観的には巡回路の長さが枝の長さの和で定義されていることから理解できる。図 1 にランダムに配置された 575 個の点によって生成され

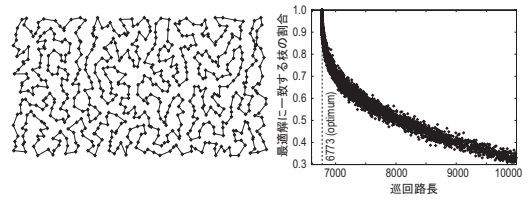


図 1 rat575 の最適解 (左) と近似解の巡回路長と最適解と一致する枝の割合の関係 (右)

た問題例 (rat575) において、適当な方法で得られた近似解の巡回路長を横軸に、最適解に一致する枝の割合を縦軸にプロットしたグラフを示す。近似解の巡回路長が最適値に近づくにつれて、近似解と最適解で一致する枝の割合がほぼ単調に増加している様子がわかる<sup>1</sup>。

TSP に対しては親を構成する枝を組み合わせる子個体を生成するような交叉法が多数考案されている。その中でも edge recombination (ER) [12] は比較的単純な交叉法である。ER ではランダムな頂点からスタートして、各頂点に接続している親の枝 (高々 4 本ある) を順次選択しながらパスを伸ばしていき、最終的に巡回路を構成する。通常、パス形成の終盤では選択可能な枝が存在せず (一度訪問した頂点は二度と訪問できないため)、この場合は未訪問点へ接続する新しい枝をランダムに選択する。ER では簡単な先読みを行い、なるべく多くの親の枝を用いて子個体を生成する工夫が導入されている。しかし、生成子個体には少数ではあるが親に含まれない“長い”枝が含まれるため、親個体がすでに良い近似解である場合に、それを改善する子個体を生成することは困難である。図 2 に ER で生成される子個体の典型例を示す。

親の枝のみを使って子個体を生成することを重視した交叉方法として、subtour exchange crossover (SXX) [16], edge exchange crossover (EXX), partition crossover (PX) [11] が提案されている (SXX は高々 2 本の新しい枝を含む)。図 2 に PX で生成された子個体の例を示している。これらの交叉方法は ER よりも効率的に質の高い子個体 (親を改善する解) を生成することができる。しかし、これらの交叉方法では親の枝のみを用いて子個体を形成するという制約により、生成可能な子個体のバリエーションが少なくなってしまうという問題点がある。

<sup>1</sup> TSP 以外の組合せ最適化問題では、普通これほど明確に POP は成立しない。文献 [3, 5] では図 1 と同様のグラフが二次割当て問題や最大クリーク問題に対して示されている。



