

共進化アルゴリズムにおける解表現と
局所探索手法に関する研究

2009年

田村 謙次

名古屋工業大学博士論文
甲第690号(課程修了による)
平成21年3月23日授与

目次

第1章 序論	1
1.1 最適化問題	2
1.1.1 厳密アルゴリズム	5
1.1.2 近似アルゴリズム	6
1.1.3 多様化と集中化	9
1.1.4 多点探索によるメタヒューリスティクス	11
1.2 進化的計算手法	12
1.2.1 進化論	13
1.2.2 進化的計算手法の種類	13
1.2.3 個体表現	15
1.2.4 進化的計算手法の拡張	17
1.3 共進化アルゴリズム	19
1.3.1 GA の遺伝的操作	20
1.3.2 適応度計算	22
1.3.3 GA のアルゴリズム	23
1.3.4 共進化	23
1.3.5 ウイルス進化論	25
1.3.6 ウイルス進化型遺伝的アルゴリズム	27
1.3.7 ウイルスの遺伝的操作	28
1.3.8 ウイルスの適応度と生命力	29
1.3.9 VE-GA のアルゴリズム	31
1.4 本研究の目的	32
1.5 本論文の構成	34

第2章	線形表現を用いたウイルス進化型遺伝的プログラミング	37
2.1	はじめに	37
2.2	従来手法	38
2.2.1	ウイルス進化型遺伝的アルゴリズム	38
2.2.2	遺伝的プログラミング	39
2.2.3	線形型遺伝的プログラミング	42
2.3	VE-LGP の提案手法	45
2.3.1	個体群	45
2.3.2	ウイルス個体の遺伝的操作	46
2.3.3	二点交叉	48
2.3.4	他の類似手法との比較	50
2.4	実験	51
2.4.1	実験環境	51
2.4.2	偶数関数生成問題とは	53
2.4.3	実験結果 1(偶数 6 ビット関数生成問題)	56
2.4.4	Ant とは	58
2.4.5	実験結果 2 (Ant)	62
2.5	まとめ	65
2.6	おわりに	66
第3章	ウイルス進化型遺伝的アルゴリズムにおける感染手法による個体進化の相違に関する一考察	69
3.1	はじめに	69
3.2	VE-GA における個体群	71
3.3	感染手法における進化の相違	71
3.3.1	予備実験	72
3.4	適応的感染手法	76
3.4.1	アルゴリズム	77
3.5	実験	80
3.5.1	結果	80
3.6	考察	81

3.7 まとめ	84
3.8 おわりに	86
第4章 結論	87
謝辞	91
参 考 文 献	93
研究業績	101

目 次

1.1 最適化の枠組み	3
1.2 TSP における経路最適化例	4
1.3 P と NP の関係の 2 つの可能性	6
1.4 メタヒューリスティクス解法の探索のイメージ	7
1.5 巡回セールスマン問題 att48 における共有部品と類似度	10
1.6 巡回セールスマン問題 att48 における類似度の評価	10
1.7 グレイコードの例	15
1.8 巡回セールスマン問題における順回路の例	16
1.9 ネットワークの分割	16
1.10 算術交叉の適用例	17
1.11 近傍モデル	17
1.12 島モデル	18
1.13 共進化モデル	18
1.14 並列化モデル	19
1.15 GA における一点交叉	21
1.16 GA における突然変異	22
1.17 GA の基本的なアルゴリズム	24
1.18 個体群の概念図	28
1.19 ウイルス感染	29
1.20 感染方法例	30
1.21 ウイルス取込	31
1.22 ウイルス感染と個体群 S	31
1.23 VE-GA におけるアルゴリズム	32
2.1 GA と GP における個体例	39

2.2 GP における交叉	40
2.3 GP における突然変異	41
2.4 GP でのポインタによる個体表現例	42
2.5 LGP における Profix 形式と Prefix 形式	43
2.6 配列による個体表現	43
2.7 配列表現における交叉	44
2.8 配列表現における突然変異	45
2.9 宿主個体群とウイルス個体群	47
2.10 個体表現における多様性と取込コスト	47
2.11 ウイルス感染	48
2.12 ウイルス取込	49
2.13 二点交叉とウイルス感染	50
2.14 二点交叉	51
2.15 解構造のパターン	51
2.16 解の性質 (問題 1) : 偶数 6bit 関数は複数の偶数 2bit 関数から構成することが できる	53
2.17 解の性質 (問題 2) : 個体の持つ遺伝子列により 1 回の行動が決定される ので繰り返し同じ遺伝子列が用いられる	53
2.18 偶数 2bit 関数の例	54
2.19 偶数 6bit 関数	54
2.20 最良宿主適合度の遷移 (実験 1)	57
2.21 世代と個体長の推移	58
2.22 ウイルス適合度の推移	59
2.23 高適合度のウイルスの例	59
2.24 Ant の環境	60
2.25 エージェントの行動	61
2.26 エージェントの行動プログラム例	61
2.27 最良宿主適合度の遷移 (実験 2)	63
2.28 世代と個体長の推移	64
2.29 ウイルス適合度の推移	65
2.30 有効遺伝子列の伝播	66

3.1	ウイルス感染	73
3.2	TSP におけるウイルス感染	74
3.3	予備実験 (25 都市)	75
3.4	各感染手法における破壊頻度	76
3.5	適応的感染の概念図	78
3.6	感染個体数変更アルゴリズム	78
3.7	システム全体の基本的なアルゴリズム	80
3.8	実験結果 (25 都市)	81
3.9	感染回数の推移 (25 都市)	82
3.10	適応度と感染回数の推移 (逆位&挿入)	83
3.11	適応度と感染回数の推移 (PMX&逆位)	84
3.12	適応度と感染回数の推移 (挿入&PMX)	85
3.13	PMX における適応度上昇個体割合の推移	85

表 目 次

1.1	2 種間における相互作用	24
1.2	ダーウィン進化論とウイルス進化論との比較	26
2.1	ポインタを用いる GP の長所と短所	42
2.2	各構造パターン同士の交叉方法	50
2.3	各手法での部分解の取扱いにおける主な相違点	52
2.4	偶数 2bit 関数の出力	54
2.5	偶数関数生成問題における終端記号と非終端記号	55
2.6	偶数関数生成問題における実験パラメータ	55
2.7	宿主平均適合度と分散 (実験 1)	56
2.8	偶数 6bit 関数生成問題における個体長	58
2.9	Ant おける終端記号と非終端記号	60
2.10	Ant における実験パラメータ	62
2.11	宿主の平均適合度と分散 (実験 2)	63
2.12	Ant における平均個体長	64
3.1	感染手法における世代と適応度の関係	77

第1章

序論

近年、計算資源の増加に伴い、進化的計算手法 (Evolutionary Computation : EC) を用いて、実世界の問題への応用が盛んに行われている。EC は遺伝的アルゴリズム (Genetic Algorithm : GA), 進化戦略 (Evolutionary Strategy : ES), および、進化プログラミング (Evolutionary Programming : EP) などの総称であり、生物進化説の枠組みを模倣することにより、適用問題における最適化や探索を実用的な時間内に行う手法の一つである。また、GA においては、解を木構造で表現することにより、構造的知識の探索を行う遺伝的プログラミング (Genetic Programming : GP) や、複数の解個体群を用いることにより、効率的な探索が実現できる共進化手法も提案されている。EC による解探索では、膨大な計算量を必要とするが、大域的な探索が可能なことや適用問題に連続性や微分可能性を仮定する必要がないなどのため、幅広い分野での応用が期待されている。しかし、実世界への応用においては、より信頼性の高い解が望まれるため、計算コストの削減と、より良質な解の探索が求められる。

本研究では、共進化アルゴリズムを用いた GP および GA における解探索手法を提案する。生命進化論の一つであるウイルス進化論を模倣し、適用問題の解候補を持つ宿主個体群と部分解を持つウイルス個体群の共進化により、効率的な解探索を行う手法であるウイルス進化型 GA (Virus Evolutionary GA : VE-GA) に基づき、解となる木構造を GA と同様に線形で解表現された GP の局所探索手法の提案と、GA において、局所的探索手法の相違により、個体進化にどのような影響が生じるかを考察し、適応的に局所探索手法を変更する手法を提案し、共進化アルゴリズムにおける解表現と局所探索手法についての議論をする。

本章では、1.1 節に最適化問題とその解法を主に近似アルゴリズムについて、その有用性と課題について述べる。1.2 節では、近似アルゴリズムの一手法である進化的計

2 第1章 序論

算手法, そして, その代表的な手法である遺伝的アルゴリズムに関して述べる. 次に, 1.3 節で遺伝的アルゴリズムの拡張モデルの一つである共進化手法について述べ, 1.4 節では, 1.1 節から 1.3 節をもとに, 本研究の目的について議論する. 最後に, 1.5 節で本論文の構成を述べる.

本論文では, まず線形表現を用いたウイルス進化型遺伝的プログラミングの提案と, その有効性を確認する. 一般的な GP における解は木構造で表現され, 木構造における遺伝的操作には多大なコストを要するため, GA のように線形で表現し, 計算機への負荷を低減した線形型遺伝的プログラミング (Linear GP : LGP) が提案されている. しかし, Linear GP において線形で表現された解個体は遺伝子列と木構造の情報を保持しているので, 共進化においても構造情報を考慮する必要がある. そこで, 解個体および部分解個体を共に線形にすることにより解探索を行う共進化型 GP を提案し, 偶数 6bit 関数生成問題とエージェント行動制御問題 (ANT) に適用する. 2つの適用問題における解の性質は異なっており, 偶数 6bit 関数生成問題は最適解が複数の部分解から構成され, エージェント行動制御問題は部分解より構成されることに大きな意味を持たないため, その性質を考慮した提案手法の比較検討を行う. さらに, 部分解の伝播手法を適応的に変更する手法を提案し, その評価を行う. 一般的な共進化型 GA において, 部分解の伝播手法は一つの手法のみであったり, ランダムに選択するなどであるが, 部分解の伝播手法により, 解個体の進化への影響が異なることから, 部分解の伝播による解適応度の変化に応じて適応的に伝播手法を変更する手法を提案し, 経路最適化問題である巡回セールスマン問題 (Traveling Salesman Problem : TSP) に適用し, 各伝播手法の組み合わせにおける評価を行う.

1.1 最適化問題

一般に, 最適化問題とは, 与えられた制約条件における目的関数の最大 (最小) 化する変数を求める問題, すなわち, 与えられた問題に対して最もよい解を発見することを目的とする問題で, 数学的に定義すると, パラメータ空間 D 上の関数として目的関数 $f(x)$ ($x \in D$) が与えられたとき, その最大値 (もしくは最小値) f^* , もしくはそのときの x の値 x^* を求める問題として定式化でき, 最大化問題は以下の式 (1.1), (1.2) のように記述できる (図 1.1[40]).

$$f^* = \max_{x \in D} f(x) \quad (1.1)$$

$$x^* = \arg \max_{x \in D} f(x) \quad (1.2)$$

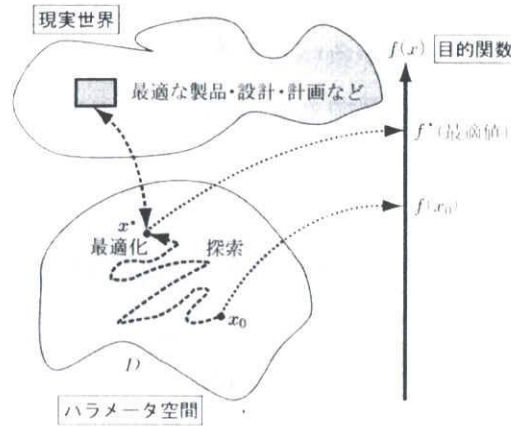


図 1.1: 最適化の枠組み

このような最適化問題は、集積回路設計、通信ネットワーク、自動プログラミング、エンジン設計、スケジューリング、工場のレイアウト、ポートフォリオ選択など、工学の分野だけでなく、経済学や社会学などのさまざまな分野に現れ、最適化の重要性は広く認識されている。さらに、近年のシステムの大規模・複雑化や設計・運用・解析・制御の性能に対する要求の高度化により、一層の効率化・高信頼化・高機能化が求められ、実用的な最適化に対する要求が高まっている。工学的に念頭においた最適化は、(1)最適化理論および最適化アルゴリズム、(2)コンピュータの計算能力、(3)実システムの大規模さ・複雑さ、(4)システムのモデリング・シミュレーション技術の4つの主要な要素が互いに強い影響を与えつつ発展し、実システムの最適化は、これらの4つの要素の適切なバランスの上で実現されることが理想的である [10]。

また、近年のコンピュータの計算能力の向上により、実用的な時間内に計算できる情報量が飛躍的に高まり、より複雑な最適化が可能になったことや、実用的な近似手法や微分可能性・連続性などを必要としないメタヒューリスティクスが開発されるなど、最適化を取り巻く環境が大きく変わりつつある。

4 第1章 序論

一方、ある集合の離散的な要素の最適な並べ方や順序を決定する組合せ最適化問題では、 NP 困難性に代表されるように、大規模な問題においては、厳密な最適解を求めることはきわめて困難であることが、計算の複雑さの理論により明らかにされてきた [5]。したがって、最悪の場合には、大規模組合せ最適化問題の厳密な大域的最適解を求めることは、今後のコンピュータ計算能力の飛躍的な向上を踏まえても、ほとんど絶望的であることを意味している。しかし、実際の問題において解を求める場合には、厳密な最適解である保証はなくても、精度が高い準最適解を実用的な時間で求めることができれば十分である場合が多いことから、コンピュータの計算能力の増大を活用した近似解法の開発が重要な課題となっている。

ここで、 NP 困難性をもつ代表的な組合せ最適化問題の一つである巡回セールスマン問題 (TSP) を例として挙げる。

巡回セールスマン問題は、いくつかの都市とその都市間の距離が与えられたとき、各都市は一回のみ経由するという条件のもとで、すべての都市を巡回し、出発した都市に戻る最短経路を求める問題である。図 1.2 は、バツ印が各都市を表し、最適化後の都市間の経路が結ばれたことを表している。

巡回セールスマン問題は、プリント基板の穴あけの経路を求める問題や、製造業において複数の機械を用いて複数の製品を生産するときの生産コストを求める問題などに帰着することができ、その他にも結晶の構造分析、データアレイのクラスタ化、ガスタービンエンジンの点検作業など多くの分野で応用されており、厳密な最適解を求めることなく開発などに応用するなど、実用的な分野への活用が可能である。

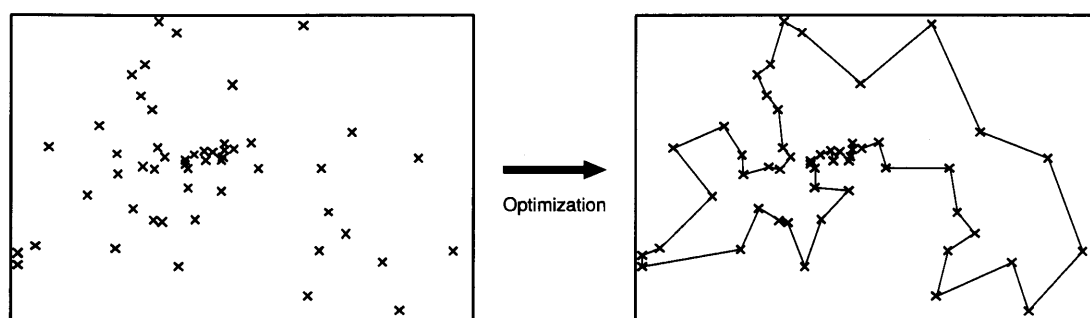


図 1.2: TSP における経路最適化例

以上のことをふまえた最適化における課題は、以下のようにまとめることができる [10]。

(1) 計算困難問題，たとえば NP 困難な大規模組合せ最適化問題などに対する実用的な近似手法の開発

(2) コンピュータパワーの飛躍的増大をふまえた，最適化アルゴリズムとモデリング・シミュレーション技術を結合した最適化の新たな枠組みの構築

(3) モデリング・シミュレーション技術との結合を前提とした最適化手法の開発と，最適化手法との結合を前提としたモデリング・シミュレーション技術の開発による最適化関連技術の発展的再構築

また，最適化のアルゴリズムは，一般的に厳密アルゴリズムと近似アルゴリズムに分類でき，従来は厳密アルゴリズムでの解探索が主流であったが，近年のコンピュータ計算能力の向上により，近似アルゴリズムの研究が盛んに行われている。

1.1.1 厳密アルゴリズム

厳密アルゴリズムは，数学を基礎とした数理計画法として構築され，線形計画法，動的計画法，分岐限定法などがあるが，いくつかの手法では数え上げになる傾向があり，解くべき問題規模が小さい場合や，問題に関する情報が十分に得られ，実用的なアルゴリズムが存在する場合などには有効であるが，それ以外の場合には，多項式時間で終了することができないこともあるので， NP 困難な問題を解くことに使用することは実用的ではない。ここで，問題の難しさを表す P や NP について説明しておく。まず，クラス P は，多項式 (Polynomial) の頭文字をとったもので，多項式時間で解くことができる解きやすい問題を示しており，多項式時間アルゴリズムを1つ見つければ，クラス P に入っていることを証明することができる。しかし，多項式時間アルゴリズムをもたないことを証明することは容易ではないので，そのような問題に対しては，答えが yes か no かのいずれかである問題である決定問題 (decision problem) において，yes であるための証拠が，入力サイズの多項式の大きさでおさえられるとき，その決定問題はクラス NP (Nondeterministic Polynomial) とよばれ，非決定性の計算機において多項式時間で解ける問題のクラスを意味する。例えば，巡回セールスマン問題の決定問題バージョンが効率的に解けるならば，クラス NP に含まれるすべての問題が効率的に解けることが証明されている。非決定性の計算機において， $P \subseteq NP$ であることは明らかであるが， $P \neq NP$ であるか否かはわかっていない (図 1.3[6])。その意味で，巡回セールスマン問題の決定問題バージョンはクラス NP の中で最も難しい

6 第1章 序論

クラスに属する問題と考えられ、そのような問題クラスを NP-完全 (NP-complete) とよび、NP-完全問題よりも難しいものは、NP-困難 (NP-hard) とよばれる。通常の巡回セールスマン問題・最大安定集合問題・ナップサック問題は、NP-困難な問題である。

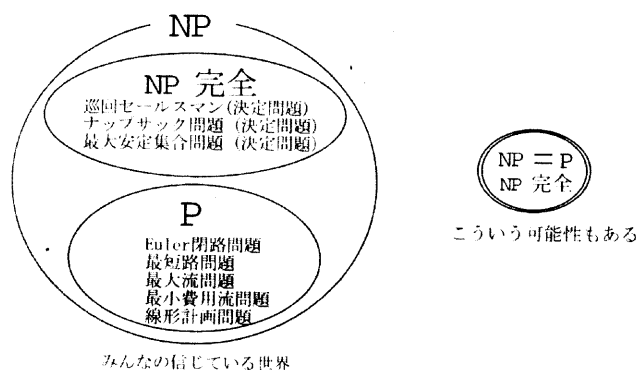


図 1.3: P と NP の関係の 2 つの可能性 (図の大きさがみんなの信じている度合を表す)

1.1.2 近似アルゴリズム

近似アルゴリズムは、メタヒューリスティクス (メタ解法, メタ戦略) ともいわれ、近年のコンピュータ計算能力の向上と最適化問題の複雑化などの要因で注目されるようになった手法で、実用的な時間内で精度の高い解を探索することができる。図 1.4[38] は、メタヒューリスティクスにおける探索のイメージを示しており、解集団がさまざまな領域の探索を行うことにより、局所的最適解や大域的最適解に辿り着いていることを表している。特定の問題に高性能なアルゴリズムは、コンピュータの計算能力が低い場合には有効であり、また、無限の計算能力を用意できるのであれば、ランダム探索や全数探索による手法が有効であるが、コンピュータの計算能力は有限であり、また、問題の特性について十分な知見が得られない未知の問題や、複雑な問題においては、メタヒューリスティクスのように幅広い種類の問題に対して、比較的安定的な性能を示すアルゴリズムが持つ役割は大きい。また、計算コストだけでなく、適用問題をコンピュータで扱えるようにするモデル化のコストも考慮する必要がある。

メタヒューリスティクスの特徴の 1 つとしては、生命・生物現象、物理現象、生物集団、社会行動現象などにアナロジーを持つものが多いことが挙げられ、以下のようなさまざまな手法が提案されている。

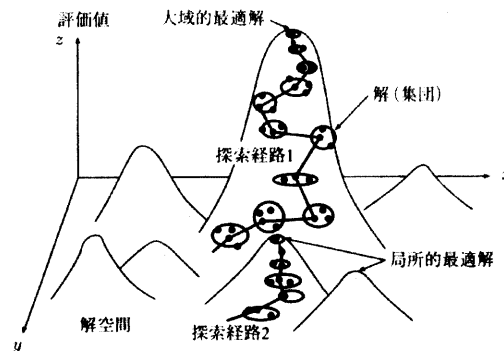


図 1.4: メタヒューリスティクス解法の探索のイメージ

シミュレーティッド・アニーリング 物理現象の焼き鈍しを模倣した探索手法である。

焼き鈍しとは、高温でエネルギー状態が高い物質の粒子がランダムに散らばっている状態から、徐々に温度を下げることにより、最も低いエネルギー状態へ到達し粒子の並びが整った形で整列される現象である。

タブーサーチ タブーリストとよばれる探索を避ける点の集合を保持しつつ探索を行う手法である。タブーリストは人間の記憶のような役割を果たし、探索した点への移動を禁止することにより、局所的探索だけでなく、大域的な探索も行うことができる。

遺伝的アルゴリズム 進化的計算手法の1つで、生物の進化過程を模倣した探索手法である。適用問題の解を遺伝子列として表現し、複数の解候補間で交叉や突然変異などの遺伝的操作による進化を施すことで解の探索を行う。

免疫アルゴリズム 生体の免疫システムがもつ、侵入した抗原に対する防御の仕組みを模倣して探索を行う手法である。免疫システムは、過去に排除した抗原を記憶することにより素早く排除する機能や、初めて侵入した抗原に対する抗体を作り排除する機能を持つことから、抗原を最適化問題の制約条件と目的関数、抗体を解とすることにより、抗原と抗体の親和度を評価値として探索を行う。

ACO(Ant Colony Optimization) 蟻の採餌行動を模倣することにより探索を行う手法である。蟻は餌を発見した際にフェロモンを地面に散布して巣まで帰ることにより、餌がある方向を仲間に知らせることができ、フェロモンは蒸発するという特性を持つので、散布されたフェロモンの濃度から最短経路が選択される。

PSO(Particle Swarm Optimization) 鳥などの群れ行動を模倣した探索手法である。情報を群れ全体で共有し、個体が独立に行動するのではなく、個体情報と群れ全体の共通情報を組み合わせることで、局所的情報と大域的情報により探索を行う手法である。

また、メタヒューリスティクスは組合せ最適化問題の発見的近似手法としてだけでなく、連続かつ多峰構造をもつ非線形最適化問題に対しても有用であることが検証され、幅広い分野への応用が期待されている。このようなメタヒューリスティクスにおけるアルゴリズムは、自由度をもったいくつかのパラメータによって構成されるのが一般的であることから、メタヒューリスティクスは「ヒューリスティックにパラメータを追加し、そこで生まれた自由度を用いて問題を巧みに解くためのテクニック」や「ユーザが設定する計算時間に応じて良い解を探索可能な近似手法」として定義されている。

さらに、メタヒューリスティクスの多くは相互作用をもつ多点探索に基づいており、探索性能・効率の向上、探索のロバスト性の向上、多目的最適化問題における非劣解の同時探索など、従来の最適化手法にはないメリットを持っている [10], [11].