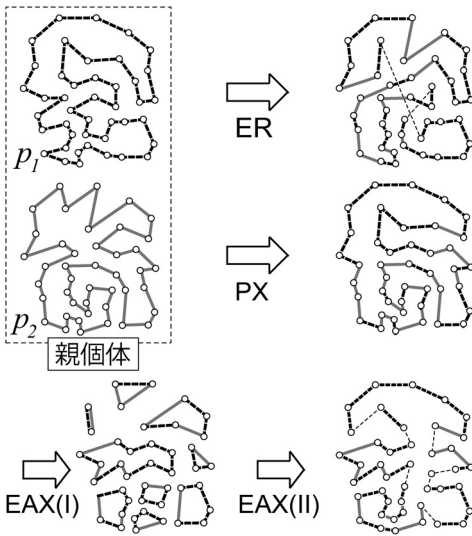


図2 TSPに対する交叉 ER, PX, EAX の例. 親にない新しい枝は細い点線で描かれている

現在, TSP に対する最も強力な交叉方法が edge assembly crossover (EAX) [13] である. 図 2 に EAX の概念図を示している. EAX は 2 つのフェーズから構成される. 第 1 フェーズでは各頂点に 2 本の枝が接続するという条件の下で, 親の枝を組み合わせて中間個体を生成する. 一般に, 中間個体は複数の部分閉路から構成される. 第 2 フェーズでは複数の部分閉路を結合して 1 つの巡回路へと修正する. ただし, 2 つの閉路は各閉路から 1 本の枝を削除して 2 本の枝を加え, 枝の長さの総和が最小になるように結合するものとする. 修正操作は最小全域木を求める方法と同様の考え方に基づいて効率的に実行することができる. EAX で生成される子個体は少数の新しい枝を含むが, そのような枝は短い枝で構成されるため, 質の良い子個体を効率良く生成することができる. また, TSP の緩和条件の下で親の枝を組み合わせるにより, 多様な中間個体および子個体の生成を可能としている.

#### 2.4 TSP に対する GA

交叉 EAX を用いた GA は, いくつかの改良を経て, 現在 TSP の最も優れた近似解法の 1 つとなっている [9]. この GA を GA-EAX と呼ぶことにする. GA-EAX は頂点数が数万~20 万の TSP の問題例に対していくつもの既知最良解を更新している. 図 3 は Mona Lisa TSP と呼ばれる 10 万頂点の問題例で, 2009 年 2 月に TSP のチャレンジ問題<sup>2</sup> として多くの強力な近似解法が適用されたが, 現在の既知最良解 (巡回路長: 5,757,191) は GA-EAX を用いて発見されている.

オリジナルの EAX では, 各頂点が 2 本の枝に接続

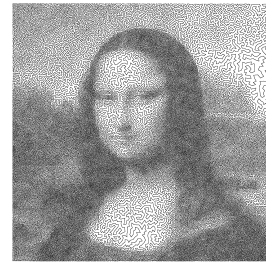


図3 Mona Lisa TSP ( $n = 100,000$ ) の既知最良解

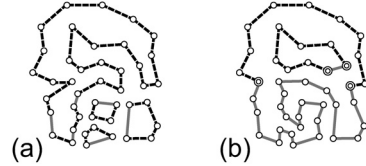


図4 EAX の拡張 (中間個体の例): (a) 局所的な EAX, (b) 部分閉路数を削減する EAX. 親ペアは図 2 と同一

するという条件下でランダムに親の枝を組み合わせて中間個体を生成する. その場合, 中間個体は多くの部分閉路で構成される (図 2 参照). しかし, 部分閉路の個数が増加するほど質の良い子個体を生成することが難しくなる傾向にある. オリジナルの EAX を改良する方法として, GA-EAX では (i) 局所的な EAX (GA の終盤まで使用), および (ii) 部分閉路数を削減する EAX (GA の終盤でのみ使用) を提案している.

局所的な EAX では, 一方の親  $p_1$  の枝を他方の親  $p_2$  の比較的少数の枝で交換することで中間個体を生成する (図 4(a) に典型例を示す). 局所的な EAX には以下の長所がある: (i) GA の終盤以外では効率的に親を改善する子個体を生成可能. これは, 中間個体の部分閉路の個数が減少することによる. (ii) DC のような生存選択と併用することで多様性維持が促進される. (iii) 子個体生成の計算コストを削減できる. 実際, 適切なデータ構造とアルゴリズムを用いて, 交叉の手続きを局所的な操作で行うことが可能となる.

部分閉路数を削減する EAX では, 親の枝を大域的に組み合わせながら部分閉路数の少ない中間個体を生成する (図 4(b) に典型例を示す). この交叉では, 中間個体において  $p_1$  から継承される枝と  $p_2$  から継承さ

<sup>2</sup> 現時点で厳密解法で解かれている最大の (自明な解を持たない) 問題例は 85,900 頂点のものであったが, 次のターゲットとしてこの問題例が選ばれた. 厳密解法の適用では近似解法で最適解 (後で証明される) を発見しておくことが重要である.