多点探索による探索性能・効率の向上 探索点間の相互作用が、単一の探索点では不可能な多様で豊かなダイナミクスを生み出すことが知られており、新たな近傍生成が、大域的探索や効率的探索の実現に寄与している。

多点探索による探索のロバスト性の向上 多点探索では、一部の探索点における評価値情報が欠落した場合にも、他の探索点を用いて探索が継続できるため、探索におけるロバスト性が高い。

多点探索による非劣解の同時探索 多目的最適化問題は、完全最適解が存在する特殊な場合を除いて、他の解に比べて劣っていない解の集合であるパレート集合が最適解となるので、多点探索では、各探索点がそのまま求める点の集合として対応付けることができる。

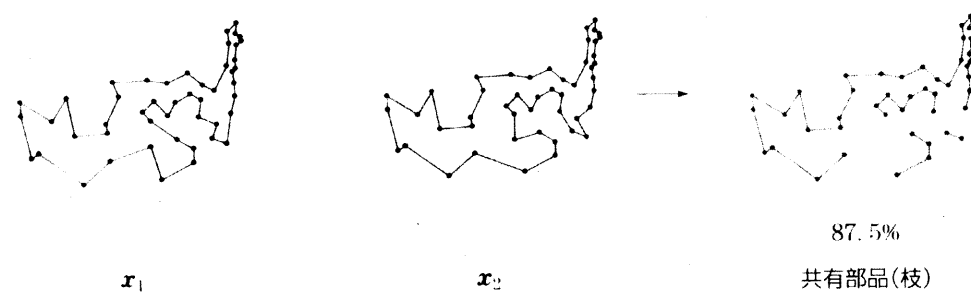
以上のようにメタヒューリスティクスは、厳密アルゴリズムと比較すると、コンピュータ計算能力が向上した今日において実用的な手法であるといえる。

1.1.3 多様化と集中化

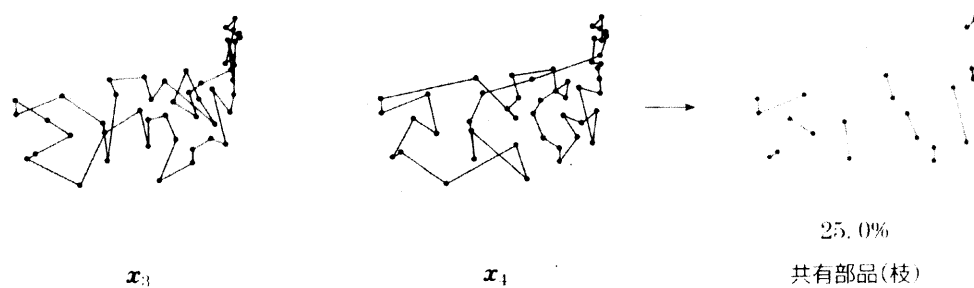
メタヒューリスティクスは、解空間全体の探索を行うわけではないので、最適解が発見されることもあれば、探索を行わなかった範囲に最適解が存在したということも起こりうるが、実用的な時間内に準最適解を発見することが目的なので、局所解に陥ることなく探索を行うことが必要である。したがって、探索範囲の集中化と多様化が重要となる。集中化とは、良い解の周辺には良い解が存在するという概念のもとで、良い解の周辺を集中的に探索することにより、さらに良い解を探すことで、これは良い解同士の構造は似通っていることが多いという性質を利用したものである。この性質は近接最適性原理 (Proximate Optimality Principle : POP) とよばれ、工学における最適化問題の解構造には、なんらかの偏りが存在しているということが経験的に知られている [10]。図 1.5, 1.6 は、文献 [11] において、巡回セールスマン問題における近接最適性原理の成立について検証したものである。図 1.5(a) は、最適解 x_1 と準最適解 x_2 における巡回路を構成する部品 (都市と都市を結ぶ道) の類似度を評価すると、87.5% が一致していることを示しており、よい解同士には高い類似性があることを示唆している。図 1.5(b) では、あまりよくない解同士を比較したときには、高い類似性がみられないことを示している。さらに、図 1.6 は、大域的最適解との類似度と二つの解同士の平均類似度の評価を表しており、Case1 では、優れた評価値をもつ解ほど大域的最適解との共有部品が増加し、Case2 では、優れた評価値をもつ解同士は、その値が優れているほど共有部品が増加することを示している。

また、多様化とは、集中化とは反対に、より広い範囲を探索することにより、まだ探索していないより良い解を探すことである。メタヒューリスティクスでは、この相反する考えである集中化と多様化をバランスよく行うことがアルゴリズムの設計における課題の一つとなる。そこでメタヒューリスティクスの各手法を比較すると、一般的なシミュレーティッド・アニーリングやタブーサーチでは、単一個体により探索を行い、遺伝的アルゴリズム、免疫アルゴリズム、ACO, PSO は複数点による探索を行う手法に分類できる。単一個体による探索を行うシミュレーティッド・アニーリングおよびタブーサーチを多様化という観点において比較した場合、複数点の探索による手法に比べて、探索している解領域から離れた未知の領域を探索するための工夫が必要となる。また、複数点による探索を行う各手法を比較すると、各手法とも集中化と多様化を組み込んだ手法となっており、解の集合を求める多目的最適化問題などにお

いて、容易に適用できるなどの利点がある。



(a) よい解 x_1 と x_2 の共有部品と類似度



(b) あまりよくない解 x_3 と x_4 の共有部品と類似度

図 1.5: 巡回セールスマン問題 att48 における共有部品と類似度

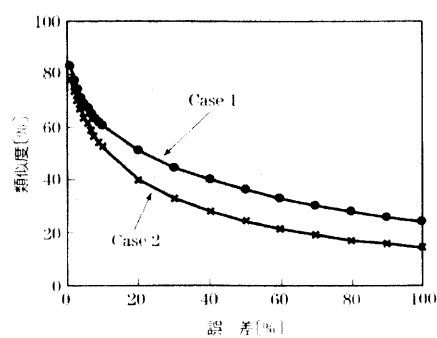


図 1.6: 巡回セールスマン問題 att48 における類似度の評価

1.1.4 多点探索によるメタヒューリスティクス

多点探索によるメタヒューリスティクスは、生命・生物現象、生物集団、社会行動現象などにアナロジーを持つアルゴリズムであり、遺伝的アルゴリズム、免疫アルゴリズム、ACO、PSO など多種多様なものが提案され、各手法ともさまざまな分野に適用され、成果を挙げているが、各アルゴリズムにおける系統的な分類・整理、本質的な相違などについては、十分な検討が行われていないのが現状である。そこで、ここではまず、最適化手法において必要な評価尺度について検討する。評価尺度には以下のようなものが挙げられる [6]。

性能に関するもの 速さ、メモリ使用量、解の良さ

一般性に関するもの 頑強性、パラメータに対する頑強性、汎用性

利便性に関するもの 単純性、実装の容易さ、拡張の容易さ、モジュール化の容易さ

報道報告に関するもの 新規性、重要性

各項目についてみていくと、性能に関するものにおいて、速さは最適解または準最適解発見までの速さ、メモリ使用量はアルゴリズムの軽さを表し小型端末などでの利用などでボトルネックとなる、解の良さはどれだけ最適解に近いかの精度の良さがある。

一般性に関するものにおいて、頑強性は問題の特性によって速さや解の良さが大きく変わらず安定した結果を算出すること、パラメータに対する頑強性は広い範囲のパラメータで安定してよい結果を算出すること、汎用性は広い範囲の問題特性に対して適用可能なことがある。

利便性に関するものにおいて、単純性は解法の理解が容易で記述が簡単なこと、実装の容易さはプログラミングが容易なこと、拡張の容易さは他の異なる種類の問題に対しても容易に拡張できること、モジュール化の容易さは部品化しやすくプログラミングなどで再利用が容易であること。

報道報告に関するものにおいて、新規性は斬新なアイデアであること、重要性は重要な問題を対象としていること。

メタヒューリスティクスの設計には、以上のような評価尺度を考慮することが必要となるが、各手法を比較するとしても、適用する問題の特性などへの依存や、そもそもの設計に柔軟性があるため短所を補うかたちでの改良が可能であることが多いため、

すべての項目において定量的な比較が難しく、明確に優位性を示すことは困難である。さらに、各手法を組み合わせたハイブリット化により多様化と集中化をバランスよく行うほうが、個々の探索手法よりも高性能であることが示されており [10]、このことは単一の手法のみに依存するべきではないことを示唆している。

ここで改めて各手法について考察してみると、ハイブリット化という点においては、遺伝的アルゴリズムに代表される進化的計算手法に優位性があると考えられる。進化的計算手法は、交叉や突然変異などの遺伝的操作による進化により解探索を行う手法であるが、各遺伝的操作は一般的に独立して扱うことが可能であり、前述の利便性に関しての単純性、実装の容易さ、拡張の容易さ、モジュール化の容易さなどの観点からも、他の手法と比較してハイブリット化が容易であるといえる。これは他の手法が進化的計算手法より性能的に劣るというわけではなく、各遺伝的操作が独立という点においてはアルゴリズムの理論的な解析も他の手法に比べて容易であり、また、最適化問題の解決においてハイブリット化が効果的であるという観点においても、進化的計算手法で得た知見を足掛かりにして、他の手法の発展に貢献が可能であると考えられる。

1.2 進化的計算手法

進化的計算手法は、交配、突然変異、淘汰などによる生物の進化メカニズムを模倣した計算技法であり、さまざまな研究分野に応用されている。進化的計算手法の研究には、次のように大きく二つの方向性がある [27] – [31]。

工学的目的 最適化問題の解法や有益な構造の生成

科学的目的 人工生命などにおける既存の生命のより深い理解、とくに生命にとって基本的な属性の理解

工学的目的に関しては、上述までの議論においてその有効性は明確である。科学的目的に関しては、生命のような複雑な現象をコンピュータでモデル化することによって、生物の振る舞いが的確に予測・制御でき、またそれらを工学的に利用することが期待される。実際に最適化問題だけでなく、学習、推論、プログラム自動生成、バイオインフォマティクスなどに応用されている。本研究では、工学的目的の立場から進化的計算手法について議論していく。

1.2.1 進化論

進化的計算手法はダーウィン進化論を主軸としているが、進化論にはダーウィン進化論をはじめとして、以下のようにさまざまな説があり [3], アルゴリズムの設計において、それらからの多くの知見を導入した手法も提案されている。しかし、生物学などにおいては、各進化論の正当性については今なお議論されており、どの説が正しいということを言うことができないのが現状であるが、本研究では、工学的な立場から、進化論の知見を利用することにより、より最適なアルゴリズムを設計することをめざしている。

ダーウィン進化論 突然変異と自然淘汰により種が進化するという説。

ラマルク進化論 親個体が学習・経験したことが子孫に遺伝するという説。

中立進化説 遺伝子の大部分の変化は、生物が生き残るために有利でも不利でもない中立的な変化であり、その変化が蓄積することにより進化するということになるという説。

連続共生説 異なる種の生物が共生することによって新しい種が誕生するという説。

今西進化論 種が生存する場所を棲み分けることで進化が起こるという説。

断続平衡説 進化は連続して起こるのではなく、長い停滞期間の後のある時期に急激に生じるという説。

ウイルス進化論 ウイルスが異なる種の間で遺伝子を運びこむことにより進化が起こるという説。

1.2.2 進化的計算手法の種類

進化的計算は、次のようにさまざまな手法が提案されており、特に遺伝的アルゴリズムや遺伝的プログラミングが盛んに研究されており、本研究では、その2つについて議論する。

- 遺伝的アルゴリズム (Genetic Algorithm)

生物の進化を模倣した確率的探索アルゴリズムであり、さまざまな組合せ最適化問題に応用され、その有効性が示されている。自然界における生物の進化過程に

においては、ある世代を形成している個体の集合 (個体群) の中で、環境への適応度の高い個体が高い確率で生き残れるように選択される。さらに、交叉や突然変異によって進化した個体を含めて、次の世代の個体群が形成されていく。GA では個体群の中に含まれる個体の数を個体群サイズとよび、各個体はそれぞれ染色体によって表現され、染色体は複数個の遺伝子の集まりによって構成されている。生物では、特定の個数の染色体の集まりによって個体が決定されているが、GA では1つの染色体で個体を表現することが多い。染色体上では各遺伝子の置かれている位置を遺伝子座とよび、各遺伝子のとることのできる遺伝子候補を対立遺伝子という。また、染色体の構造は遺伝子型とよばれ、遺伝子によって定まる個体の形質は表現型とよばれる。

- 進化的戦略 (Evolution Strategy)

染色体を実数ベクトルで表現し、ベクトルの各要素に等しく変移を施した上で淘汰と行うことで、システムの最適パラメータを得る手法。

- 進化的プログラミング (Evolutionary Programming)

入出力値の関係によって適応度が評価されるオートマトン集団を変移によって進化させる手法。

- 遺伝的プログラミング (Genetic Programming)

GP はGA から着想を得て考えられた手法で染色体を構造的に表現し、複雑な数式や概念、関係などを木構造で扱えるようにした手法でロボット制御、マルチエージェント学習、システム同定問題などのさまざまな分野で応用されている。アルゴリズムはGA と同様に淘汰、交叉、突然変異を繰り返すことにより解の探索を行なう。GA と異なる点の一つはGA では個体が線形で表現されているのに対しGP は個体を木構造で表現されているので、GA とは異なる方法で遺伝的操作を行わなければならないという点が挙げられる。

以上のような進化論・進化的計算手法を基に、さまざまな性能改善手法による拡張が行われており、それらには次項のようなものがある。

1.2.3 個体表現

一般的な GA において扱う情報は、遺伝子型 (GTYPE) と表現型 (PTYPE) からなり、GTYPE は遺伝的操作である交叉・突然変異などの操作対象であり、PTYPE は GTYPE の適用問題における発現であり、適応度が決定される。

一般に GTYPE は文字列を並べることにより表現されるが、どのような文字列で表現するかは設計者が決定することとなり、同じ問題でも設計者によって個体表現の仕方は異なることもある。したがって、GA の個体表現には設計者の経験的な技術や解くべき問題に応じて、解を遺伝子列として適切に符号化する必要があり、これらは解探索性能に影響を与える。個体表現には、順列構造、ネットワーク構造、木構造などの構造を持つものや、可変長にして世代交代が進むにつれて文字列を複雑化させ、長くするアルゴリズムなども提案されている。以下に主な表現方法をまとめる [40]。

グレイコード 2進数による符号化では、桁上がりによる 0 と 1 の分布が大きく変化するため、突然変異により非連続性が生じる場合があり、探索性能に影響が生じる。そこで、1つのビットによる変化で非連続性が生じることがないようにグレイコードによる符号化が用いられる。しかし、有効な遺伝子列の並びが明確にならないため交叉による探索がうまくいかなる可能性がある。図 1.7 は、10進数の 2進コード、グレイコードへの対応を示しており、2進コードにおいて桁上がり (例えば、0111 から 1000) が生じた場合、大幅な遺伝子の変化が生じるが、グレイコードではそのような変化が生じないことが分かる。

数	2進コード	グレイコード	数	2進コード	グレイコード
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

図 1.7: グレイコードの例

順列構造 巡回セールスマン問題などの順番を扱う問題において、その順列を直接遺伝子列とする符号化。巡回セールスマン問題のように、一つの都市を 1 回のみ巡回するという制約がある場合には、一般的な交叉をすると、2 回巡回する都市や 1

度も巡回しない都市などが生じ、解としては成立しないこともあるので、順列をリスト形式の文字列に変換する手法や解として成立させるようにした部分写像交叉や順序交叉などが提案されている。図 1.8 は、巡回セールスマン問題における遺伝子型を示しており、都市 a, c, b, e, d の順番に巡回していることを表している。

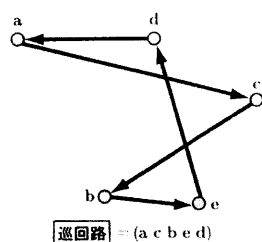


図 1.8: 巡回セールスマン問題における順回路の例

木・グラフ・ネットワーク構造 人工知能における知識表現などにおいて頻繁に用いられる構造の符号化。ノードやアークが遺伝子として表現されるが、単純な交叉や突然変異による遺伝的操作では、構造が崩れる場合もあるので、そのようなことが起こらない手法を用いる必要がある。図 1.9 は、ネットワークで表現された遺伝子 p , q が、それぞれ部分集合に分割されるようすを表している。

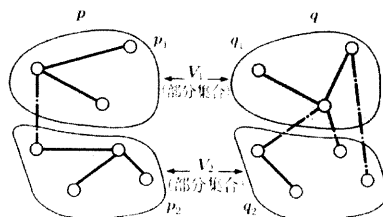


図 1.9: ネットワークの分割

実数値での符号化 2進数などの離散的な変数を用いるのではなく、実数値をベクトルとして符号化。BLX- α , UNDX など、実数の連続性などを活かすための遺伝的操作が提案されている。図 1.10 は、遺伝子 x , y を用いて、 \hat{x} および \hat{y} が生成されることを表している。

このように、解の表現にはさまざまな方法がある。本論文では、順列構造による個

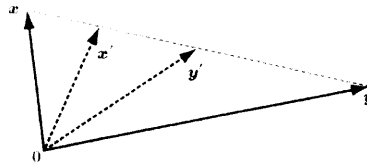


図 1.10: 算術交叉の適用例

体表現と木構造による個体表現において探索の集中化を行う手法を提案する。同手法の詳細については、第2章および第3章を参照されたい。

1.2.4 進化的計算手法の拡張

進化的計算手法で高品質な解を得るためには、多様性を保つために十分な個体数を用意する必要があるが、それにより計算機への負荷が増加し、解探索の効率低下を招くおそれがある。そこで、多様性を維持するために、個体同士の交配を制限する手法や、複数の個体群を用いる手法などが提案されている。また、進化的計算では複数の個体を独立に扱うことが可能であるので、計算機への負荷を軽減するために並列化したモデルも提案されている。以下に、その主な手法を示す [31][51]。

近傍モデル 個体が格子状に配置され、交叉や選択は、基本的に隣接する個体間で行われるモデル。図 1.11 では、楕円が各個体を示しており、線で結ばれた隣接個体同士とのみ交叉や選択などの遺伝的操作が行われることを表している。

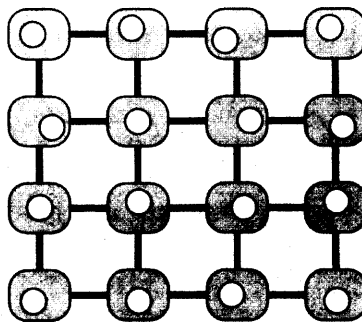


図 1.11: 近傍モデル

島モデル 1つの島に生息する個体群間で交叉や選択が行われることにより、各島で独自の進化が進むが、一部の個体において島間の移住が行われるモデル。図 1.12

は、複数の個体が存在する島同士が線で結ばれており、隣接した島間においてのみ交叉や選択などの遺伝的操作が行われることを表している。

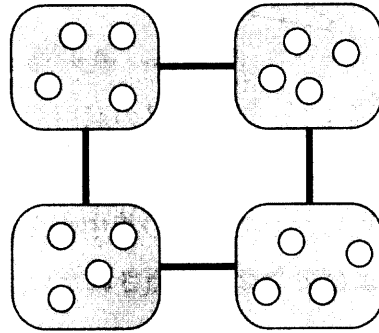


図 1.12: 島モデル

共進化モデル 複数の個体群から構成され、個体群内での個体進化と個体群間での相互作用により進化が行われるモデル。図 1.13 は、環境・種・個体が存在し、種は環境からの影響と他の種からの影響(競争, 共生, 搾取など)を受け、それによる相互作用により進化することを表している。共進化については、次項で詳述する。

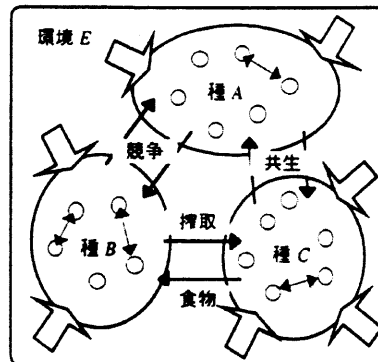


図 1.13: 共進化モデル

並列化モデル 複数のコンピュータを使用した並列化により解探索を行う。適応度計算の並列化や、島モデルなどにおいて独立した島としての並列化などのモデルがある。図 1.14 は、個体評価を一つのコンピュータで行うのではなく、複数のコンピュータに分散させて並列に行うことを表している。

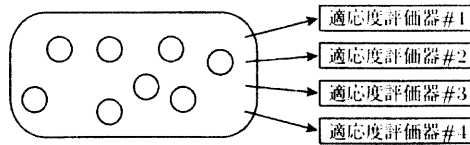


図 1.14: 並列化モデル

ここまで述べてきたように進化的計算手法は、その大きな枠組みである進化論、解となる個体の表現、個体が生存する環境などに相当する拡張モデルがあり、実際の解探索が行われる前に、適用問題の解進化が行われる個体の表現や生存環境の設定が必要となる。それらにはさまざまな手法が提案され、多くの研究がなされており、適用問題・個体表現・拡張モデルの組合せが解探索性能を左右する要因ともなるが、逆にそれが進化的計算手法の解探索や適用可能な問題範囲の広さなどの柔軟性をもたらし、ことなどからより詳細な研究が求められている。特に共進化は我々生命の進化要因の一つであると考えられていることから、共進化による解探索の研究は進化的計算手法のさらなる性能向上に寄与できると考えられる。

1.3 共進化アルゴリズム

遺伝的アルゴリズム (GA) は生物の進化を模倣した確率的探索アルゴリズムであり、さまざまな組合せ最適化問題に応用され、その有効性が示されている。自然界における生物の進化過程においては、ある世代を形成している個体の集合 (個体群) の中で、環境への適応度の高い個体が高い確率で生き残れるように選択され、さらに、交叉や突然変異によって進化した個体を含めて、次の世代の個体群が形成されていく。また、適応の概念には以下のようなものがあるが [41][42]、最適化問題における遺伝的アルゴリズムの適応は、主として進化や学習による適応に位置づけられる。

進化的適応 生物が、環境変化に対応するために行う、非常に長期間に渡る遺伝的な調整過程

生理的適応 個体が、気候や食物の質の変化などに対応するために行う、生理的過程を通しての調整過程

感覚器的適応 感覚器が、検出対象である刺激の強度変化に対応するために行う調整

過程

学習による適応 動物が、多種多様な環境の変動への対処を可能とする過程

GAでは個体群の中に含まれる個体の数を個体群サイズとよび、各個体はそれぞれ染色体によって表現され、染色体は複数個の遺伝子の集まりによって構成されている。生物では、特定の個数の染色体の集まりによって個体が決定されているが、GAでは1つの染色体で個体を表現することが多い。染色体上では各遺伝子の置かれている位置を遺伝子座といい、各遺伝子のとることのできる遺伝子候補を対立遺伝子という。また、染色体の構造は遺伝子型とよばれ、遺伝子によって定まる個体の形質は表現型とよばれる。

1.3.1 GAの遺伝的操作

一般的なGAの遺伝的操作には淘汰、交叉、突然変異があり、それらは適用する問題に応じて適切な方法を選択することが必要であり、次項のようなさまざまな手法も提案されている。

淘汰

適応度の高い個体が次の世代に、より多くの子孫を残すという自然淘汰の考えでは、個体の中で問題(環境)への適応度の良いものは増殖し、逆に悪いものは淘汰する。このような考え方のモデルとしてルーレット戦略が提案されているが、その他にも以下のような方法や期待値戦略やエリート保存戦略なども提案されている。ただし、適応度が高いエリートばかりが多産で生き残りやすいシステムでは、環境の変化に適応出来ず、頑健性がないシステムとなる。したがって、適応度が低い個体も次の世代に残す必要もある。

ルーレット戦略 個体の適応度に応じた確率で次世代の個体を選択する戦略で、適応度が大きな個体を選ばれる確率が高くなり、適応度の低い個体は選択されにくくなる。この戦略の問題点として、確率的に個体が生成されるため、その確率のゆらぎにより集団内で最良の個体が失われる可能性や、集団が収束に近づいたとき、個体間で適応度の差が少なくなり、選ばれる確率がほぼ同じになるなどがある。

ランク戦略 個体の適応度そのものを用いるのではなく、個体を適応度の高い順に並べてその順番をもとに選択を行う戦略で、選択する個体を関数などを導入することにより、選択の圧力を制御することができる。

トーナメント戦略 s 個の個体を組にして、それらの間で適応度の高いものを選択する戦略で、トーナメントのサイズ s を変化させることにより、選択の圧力を制御することができる。

交叉

一般的に交叉とは個体群から二つの個体を選択し、その個体をそれぞれ親1、親2とした時に2つの親の染色体を組み替え、2つの子孫を生成する操作である。

一点交叉 (図 1.15) では、染色体からランダムに交叉点を決定し、親1と親2の交叉点以降に遺伝子列を交換した個体の子孫とする。その他にも以下のような方法や交叉には他にも以下のような方法や部分一致交叉、順序交叉、周期交叉なども提案されている。

多点交叉 一点交叉では1つの交叉点を選択し、その間の遺伝子列を交換する方法が、多点交叉では複数の交叉点を選択し、その間の遺伝子列を交換する

一様交叉 多点交叉のように交叉点を用いるのではなく、0と1からなるテンプレートをランダムに生成し、それにもとづいて互いの文字を交換するかどうかを決定する方法である。

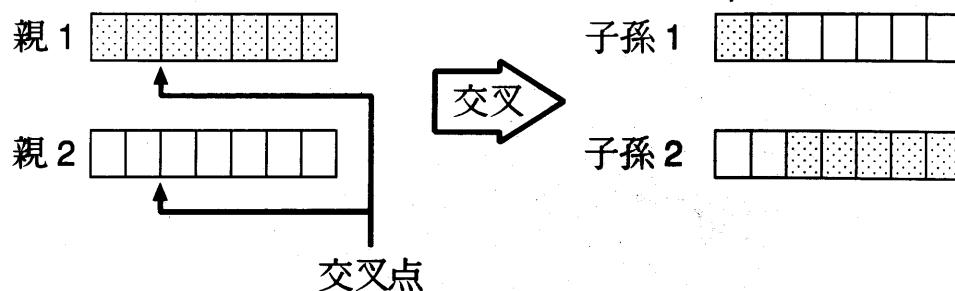


図 1.15: 一点交叉

突然変異

突然変異は遺伝子を一定の確率で変化させる操作である。突然変異が無いと初期の遺伝子の組合せ以外の解空間の探索が行なえないので、求めることのできる解にも限界がある。しかし、突然変異によって有効な解の並びが破壊されてしまう可能性もあるので、解探索において補助的に使われる。一般的に突然変異を行なう確率は一定であるが、確率が高すぎるとランダムサーチとなってしまうので設計者が適切に設定する必要がある。代表的な手法には以下のようなものがある (図 1.16)。

変異 選択された遺伝子座を対立遺伝子に変える。

逆位 染色体から2つの遺伝子座を選択して、それぞれを入れ換える。

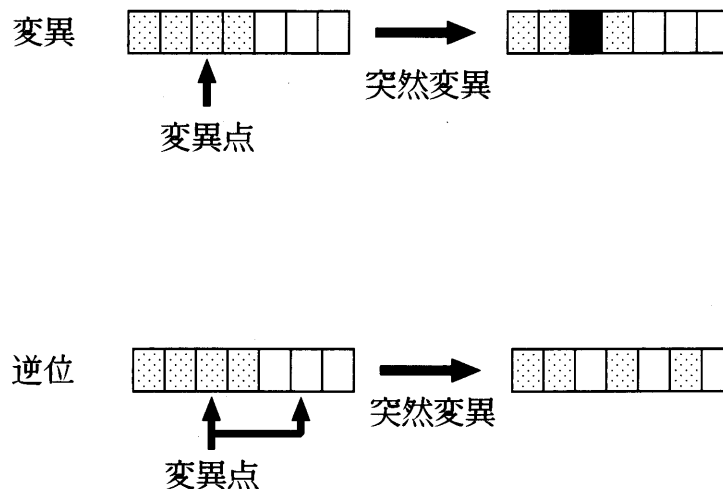


図 1.16: 突然変異

1.3.2 適応度計算

適応度は与えられた環境 (問題) に対して個体がどれだけ適応しているか (解として適切か) を算出したものである。したがって適応度を算出する式は問題に応じて設計者が決定する必要があるので設定した式によってもシステムの性能に影響が現れる。適応度の値をもとに環境に適応できた個体は生き残り増殖することができ、適応できない個体は淘汰され消滅する。また、適応度は適合度と呼ばれる場合もあり、適応度が

「環境への適応度合」という生物学の用語をもとにしているのに対して、適合度は「目的（関数）への適合度合」を意味し、ともに fitness に対応する用語である [40]。進化的計算の分野においては、特に区別なく用いられることが多いが、本研究では、遺伝的プログラムにおいては、目標となるプログラムにどれだけ近い出力が得られるかという意味で適合度、遺伝的アルゴリズムにおいては、どれだけ与えられた条件において好ましい値が得られるかという意味で適応度を用いることとするが、特に異なる意味で用いているわけではないことに注意されたい。

1.3.3 GA のアルゴリズム

GA は図 1.17 のようなアルゴリズムを世代や適応度などの設定した終了条件を満たすまで繰り返す。GA では淘汰によって局所的に解の探索を行ない、交叉や突然変異によって解の探索範囲を広げる操作を繰り返すことによって最適解の探索を行なう。

個体群生成	個体の染色体をランダムに N 個生成する。
淘汰	適応度の高い個体ほど子孫を残せるようにする。
交叉	設定された交叉確率や方法により新たな個体を生成する。
突然変異	設定された突然変異率や方法により新たな個体を生成する。
適応度計算	問題に応じて設定された計算方法に基づいて計算を行なう。
終了条件	条件には世代や設定した適応度などが用いられる。

1.3.4 共進化

共進化とは、相互に影響を与えながら 2 種類以上の生物が進化することであり、実際の生態系では、複数の種が存在し、それぞれが進化することによって、他の種における環境変化が生じ、それがさらに互いの環境に動的变化を促す。進化順としては、競争、寄生（片利共生）、共生（協調）の順に進化したと考えられている。一般的な遺伝的アルゴリズムはダーウィン進化論のような、単一個体群（種）と環境における相互作用による進化システムを基にしている。一方、共進化におけるアルゴリズムは複数の個体群から構成されており、個体群内および個体群間における進化により、他の個体群の適応度景観が変化する。基本的に、共進化では、協調と競争という 2 種類の基礎的な相互作用に基づいており、種間の利益・不利益により相互作用の形態が表 1.1 のように分類される [51]。

また、さらに 2 種間の関係では、共生と競争に分類が可能であり、共生は、利益・

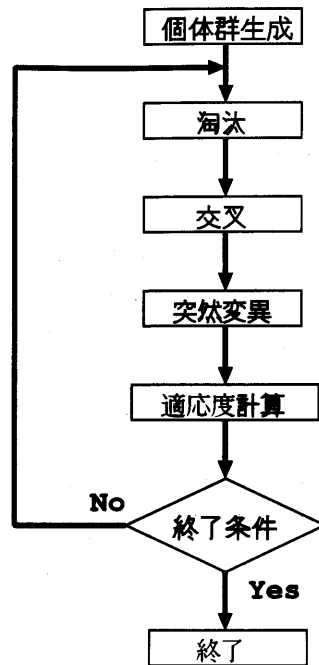


図 1.17: GA の基本的なアルゴリズム

表 1.1: 2 種間における相互作用

		種 B の種 A に対する影響		
		+(利益)	0	-(不利益)
種 A の種 B に対する影響	+	相利共生	片利共生	寄生
	0	片利共生	中立	片害作用
	-	寄生	片害作用	競争