れる枝が切替わる点 (図の2重丸) の個数が少なくなるように親の枝を組み合わせる<sup>3</sup>。切替わり点の個数の最小化を簡単な局所探索を用いて近似的に行うことで、低コストで部分閉路数が少なくなるような親の枝の組み合わせ方を決定することができる。

GA-EAX では多様性維持戦略として TDGA のアイデアを導入している。TDGA は生存選択の操作に多くの計算コストが必要となるが、DC の枠組みと組み合わせることでこの問題点を解消している。局所的な EAX で生成される子個体は親  $p_1$  に類似しているため、各子個体で  $p_1$  を置換えた際の  $E$  と  $H$  の変化量  $\Delta E$  と  $\Delta H$  を子個体と  $p_1$  の差分情報のみから局所的に計算することが可能となる。各両親ペアに対し、 $\Delta F = \Delta E - T\Delta H$  を最小とする子個体を選択して  $p_1$  を置換える ( $p_1$  を改善する場合のみ) ことで、低コストで効果的な多様性維持を実現している。

### 3. Memetic Algorithm

memetic algorithm (MA) は GA と局所探索を融合した最適化法である<sup>4</sup>[2]。GA は多点探索の特徴を活かして大域的探索 (解空間の広い範囲を探索すること) に優れた近似解法を構築することができるが、局所探索能力の弱さが指摘されている。これは GA で用いる交叉オペレーターの局所探索能力の欠如に起因する (TSP の交叉 EAX は例外である)。通常、交叉で生成された子個体は、それを初期解として局所探索を適用することで、容易に目的関数を改善することができる。MA は GA の多点探索の長所を活かしつつ、局所探索能力も強化した探索手法といえる。MA はさまざまな組合せ最適化問題へ適用されているが、TSP [10]、二次割当て問題 (quadratic assignment problem, QAP) [5]、グラフ彩色問題 (graph coloring problem, GCP) [4]、最大クリーク問題 (maximum clique problem, MCP) [3]、車両配送問題 (vehicle routing problem, VRP) [7, 8] などの代表的組合せ最適化問題で、特に良い結果が得られている。

MA の枠組みは、GA の枠組み (2.1 項) において子個体の集合  $P''(t)$  を構成するそれぞれの個体に局所探索を適用すると考えればよい。本節では高性能な

<sup>3</sup> 切替わり点の個数が  $2k$  なら部分閉路数は  $k-1$  以下となる。

<sup>4</sup> このような手法は遺伝的局所探索法 (genetic local search, GLS) と呼ばれる。MA は GA のような多点探索型アルゴリズムを対象問題の知識を導入する探索手法というコンセプトで提案されたが、実質的には GLS と同じといってもよい。

MA を構成するための設計方針を解説した後、GCP と VRP に対して提案された高性能 MA を紹介する。

#### 3.1 MA の設計方針

高性能な MA を構築するためには局所探索の設計が重要である。基本的には単独で高性能な局所探索を用いるほど MA の性能も向上する傾向がある。また、局所探索の代わりにシミュレーテッド・アニーリングやタブー探索のようなメタ戦略を用いた MA も多数提案されている。ただし、MA では多くの回数の局所探索を実行するので、1 回の局所探索に要する計算時間が長くなりすぎないようにする必要がある。

次に交叉設計について考える。MA では交叉は局所探索のための初期解生成オペレーターととらえることができる。POP の仮定に従えば、MA で用いる交叉も GA の交叉と同じ設計指針に基づいて設計すればよい。ただし、必ずしも交叉で生成される子個体自体の質が良い必要はなく、重要と考えた形質をある程度適度に継承して子個体を生成するような比較的単純な交叉を用いても良い成果が得られる場合が多い。例えば、2.3 項で紹介した TSP の交叉 ER を用いた GA はそれほど良い探索性能を示さないが、ER で生成した子個体に 2-opt 近傍 (現在の解から 2 本の枝を除き 2 本の枝を加えて得られる解集合) を用いた局所探索を適用することで、探索性能が飛躍的に改善される。

一般に、組合せ最適化問題の解には何らかの制約が課されているが、制約が厳しいほど実行可能解を発見すること自体が困難となる。特に、交叉で実行可能解を生成することは (局所探索における近傍操作で実行可能解を生成することに較べて) 難しい場合が多い。そのような場合、ある程度簡単な交叉で実行不可能解を生成し、それを実行可能解へ修正する方法が有効である。以下ではそのような例も紹介する。

#### 3.2 GCP に対する MA

与えられたグラフ  $G = (V, E)$  に対して、 $k$  種類の色を用いて隣接する頂点どうしが異なる色になるような彩色の仕方を  $k$ -彩色 ( $k$ -coloring) と呼ぶ。GCP は  $k$  を最小とする  $k$ -彩色を見つける問題である。

[1] で提案された GCP に対する MA は単純でありながら強力な近似解法である。この MA の特徴的な構成要素は交叉である greedy partition crossover (GPX) と GCP 用のタブー探索の使用である。一方、複製選択にはランダム選択を用い、生存選択には家族内淘汰の枠組みで悪いほうの親を子個体で置き換えるという単純なものを用いている。

GCP に対するメタ戦略では探索点 (解) を頂点の分

割  $\pi = (V_1, V_2, \dots, V_k)$  で表すことが多い。各  $V_i$  は同一色で塗られた頂点集合である。ただし、 $\pi$  は実行不可能解を含むことに注意されたい。基本的な探索法として、現在の探索までに見つかっている  $k$ -彩色の  $k$  の最小値を  $k^*$  として、 $k = k^* - 1$  となる  $k$ -彩色を探索する方法がよく用いられる（見つかったら、 $k^*$  を 1 だけ減らして同様の探索を繰り返す）。ここで、探索点  $\pi$  の評価関数として  $f(\pi) = \{e \in E \mid e \text{ の両端が同じ集合} \}$  がよく用いられる。すなわち、 $f(\pi)$  は両端の頂点が同色である枝の数である。ここで紹介する MA もこの枠組みに従って探索を行う。

GCP では頂点の分割を形質と考えるのが自然である。交叉 GPX では親となる 2 つの解  $p_1 = (V_1^1, V_2^1, \dots, V_k^1)$  と  $p_2 = (V_1^2, V_2^2, \dots, V_k^2)$  を構成する  $V_i^s$  ( $i = 1, 2, \dots, k, s = 1, 2$ ) の中から  $k$  個を組み合わせさせて子個体を生成することが基本となる。すべての頂点を被覆するような集合の組合せ（親と同一となるものは除く）を見つけることがひとつの理想であるが、GPX では貪欲的な方法を用いてなるべく多くの頂点を被覆するような集合の組み合わせを決定する。具体的には以下の手続きで 1 つの子個体  $c = (V_1, V_2, \dots, V_k)$  を生成する。

#### GPX の手続き：

1. 親  $p_1$  から最大サイズの  $V_i^1$  を選び  $V_1$  とする。  $V_1$  の頂点を  $V_i^s$  ( $i = 1, 2, \dots, k, s = 1, 2$ ) から削除。
2. 親  $p_2$  から最大サイズの  $V_i^2$  を選び  $V_2$  とする。  $V_2$  の頂点を  $V_i^s$  ( $i = 1, 2, \dots, k, s = 1, 2$ ) から削除。
3. ステップ 1, 2 と同様の操作を交互に繰り返して  $V_i$  ( $i = 3, \dots, k$ ) を生成。
4.  $V_i$  ( $i = 1, \dots, k$ ) に含まれていない頂点をランダムにいずれかの集合に割り当てる。

GPX で生成した子個体 (1 個体) には GCP 用の強力なタブー探索 (tabucol と呼ばれる) が適用される。tabucol は基本的なタブー探索で、近傍は 1 つの頂点を別の集合に移動することで得られる解集合として定義される。

上述の MA を改良する方法が提案され、さらに良い探索性能が得られている [4]。改良された MA では (i) 交叉 GPX の多親への拡張、および (ii) 多様性維持を考慮した生存選択を導入している。