不利益の関係から相利共生，片利共生，寄生に分類が可能である。競争もまた，特殊な場合に片害作用がある。2 種間の共進化のモデルには次のようなものが考えられる。各式は種 A の個体を x_i ，種 B の個体を y_j としたときの適応度関数を表している。本研究では，2 種類の個体間での相利共生モデルを扱う。

- 相利共生モデル

一方の種の適応度が増加するとき，他方の種の適応度が増加するモデルであり，一般的に以下のような適応度関数を用いる。多くの決定変数をもつ複雑な最適化問題などに適用されてきた。

$$\max_i \text{fitness}_A(x_i) = f(x_i, y_j) \quad (1.3)$$

$$\max_j \text{fitness}_B(y_j) = f(x_i, y_j) \quad (1.4)$$

- 競争モデル

一方の種の適応度が増加するとき、他方の種の適応度が減少するモデルあり、極端な場合では以下のような適応度関数を用いる。ゲーム理論におけるゼロ和二人ゲームに相当し、多目的最適化問題などに適用されてきた。

$$\max_i \text{fitness}_A(x_i) = f(x_i, y_j) \quad (1.5)$$

$$\max_j \text{fitness}_B(y_j) = -f(x_i, y_j) \quad (1.6)$$

- 寄生モデル

最大化と最小化等の以下のような目的の異なる適応度関数を用いる。被食者-捕食者の相互作用モデルであり、テスト-解法問題やシミュレーションに基づく最適設計問題などに適用されてきた。

$$\max_i \text{fitness}_A(x_i) = f(x_i, y_j) \quad (1.7)$$

$$\min_j \text{fitness}_B(y_j) = f(x_i, y_j) \quad (1.8)$$

1.3.5 ウイルス進化論

進化論にはさまざまな説があることを前節で述べたが、ここでは共進化により、生命が進化したとするウイルス進化論について述べる [1][2]。ウイルス進化論において主軸にしている点は”進化は遺伝子の水平移動により起こる”ということである。種間で水平移動した遺伝子は、その子孫へ受け継がれることがわかっている。以上のことから、ウイルス進化論では、ウイルス感染による生物の個体間の遺伝子の水平移動とその子孫への遺伝子の垂直移動が進化のプロセスとなっている。

このようなウイルスの特性を利用したものとしては、インシュリンの生成がある。インシュリンは、人の膵臓で作られているホルモンで、糖尿病はそのインシュリンが分

泌されなくなる病気であり，治療のためインシュリンを注射する必要がある．それにはヒトのインシュリンを使うのが最適であるが，直接インシュリンをヒトから集めるわけにはいかない．そこで，ヒトのインシュリンを作る遺伝子をヒトの染色体から切り取って，ウイルスの遺伝子につなげ，今度は，ウイルスがその遺伝子を大腸菌へ運び，その遺伝子を持つ大腸菌が複製されることにより，大量のインシュリンを作ることが可能となるなどの応用が実際に行われている [1]．

ダーウィン進化論との比較

ダーウィン進化論は生物学の世界では広く知られているが，その説には大きな問題がある．それはダーウィン進化論は突然変異の積み重ねで生物が進化したとしているが，もしそうならば進化の過程において現在の形までたどり着くまでの化石が存在するはずだが今のところそのような化石は見つかっていない．そこで提唱されているのがウイルス進化論である．この仮説ではウイルス感染によって遺伝子を変更され全く異なった特徴を持つ生物になってしまうこともあり得るので，ダーウィン進化論における化石が存在しないという不都合を説明することができる．

進化論	進化の要因
ネオ・ダーウィニズム	突然変異の積み重ね
ウイルス進化論	ウイルスによる遺伝子列の組み込み

表 1.2: ダーウィン進化論とウイルス進化論との比較

ウイルス進化論におけるウイルスの役割

ウイルスの中には感染すると宿主の DNA の一部を取り込むことがあるものが存在し，そのウイルスが他の宿主へ感染することで元の宿主の遺伝子を他の宿主に運ぶことがある．このようにウイルスは DNA を運ぶのでベクターともよばれている．種の進化は，その種が属する遺伝子の変化であり，ウイルス進化論では，ウイルス等のベクターが遺伝子の変化をもたらし，遺伝子の水平移動が進化の原因とされる．

1.3.6 ウイルス進化型遺伝的アルゴリズム

ウイルス進化型遺伝的アルゴリズム (Virus Evolutionary Genetic Algorithm : VE-GA) は、適用問題に対する解を持つ宿主個体群と、その部分解を持つウイルス個体群から成り、互いの共進化により効率的な解探索を実現する手法の一つである。一般的な GA はダーウィン進化論をもとにしたアルゴリズムを用いているが、VE-GA はウイルス進化論をもとにしたアルゴリズムである。VE-GA は基本的に 2 種類の個体群から構成され、交叉、突然変異、淘汰の他にウイルスの遺伝的操作が導入された GA である。解候補として表される個体群を宿主個体群、部分解をもつ個体群をウイルス個体群とし、2つの個体群の相互作用による共進化に基づいている。一般的な進化的計算は単一の個体群で交叉や突然変異、淘汰を行なうことにより個体群を進化させるが、共進化は複数の個体群から構成されており、個体群内での個体の進化、さらには個体群間での相互作用で進化するものであり、ある個体群の進化により他の個体群の適応度の景観が変化するシステムである。またウイルスが持つ機能は宿主の遺伝子列を取り込み、それを他の宿主に逆転写し、遺伝子の水平移動を行なうものとしている。VE-GA では交叉は大域探索、感染は局所的探索として用いられる。VE-GA の特長は GA ではスキーマを交叉や淘汰によって個体間に伝播しているのに対し、ウイルスの遺伝的操作によって迅速に生成・伝播し、計算コストを削減できることである。

個体群の定義

VE-GA は宿主個体群とウイルス個体群が存在し、それぞれ以下のような特徴を持っている。個体群の概念図を図 1.18 に示す。

- 宿主個体群
 - － 解候補として表される個体群
 - － 個体長を L とする (固定長)
 - － 環境に対する適応度を持つ
- ウイルス個体群
 - － 部分解を持つ個体群

- ウイルス v の個体長 L_v は可変であり，宿主の進化と共に個体長が伸びていき，宿主個体より短いものとする ($L_v \leq L$)
- 環境に対する適応度と生命力を持つ

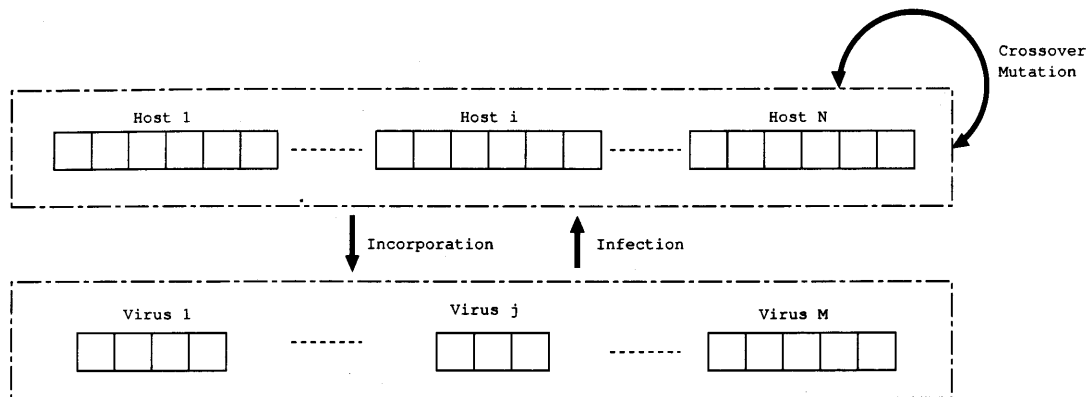


図 1.18: 個体群の概念図

1.3.7 ウイルスの遺伝的操作

宿主個体群は，一般的な GA と同様に交叉・突然変異・淘汰により，宿主個体群のみでの進化が行われる．それと同時に，ウイルス個体独自の遺伝的操作で宿主個体と共進化することにより解探索を行う．ウイルスの遺伝的操作には以下のように感染と取込がある．

ウイルス感染

感染とはウイルスの持つ遺伝子列を宿主個体に転写する操作である．この操作によって宿主は感染前に持っていた遺伝子列を失い，その部分にウイルスの遺伝子列を持つことになる (図 1.19)．また交叉や淘汰によって数世代かけて遺伝子列を個体群内に広めるのではなく，他の宿主個体に転写することにより，速やかに個体群内に広めることが可能になっている．しかし，ウイルス感染は宿主個体群の多様性欠如の原因になり得るのでウイルスの個体数や感染数を適切に設定する必要がある．

また，ウイルス感染方法は単純にウイルスの遺伝子列を宿主に逆転写するだけではなく，交叉と同様に適用する問題に応じてその方法を変える必要がある．代表的な操

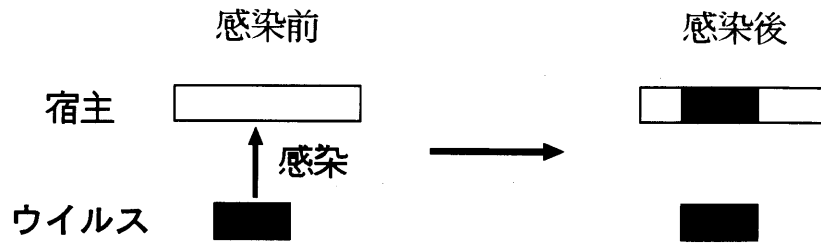


図 1.19: ウイルス感染

作として、以下のようにバイナリ表現された宿主に対する感染，PMX 型，逆位型などがある (図 1.20).

ウイルス取込

取込とは，ウイルスの初期生成および生命力が 0 になった時に宿主個体の遺伝子列の一部をウイルスに取り込む操作である (図 1.21). 取込を行った際にはウイルスの適応度と生命力は設定された初期値に戻される．この操作により，ウイルスの遺伝子列が宿主の進化に貢献しない場合や改悪するなどの場合に，新たな遺伝子列をもつこととなる．したがって，宿主個体は，ウイルスとの共進化および宿主同士の遺伝的操作により，より良い解を保持していれば，その遺伝子を取り込むこととなる．

1.3.8 ウイルスの適応度と生命力

ウイルス i は適応度 $fitvirus_i$ と生命力 $life_{i,t}$ とよばれるパラメータを持つものとし， $fitvirus_i$ は宿主に感染することによって得られる宿主個体の適応度の変化量 (改善度) で表すこととする．また感染された宿主の集合を個体群 S とし， $fithost_j$ をウイルス感染前の宿主の適応度， $fithost'_j$ を感染後の適応度とし，式 (1.9), (1.10) によって計算される．ただし，最大化問題と最小化問題では，式 (1.10) における感染前と感染後の適応度を入れ替える必要がある．

図 1.22 は，式 (1.9) におけるウイルス個体と宿主個体についての概念図であり，一つのウイルスが複数の宿主に感染し，その宿主の群れを宿主群 S とすることを表している． $life_{i,t+1}$ は生命力減衰 r ($0 < r < 1.0$) とウイルスの適応度，前世代 t での生命力とで決定され (式 1.11)，この値が 0 になったらウイルスは新しい宿主を選び，取込を行うことによって新しい遺伝子列を持つ個体となる．生命減衰率は過去の改善度に重

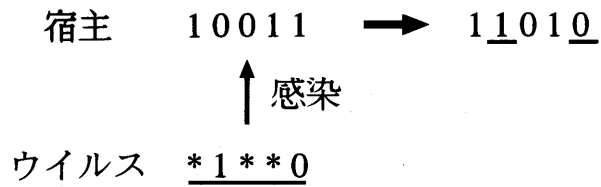
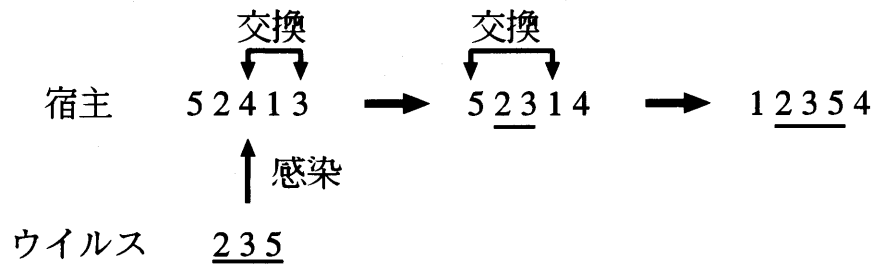
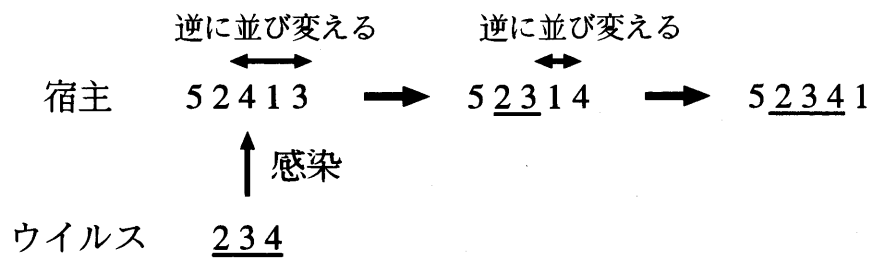
バイナリ表現された宿主に対する感染PMX型逆転写による感染逆位型逆転写による感染

図 1.20: 感染方法例

み付けを与えるための係数であり r が小さいほど前世代における改善度は考慮されなくなる。

$$fitvirus_i = \sum_{j \in S} fitvirus_{i,j} \quad (1.9)$$

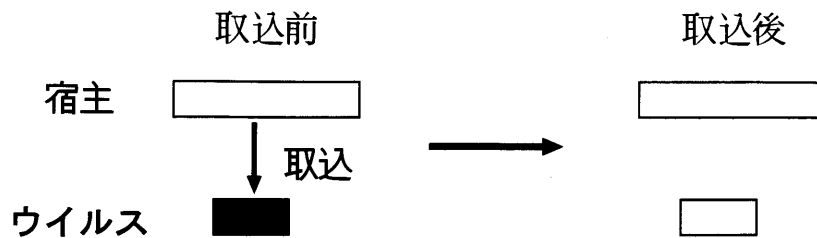


図 1.21: ウイルス取込

$$fitvirus_{i,j} = fithost_j - fithost_j \quad (1.10)$$

$$life_{i,t+1} = r \times life_{i,t} + fitvirus_i \quad (1.11)$$

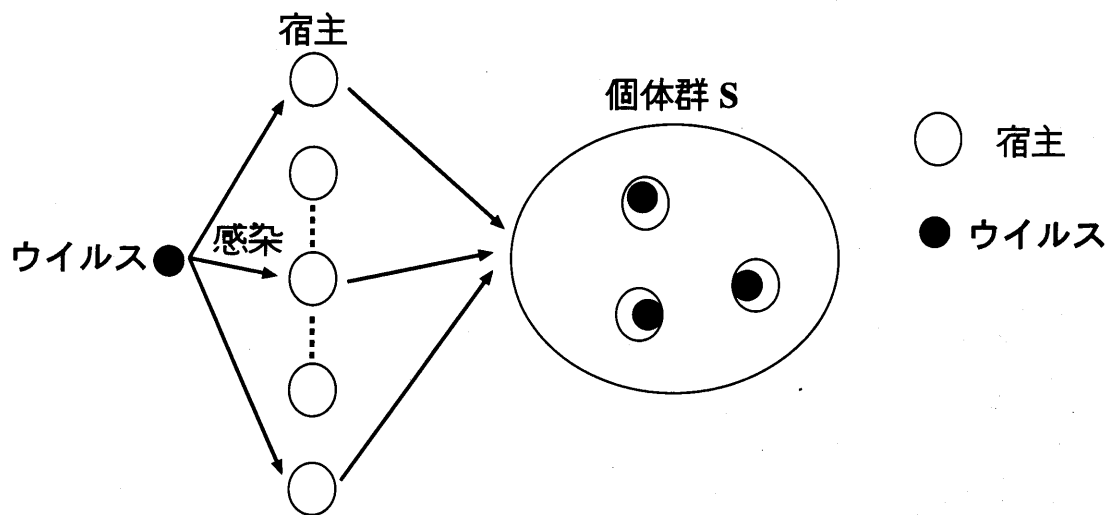


図 1.22: ウイルス感染と個体群 S

1.3.9 VE-GA のアルゴリズム

図 1.23 は、VE-GA の基本的なアルゴリズムである。一般的な GA のアルゴリズムと異なる処理はウイルスによる感染・取込操作および、ウイルスの適応度、生命力の算出であり、基本的な処理の流れは GA と同様である。

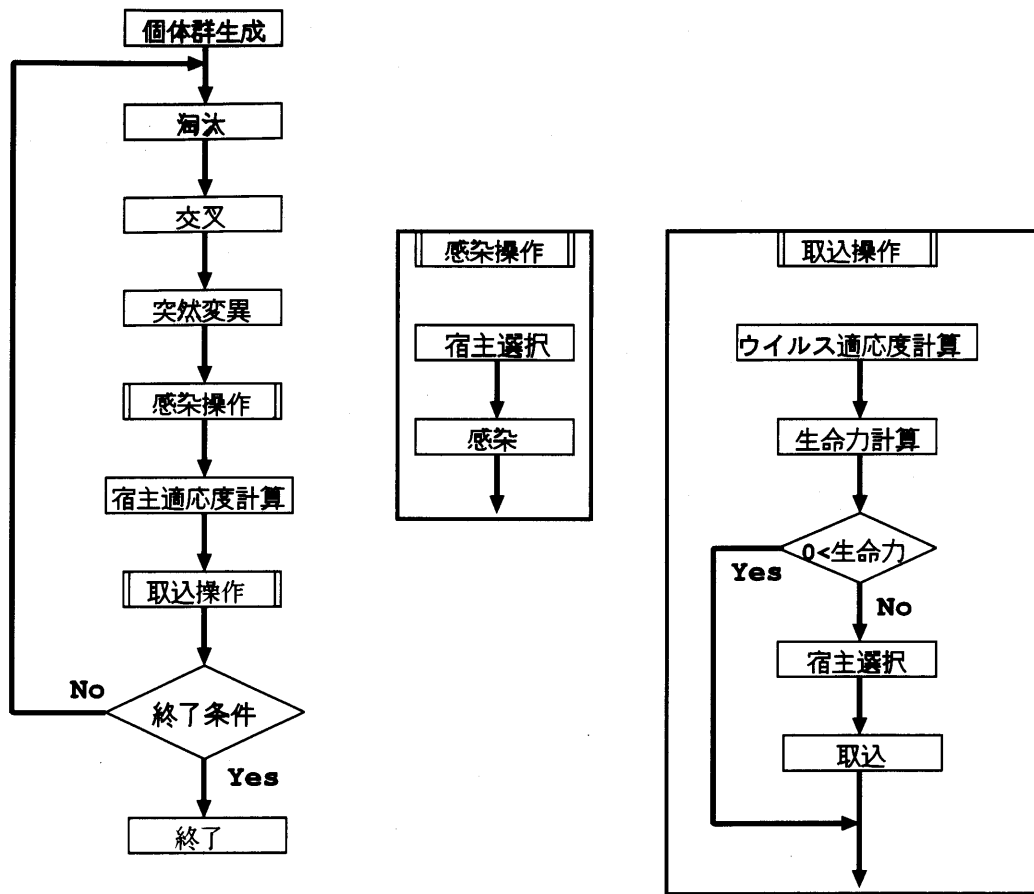


図 1.23: VE-GA におけるアルゴリズム

ここまで述べてきたように、遺伝的アルゴリズムおよび共進化モデルであるウイルス進化型遺伝的アルゴリズムの遺伝的操作には淘汰、交叉、突然変異があり、さらにその中においても探索性能や適用問題に応じた手法の違いがある。また、ウイルス進化型遺伝的アルゴリズムにおいては感染、取込などの遺伝的操作もある。1.2節で述べたことと同様に、これらの構成要素により進化的計算手法に柔軟性をもたらしめているが、その有効性についてはさらなる研究が必要であると考えられる。

1.4 本研究の目的

前節までは最適化問題、進化的計算手法、共進化アルゴリズムについて述べたが、本節では本研究の目的について述べる。まず第 1.1 節では、最適化問題は工学・経済学・社会学などに応用可能な重要な問題であり、その解法には厳密アルゴリズムと近似ア

ルゴリズムが存在し、厳密アルゴリズムは大規模・複雑な問題や効率的な解法が未知の問題に対しては実用的ではなく、一方、近似アルゴリズムは近年の計算機能力の向上により、厳密アルゴリズムでは困難な問題においても準最適解の発見が可能な実用的な手法であることを述べた。次に、近似アルゴリズムでの解探索には、解空間の探索範囲を広げて探索するための多様化と、解が存在する可能性が高い範囲を集中的に探索する集中化が重要であり、シミュレーティッド・アニーリングやタブーサーチに代表される単点探索手法においては、多様性を維持するための工夫が必要であるが、遺伝的アルゴリズム、免疫アルゴリズム、ACO、PSOなどの多点探索手法は、複数の個体により解の多様化の実現が容易であるなどの利点がある。さらに、多点探索による近似アルゴリズムでは、評価尺度として解探索の速さや解の精度などの他にも利便性として、実装・拡張・モジュール化の容易さがあるが、第1.2節で述べたように、遺伝的アルゴリズムに代表される進化的計算は、交叉・突然変異・淘汰などの遺伝的操作が独立していることにより、近傍・島・共進化・並列化モデルのような拡張も容易で、利便性の要素を兼ね備えた手法であり、そのため他の近似アルゴリズムにおける知見の実装も比較的容易に行うことができる。

進化的計算手法は、その手法自体が進化途上の手法であり、多様化、集中化、個体表現(解表現)、拡張モデル、遺伝的操作などにおいても、さらなる研究が求められている。進化的計算手法は多点による探索を行うという点で、初めから多様性を持っているといえる。そこで本論文では、拡張モデルである共進化における個体表現および遺伝的操作による集中化、すなわち、共進化アルゴリズムにおける解表現と局所探索手法を観点として、解探索性能の向上を目的としている。

我々の実社会には、経済や法律など我々人類が築き上げてきたシステムが存在するが、それらは人工的に作られたにも関わらず、その複雑な現象を我々が理解し、制御することは困難であるのが現状である。そこで、生命の進化システムの知見を利用し、我々が容易に想像することが困難なシステムの成り立ちを理解しようとする研究も盛んに行われている。第1.2節において、生命の進化論にはさまざまな種類があることを述べたが、我々生命の進化は、さまざまな種や自然環境との相互作用により生じたと考えられており、それを模倣した共進化アルゴリズムを用いることは、最適化問題を解く上でも有意義であると考えられる。また、第1.3節で述べた本論文で用いる共進化アルゴリズムであるウイルス進化型遺伝的アルゴリズムは、個体が適用問題の部分解を持つことにより、解の局所的な領域を集中化して探索を行うことが可能である。そ

ここで、進化的計算の性能を左右する個体表現においては、解が木構造で表現される遺伝的プログラミングでの共進化アルゴリズムを提案する。個体表現の研究については、第2章を参照されたい。さらに、遺伝的操作には第1.2節、第1.3節のようにさまざまな手法があり、特に解である遺伝子列を直接操作する交叉は解の適応度に直接的な影響を与える。そこで、遺伝子列を直接操作して局所的な探索を行う遺伝的操作である感染操作手法を適応的に選択する共進化型遺伝的アルゴリズムを提案する。感染操作による局所探索の研究については、第3章を参照されたい。

1.5 本論文の構成

本論文では、共進化アルゴリズムにおける解表現と局所探索手法を観点として、GPおよびGAにおける解探索手法を提案する。すなわち、生命進化論の一つであるウイルス進化論を模倣し、適用問題の解候補を持つ宿主個体群と部分解を持つウイルス個体群の共進化により、効率的な解探索を行う手法であるウイルス進化型GA (Virus Evolutionary GA : VE-GA) に基づき、解となる木構造をGAと同様に線形で解表現されたGPにおける局所探索手法の提案と、GAにおいて、局所的探索手法の相違により、個体進化にどのような影響が生じるかを考察し、適応的に局所探索手法を変更する手法を提案する。

まず、第2章では、線形表現を用いたウイルス進化型遺伝的プログラミングの提案と、その有効性を確認する。一般的なGPにおける解は木構造で表現され、木構造における遺伝的操作には多大なコストを要するため、GAのように線形で表現し、計算機への負荷を低減した線形型遺伝的プログラミング (Linear GP) が提案されている。しかし、Linear GPにおいて線形で表現された解個体は遺伝子列と木構造の情報を保持しているので、共進化においても構造情報を考慮する必要がある。そこで、解個体および部分解個体を共に線形にすることにより解探索を行う共進化型GPを提案し、偶数6bit関数生成問題とエージェント行動制御問題 (ANT) に適用する。2つの適用問題における解の性質は異なっており、偶数6bit関数生成問題は最適解が複数の部分解から構成され、エージェント行動制御問題は部分解より構成されることに大きな意味を持たないため、その性質を考慮した提案手法の比較検討を行う。

次に、第3章では、部分解の伝播手法を適応的に変更する手法を提案し、その評価を行う。一般的な共進化型GAにおいて、部分解の伝播手法は一つの手法のみであっ

たり，ランダムに選択するなどであるが，部分解の伝播手法により，解個体の進化への影響が異なることから，部分解の伝播による解適応度の変化に応じて適応的に伝播手法を変更する手法を提案し，経路最適化問題である巡回セールスマン問題 (Traveling Salesman Problem : TSP) に適用し，各伝播手法の組み合わせにおける評価を行う．

最後に，第4章では，本研究の成果と今後の展望について総括する．

第2章

線形表現を用いたウィルス進化型遺伝的プログラミング

生命システムは複雑な働きを担いながらも、柔軟かつロバストなシステムを実現している。そのような自然界のシステムを模倣したものに進化的計算がある。進化的計算は生命進化を模倣したシステムで、遺伝的アルゴリズム (Genetic Algorithm:GA) や遺伝的プログラミング (Genetic Programming:GP) などがあり、最適化や学習などに適用されている。特に遺伝的プログラミングは木やグラフなどの構造的な解表現が扱えるため、プログラム生成や学習、推論などにも応用が可能である。しかし、構造を持つことによる計算量の増加などの問題もある。また、進化的計算では、局所的な解探索は良質な解を得るための重要な要素の一つであるが、解が構造を持つ場合において、解の操作には遺伝子情報だけではなく構造情報も考慮する必要があるなどの課題がある。そこで本章では、構造的に表現された解を線形で表現した遺伝的プログラミングにおいて、部分解を持つ個体を導入した共進化型遺伝的プログラミングを提案し、効率的な解探索を実現することができたことを述べる。

2.1 はじめに

近年、進化的計算手法は多種多様な問題へ応用されており、数理的手法を利用したものなど様々な改良が加えられてきた [29], [30], [71]。それらの主なものに Genetic Algorithm (GA) や Genetic Programming (GP) がある。GA や GP のプロセスは主に交叉と突然変異による環境適応と選択による適者生存にあるが、効率的な探索を実現するための重要な要素の中に有効な部分解 (スキーマ) の発見とそれらの個体間への伝播がある。これは進化的計算手法において最も重要なテーマの一つである。なぜなら、

得られたスキーマを個体間に広めることにより最適解を効率的かつ正確に求めることができるからである。このような観点から、スキーマの探索を有効に行う手法としてウイルス進化型遺伝的アルゴリズム (Virus-Evolutionary Genetic Algorithm; VE-GA) が提案されている [51], [53], [62], [63]。VE-GA は宿主個体群とウイルス個体群を用いた共進化アルゴリズムであり、ウイルス感染と取込により局所的なスキーマの探索と宿主間へのスキーマの伝播を実現している。また、GP は人手によるプログラミングなどの労力を軽減できるとして期待されている手法の一つである。しかし、GP における解の探索には膨大な計算コストが掛かり、高機能な CPU やメモリ等の計算機能力を必要とするという問題点がある。それらを解決する手法として線形表現による GP (Linear Genetic Programming In C; LGPC) が提案されている [72]~[74]。LGPC では個体を木構造ではなく GA と同様に一次元 (線形) で個体を表現することにより、GP において問題点となる木構造の探索コストやメモリ消費量を軽減し、解探索の高速化を実現する手法の一つである。

本章では線形で表現した宿主・ウイルス個体群を用いた GP においてスキーマを効率的に広める VE-LGP (Virus-Evolutionary Linear Genetic Programming) を提案し、数値シミュレーションにおいて従来手法と比べて有効な解が得られたことを報告する。

以下、次節では VE-GA と LGPC の概要を述べ、3 節で提案手法を紹介する。そして、4 節では従来手法との比較実験とその結果について述べ、最後に本章のまとめを行う。

2.2 従来手法

2.2.1 ウイルス進化型遺伝的アルゴリズム

VE-GA では宿主個体群とウイルス個体群という二種類の個体群が互いの相互作用による共進化により解の探索を行う。宿主は適用する問題における解候補を遺伝子列に持ち、ウイルスは部分解を遺伝子列としている。部分解は対象とする問題の解の一部しか表現していないので適用問題の解としては成立しない。ウイルスは感染操作により自身の遺伝子列を宿主にコピーを行い、取込操作により宿主の遺伝子列の一部を自身の遺伝子列とする。これらの操作により VE-GA では部分解を扱う局所探索を行い、その結果として発見されたスキーマを感染により宿主個体群に高速に伝播させる。一方、宿主同士は交叉により大域的な探索を行う。宿主はより良い解候補の探索を行い、

ウイルスは宿主の適応度を向上させる部分解の探索を行う。

2.2.2 遺伝的プログラミング

GP は GA から着想を得て考えられた手法で染色体を構造的に表現し、複雑な数式や概念、関係などを木構造で扱えるようにした手法でロボット制御、マルチエージェント学習、システム同定問題などのさまざまな分野で応用されている。アルゴリズムは GA と同様に淘汰、交叉、突然変異を繰り返すことにより解の探索を行なう。

GA と異なる点の一つは GA では個体が線形で表現されているのに対し GP は個体を木構造で表現されている (図 2.1) ので、GA とは異なる方法で遺伝的操作を行わなければならないということである。

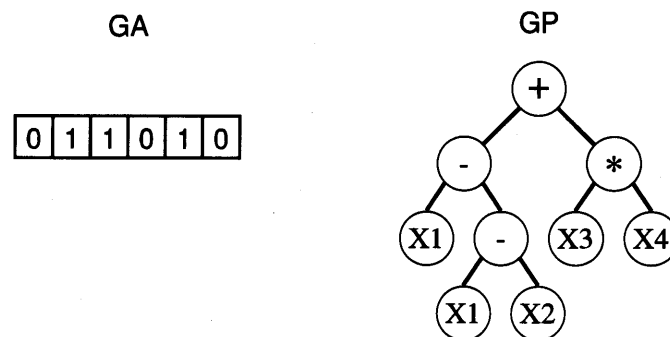


図 2.1: GA と GP における個体例

GP における遺伝的操作

GP では GA と同様に交叉や突然変異によって個体を進化させる。しかし一般的な GP は木構造で表現されているので GA とは異なる方法で遺伝的操作を行わなければならない。

交叉

GP で用いられる交叉は親 1、親 2 のそれぞれに染色体からランダムに交叉点を選択し、選択されたノードを根とした部分木を交換する操作である (図 2.2)。

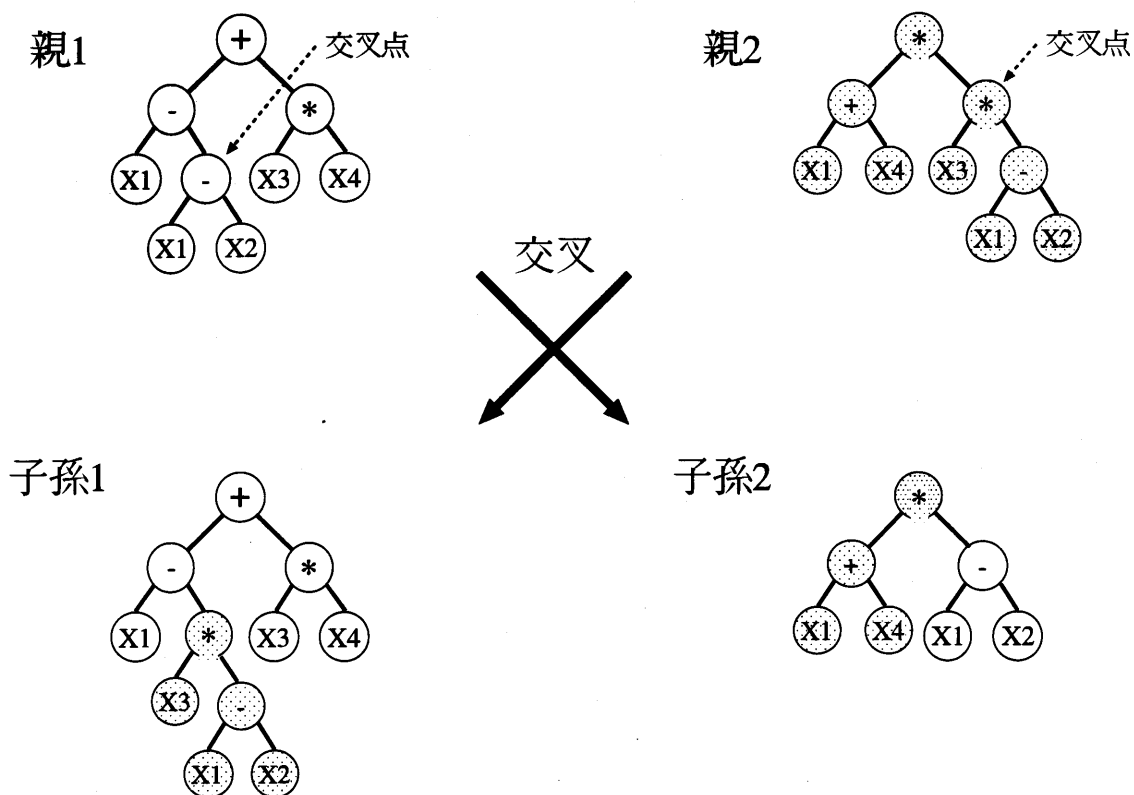


図 2.2: GP における交叉

突然変異

GP で用いられる突然変異は GA と同様な考え方で行なわれるが、GP は木構造なのでノードの変化は次のような場合がある (図 2.3).

1. 非終端ノード → 非終端ノード
2. 非終端ノード → 終端ノード
3. 終端ノード → 非終端ノード
4. 終端ノード → 終端ノード

非終端ノードから終端ノードに変化する場合は終端ノードを根とする部分木を削除する必要があり、逆に終端ノードから非終端ノードの場合には部分木を生成しなければならない。

GP の問題点

一般的に C 言語などを用いる場合、GP では木構造にポインタを用いることにより表現する。そうすることによって交叉や突然変異の遺伝的操作をポインタが指す場所

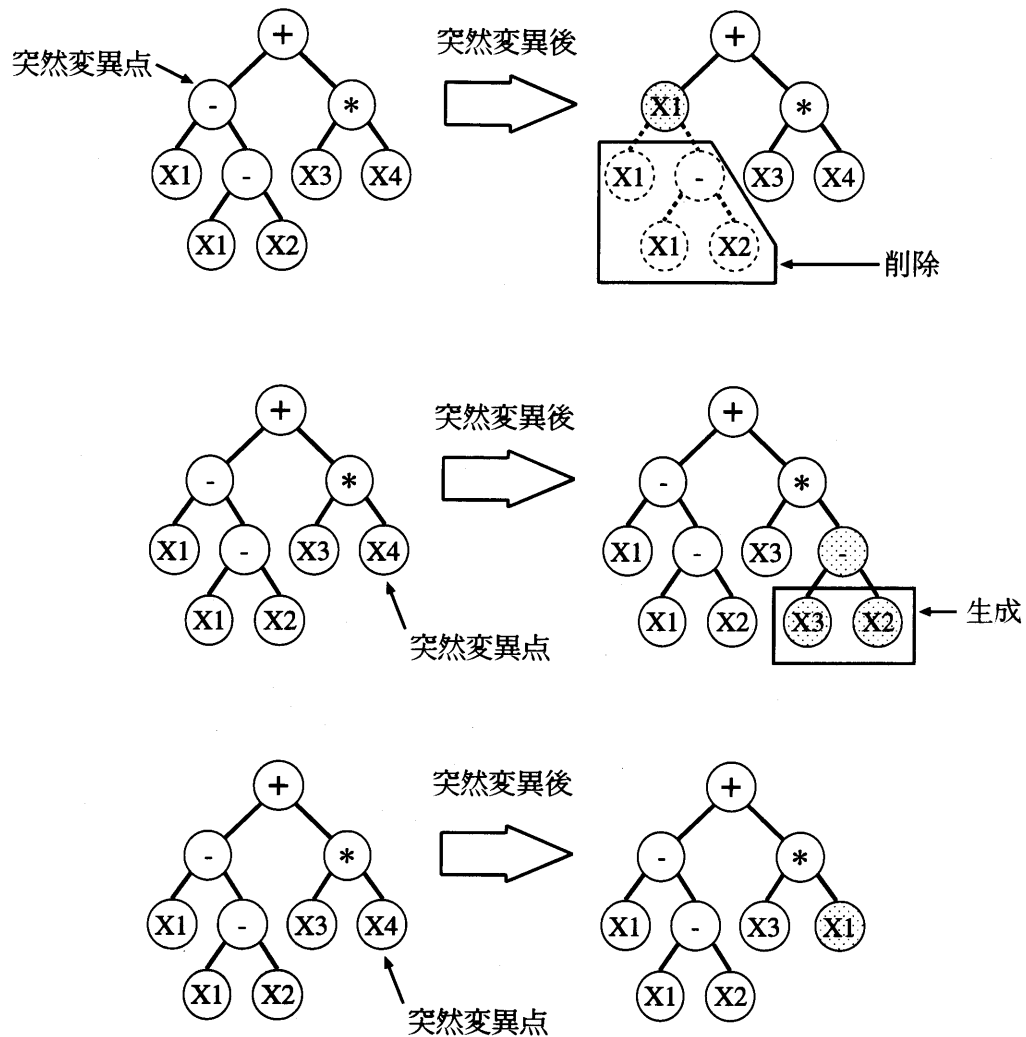


図 2.3: GP における突然変異

を変更することによって容易に行なえるという利点がある。しかし交叉点や突然変異を行なう位置の探索、個体の評価をする際にはポインタを順にたどらなければならず、その探索時間に多くのコストを要する。またポインタに要するメモリ量も考慮する必要がある。GP は多様性を確保するために多くの個体が必要なので、大規模な問題には計算機の能力が低いと解が得られるまでに多くの時間を要してしまう。

2.2.3 線形型遺伝的プログラミング

一般的な GP において個体は木構造を持ち、C 言語で実装した場合はそれをポインタで表現することが多い。ポインタで表現することによって交叉や突然変異の遺伝的操作を、ポインタが指す場所を変更することによって容易に行なえるという利点がある。しかし、ポインタを用いた場合、計算機のメモリ量の問題や、遺伝的操作を施す位置の探索や、個体の評価をする際にはポインタを順番に辿る必要があり、その探索時間に多くのコストを要することになる (図 2.4)。表 2.1 にポインタによる長所と短所を示す。個体の十分な多様性を確保するためには多くの個体が必要となるので、大規模な問題には計算機的能力が低いと解が得られるまでに多くの時間を要するという課題がある。その短所を解決する手法の一つに LGP が提案されており、遺伝的操作においても一般的 GP とは異なるにもかかわらず解探索において同等の性能を示し、高速かつ低メモリ消費で解の探索が行なえるということが示されている。

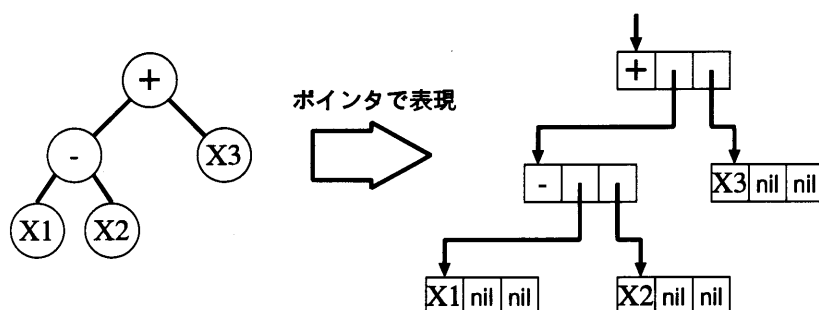


図 2.4: GP でのポインタによる個体表現例

長所	プログラムの構造を直感的に理解しやすい
	ポインタの付け替えによって交叉・突然変異が可能
短所	ポインタ分だけメモリ消費量が多い
	遺伝的操作や個体の評価の際にポインタをたどる必要がある

表 2.1: ポインタを用いる GP の長所と短所

一般的な GA での個体表現は線形、GP での個体表現は木構造である (図 2.1) が、LGPC では、個体を配列で表現することによりメモリー量の節約と探索時間の削減を実現している。

個体表現の方法としては、引数の後に関数を書く Postfix 形式と、その逆の Prefix 形式が考えられるが、Postfix 形式では If-then-else などの制御文の実装が困難なので LGP では Prefix 形式が採用されている (図 2.5).

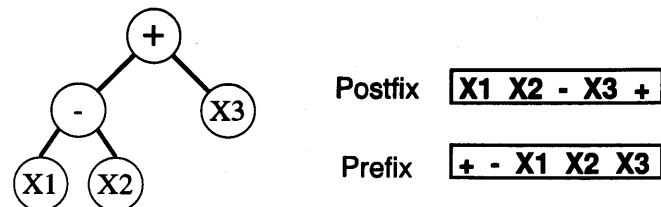


図 2.5: LGP における Postfix 形式と Prefix 形式

配列により線形で表現された個体は、木構造と線形構造とで一意に変換可能である。また、一つの個体は遺伝子情報と構造情報を持つことになり、この構造情報を持つことにより交叉や突然変異を行っても生成される子個体は解としての構造を持つことができる (図 2.6)。構造情報はスタックに対するプッシュとポップの回数の差をスタックカウントとして導入している。木構造として構文的に正しい個体ではスタックカウントの合計が必ず 1 になる。また部分木についても同様に 1 となる。

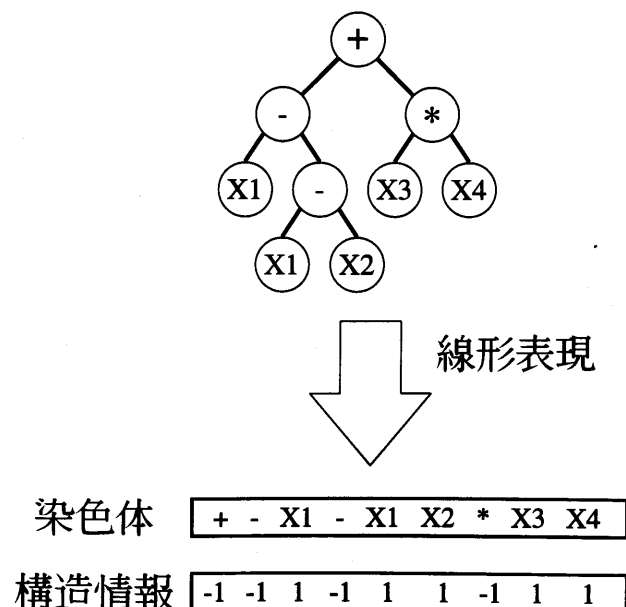


図 2.6: 配列による個体表現

LGP における一点交叉や突然変異は GP とは異なり GA の一点交叉・突然変異に似

た操作を行なう。操作は以下のような手順で行なう (図 2.7, 2.8)。

- 一点交叉

- (1) 個体群から親となる個体を 2 個体選択する。
- (2) 構造情報の先頭から順に両個体の同じ遺伝子座において異なる値が見つかるまで探索する (構造が全く同じ場合には再後尾まで探索)
- (3) 先頭からの構造情報が一致する染色体からランダムに 1 点を選択し、それを交叉点とする
- (4) 交叉点で区切られた親個体同士の遺伝子列を交換し、生成された個体を子孫とする

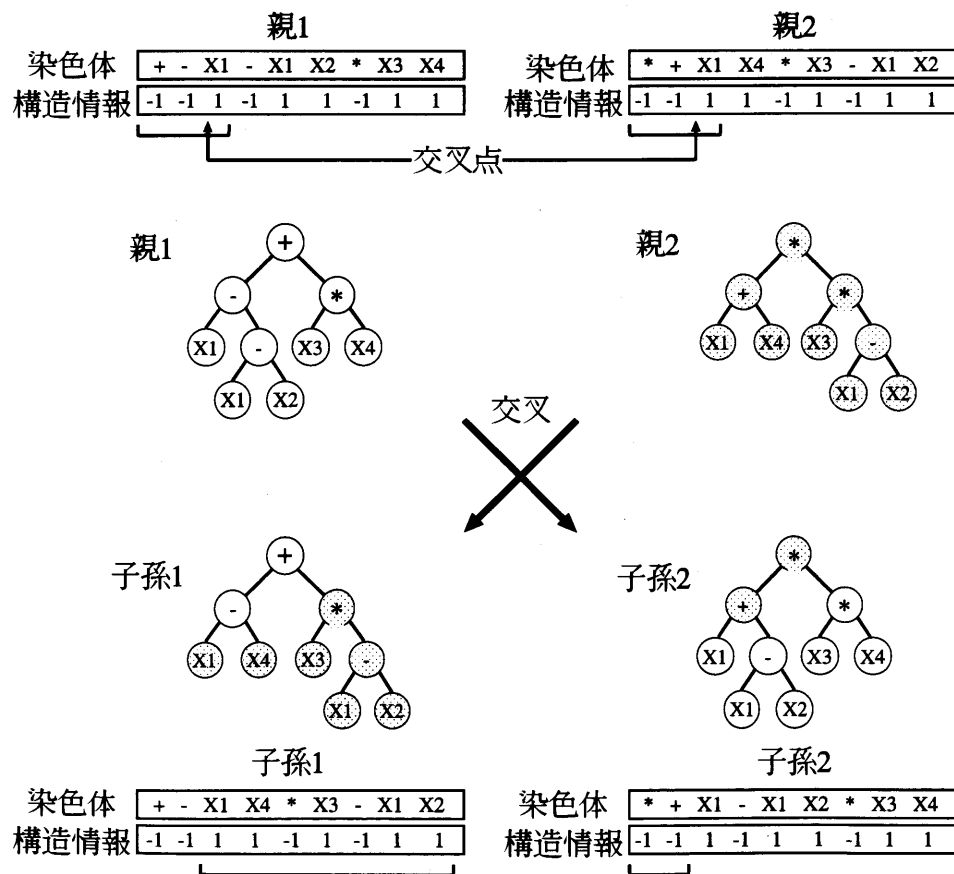


図 2.7: 配列表現における交叉

- 突然変異

- (1) 子孫の遺伝子列をランダムに選択

- (2) 選択された遺伝子座を構造情報と矛盾がないように終端記号は終端記号をランダムに選び、非終端記号の場合も同様に非終端記号をランダムに選択する

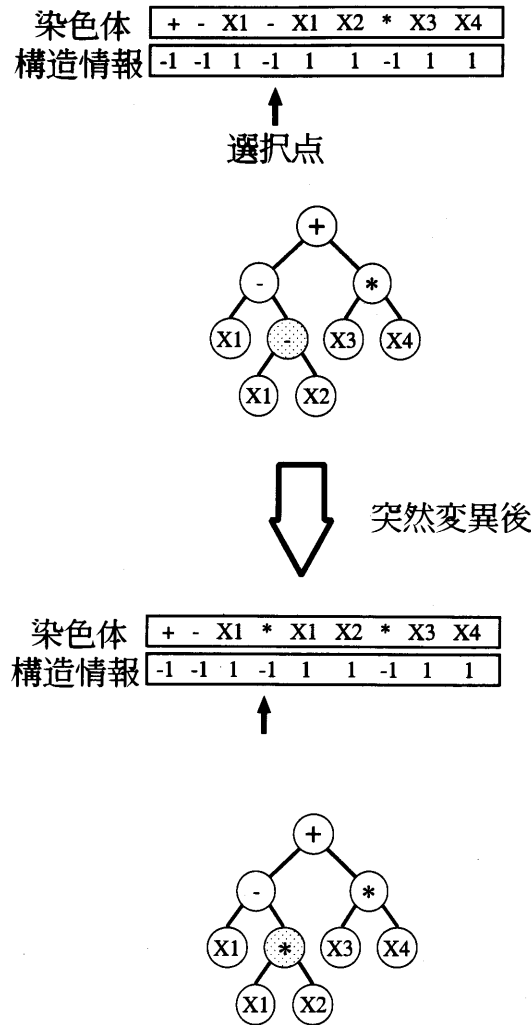


図 2.8: 配列表現における突然変異

2.3 VE-LGP の提案手法

2.3.1 個体群

VE-LGP では VE-GA と同様に宿主個体群とウイルス個体群の二種類の個体群を用いる (図 2.9). また, 各個体は LGP と同様に線形の遺伝子列と木構造を表現するための構造情報を持っている. LGP を用いることによる優位性としては, 通常の GP では

得られない部分木の取得や感染・取込操作における計算コストを削減することである。例として図 2.10 に示すように、[2-3-4][5-6-7] という遺伝子列を取り込む場合、木構造ではすべてのノードを経由して、[1] だけ除外するなどの特別な操作が必要になりますが、線形では容易に取込操作が可能となる。多様な遺伝子列を生成することが探索において必ずしも効果的というわけではないが、有効ではないウイルス個体は淘汰されるため、提案手法において多様な遺伝子列を持つことで解が改悪される可能性が低いと考えられる。また、取込時におけるコストを考えた場合、木構造では $2 \rightarrow 3 \rightarrow 2 \rightarrow 4$ という経路を辿ることにより遺伝子列が選択されるのに対し、線形では $2 \rightarrow 3 \rightarrow 4$ という経路だけで選択を行うことができる。さらに、感染時においてもノードの巡回が不要なためコストを削減することが可能である。宿主個体数はウイルス個体数より十分大きくして、感染により多様性が失わないようにする。また、ウイルス個体はランダムに選択された宿主の遺伝子列から、部分解として取り込む列をランダムに選択することにより生成されるので、個体長は取込を行なった宿主によりは短くなるが、その他の宿主より長い場合もある。また、その個体長は初期世代では短いが世代が進むにつれて長くなるものとする。宿主個体の遺伝子列は木構造となっているが、ウイルス個体はランダムに選択された部分解なのでかならずしも木構造となるわけではない。したがって、ウイルスの遺伝的操作は宿主の木構造を保つため次のように行う。

2.3.2 ウイルス個体の遺伝的操作

宿主個体は一般的な GP と同様に交叉と突然変異により進化するが、LGPC のように遺伝子列を線形で表現した個体に GA と同様な手法の交叉や突然変異などの遺伝的操作を施すとほとんどの場合は木構造ではなくなってしまう。そこで LGPC では構造情報を持つことによりそれを回避している。VE-LGP においてもウイルス個体が遺伝子列と構造情報を持つことにより、ウイルスの遺伝的操作を行っても、木構造が保たれるようにした。ウイルスは VE-GA と同様に適合度と生命力を持ち、宿主の改善度によってその値が求められ、その計算式も VE-GA と同じものを用いる。また、アルゴリズムは GP にウイルスの遺伝的操作を加えたものとしている。

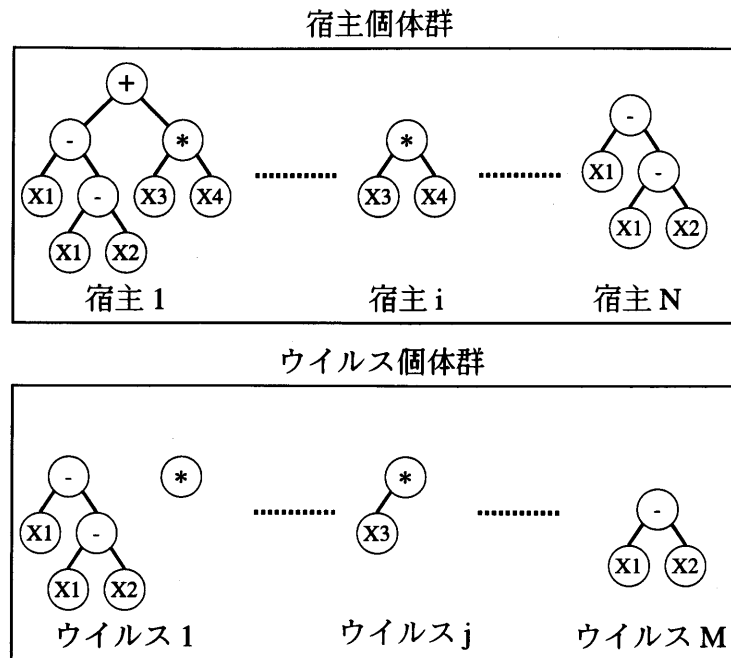


図 2.9: 宿主個体群とウイルス個体群

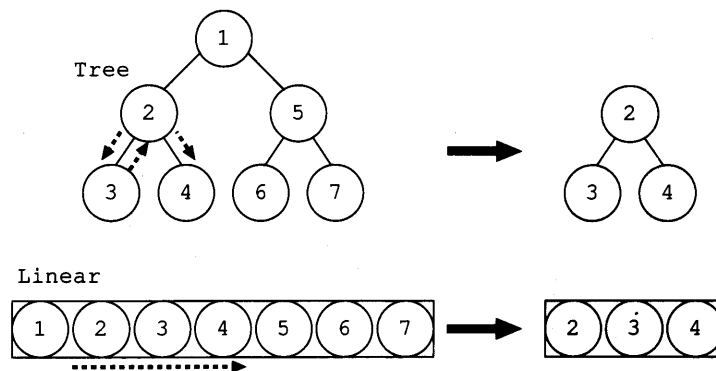


図 2.10: 個体表現における多様性と取込コスト

ウイルス感染

ウイルス自身の遺伝子列を宿主にコピーする遺伝的操作をウイルス感染と呼ぶ。宿主個体の遺伝子列は木構造を表しているので、構造が一致しない位置にウイルス感染が行われると宿主の遺伝子列が木構造ではなくなってしまう。従ってウイルスが持つ構造情報と一致する位置にウイルス感染を行うこととする。ウイルスは選択した宿主のランダムな位置から自身と同じ構造を持つ部分を探索し、一致した場合にウイルス