(i) については、親数を  $m$  ( $\geq 2$ ) として、オリジナルの GPX の手続きにおいて、最大サイズの  $V_i^s$  ( $i = 1, 2, \dots, k, s = 1, \dots, m$ ) を順次選択することを基本とする（親は順番に選ぶ必要はない）。親数を増やすことで、GPX のステップ 4 でランダムに彩色される頂点

数の削減が期待できる。

(ii) については、現集団中に生成子個体 (1 個体) を加えた個体群を  $P$  として、集団の多様性を考慮した評価関数を用いて削除する 1 個体を決定する。個体群  $P$  の各個体  $\pi$  は評価関数  $g(\pi) = f(\pi) + e^{\beta/D_P(\pi)}$  で評価される (最小化)。ここで、 $D_P(\pi)$  は  $\pi$  と  $P$  を除く  $P$  中の個体との最小距離、 $\beta$  はパラメーターである。個体  $\pi$  は  $P$  中に類似個体が存在すると評価値が悪化することになる。ここで、任意の 2 つの解  $\pi_1$  と  $\pi_2$  の距離は、1 つの点を別の集合へ移動する操作を 1 ステップとして、一方の解から他方の解へ遷移するための最小ステップ数と定義される。この距離は 2 部グラフの最大マッチング問題として定式化して計算することができる。

### 3.3 VRP に対する MA

VRP はデポと呼ばれる集配所にある荷物をあらかじめ指定された  $n$  個の顧客に配送する際の最適経路を発見する問題である。VRP にはさまざまな種類の問題があるが、ここでは最も基本的な Capacitated VRP (CVAP) を説明する。与えられたグラフ  $G = (V, E)$  に対して、頂点集合を  $V = \{0, 1, \dots, n\}$  とする。頂点 0 はデポ、それ以外の点は顧客に対応する。各顧客  $i$  に配送する荷物の重さ  $q_i$  と配送車の最大積載量  $Q$  は既知である。図 5 の左側には 2 つの配送例が描かれているが、4 台の配送車で配送を行っている。1 台の配送車の配送経路をルートと呼ぶ。配送経路には制約として、(i) 各ルートはデポを含む閉路である、(ii) 各ルート上の荷物の重さの和は  $Q$  以下である (容量制約)、(iii) すべての顧客を訪問する、ことが課される。目的関数は全ての配送車の総走行距離である (配送車数は決定変数となる)。

[8] で提案された MA は CVRP やそのほかの種類 VRP でも良い成果を挙げている。この MA は探索の枠組みは単純であるが、形質遺伝性を重視した交叉である SREX と交叉で生成された実行不可能解を実行可能解へ修正する方法 (修正操作) が特徴的である。以下ではこれらの概要を紹介する。

SREX は解を構成するルートを形質と考え、親個体のルートを組み合わせて子個体を生成することを基本とする。以下に SREX の手続きを示す。また、図 5 に SREX の概念図を示す。本稿では説明を簡略化するため、車両数 (ルート数) は固定数  $m$  であるとする。