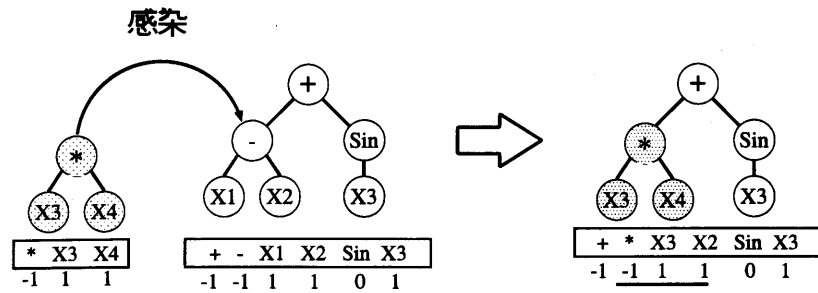


図 2.11: ウイルス感染

感染を行う。すべての構造情報と不一致な場合はウイルス感染は行わないものとする。一般的な GA では個体長は固定されているが、GP では可変長である。したがって、ウイルス感染時に選択された宿主個体がウイルス個体の個体長よりも短い場合もある。そのような場合もウイルス感染は行わないものとする。構造情報が異なる場合でも接ぎ木などにより感染を行うことも考えられるが、それにより宿主の遺伝子列が巨大になりすぎてしまったり、感染操作に膨大な時間を要してしまうことも考えられるので、本論文では接ぎ木はしないものとした。図 2.11 は構造情報が一致する時と不一致の場合の感染について示している。virus_i は宿主と同じ構造を持つので感染を行うことができるが、virus_j は宿主が同じ構造を持っていないので感染を行うことができない。

ウイルス取込

ウイルスが宿主の遺伝子列をコピーして自身の遺伝子列とする遺伝的操作をウイルス取込と呼ぶ。この際、以前に保持していた遺伝子列は失うものとする。VE-LGP では図 2.12 に示すように VE-GA と同様な方法で遺伝子列とその構造情報のコピーを行うが、取込の位置はランダムに決定されるので、ウイルスの遺伝子はかならずしも木構造となるわけではない。virus_i のように不連続な構造を持つ場合や virus_j のように木構造となる場合もある。

2.3.3 二点交叉

ウイルス感染は選択した宿主遺伝子列を部分的に置き換え、新たな遺伝子列とする操作なので、二点間の遺伝子列の置き換えを行う二点交叉と類似している。このような観点からウイルス感染は特殊な二点交叉と行うことができる (図 2.13)。したがって、

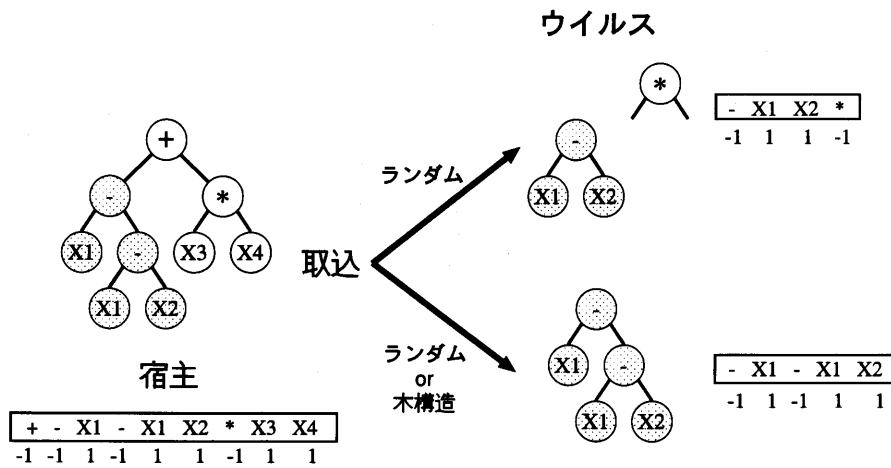


図 2.12: ウイルス取込

ウイルス感染の有効性を示すためには二点交叉との比較も必要との考えから、従来手法では提案されていなかった二点交叉を提案する。二点交叉における交叉点位置の選択方法は、始めに従来の一点交叉と同様な方法で一点目の交叉点を選択し、二点目は図 2.14 のように遺伝子列の末尾から一点目の点を選択した方法と同様な操作により交叉点を選択する。構造情報が一致する部分から交叉点を選択しているので一点交叉と同様に木構造が破壊されることはない。また、適用する問題における非終端記号の引数により、かならずしも構造が一致して、二点交叉が行われるわけではないが、終端記号部分の構造は常に一致する部分があるので、二点交叉ができない場合は一点交叉が行われる。このことを図 2.15 と表 2.2 を用いて示す。図 2.15 は、本論文で解が取り得る木構造のパターン (次章以降での実験において、実験 1 では Pattern 1～4、実験 2 では Pattern 1 のみが存在する) を表しており、表 2.2 は、図 2.15 における各パターン同士の交叉方法を表している。表中の (1)～(4) は、図における Pattern1～4 を表している。例えば、○になっている (1) と (1) では、2 点交叉が可能であることを表している。また、(1) と (2) では、葉に最近の非終端記号の引数が異なるので、交差点として終端記号のみが選択されるという制限が生じる。さらに、(1) と (3) では、根にある非終端記号の引数が異なるので、葉の部分でのみ交叉点となり、1 点交叉となる。以上のことから、2 点交叉においては常に 1 点または 2 点交叉が行われる。

	(1)	(2)	(3)	(4)	
(1)	○	△	×	×	○：2点交叉
(2)	△	○	×	×	△：制限付2点交叉
(3)	×	×	○	△	×
(4)	×	×	△	○	×

表 2.2: 各構造パターン同士の交叉方法

2.3.4 他の類似手法との比較

提案手法のように部分的に解を用いる手法に, Modular Acquisition(MA) や ARGP などがあるが, それらとの相違を表 2.3 に示す. また, 優位性としては, MA, ARGP のモジュールは進化の過程において解個体の交叉により次第に伝播と淘汰が行われるのに対し, 提案手法では部分解(ウイルス)の適合度による淘汰が行われ, 有効な解を持つものが生き残り, その遺伝子列を宿主間に広め, そうでないものは淘汰されるので迅速に有効な解を個体間に広めることができると考えられる.

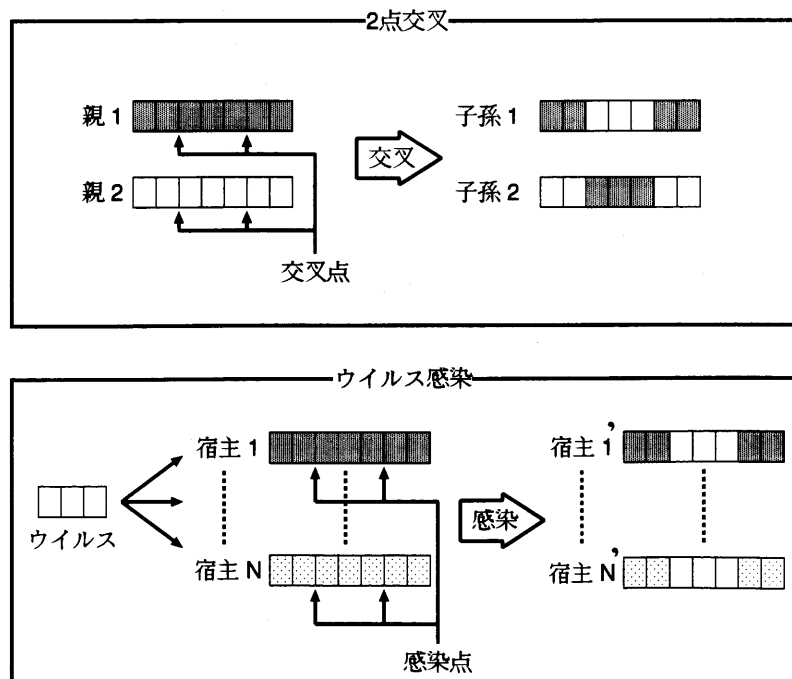


図 2.13: 二点交叉とウイルス感染

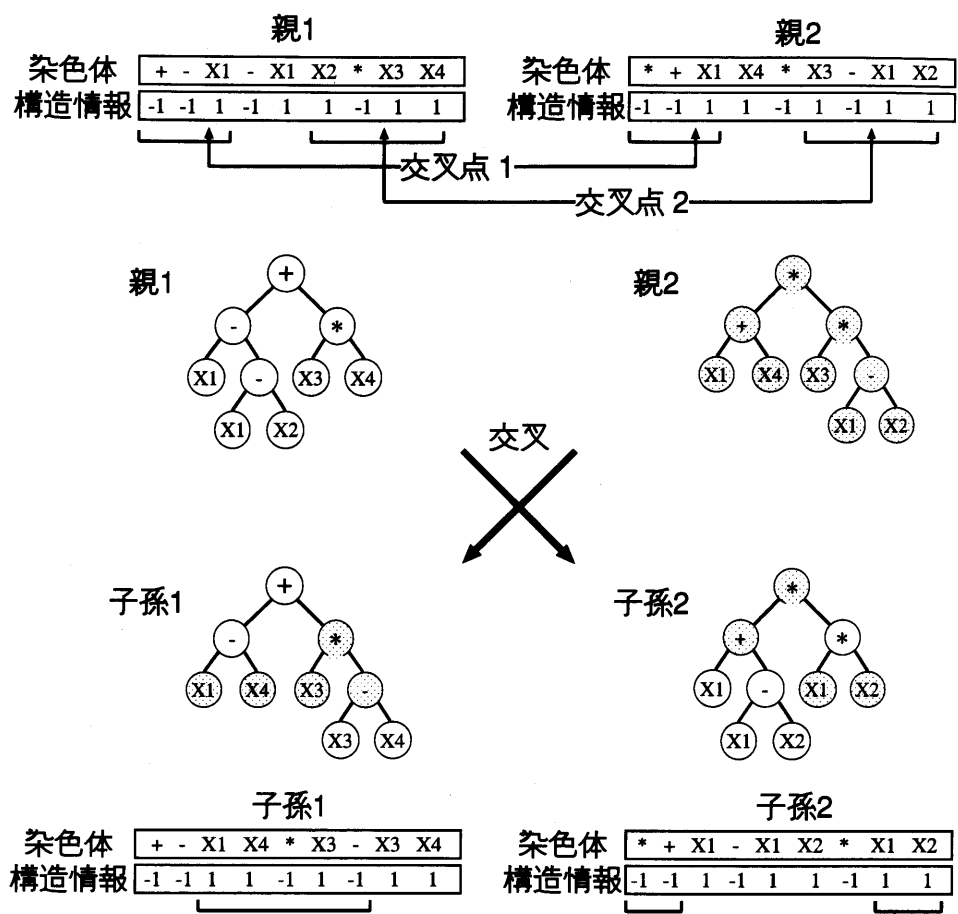


図 2.14: 二点交叉

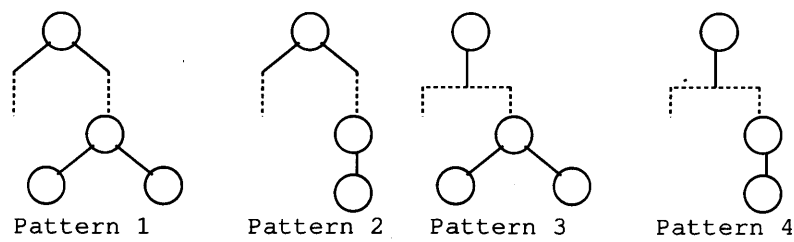


図 2.15: 解構造のパターン

2.4 実験

2.4.1 実験環境

本論文では提案手法と従来手法において数値シミュレーションによる比較実験を行った。宿主個体同様、ウイルス個体も淘汰されることにより有効な遺伝子列を持つ個体

表 2.3: 各手法での部分解の取扱いにおける主な相違点

手法	選択方法	淘汰方法
MA	ランダム	解個体の適合度により淘汰
ARGP	最良個体	解個体の適合度により淘汰
VE-LGP	ランダム	部分解の適合度により淘汰

が生き残り、それを宿主間に伝播させる働きが期待できるが、有効な遺伝子列を持っていてもかならずそれが有効な働きができる遺伝子座に感染できるとは限らないので宿主の進化に貢献できず、逆に適合度を下げてしまう場合もある。しかし、最適解が類似の部分解により構成されている問題であればウイルスがその部分解を取り込むことにより、宿主間にそれを広める確率が増加することが期待できる。以上のことより、本論文では最適解が部分解により構成され得る偶数6ビット関数生成問題（問題1）と部分解より構成されることに意味をもたない問題としてANT（問題2）を扱った[71], [75]。問題1において、宿主の最適解は部分解の集合により表現することができる（図2.16）。一方、問題2では宿主個体は遺伝子列を繰り返し使用することにより適合度が求められるので部分解の集合による表現に意味を持たず、むしろ部分解を繰り返し使用するような遺伝子列は複雑さを増し、解の探索に悪影響となると考えられる（図2.17）。したがって、問題1のような構造を最適解とする問題に対して提案手法はウイルス感染により効率良くスキーマの伝搬が行え、より最適な解が得られると期待できる。一方、問題2においてはウイルス感染による効果はあまり期待できないと考えられる。しかし、未知な問題においてその解の性質をあらかじめ予測することは困難な場合が多いので、適用した手法が解の性質にかかわらず改悪しないことが必要である。両問題は共に適合度の値が0に近いほど最適な解である。実験回数はそれぞれの手法で20回とし、また、各パラメータは試行錯誤により決定した。以下に示す実験結果において、交叉方法の'1 or 2', '1', '2', はそれぞれ'一点交叉または二点交叉', '一点交叉', '二点交叉'を表している。'1 or 2'ではランダムに'1'または'2'が選択されることを意味している。

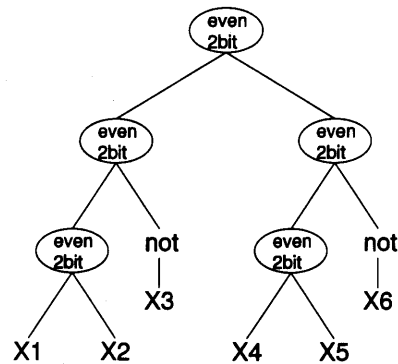


図 2.16: 解の性質 (問題 1) : 偶数 6bit 関数は複数の偶数 2bit 関数から構成することができる

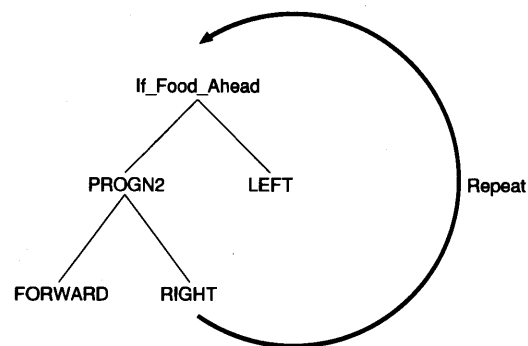


図 2.17: 解の性質 (問題 2) : 個体の持つ遺伝子列により 1 回の行動が決定されるので繰り返し同じ遺伝子列が用いられる

2.4.2 偶数関数生成問題とは

偶数関数生成問題と入力される変数の値 (0 または 1) において 1 の数が偶数ならば 1 を出力し, そうでなければ 0 を出力する問題である。

表 2.4 のような入力と出力が与えられた時にすべての入力に対して正しい出力を行なう図 2.18 のような関数が解の一つである。

偶数関数生成問題における解の性質

偶数関数生成問題は図 2.19 のように複数の偶数 2bit 関数の集まりで構成することができ, ウイルス感染が有効に行なえる場所が多いと考えられるので感染による宿主の

入力 1	入力 2	出力
0	0	1
0	1	0
1	0	0
1	1	1

表 2.4: 偶数 2bit 関数の出力

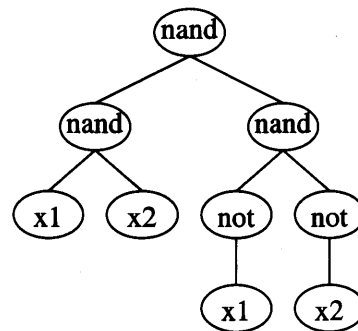


図 2.18: 偶数 2bit 関数の例

適合度上昇が期待できる。

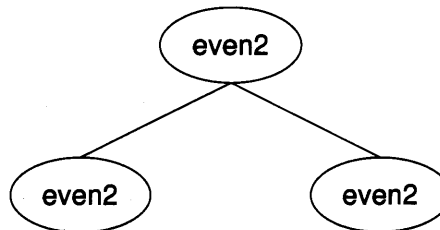


図 2.19: 偶数 6bit 関数

終端記号と非終端記号

終端記号，非終端記号として用いる記号は以下の表 2.5 とする。非終端記号は論理回路における論理ゲートを表し，終端記号は回路に入力される変数を表す。 n は入力される変数の数で， $n = 2$ ならば偶数 2bit 関数である。

非終端記号	and	or	nand	nor	xor	not	終端記号	x_1	\dots	x_n
引数	2	2	2	2	2	1				

表 2.5: 偶数関数生成問題における終端記号と非終端記号

適合度計算

宿主 i の適合度 $fithost_i$ は入力 X_{j1}, \dots, X_{jn} に対して出力される値 $f(X_{j1}, \dots, X_{jn})$ とその入力に対する正しい出力 Y_j との誤差の絶対値をすべての入力の組合せについて足したものとする。したがって適合度の値は小さいほど良く、適合度が 0 の個体が最適解とみなされる。一方、ウイルスの適合度は、宿主の適合度を改善した値の和とする。

$$fithost_i = \sum_{j=1}^{2^n} |Y_j - f(X_{j1}, \dots, X_{jn})| \quad (2.1)$$

偶数 6 ビット関数生成問題における実験パラメータ

実験パラメータは表 2.10 の値とする。VE-LGP は LGP にウイルス個体数とウイルス感染数、世代ギャップを加えたシステムとする。世代ギャップは感染を行なう世代間隔を表し、世代ギャップ = 3 ならば 3 世代ごとに感染操作を行なうものとする。これらの値は思考錯誤的に決定した。

宿主個体数	1000
ウイルス個体数	100
ウイルス感染数	3
交叉率	0.8
突然変異	0.1
世代ギャップ	3
世代数	200

表 2.6: 偶数関数生成問題における実験パラメータ

表 2.7: 宿主平均適合度と分散 (実験 1)

technique	LGP			VE-LGP		
crossover method	1or2	1	2	1or2	1	2
average fitness	4.0	5.9	5.3	2.6	3.7	3.6
dispersion	3.10	3.40	3.60	1.84	2.06	1.58

2.4.3 実験結果 1 (偶数 6 ビット関数生成問題)

表 2.7 は実験 1 の各手法における宿主の平均適合度とその分散の結果を示している。適合度においては各手法とも '1 or 2' の交叉方法が最も良い値を示し、次いで '2', '1' という順となった。また、提案手法はどの交叉方法においても従来手法より良い結果を得ることができた。分散においても提案手法の方が従来手法より小さな値を示しており、提案手法の方が安定した探索を行うことができたと言える。図 2.20 は各世代での最良宿主の適合度推移を示している。いずれの手法においても約 30 世代までは顕著な違いはなく比較的なだらかに進化の曲線を描いるが、その後は提案手法が進化の速度を早めている。ウイルスが有効な遺伝子列を持つためには取込元である宿主個体が進化することにより、有効な遺伝子列を持つことが必要である。したがって、LGP において '1 or 2' が最も良好な結果を得ることができたことからわかるように VE-LGP においても '1 or 2' が最も良い結果が得られたと言える。さらに、提案手法が良い結果を得られた要因には宿主とウイルスの共進化があると考えられる。すなわち、宿主間における交叉や突然変異による進化により生成された有効な遺伝子列をウイルスが取り込み、それを宿主間に伝播することにより、さらに宿主が進化するということである。また '1' では構造が同じ遺伝子列から交叉点を選択するので、線形表現された染色体の先頭部分の探索は初期の世代から積極的に行なうことができるが、遺伝子列の後方部分の探索は進化が収束して遺伝子列の構造が比較的一様になる世代の後半に行なわれるので、初期状態において遺伝子列の後方部分に比較的有益な部分解を持つと、その遺伝子列を持つ個体に初期収束してしまう可能性があると考えられる。一方、'2' では遺伝子列の先頭と後方からの探索を行っているので '1' に比べ遺伝子列全体の探索をバランス良く行なえるが、木構造として有益な意味を持つ部分を '1' に比べて破壊してしまう可能性も高い。したがって '1 or 2' はそれぞれの特長をうまく生かすことができたのではないかと考えられる。

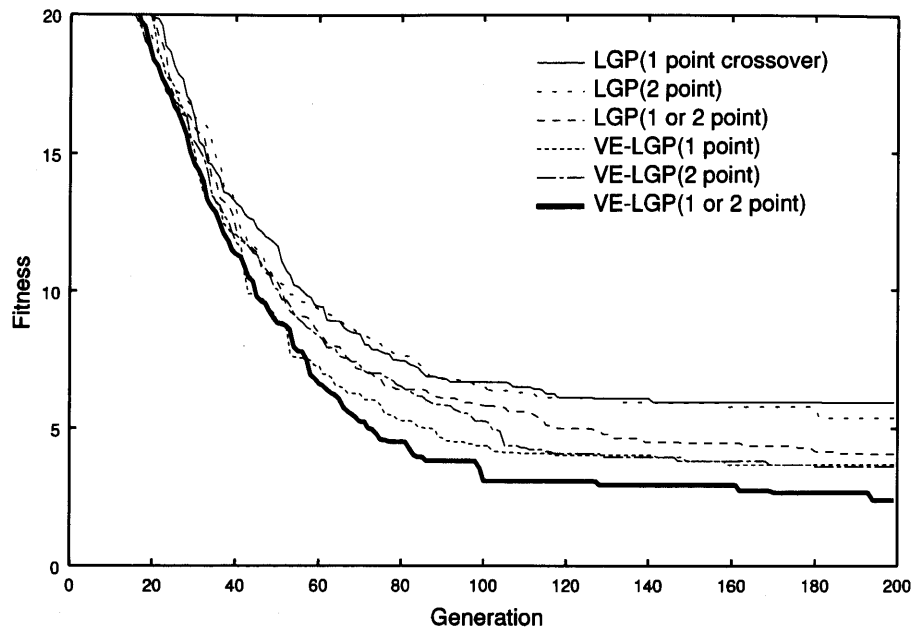


図 2.20: 最良宿主適合度の遷移 (実験1)

表 2.8 は宿主の平均個体長とその分散の結果を示している。平均個体長に関しては LGP, VE-LGP とともに交叉の方法に関わらずほぼ同様な数値を示しているが、VE-LGP の方がいくぶん大きな値となっており、分散に関しても提案手法の方が値が大きく不安定だと言える。図 2.21 は各世代での宿主の平均個体長の推移を示している。初期世代はどの手法においてもほとんど同様な曲線を描きながら上昇しているが、約 15 世代付近からそれぞれ独自の進化をしていることがわかる。また、交叉方法における推移を比較すると '1 or 2' では世代の後半以降でも変化し続けていることがわかる。これは解の探索が積極的に行われていると考えられる。さらに提案手法の '1 or 2' においては終盤までわずかながらも変化している。これはウイルス取込と感染により、交叉だけでは起こりえない遺伝子列の変化が生じ、新たな個体長を持つ宿主が生まれては選択、淘汰を繰り返すことにより、わずかながらも変化を続けていると考えられる。

図 2.22 は各世代でのウイルスの平均適合度の推移を示している。宿主の進化とともにウイルスが有効な遺伝子列を宿主間に広めることによってウイルスの適合度も上昇しているといえる。'1 or 2' では世代の中盤 (80 から 120 世代) において他の交叉方法より適合度が停滞しているにもかかわらず図 2.20 においては急激な進化が起こっている世代がある、このことから '1 or 2' の方が良い結果を得られる要因を示すことの手が

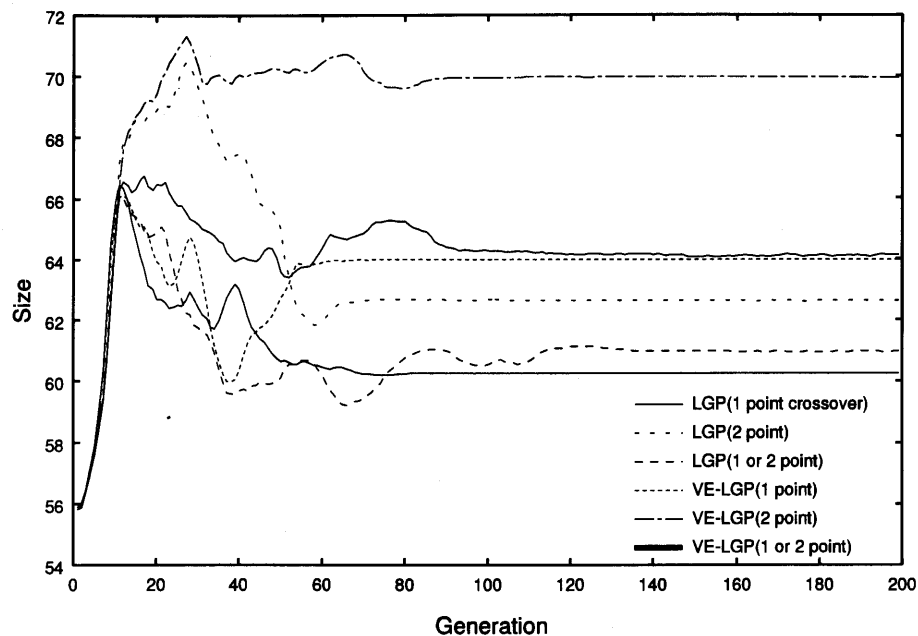


図 2.21: 宿主個体長の推移 (実験 1)

かりになるのではないかと推測されるのでさらなる解析が必要である。また、図 2.23 に適合度が高かったウイルスの例を示す。

2.4.4 Ant とは

Ant とは図 2.24 のような 32×32 のトーラス状の平面に 89 個の餌があり、エージェントが S をスタートして限られたエネルギーの範囲内で、できるだけ多くの餌を獲得するためのプログラムを生成する問題である。エージェントは向きを変えながら前方に 1 マスずつ進み、自分の前方に餌があるかどうかは知覚することができる。

表 2.8: 宿主の平均個体長と分散 (実験 1)

technique	LGP			VE-LGP		
crossover method	1 or 2	1	2	1 or 2	1	2
average gene size	61.2	60.2	62.9	61.8	64.0	69.9
dispersion	45.7	38.8	43.0	47.8	67.3	43.1

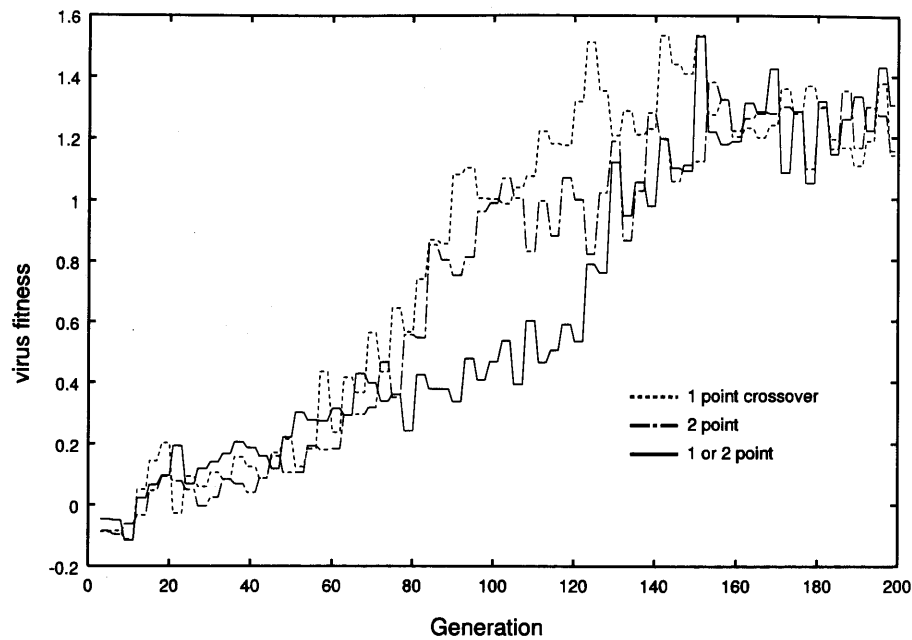


図 2.22: ウイルス適合度の推移 (実験 1)