#### SREX の手続き：

1. 2 つの親個体 ( $\sigma_A^p, \sigma_B^p$ ) を構成するルートから  $m$  個を選択し、重ね合わせる。



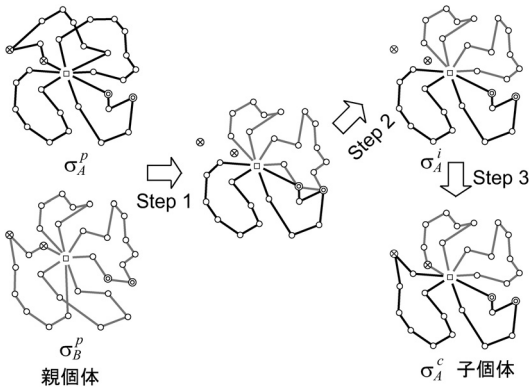


図5 交叉 SREX の概要

- 重複して訪問される点 (図の二重丸の点) は親 A から選択されたルートに残し, 親 B から選択されたルートではスキップする. 得られた中間的な解を  $\sigma_A^i$  とする. また, 親 A と親 B の役割を逆にして得られる中間的な解を  $\sigma_B^i$  とする (図では省略).
- $\sigma_A^i$  および  $\sigma_B^i$  のそれぞれに対し, 孤立している点 (図の×印の点) をランダムな順で選択し, 最も総走行距離の増加が少なくなる個所に挿入する. 生成された解をそれぞれ子個体  $\sigma_A^c$  と  $\sigma_B^c$  とする.

SREX のステップ1では組み合わせるルートを決めるが, なるべく孤立している点 (図の×印の点) が少なくなるようなルートの組み合わせをタブー探索 (局所探索でもよい) を用いて決定する. これは, ステップ3で孤立点を挿入した後に得られる子個体は, 孤立点の数が増えるほど容量制約に違反してしまうためである. また, タブー探索を複数回実行することで, 複数個の子個体を生成することも可能である.

一般に, SREX で生成される子個体は容量制約に違反する. そこで, 修正操作を用いて実行可能解への修正を行う. 修正操作は  $F(\sigma) = f(\sigma) + \alpha F_p(\sigma)$  を目的関数とした局所探索を用いる. ここで,  $f(\sigma)$  はオリジナルの目的関数 (総走行距離),  $F_p(\sigma)$  はペナルティー関数で容量制約違反量を表す. ペナルティー関数は各ルートの荷物の超過量 (ルート上の荷物の総重量  $-Q$ ) の和と定義するのが自然である.  $\alpha$  は重み係数である. 近傍には VRP では標準的な 2-opt, relocation, exchange を用いる (図6参照). 修正操作で実行可能解が得られない場合もあるが, この場合には生成子個体は捨てられる.

修正操作後に得られた実行可能解に対しては, 通常の MA の枠組みに従い局所探索が適用される. 局所探

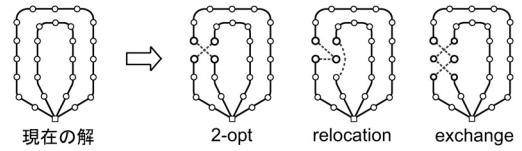


図6 VRP の局所探索における近傍

索では上述の3つの近傍を用い, 目的関数を  $f(\sigma)$  として実行可能解のみを探索する. 世代交代モデルには生存選択にランダム, 複製選択に家族内選択をベースとした単純な方法を用いている.

### 3.4 パス再結合法

MA では交叉で生成した子個体に局所探索を適用することで, 親の形質を組み合わせつつ局所的にも最適化された子個体を生成する. パス再結合法 (PR) [6] は別のアプローチで同様の効果を狙った手法である.

PR では親ペア  $(p_1, p_2)$  から局所探索用の近傍を用いて実行可能解の系列  $\{x_1 (= p_1), x_2, \dots, x_r (= p_2)\}$  を生成し, その中で最良の解を子個体とする. ただし,  $x_{i+1}$  は  $x_i$  の近傍の中から  $d(x_{i+1}, p_2) < d(x_i, p_2)$  となるように選択する. ここで,  $d(x, y)$  は2つの解  $x, y$  の間の距離を表す. 距離は問題に応じて定義する必要があるが, 例えば TSP であれば2つの解の間で異なる枝数などと定義するのが自然である. 上記のルールに従って解の系列  $x_i (i = 1, 2, \dots)$  を生成することで,  $x_i$  は  $p_1$  からスタートして次第に  $p_2$  に接近していく. ここで,  $x_{i+1}$  を  $x_i$  の近傍から選択する際, 目的関数値が最も良いものを選択することで (確率的な選択でもよい), 親の形質を組み合わせつつ局所的にも最適化された子個体を生成することが期待される. また, PR で生成された子個体に局所探索を適用する場合もある.