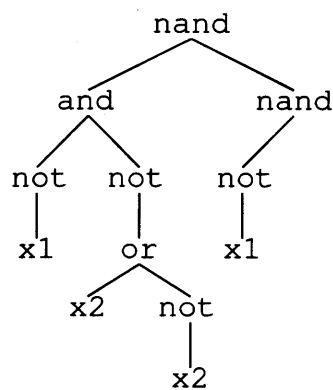


図 2.23: 高適合度のウイルスの例

Ant における解の性質

Ant では、解となる遺伝子列がエージェントの行動の度に繰り返し使用されるので、偶数関数問題とは異なり、複数の部分解により、解を構成することに明確な意味付けをすることはできない。

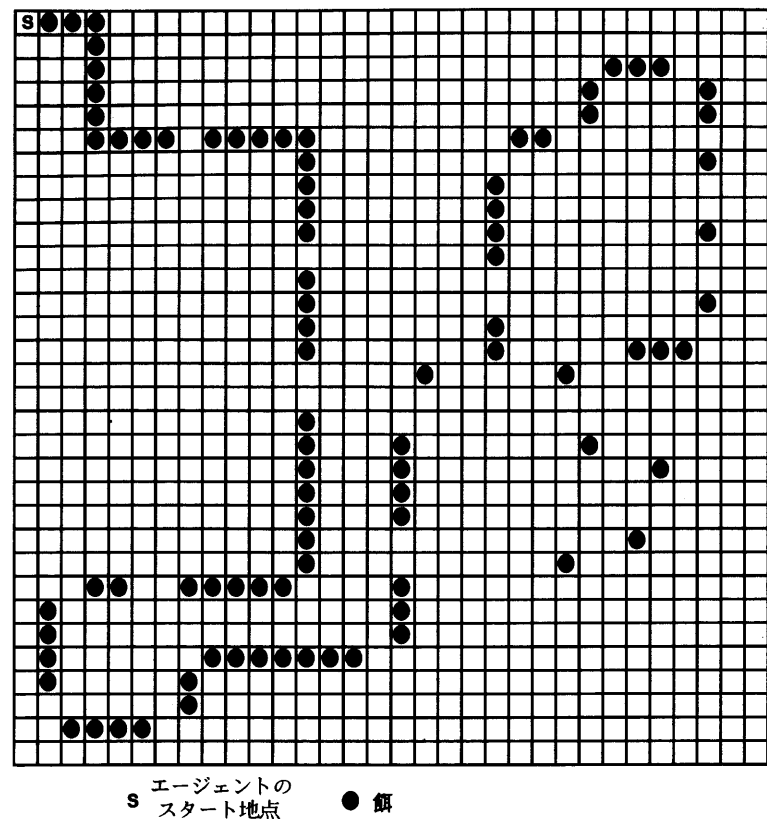


図 2.24: Ant の環境

終端記号と非終端記号

終端記号、非終端記号として用いる記号は表 2.9 とし、図 2.25 に行動を図示する。

非終端記号	引数	意味
<i>If_Food_Ahead</i>	2	目前のマスに餌があれば第 1 引数、 ない時は第 2 引数を実行
PROGN2	2	引数を順に実行

終端記号	意味
RIGHT	右に 90° 旋回
LEFT	左に 90° 旋回
FORWARD	1 マス前に進む

表 2.9: Ant における終端記号と非終端記号

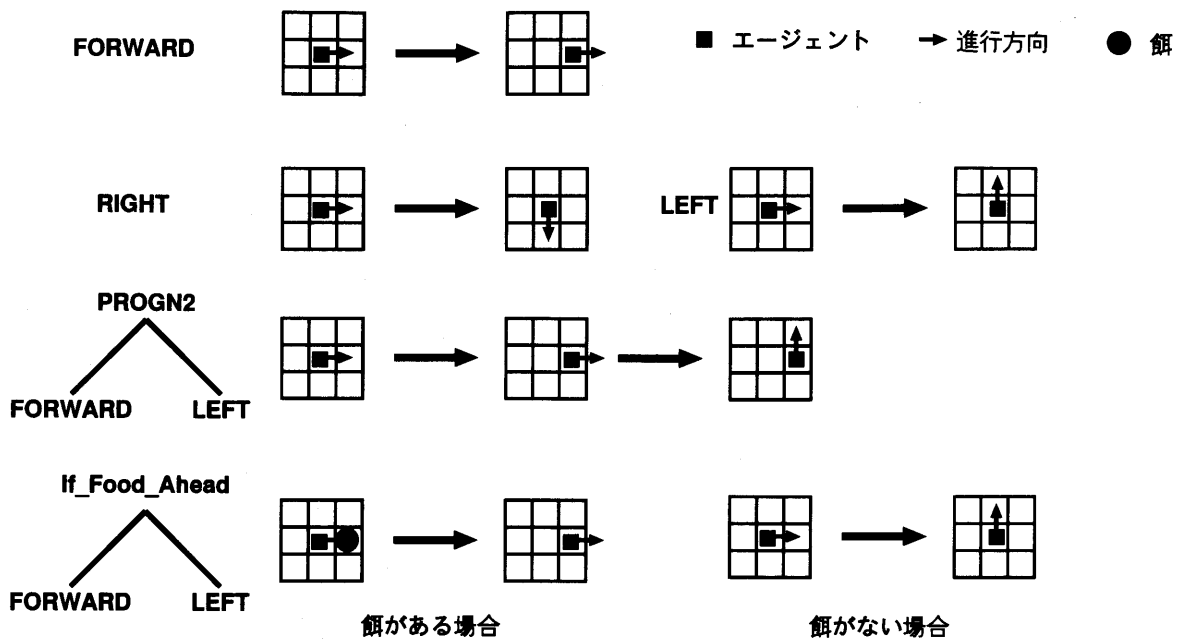


図 2.25: エージェントの行動

エージェントの行動プログラム

エージェントは図 2.26 のような行動プログラムをエネルギーが 0 になるまで繰り返す。この行動プログラムでは *If_Food_Ahead* により目前に餌がある場合は第 1 引数の *PROGN2* が選択され *FORWARD*, *RIGHT* の順に行動を行ない、再び *If_Food_Ahead* に戻って行動を繰り返す。

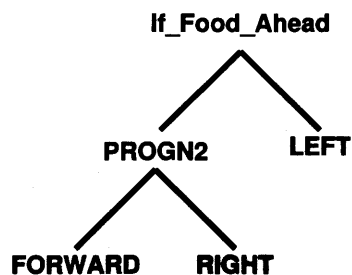


図 2.26: エージェントの行動プログラム例

適合度計算

エージェントは一回行動するたびにエネルギーを1消費し、エネルギーの残量が0になったら適合度が計算される。宿主 i の適合度 f_{ithost_i} は初期状態でフィールド上に置かれた餌の数 $FOOD$ と獲得した餌の数 $food_i$ の差で求められる。したがって適合度の値は小さい個体ほど解として適切とみなされる。餌は一度獲得されるとフィールド上から消されるので適合度が0の個体が最適解となる。ウイルスの適合度は、宿主の適合度を改善した値の和とする。

$$f_{ithost_i} = FOOD - food_i \quad (2.2)$$

Ant における実験パラメータ

実験パラメータは表 2.10 とする。VE-LGP は LGP にウイルス個体数、ウイルス感染数、世代ギャップを加えたシステムとする。これらの値は思考錯誤的に決定した。

宿主個体数	1000
ウイルス個体数	100
ウイルス感染数	2
交叉率	0.8
突然変異	0.1
世代ギャップ	5
世代数	100
初期エネルギー	400

表 2.10: Ant における実験パラメータ

2.4.5 実験結果 2 (Ant)

表 2.11 は実験 2 における宿主の平均適合度とその分散の結果を示している。各手法における結果に著しい相違はないが、'1' については各手法とも、他の交叉方法に比べて良好な解を得ることはできなかった。分散からも安定して有効な解が得られないということがわかる。提案手法と従来手法を比較すると、'1 or 2' と '2' では、やや提案手法の方が良好な結果を示しているものの、大きな差はない。しかし、'1' については提

案手法の方が良好な解が得られたと言える。この点はさらなる解析が必要である。図 2.27 は各世代での最良適合度の推移を示している。従来手法、提案手法ともに '1' を用いた手法では他の交叉方法に比べて極端に悪い結果となっているが、'2' や '1 or 2' ではほぼ同様な遷移となっている。結果としては提案手法のほうがわずかに良い結果を示しているが、この結果を見る限り解探索の性能に関してはほとんど同等と言える。しかし、これはウイルスを用いても改悪することがないということを示唆しているので、提案手法は適用問題における解の性質によらず使用できる可能性があることを示している。

表 2.12 は宿主の平均個体長とその分散を示し、図 2.28 は各世代での宿主の平均個体長の推移を示している。図 2.28 の初期世代においてはどの手法もほとんど同じ曲線を

表 2.11: 宿主の平均適合度と分散 (実験 2)

technique	LGP			VE-LGP		
crossover method	1or2	1	2	1or2	1	2
average fitness	15.0	26.1	15.7	14.7	20.5	15.1
dispersion	5.90	11.77	4.32	4.66	9.09	5.64

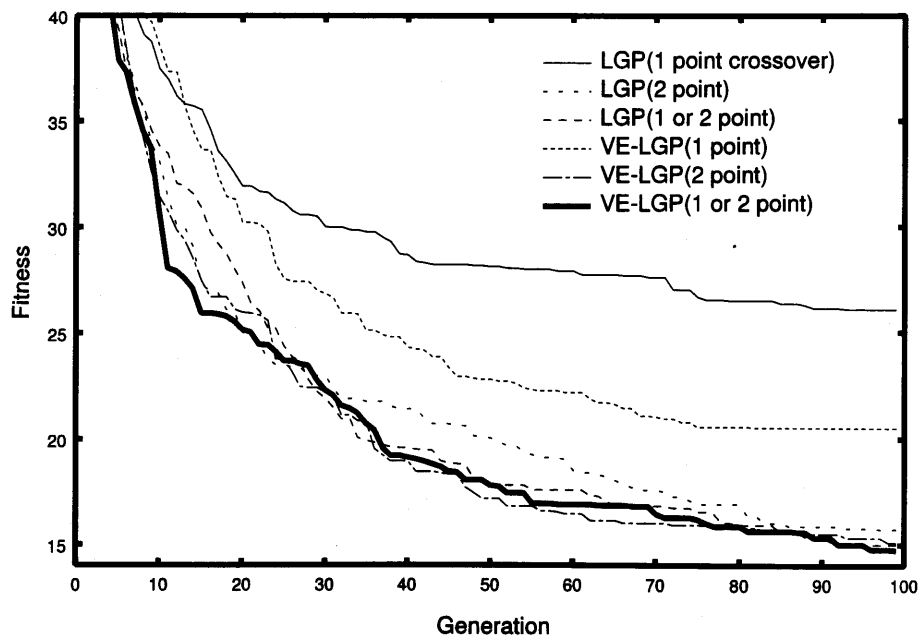


図 2.27: 最良宿主適合度の遷移 (実験 2)

描きながら上昇しているが、10世代を過ぎたところから極端にそれぞれ独自の進化をしている。平均個体長とその分散はともにそれぞれの手法でほとんど変わらない結果となった。

図2.29は各世代でのウイルスの平均適合度の推移を示している。ウイルスの適合度は世代を経るにつれて高い値にはなっているが宿主の平均適合度の上昇にはあまり貢献できていない。これは有効な遺伝子列を持っていない個々の宿主に対しての進化には貢献しているが、そのことが新たな最適解の探索には繋がっていないと言える。

表 2.12: 宿主の平均個体長と分散 (実験2)

technique	LGP			VE-LGP		
crossover method	1 or 2	1	2	1 or 2	1	2
average gene size	25.5	25.1	24.8	23.7	23.5	26.1
dispersion	20.8	23.9	15.0	24.4	22.1	15.4

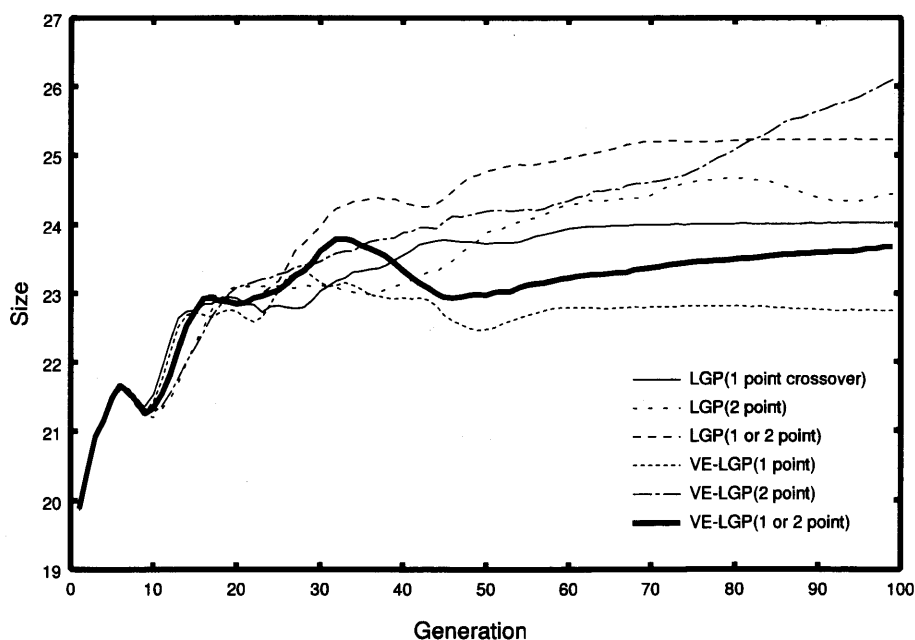


図 2.28: 宿主個体長の推移 (実験2)

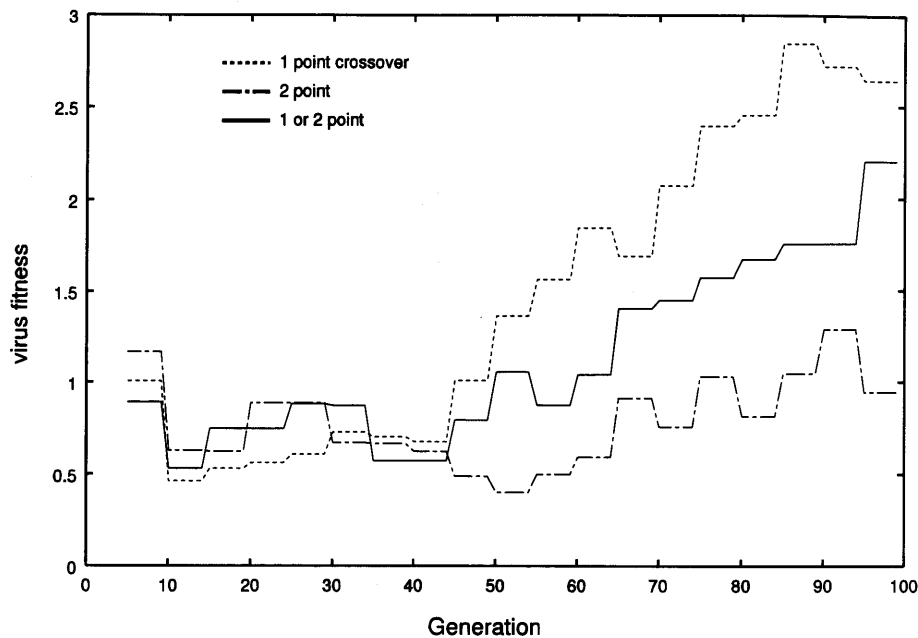


図 2.29: ウイルス適合度の推移 (実験 2)

2.5 まとめ

実験 1, 実験 2 の結果より提案手法は従来手法に比べ良好な解を得ることができたとと言える。実験 1 においては宿主とウイルスの共進化が効果的に行われたと考える。一方、実験 2 においては提案手法がやや良好という結果にとどまった。それは問題 1 は問題 2 とで求められる解の性質が異なり、部分解の集合により形成されるものでないことが原因であると考えられる。しかし、ウイルス個体による遺伝的操作が効果的に働かないと考えられる問題においても改悪することなく解の探索を行うことができたことは解の性質が未知の問題においても提案手法が適用できる可能性があることを示唆している。

また、図 2.21, 2.28 において、遺伝子列の長さが増大するブロートが抑えられているように見えるが、これはウイルスがスキーマを保持していた場合、遺伝子として機能しないイントロンを含むものより、イントロンを含まないものの方が個体長が短く、宿主の構造と一致する確率が高くなり、その遺伝子列を伝播させやすいのでブロートの原因となるイントロンが個体間に広まる前にスキーマのみを伝播させる確率が高いと考えられる。イントロンを含んでいる個体長が長い個体の場合には、宿主と構造が

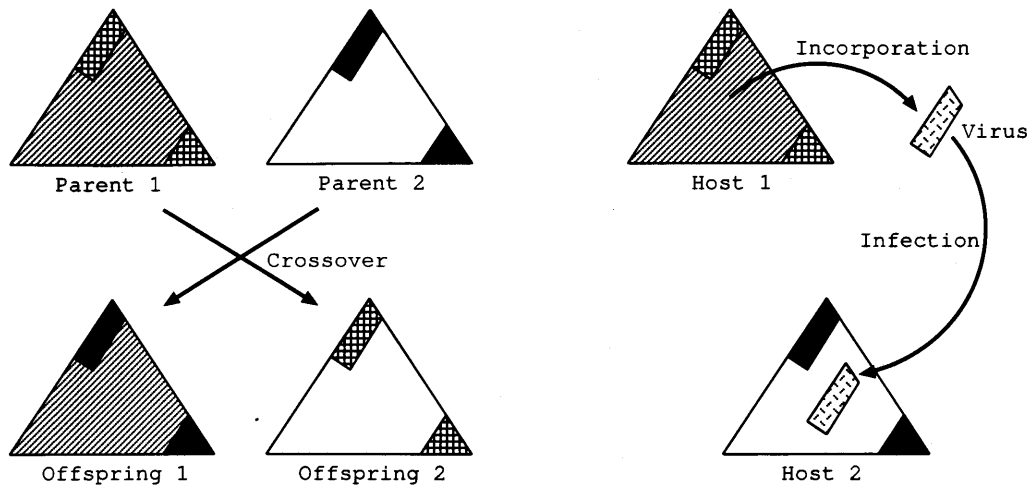


図 2.30: 有効遺伝子列の伝播

一致せずに感染が行われない可能性が高く、さらに、すでに個体間にスキーマが伝播した後には宿主の適合度に影響を与えないので、そのようなウイルスは淘汰される。また、ウイルスに感染されても宿主の個体長は変化しないので、個体長を増大させることなくスキーマを広めることができる。例として図 2.30 のように宿主 1 の遺伝子列の部分に有効な遺伝子列が含まれていると仮定すると、交叉による伝播では、有効な遺伝子列はイントロンもいっしょに個体間に広まってしまう可能性が高いが、感染による伝播では個体の大きさは変化しない。しかし、ウイルスが有効な遺伝子列のみを取り込む可能性はあるが、その確率的には非常に低いため、有効な遺伝子列が交叉により、ある程度個体間に広まった後にウイルスの働きによる進化が起これと考えられる。以上のことより、提案手法ではスキーマの伝播をウイルスで個体長を変化させず迅速に行うことにより、交叉による伝播が行われた子孫が生き残る優位性が低くなるためブロートを抑える働きができると考えられる。

2.6 おわりに

本章では線形表現を用いた遺伝的プログラミングにウイルス個体群を導入することにより有効遺伝子列の速やかな伝播を行なう手法を提案し、その有効性を確認するために偶数関数生成問題とエージェント行動制御問題 (ANT) によるシミュレーション実験を行い、その有効性を確認した。提案手法は従来手法に比べ偶数関数生成問題で

は平均的に最適な解を得ることが出来たが、問題の性質の異なる ANT では有効に働くことができなかった。さらに今後の課題としては解の性質が異なる問題への適用・解析、収束時間の比較、より大規模な問題における有効性を検討する必要がある。

第3章

ウイルス進化型遺伝的アルゴリズムにおける感染手法による個体進化の相違に関する一考察

遺伝的アルゴリズムでは交叉や突然変異などの遺伝的操作により、最適な解の探索を行うが、各遺伝的操作にはさまざまな手法がある。交叉においては一点交叉をはじめとして、二点交叉や一様交叉などが提案されているが、適用問題における解の性質により、その手法の解探索性能が異なるので、適用問題に応じて交叉方法を選択する必要がある。しかし、あらかじめ解の性質を事前に知ることが困難な場合や、交叉においても大域的な探索の要素と局所的な探索の要素があることから、進化の状況にあわせて交叉方法を変更した方が効果的な場合など、初期世代から交叉手法を固定するのではなく、適応的に交叉方法を選択する手法などが研究されている。以上のことより、部分解の局所探索を行うウイルス進化型の進化的計算においても、ウイルス感染と宿主の交叉とでは、遺伝子列に与える構造的変化は類似していることから、交叉と同様に進化の状況にあわせて方法を変更することで、より効果的な探索が行えることが期待できる。そこで本章では、進化状況に応じて感染手法を適応的に変化させる手法を提案し、従来手法と比較して良好な解が得られたことを述べる。

3.1 はじめに

遺伝的アルゴリズム (Genetic Algorithm : GA) はダーウィン進化論を模倣した確率的探索手法の一つで、さまざまな組合せ最適化問題において適用されてきた [30]。しかし、GA には早い世代で多様性が損なわれてしまう初期収束や個体間に有効な遺伝子列であ

るスキーマを効率的に広めるために適用する問題の特性に合わせた交叉や突然変異と呼ばれる遺伝的操作が行われる必要があるという問題点がある。また、進化論には今西進化論、断続進化論などの多くの進化論があり、その中の一つにウイルス進化論がある。ウイルス進化型遺伝的アルゴリズム (Virus Evolutionary Genetic Algorithm: VE-GA) は、ウイルス進化論に基づいた最適化手法の一つで、多くの研究が成されている [51] ~ [54], [56], [57], [60], [62], [64], [65]。VE-GA は適用問題の解候補となる宿主と、部分解となるウイルスを遺伝子列として持つ二つの個体群から成り、それらの相互作用による共進化により、大域的探索と局所的探索を行う。VE-GA ではウイルスの遺伝的操作である感染と取込により局所探索を行い、スキーマを個体間に高速に広めることができる。ウイルスは自身の遺伝子列を感染操作により宿主にコピーし、取込操作により宿主から部分解を取り込み、それを自身の遺伝子列とする。VE-GA では交叉による大域的な探索と、感染・取込による局所的な探索が行われる。宿主はより高い適応度を得るための解探索を行い、ウイルスは宿主の適応度を上げるための部分解の探索を行う。また、一般的な GA の遺伝的操作は選択・交叉・突然変異から成り、交叉はその中でも重要な役割を持ち、一点交叉、二点交叉、一様交叉などさまざまな手法が提案されている。一様交叉の方が一点交叉や二点交叉よりも親子の遺伝子的距離が大きいことが報告されているなど、それらの手法における解探索能力はそれぞれ異なり、適用する問題の性質や遺伝子のコーディング、個体数、進化の状況にあわせた適切な手法を選択することが重要であり、初期世代においては、遺伝子の多様性を広げることにより探索範囲を広げ、最適解が存在する領域の探索を行う必要がある。GA の施行中に適切な交叉方法を選択する手法が報告されている [78] ~ [84]。したがって、個体の遺伝子列を部分的に変化させるという点において交叉と類似している感染は、交叉同様に重要であると考えられ、宿主に感染する際に、世代途中で適切な感染手法を選択することにより、効率的な探索を行うことが期待できる。

本章では、感染手法による個体進化の相違を比較、進化の状況により適応的に感染手法を切り替える一手法である適応的感染手法を提案し、数値シミュレーションによる従来手法との比較を行ったことを報告する。以下、次節では VE-GA における個体群、第3節に感染による進化の相違と予備実験、第4節では適応的感染手法についての説明を行う。そして、第5節では実験結果について述べ、第6節で考察。最後に本章のまとめを行う。

3.2 VE-GA における個体群

VE-GA では宿主個体群とウイルス個体群の二種類の個体群を用いる。遺伝的操作は一般的な GA と同様に交叉と突然変異が宿主に適用され、ウイルスの遺伝的操作には感染と取込がある。感染操作は有効な遺伝子列を宿主間に広める役目を持ち、交叉が大域的な遺伝子列の探索を行う遺伝的操作であることに対して、感染は局所的探索を行う操作である。また、宿主の遺伝子列をウイルス自身の遺伝子列とする操作を取込と呼ぶ。ウイルス個体はランダムに選択された宿主の遺伝子列から、部分解として取り込む列をランダムに選択することにより生成され、その個体長は初期世代では短い世代が進むにつれて長くなるものとする。ウイルスの遺伝子列は宿主の遺伝子列からランダムに生成されるので、それが有効な部分解である場合とそうではない場合が生じる。それを評価するためにウイルスは宿主と同様な適応度および現在の遺伝子列を保持するための生命力とを持っている。適応度は感染を行った宿主における感染後と感染前の適応度差の総和として求められる。したがって、感染により宿主の適応度が上がれば、進化に貢献していることとなり、ウイルスの適応度も上がる。生命力は前世代での生命力と減衰率との積および適応度との和により求められる。したがって、宿主の適応度への影響が無くても減衰率により、世代を経るにつれて生命力が減少することとなる。生命力が0以下になれば、その遺伝子列は有効に働かないと評価され、取込により新たな遺伝子を持つこととなる。各個体群において、宿主は高い適応度を持つ解候補を探索し、ウイルスは宿主の適応度を上げる部分解を探索する役割を持つこととなる。

3.3 感染手法における進化の相違

GA において交叉は最も重要な遺伝的操作の1つであり、一点交叉、二点交叉、一様交叉などさまざまな手法が提案されている。しかし、一様交叉の方が一点交叉や二点交叉よりも親子の遺伝子的距離が大きいことが報告されているなど、それらの手法においては解探索の特徴はそれぞれ異なり、適用する問題の性質や遺伝子のコーディング、個体数、進化の状況にあわせた適切な手法を選択することが重要であり、GA の施行中に適切な交叉手法を選択する手法も報告されている [78]~[84]。したがって、遺伝子列の局所的な部分を入れ替える感染は、交叉と類似の働きを持ち、進化において重

要な役割と果たすと考えられる。また、ウイルスは感染により、自身の遺伝子列を宿主間に広めるが、その際には巡回セールスマン問題 (Traveling Salesman Problem: TSP) のように、単純に交叉などの遺伝的操作を行うと致死遺伝子が発生する問題においては、そのような遺伝子が生成されないように、感染が行われた部分以外にも遺伝子の変化が生じる場合がある。したがって、感染はスキーマの伝播だけでなく、感染が行われた部分以外のスキーマとなりうる重要な遺伝子列を破壊してしまう可能性が生じる。そこで、感染操作によりスキーマを広めるだけでなく、進化の状況にあわせて、感染が行われた部分以外の遺伝子列に多様性を持たせる手法や感染部分以外は極力変化が生じないようにする手法を取り入れることが考えられる。すなわち、初期世代にはスキーマの探索と多様性を広げる操作を行い、後の世代ではスキーマの維持と探索空間の局所的探索を行うことが解探索において有効に作用すると考えられる。VE-GA の感染手法は逆位や挿入、PMX などの手法が提案されており (図 3.1)、逆位感染はウイルスの遺伝子列にしたがって、宿主において対応する遺伝子座までの遺伝子列の並びを逆順にする手法である。挿入感染はウイルス遺伝子列先頭の遺伝子座と一致する宿主の遺伝子座から、順次ウイルスの遺伝子列を挿入し、挿入された遺伝子列と重複する宿主の遺伝子座を削除することにより、解としての整合性が保たれる。PMX はウイルス遺伝子座と一致する宿主の遺伝子座を交換する手法である。図 3.1 のように、宿主とウイルスの遺伝子列は同一のものであっても、感染の方法の違いにより、感染後の宿主遺伝子列が異なる場合が生じ、それが進化に影響を与えることが考えられる。図 3.1 を巡回セールスマン問題の巡回路で表現すると、図 3.2 のように、同一の巡回路を持つ宿主とウイルスの間で感染が行われても、逆位・挿入・PMX のそれぞれの手法で、感染後の巡回路に違いがあることがわかる。