## 4. おわりに

MA は多点探索型アルゴリズムの長所である大域的探索能力に局所探索能力を合わせ持つ高い潜在能力を持ったメタ戦略である. 一般に, 強力な近似解法を構築するためには対象問題の性質を考慮した設計が必要となるが, MA の枠組みでは交叉や局所探索の設計で問題の性質を考慮する必要がある. 本稿では交叉の設計指針について説明したが, 局所探索の設計に関しては本特集の「メタヒューリスティクス事始め—まずは局所探索法から—」などを参照されたい. 世代交代モデルは汎用的な枠組みであるが, 適切な世代交代モデルの設計には, 基本的な手法を知ったうえで程度度の試行錯誤が必要である. 本稿が読者が MA を設計す

る際の助けとなれば幸いである。

#### 参考文献

- [1] P. Galinier and J.-K. Hao, “Hybrid evolutionary algorithms for graph coloring,” *Journal of Combinatorial Optimization*, **3**(4), 379–397, 1999.
- [2] F. Glover, M. Laguna, and R. Marti, “Scatter search and path relinking: advances and applications,” In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, Springer, 1–36, 2003.
- [3] K. Katayama, A. Hamamoto, and H. Narihisa, “An effective local search for the maximum clique problem,” *Information Processing Letters*, **95**(5), 503–511, 2005.
- [4] Z. Lü and J.-K. Hao, “A memetic algorithm for graph coloring,” *European Journal of Operational Research*, **203**(1), 241–250, 2010.
- [5] P. Merz and B. Freisleben, “Fitness landscape analysis and memetic algorithms for the quadratic assignment problem,” *IEEE Transactions on Evolutionary Computation*, **4**(4), 337–352, 2000.
- [6] P. Moscato and C. Cotta, “A modern introduction to memetic algorithm,” In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, Springer, 2nd edition, 141–184, 2010.
- [7] Y. Nagata and O. Bräysy, “Edge assembly-based memetic algorithm for the capacitated vehicle routing problem,” *Networks*, **54**(4), 205–215, 2009.
- [8] Y. Nagata and S. Kobayashi, “A memetic algorithm for the pickup and delivery problem with time windows using selective route exchange crossover,” In *Proc. of the 11th Int. Conf. on Parallel Problem Solving from Nature*, 536–545, 2010.
- [9] Y. Nagata and S. Kobayashi, “A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem,” *INFORMS Journal on Computing*, **25**(2), 346–363, 2013.
- [10] H. Nguyen, I. Yoshihara, K. Yamamori, and M. Yasunaga, “Implementation of an effective hybrid GA for large-scale traveling salesman problems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **37**(1), 92–99, 2007.
- [11] D. Whitley, D. Hains, and A. Howe, “A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover,” In *Proc. of the 11th Int. Conf. on Parallel Problem Solving from Nature*, 566–575, 2010.
- [12] D. Whitley, T. Starkweather, and D. Fuquay, “Scheduling problems and traveling salesman: the genetic edge recombination,” In *Proc. of the 3rd Int. Conf. on Genetic algorithms*, 133–140, 1989.
- [13] 永田裕一, 小林重信. 巡回セールスマン問題に対する交叉: 枝組み立て交叉の提案と評価, *人工知能学会誌*, **14**(5), 848–859, 1999.
- [14] 応用調査専門委員会編, 進化技術ハンドブック第I巻, 近代科学社, 2010.
- [15] 佐藤浩, 小野功, 小林重信, 遺伝的アルゴリズムにおける世代交代モデルの提案と評価, *人工知能学会誌*, **12**(5), 734–744, 1997.
- [16] 山村雅幸, 小野貴久, 小林重信, 形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法, *人工知能学会誌*, **7**(6), 1049–1059, 1992.
- [17] 前川景示, 森直樹, 玉置久, 他, 熱力学的選択ルールを用いた巡回セールスマン問題の遺伝的解法, *計測自動制御学会論文集*, **33**(9), 939–946, 1997.