3.3.1 予備実験

実験環境

始めに予備実験として、各感染手法を比較するために、逆位、挿入、PMX を単独で施行した時にどのような違いがあるのかの比較実験を行った。それぞれのシステムは、VE-GA において交叉方法を部分一致交叉とサイクル交叉、突然変異方法はランダムに選択した 2 点の遺伝子座を交換する手法と逆位を用いた。適用問題は、致死遺伝子の発生を回避するために、感染操作が行われる遺伝子列以外にも変化が生じる問題の一

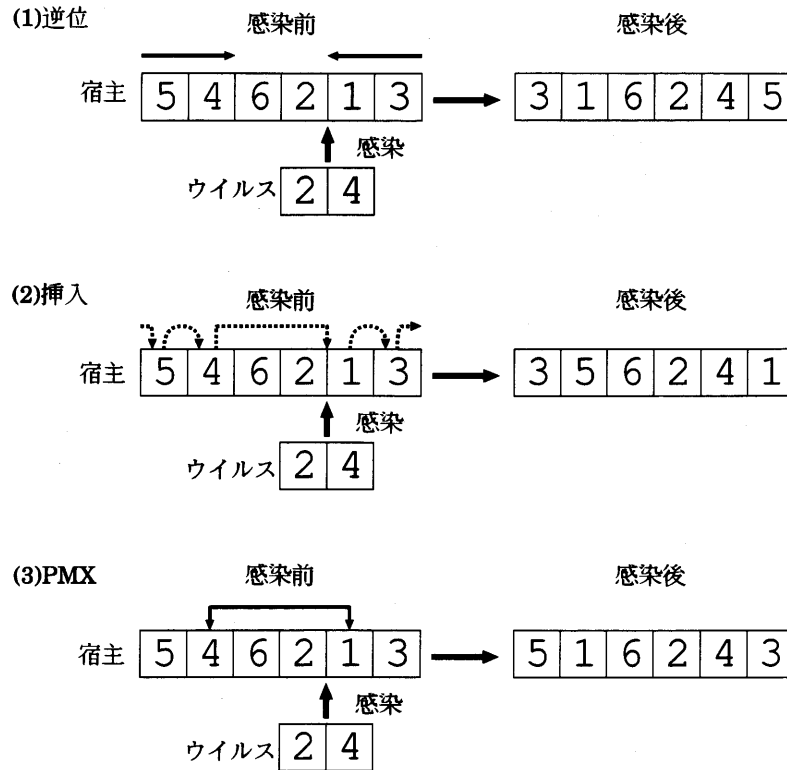


図 3.1: ウイルス感染

つである TSP を使い、都市数 25 および 52 において各 50 回の実験を行った。各パラメータは、宿主個体群の個体数は 25 都市では 200, 52 都市では 400, 交叉率 0.8, 逆位率 0.2, 突然変異率は 0.001 とし、ウイルス個体群の個体数は 25 都市では 10, 52 都市では 20, 感染個体数を 5, 世代ギャップを 10 とした。宿主個体数はウイルス個体数より十分大きくして、感染により多様性が失わないようにする。また、感染個体数とはウイルスが 1 世代で感染を行う宿主の個体数を表しており、本稿ではこのパラメータを 25 都市において試行錯誤的に決定し、52 都市においては宿主個体数に対する総感染数が 25 都市と同比率になるように設定した。全世代を通して感染個体数は一定であるが、宿主個体群の多様性が失われると、感染操作が行われても宿主遺伝子列に変化が生じないため、感染操作による宿主およびウイルスの適応度への影響は相対的に増減することとなる。また、世代ギャップはウイルス感染が行われる世代間隔で、5 の場合は 5 世代毎に感染が行われることを表す。予備実験において感染方法は全世代で変更を行わないものとする。

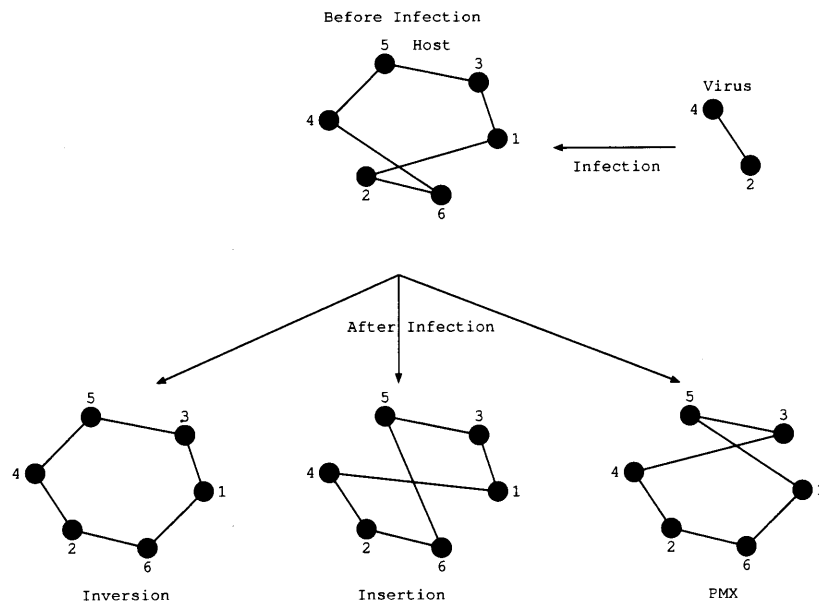


図 3.2: TSP におけるウイルス感染

実験結果

図 3.3 は 25 都市での各手法の 50 回の試行における最良宿主個体適応度の平均値推移を示している。この結果を世代における最良適応度の観点から見てみると、0 世代から 1000 世代付近までは PMX, 1000 世代付近から 1800 世代付近までは挿入, 1800 世代付近以降は逆位が最適な値を示している。また、挿入と PMX を比較した場合、世代の初期 (0 世代から 1000 世代付近) と終盤 (3600 世代付近から最終世代) においては PMX, 中盤 (1000 世代から 3600 世代付近) においては挿入の方が良好な値を示している。さらに、逆位と挿入、逆位と PMX との比較においても、適応度の推移を示す曲線の交差が生じる。これらのことより、感染手法が異なることで進化の過程が異なると言える。すなわち、宿主個体群の進化状況により、進化に貢献するための適切な感染手法があり、それを用いることにより効率的な解探索が行われることが期待される。52 都市の実験結果においても同様な傾向が見られた。以上のことより、感染手法の違いにより進化速度と得られる結果が異なることがわかった。また、適用する問題の規模が異なる場合でも各適応度推移の曲線が交差する現象が生じることも確認した。これらの相違の原因としては、TSP は各都市を 1 回のみ巡回する問題なので、致死遺伝子が生成されないように、ウイルスが感染を行う部分解以外の遺伝子列にも変化が生じる。し

たがって、感染の際には部分解以外の遺伝子列の解破壊が生じることとなる。各感染手法における解の破壊は、感染前の宿主遺伝子列に、ウイルス遺伝子列の並びが存在せず、かつ、感染の際に影響が及ばないように、各遺伝子座が十分に離れているような一般的な状態において、頻度が高い順にPMX・挿入・逆位となる。これらのことを図3.4に示すと、△はウイルス部分の遺伝子列と同様な並びとなる箇所、×はウイルス部分以外の遺伝子列に並びの変化が生じた箇所を表し、ウイルスが持つ遺伝子列が有効な部分解であると仮定した場合には、△は有効な変化であり、×は未知の並びとなるが、遺伝子列に変化が生じていることから、この議論においては遺伝子列が破壊されたものとする。感染が行われた時点での×の数が多い順にPMX・挿入・逆位となり、すなわち、それが破壊度の高い順となる。下図では、×の下に点線のある×が、感染後の×の数に該当し、PMXは5、挿入は3、逆位は2となる。また、図3.3が示すように、各感染手法の世代において得られる適応度は、初期世代から高い順に、PMX、挿入、逆位となっている。また、世代を経るにつれて、表3.1のように、各感染手法の世代において得られる適応度の大小関係が変化することがわかる。このように、感染における解の破壊と適応度には何らかの相関があると考えられる。それは、初期世代

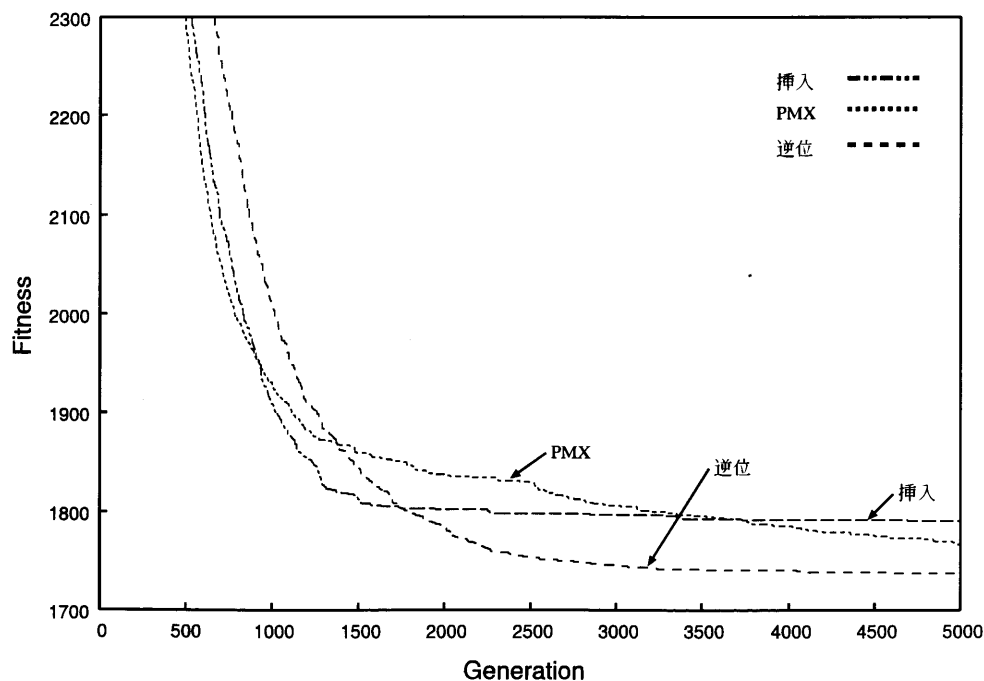


図 3.3: 予備実験 (25 都市)

においては有効な遺伝子列が少なく、感染によって有効な解が破壊される頻度は低い、したがって遺伝子列の頻繁な入れ替えが最も起こる PMX が解空間を広めることに働き、良好な解が得られ、中盤においては、PMX は探索された解の破壊を伴うので、相対的に挿入が良好な値を示し、破壊度が低い逆位もゆつくりとではあるが、良好な解の探索を行う。終盤では逆位がもっとも良好な結果となり、最終的には良好な順に逆位、PMX、挿入となっているが、解の破壊は新たな解候補の探索に貢献することができるので、より解破壊が起こりやすい PMX の方が挿入よりも良好な解探索が行えたと考えられる。しかし、これらの結果は適応度の平均値を示しているので、実験毎または適用する問題規模によっては進化の速度が異なり、単純に世代を区切り、感染手法を変更することはできないので、感染手法を自動的に変更する必要性が生じる。

3.4 適応的感染手法

前節の予備実験結果を踏まえて、動的に感染手法を選択する一手法を提案する。提案システムでは、一度に宿主に感染できる個体数は、式 (3.1) のように一定数であり、

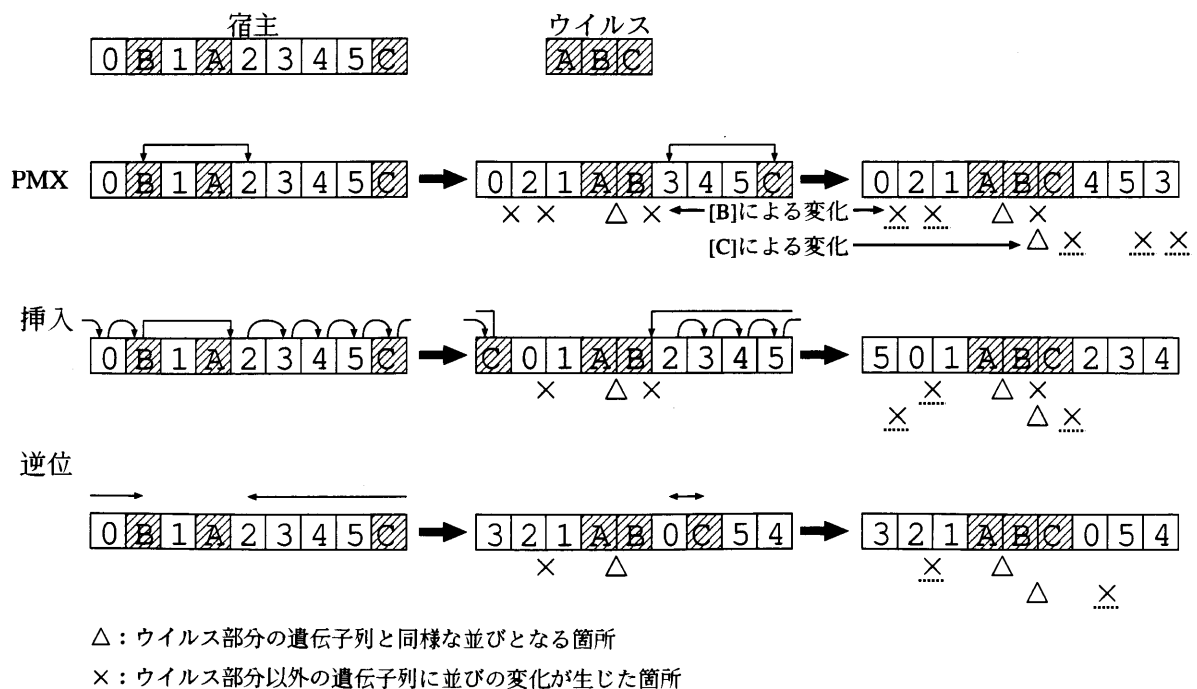


図 3.4: 各感染手法における破壊頻度

表 3.1: 感染手法における世代と適応度の関係

適応度	世代：0 ～				
低 ↑ 高	逆位	逆位	PMX	PMX	挿入
	挿入	PMX	逆位	挿入	PMX
	PMX	挿入	挿入	逆位	逆位

逆位, 挿入, PMX のうち 2 種類の手法を感染手法 1, 感染手法 2 のペアとし, そのペアとなった感染手法で感染を行うものとする. inf_s が感染する宿主個体数の和を表し, inf_1, inf_2 はそれぞれ感染手法 1, 感染手法 2 による感染の回数, inf_r は感染手法 1 または感染手法 2 がランダムに選択される回数を表している. inf_r は一つの感染手法に偏ることなく, 適切な感染手法に切り替えるための役割を持つ.

$$inf_s = inf_1 + inf_2 + inf_r \quad (3.1)$$

$$(0 \leq inf_1, 0 \leq inf_2, 1 \leq inf_r)$$

適応的感染の概念図を図 3.5 に示す. この図では, 感染手法 1 が 1 個体, 感染手法 2 が 2 個体, ランダムが 1 個体に適用されていることを示している. 各感染手法の感染個体数は図 3.6 のアルゴリズムに従って, ウイルスの適応度に応じて決定される. それぞれの感染手法時におけるウイルスの適応度は, 感染手法 1 は $fitvirus_inf1$, 感染手法 2 は $fitvirus_inf2$ として表している. 感染手法 1 により感染した時の方が適当度が高い場合は, 感染手法 1 による感染回数を増加させ, 感染手法 2 による感染回数を減少させる. 感染手法 2 の方が高い場合は, その逆となる. 既存研究における GA 施行中に適切な交叉方法を見つける手法と提案手法との相違としては, ウイルス個体群によって有効な部分解が, より適切な感染手法により宿主間に高速に伝播可能なことが挙げられる.

3.4.1 アルゴリズム

本システムにおける全体の基本的なアルゴリズムは 1 世代において部分的に個体の入れ替えが行われる定常状態 GA (steady state GA:SSGA) にウイルスの遺伝的操作を追加した図 3.7 のようになっており, 各操作の詳細については以下のようになっている.

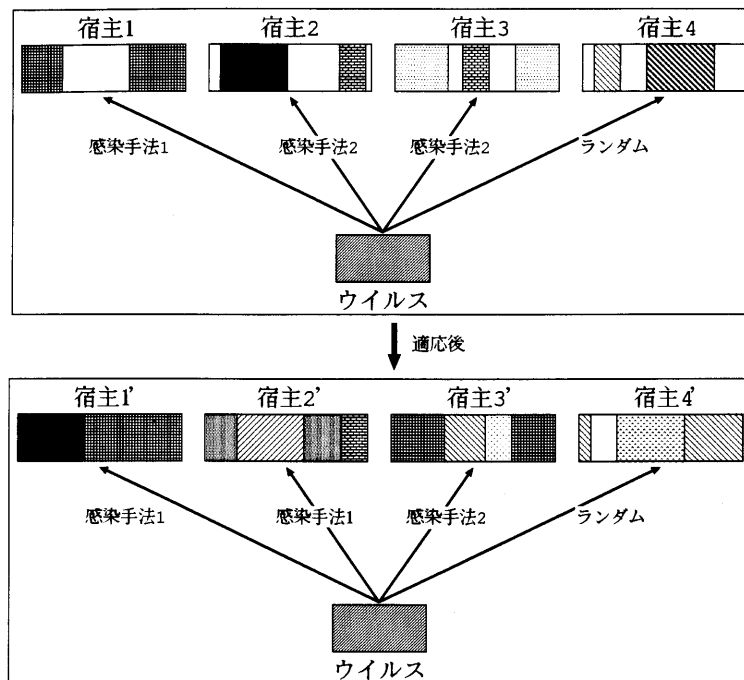


図 3.5: 適応的感染の概念図

```

if(fitvirus_inf1 > fitvirus_inf2 &&
   inf_1 < inf_s - inf_r){
    ++inf_1;
    --inf_2;
} else
if(fitvirus_inf1 < fitvirus_inf2 &&
   inf_2 < inf_s - inf_r){
    --inf_1;
    ++inf_2;
}
    
```

図 3.6: 感染個体数変更アルゴリズム

- Initialization
 - － 初期化. ランダムに生成された遺伝子列を持つ宿主と宿主の遺伝子列の一部を持つウイルスを生成
- Selection, Crossover, Mutation
 - － 選択, 交叉, 突然変異

- Infection
 - － 宿主をランダムに選択して感染
- Calculate
 - － 宿主の適応度, ウイルスの適応度, 各手法の感染個体数の値を求める
- fithost, fitvirus
 - － 宿主の適応度, ウイルスの適応度
- inf_s
 - － ウイルスの適応度により変更される各手法の感染個体数
- life
 - － ウイルスの生命力
- Incorporation
 - － 宿主をランダムに選択して取込
- Replacement
 - － 淘汰
- Termination-condition
 - － プログラムの終了条件. 設定された世代まで実行されたら終了

本アルゴリズムでは, 宿主個体群は一般的な GA と同様に交叉により大域的な探索を行い, 突然変異により解空間の拡大を行う. 交叉や突然変異を用いず, ウイルスの遺伝的操作のみで解探索を行う手法も提案されているが, ウイルス遺伝子列は宿主からの取込により生成されるので, ウイルスが有効な解を保持するためには宿主が有効な解を保持していることが前提となる. したがって, 本稿では宿主個体群において交叉および突然変異を用いる手法を採用した. また, ウイルス個体群は淘汰が行われた宿主, すなわち, 有効な遺伝子列を保持することによって生き残った宿主から, 取込により部分解を取得し, さらに感染を行うことで宿主の適応度に変化が生じた度合いにより, その部分解の評価される.

```

Begin
  Initialization
  Repeat
    Selection
    Crossover
    Mutation
    Infection
    Calculate{
      fithost
      fitvirus
      inf_s
      life
    }
    Replacement
    Incorporation
  Until Termination-condition = True
End.

```

図 3.7: システム全体の基本的なアルゴリズム

3.5 実験

本論文では提案システムを SSGA と各感染手法 (単独の感染手法, 適応的感染手法) の VE-GA とで比較実験を行った。単独の感染手法は, 逆位, 挿入, PMX, 適応的感染手法は, 逆位&挿入, PMX&逆位, 挿入&PMX である。A&B のように, & で表現されている手法は, 感染手法 A と B による適応的感染手法を表している。また, パラメータの設定は, 式 (3.1) における初期値は inf_s を 5 とし, $inf_1 = 2, inf_2 = 2, inf_r = 1$ とし, その他のパラメータについては予備実験と同様の値を用いた。

3.5.1 結果

表 1 に各実験の結果を示す。各値は 25 都市および 52 都市における, 50 回の試行の平均距離, 最短距離, 最長距離の値を示している。適応的感染手法は SSGA や組合せにおける各単独による感染手法と比較して良好な結果が得られている。適用問題は TSP なので, 数値が低いほど良好な解ということになる。また, 図 3.8 は 25 都市における各手法の最良適応度の平均推移を示している。SSGA は他の手法に比べて進化の速度が早い, 初期収束に陥っており, 最終的に得られる値は他の手法に比べて悪いものとなっている。一方, 他のウイルスを用いた手法では初期世代において, SSGA と比較して進化の速度は遅く, 急激な進化は起こらないが, 断続的な進化が起こっている。

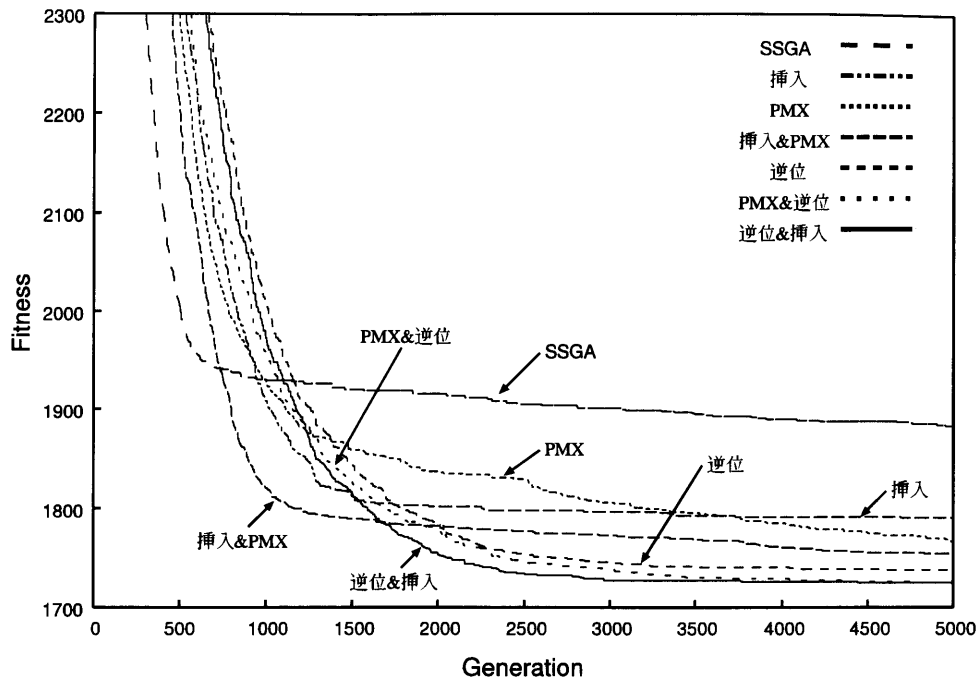


図 3.8: 実験結果 (25 都市)

ことがわかる。また、適応的感染手法においては各組合せにおける単独の感染手法と比べて改悪することなく、進化速度および探索される解においても良好な解が得られている。

表 1. 実験結果

		SSGA	逆位	挿入	PMX	逆位&挿入	PMX&逆位	挿入&PMX
25 都市	平均	1882	1737	1790	1766	1725	1725	1753
	最短	1703	1703	1703	1703	1703	1703	1703
	最長	2198	1839	1940	1953	1811	1894	1934
52 都市	平均	3241	2997	3150	3137	2991	2995	3102
	最短	3015	2877	2947	2915	2860	2923	2860
	最長	3602	3229	3500	3473	3228	3335	3200

3.6 考察

図 3.9 は各適応的感染手法における任意の試行について、A&B における A の感染回数の推移を示している。ただし、その他の試行においても同様な変動傾向が見られた。

感染個体数は一定なので、A&BにおけるBの感染回数は、Aとは対称な推移を辿ることとなる。

感染回数は挿入&PMXにおいては挿入の増加、PMX&逆位においてはPMXの減少となっており、共にPMXが減少しているという結果となった。また、逆位&挿入においては、0世代から200世代、500世代から1300世代、2500世代から最終世代においては逆位の感染回数が増加傾向にあり、その他の世代においては減少、すなわち挿入が増加傾向にあることがわかる。このことから逆位&挿入においては、各感染手法の感染回数は増加と減少と繰り返していることがわかる。これらのことより、逆位や挿入が有効に働き、PMXは有効に働いていないように見えるが、表1や図3.8の結果からは、PMXは挿入よりも良好なことや単独の逆位や挿入のみよりもPMXを組合せた手法の方が良好であることから、PMXにより探索能力が向上していると考えられる。

各適応的感染手法における適応度と感染回数を重ねたものを図3.10から図3.12に示す。図3.10において、初期世代では挿入が進化速度の点においては優位な適応度を示しており、最終的には逆位の方が良好な結果を示しているが、適応度の推移と感染回数の推移との相関性は見られない。また、全体的には逆位の感染回数が多い傾向にあ

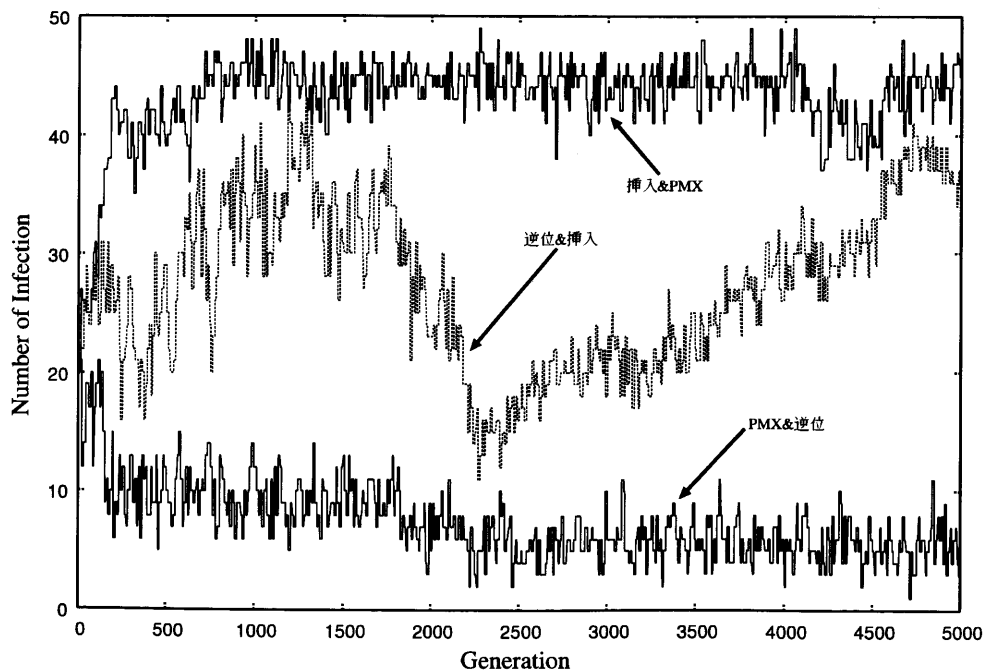


図 3.9: 感染回数の推移 (25 都市)

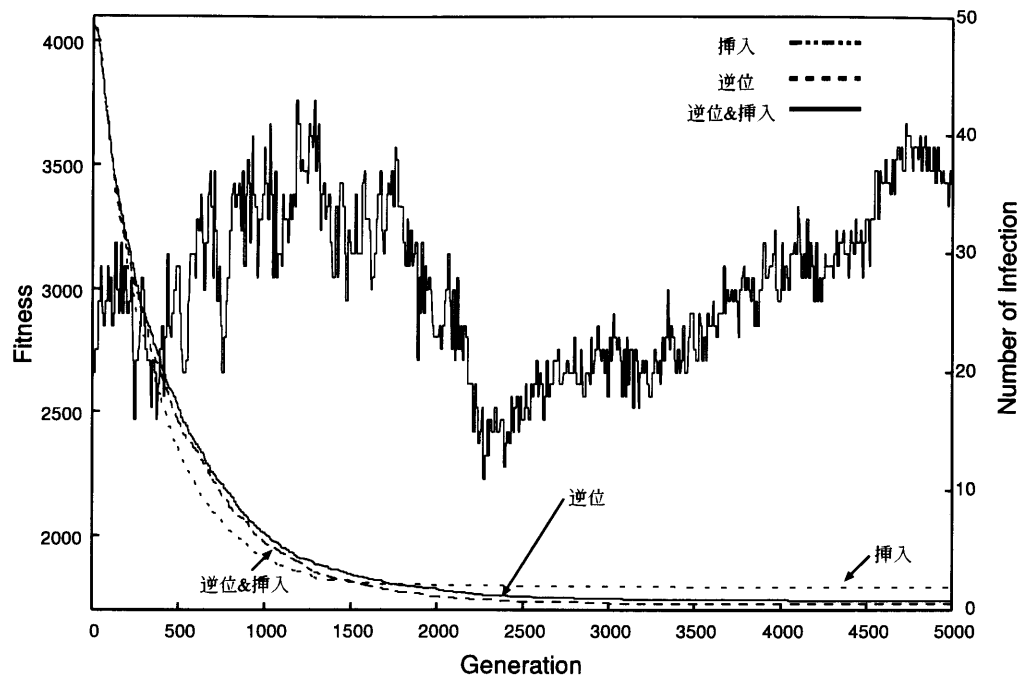


図 3.10: 適応度と感染回数の推移 (逆位&挿入)

るが、挿入の感染回数が増加する部分もあるので、逆位による進化が停滞する場面では挿入により新たな領域への解探索が行われ、逆位&挿入が良好な結果になるのではないかと考えられる。

図 3.11 において、探索結果は PMX よりも逆位の方が良好であるが、PMX に関しては初期世代の進化速度の点では逆位よりも優位であるので、初期世代においては PMX の感染回数が多くなると考えられるが、そのような結果にはならなかった。また、図 3.12 において、解探索の初期世代と終盤世代では PMX、中盤世代では挿入が優位な結果を示しているので、初期世代においては PMX の感染回数が多くなると考えられるが、図 3.11 と同様に PMX の感染回数は減少する傾向にあった。また、図 3.13 は、PMX&逆位における PMX の感染回数に対する適応度上昇個体の割合の推移を示しており、初期世代と最終世代付近において割合が上昇していることがわかる (挿入&PMX においても同様な傾向となった) が、割合としては概ね 50% 以下となるため、PMX はどのような個体に対しても平均的に良好な進化を促すのではなく、一部の個体に対して良好に働いていると考えることができる。

以上のことより、単独の感染手法だけで進化を行うよりも、複数の感染手法を組み

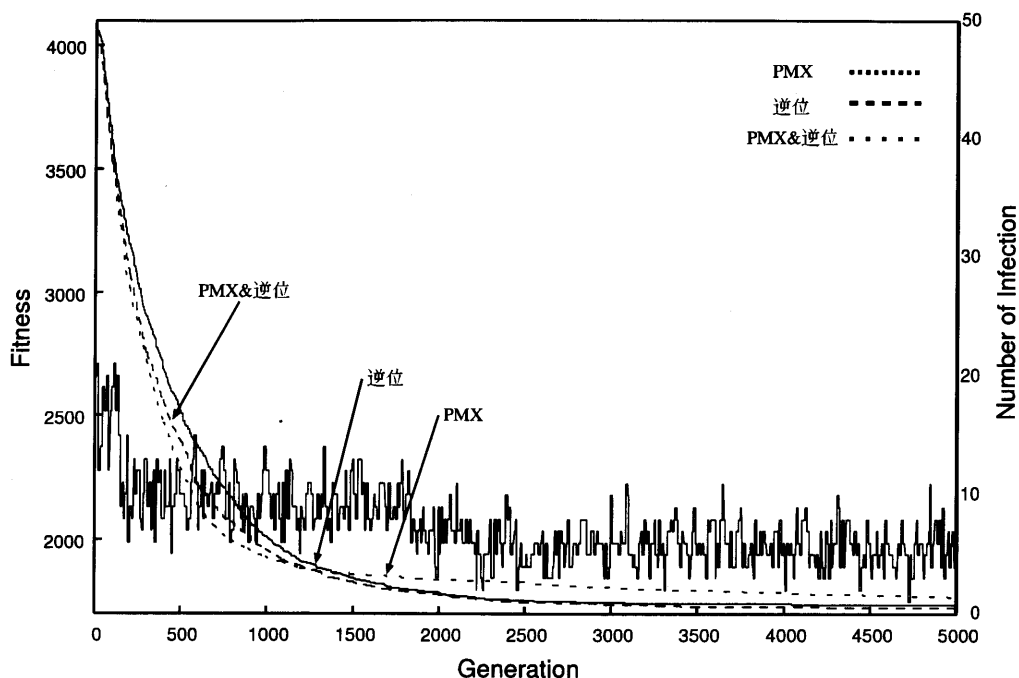


図 3.11: 適応度と感染回数の推移 (PMX&逆位)

合わせることで、良好な解が得られることがわかった。また、PMX を用いた場合には、逆位や挿入の適応度上昇個体割合も同様な上昇傾向を持つことから、PMX は他の感染手法の進化を促進する働きがあることがわかった。

3.7 まとめ

本章では、感染手法による個体進化の比較、進化状況に応じて適応的に感染手法を選択する共進化型遺伝的アルゴリズムにおける一手法を提案し、巡回セールスマン問題によるシミュレーション実験を行い、感染手法による個体進化の相違と感染手法を組み合わせることにより従来手法と比較して有益な解が得られることを確かめた。また、感染手法の違いにより宿主の進化に与える影響が異なることや適応的感染手法のように複数の感染手法を組み合わせたアルゴリズムは、単独の感染手法を用いた場合とは異なる進化をすることがわかった。このことより、さらに適切かつ高速に感染手法を切り替えることが出来れば、より早い世代により良好な解の探索を行うことが期待できると考えられる。また、さらに表1の実験結果における適応度の優位性が低い

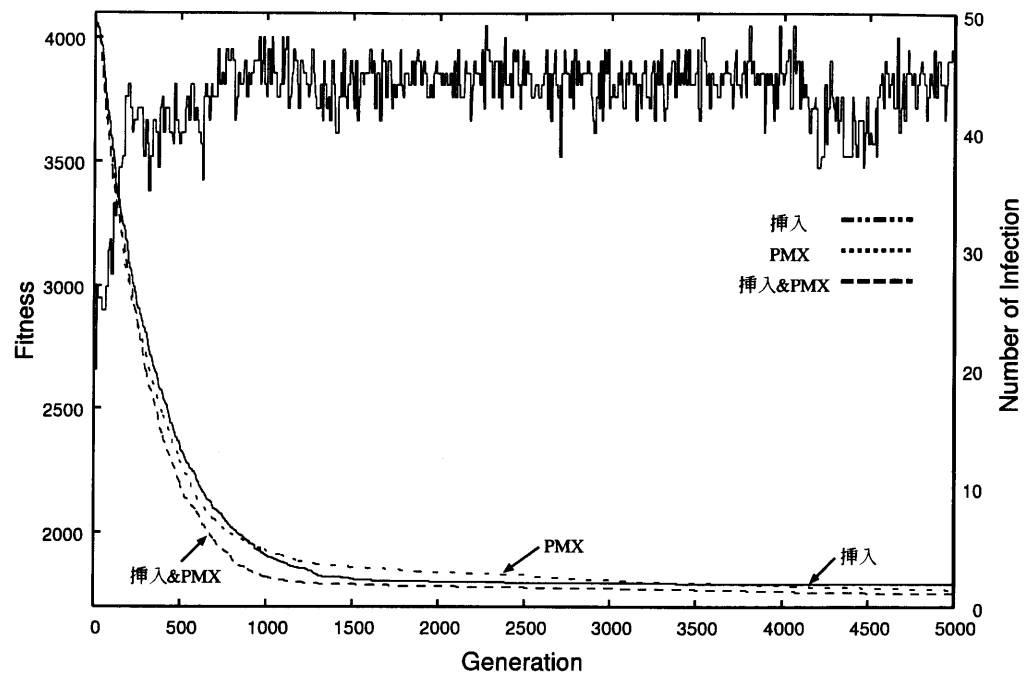


図 3.12: 適応度と感染回数の推移 (挿入&PMX)

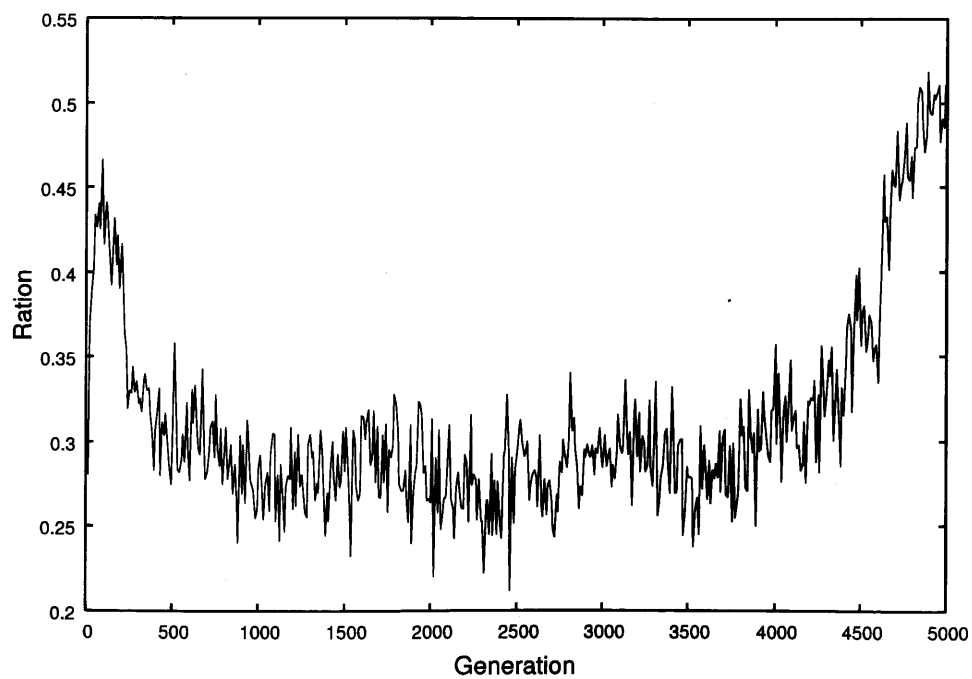


図 3.13: PMX における適応度上昇個体割合の推移

手法も、図3.8が示すように感染手法の違いは進化の速度に影響を与えることから、適用問題の条件によってその優位性は異なる。例えば、実時間内で有用な経路を探索する場合などの、時間的制約がある問題では、早い世代で有効な解が探索できる手法に優位性がある。したがって、最終的に探索される適応度による評価だけでなく、解探索までの速さで評価することも考えられる。

3.8 おわりに

本章では2種類の感染手法における適応的な手法選択についての実験を行ったが、3種類による手法も考えられる。そこで、3種類の感染頻度を各2回に固定し50回の試行した結果、各都市における平均、最短、最長は、25都市では、1831, 1703, 2079, 52都市では、3150, 2887, 3384となった。結果としては、他の感染手法と比べて優位な結果とはならなかった。これは解の破壊度が大きい手法と小さい手法が常に適用されることになるため、宿主の適応度を改悪する感染手法でも、感染手法の変更が行われず、ウイルスはその遺伝子列を別の遺伝子列に入れ替え、再び改悪する手法での感染を行い続けることが原因であると考えられる。したがって、3つの感染手法を適応的に変更することが必要となるが、どのタイミングで、いずれの感染手法を選択するのかが課題となる。その方法としては、

(1)PMX, 挿入, 逆位の順に頻度を高くする手法: 各手法を独立して適用した結果からは、初期世代よりPMX, 挿入, 逆位の順に適応度が高くなっているため、その順に頻度が増加するようにする。

(2) 他の2種類の手法をランダムに選択する手法

(3) 他の2手法の中で最も適応度の向上されているものを選択する手法

(4) 初期世代から、有効に働いている手法の情報を蓄積し、統計的に有効だと判断されるものを選択する手法

などが考えられるが、各手法を実験により比較する必要がある。

さらに今後の課題としては、収束までの時間や他の最適化手法との比較、異なる問題規模のTSPや他の最適化問題への適用などがある。

第4章

結論

進化的計算手法である共進化アルゴリズムにおいて、適用する問題の解表現および、その探索を行う局所的探索手法は良質な解を求める上で重要な要素である。また、進化的計算手法は最適化問題などに適用されることが多いが、最適化問題は工学的な分野に留まらず、さまざまな分野において見られる問題であり、そのため、効率的な解探索手法の確立は、多くの分野に貢献することが可能な研究分野の一つである。

第1章において、最適化問題に対する最適化手法であるメタヒューリスティクスの一つの手法である進化的計算について詳述した。まず、最適化問題におけるメタヒューリスティクスの重要性について述べ、生物進化説の枠組みを模倣した進化的計算手法が、適用問題における最適化や探索を実用的な時間内に行うことに有効であることを説明した。さらに、効率的に探索を行うための適用問題に対する解の表現手法や局所探索手法について述べ、コンピュータの計算能力の向上などのさまざまな要素を考慮した手法が要求されていることを説明した。

第2章においては、遺伝的プログラミングによる進化的計算において部分解を持つ個体を導入することによる共進化アルゴリズムを提案した。適用問題における解の表現に木やグラフなどの構造的なものを扱うことができる遺伝的プログラミングには、構造を持つことによる計算量の増加などの課題があるが、構造的に表現された解を線形で表現した遺伝的プログラミングにおいて、部分解を持つ個体を導入した共進化型遺伝的プログラミングを提案し、効率的な解探索を実現することができたことを確認した。

第3章では、共進化アルゴリズムにおいて部分解を持つ個体における遺伝的操作手法の適応的に変更するアルゴリズムを提案した。遺伝的アルゴリズムでの交叉の違いによる解探索性能の相違が生じることから、共進化アルゴリズムにおける適用問題の

部分解を持つウイルス個体の遺伝的操作である感染操作において、進化状況に応じて感染手法を適応的に変化させる手法を提案し、従来手法と比較して良好な解が得られることを確認した。

以上のように、本論文では、メタヒューリスティクスの一つである進化的計算手法における、その代表的な手法である遺伝的プログラミングおよび遺伝的アルゴリズムの共進化手法において、解表現と局所的探索手法について議論を展開し、適用問題における解の探索性能に関する提示が行えたと考える。今後の研究の展望としては、今後益々、最適化などを要求される課題は大規模・複雑化すると考えられるため、実用的な解を求めるためのより柔軟でロバストな手法の構築を目指す考えである。

最後に、今後の応用が期待され、最適化研究において考慮に入れておくべき科学技術についての説明をし、結びとしたい。

コンピュータの計算能力の向上により、これまで人間の能力では困難な問題に対して多くの解決が成されてきた。しかし、科学の発展によるさらなる問題の発見や、社会システムの複雑化などによるシステム間の相互作用の発生などにより、生命科学・生態系・経済などまだまだ多くの解決を要する分野がある。したがって、現代社会において応用が期待される分野にも視野を広げた手法の開発が必要となる。

以下に応用が期待される分野と研究課題について代表的なものを挙げる。

応用が期待される分野には以下のようなものがある。

- 多目的最適化問題

現実世界において人の手による答えの導出が困難な問題の多くは多目的最適化問題であり、その解を効率的に求めることは社会に貢献できる応用が期待できる分野の一つである。進化的計算は多点探索であるため、多目的最適化問題の解であるパレート集合を表現することが容易である。

- ロボットシステム

ヒューマノイドに代表されるロボットシステムでは、ロボット自体が大規模で複雑なシステムであるだけでなく、求められるタスクも従来の産業用機械のような決まったものではなく、動的に環境や目標が変化する状況に対応できるものが期待されている。このようなロボットシステムを人間がゼロから設計しても意図した通りのシステムを構築することは困難である。そこで進化的ロボティクスなど、生命進化のプロセスを模倣した進化的計算の応用が期待されている。

- 生命科学

生命科学分野の発達により、生命システムの仕組みについて多くの解明が成されているが、それらは多くの要素の相互作用から成りたっているため容易には解析出来ない。タンパク質の二次元構造予測や代謝系ネットワークの推定などにおいて進化的計算が応用されている。

研究課題としては以下のようなものがある。

- 並列化

進化的計算は複数の解個体による多点探索のため、比較的容易に並列化を行うことが可能である。並列化には、適応度計算の並列化・遺伝的操作の並列化・問題分割による並列化などが挙げられるが、適用問題に応じた設計をする必要がある。

- ハイブリット化

メタヒューリスティクスによる最適化問題における解探索では、大局的探索と局所的探索のバランスをとることが重要である。そこでメタヒューリスティクス同士を組み合わせるハイブリット化することにより、他の手法の長所を取り入れ、適用手法の短所を補うことにより、効率的な探索手法の設計を考える必要がある。

- 対話型進化的計算

解に対する評価関数の定義が困難な問題において、人間がその評価に対話的に加わることで淘汰の行う手法であり、デザインや音楽などの評価が主観的な芸術分野で用いられることが多いが、多目的最適化問題における解の選択など、絶対的な解評価が行えない問題への適用が考えられる。

- パラメータ設計

進化的計算では、符号化・交叉・突然変異・淘汰などの手法の選択の他にも、個体数・交叉率・突然変異率などの数値的なパラメータも存在し、パラメータ設計の自由度が高いことは進化的計算の利点ではあるが、その設定値により適用問題の解決能力に大きな影響を及ぼすこともある。したがって、パラメータ設計の方法論についても、実システムへの適応においては重要である。

- 理論的解析

進化的計算における理論的解析は少なく、代表的なものは有効な遺伝子列であるビルディングブロックが進化により組み合わさることにより最適な解探索を行われるというスキーマ定理である。また近年、すべての評価関数に適用できる効率的なアルゴリズムは存在しないという NFL 定理 (No Free Lunch Theorem) なども報告されており、進化的計算の理論的解析も必要である。

進化的計算の研究においては、効率的な解探索だけでなく、その設計パラメータの柔軟性を活かしたシステム構築が必要である。すなわち、他の分野との相互作用や適用問題において要求される解の信頼性などを考慮した設計をする必要がある。

謝辞

私は、名古屋工業大学工学部知能情報システム学科 伊藤英則教授の研究室に在籍し、情報通信システムに関する研究に従事してまいりました。その間に、多くの方々から数え切れないほどのご指導とご協力を賜りましたことをここに記し、感謝の意を表したく思います。

伊藤英則教授からは本研究の機会を与えていただき、本研究をすすめるにあたり終始適切にご指導を頂きました。先生からは、一つの物事をさまざまな視点から客観的に考察するという、研究者として必要な姿勢を始め、私が研究室に配属になった当時より、公私にわたり様々な相談にのっていただきました。これらのことは今後の私の研生活を支える礎になると確信しております。心より感謝申し上げます。

本論文をまとめるにあたり、貴重な御示唆を御指導を頂いた名古屋工業大学大学院の高橋直久教授、犬塚信博教授に深く感謝致します。

さらに、中村剛士准教授をはじめとして、加藤昇平准教授、武藤敦子助教からは、私の研究室配属時より、研究においてさまざまな観点からのアドバイスやご指導をいただきました。心より感謝いたします。

相山女学園大学准教授の鳥居 隆司からは、本論文の執筆の際に、本論文の内容に対して 有益なご助言をいただきました。先生のご指導をいただかなければ、現在の本論文にはなりませんでした。深く感謝いたします。

また、本研究について多くの助言を下さいました伊藤研究室の皆様にも厚くお礼申し上げます。さらに、貴重な時間を割いて、研究室内のコンピュータ環境の整備やセキュリティ対策などに従事した管理者の方々にも重ねてお礼申し上げます。

最後に、名古屋工業大学において研究する機会を与えてくれて、私の人生を暖かく見守り続けてくれた家族にお礼を述べます。

参考文献

- [1] 中原英臣, 佐川峻: ウイルス進化論, 早川書房 (1996)
- [2] 中原英臣, 佐川峻: 「進化論」を楽しむ本, PHP 文庫 (2001)
- [3] 河田雅圭: はじめての進化論, 講談社 (1990)
- [4] Stuart Russell, Peter Norving: エージェントアプローチ 人工知能 第2版, 共立出版 (2008)
- [5] 茨木俊秀: アルゴリズムとデータ構造, 昭晃堂 (1996)
- [6] 久保幹雄: 組合せ最適化とアルゴリズム, 共立出版 (2000)
- [7] 白石洋一: 組合せ最適化アルゴリズムの最新手法, 丸善 (2002)
- [8] 長尾智晴: 最適化アルゴリズム (Optimization Algorithms), 昭晃堂 (2000)
- [9] 山本芳嗣, 久保幹雄: 巡回セールスマン問題への招待, 朝倉書店 (1997)
- [10] 安田恵一郎: メタヒューリスティクスの現在と未来, 計測と制御, Vol.47, No.6, pp.453-458 (2008)
- [11] 相吉栄太郎, 安田恵一郎: メタヒューリスティクスと応用, 電気学会 (2008)
- [12] 石亀篤司: Particle Swarm Optimization -群れでの探索-, 計測と制御, Vol.47, No.6, pp.459-465 (2008)
- [13] 筒井茂義: アントコロニー最適化手法, 計測と制御, Vol.47, No.6, pp.466-472 (2008)
- [14] 筒井茂義: ACO: アントコロニー最適化, システム/制御/情報, Vol52, No.10, pp.390-398 (2008)

- [15] Eric Bonabeau, Marco Dorigo, Guy Theraulaz : Swarm Intelligence From Natural to Artificial Systems, Oxford Univ Pr(1999)
- [16] 小野功:グリッド向け遺伝的アルゴリズムによる大規模最適化, 計測と制御, Vol.47, No.6, pp.473-479(2008)
- [17] 松村嘉之, 大倉和博, 藤本 典幸:進化計算のためのグリッドコンピューティング, システム/制御/情報, Vol52, No.10, pp.382-389(2008)
- [18] 小林重信:実数値 GA の新展開, システム/制御/情報, Vol52, No.10, pp.354-361(2008)
- [19] 棟朝雅晴:進化計算におけるリンケージ同定, システム/制御/情報, Vol52, No.10, pp.362-367(2008)
- [20] 片山謙吾, 石淵久生:Memetic アルゴリズム, 計測と制御, Vol.47, No.6, pp.487-492(2008)
- [21] 野々部宏司, 柳浦陸憲: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, Vol.47, No.6, pp.493-499(2008)
- [22] Carlos A. Coello, David A. Van Veldhuizen, and Gary B. Lamont : Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers(2002)
- [23] 佐久間淳, 小林重信:確率分布推定に基づく実数値 GA の新展開, Vol18, No.5, pp.479-486(2003)
- [24] Dario Floreano, Claudio Mattiussi : Bio-Inspired Artificial Intelligence, MIT Press(2008)
- [25] David E.Goldberg : Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley
- [26] David E.Goldberg : The Design of Innovation,Kluwer Academic Publishers, 2002.
- [27] 伊庭斉志: 遺伝的アルゴリズムと進化のメカニズム』, 岩波書店 (2002)

- [28] 伊庭斉志：進化論的計算手法, オーム社, 2005
- [29] 伊庭斉志：遺伝的プログラミング入門, 東京大学出版会 (2001)
- [30] 坂和正敏・田中雅博：ソフトコンピューティングシリーズ1 遺伝的アルゴリズム, 日本ファジィ学会 (1995)
- [31] 吉田紀彦：『やわらかい情報処理 -生物に学ぶ最適化技法-』, サイエンス社 (2003)
- [32] 北野宏明：遺伝的アルゴリズム, 産業図書株式会社 (1993)
- [33] 北野宏明：遺伝的アルゴリズム 2, 産業図書株式会社 (1995)
- [34] 北野宏明：遺伝的アルゴリズム 3, 産業図書株式会社 (1997)
- [35] 電気学会 GA・ニューロを用いた学習とその応用調査専門委員会：学習とそのアルゴリズム, 森北出版 (2002)
- [36] 長尾智晴, 安居院猛：ジェネティックアルゴリズム, 昭晃堂 (1996)
- [37] 三宮信夫, 喜田一, 玉置久, 岩本貴司：遺伝アルゴリズムと最適化, 朝倉書店 (1998)
- [38] 大内東ほか：生命複雑系からの計算パラダイム -アントコロニー・DNA コンピューティング・免疫システム-, 森北出版 (2003)
- [39] John H.Holland：遺伝アルゴリズムの理論 -自然・人工システムにおける適応-, 森北出版 (2005)
- [40] 棟朝雅晴：遺伝的アルゴリズム その理論と先端的手法, 森北出版 (2008)
- [41] R.Pfeifer, C.Scheir：知の創成, pp.93-95, 共立出版 (2001)
- [42] 矢吹太郎, 伊庭斉志：遺伝的アルゴリズムから人工生命へ, Computer Today, No.113, pp.44-51(2003)
- [43] 伊庭斉志, 『進化論的計算手法の最近の研究動向』, 電子情報通信学会技術研究報告, vol.98, No.58, pp.45-49(1998)
- [44] 伊庭斉志, 『遺伝的プログラミングと進化論的な学習』, 人工知能学会誌, vol.9, No.4, pp.512-517(1994)

- [45] 伊庭斉志：進化的コンピューティング，人工知能学会誌， Vol.23, No.4, pp.559-568(2008)
- [46] 萩谷昌己：生命的コンピューティング，人工知能学会誌， Vol.23, No.3, pp.404-414(2008)
- [47] 特集：進化技術とその応用，電気学会論文誌 C， Vol.128, No.3, pp.334-468(2008)
- [48] David E.Goldberg： 実世界における遺伝的・進化的アルゴリズム， 情報処理学会誌,Vol.40,No.6,pp.623-627(1999)
- [49] 寺野隆雄：道しるべ:GA を使いこなすには， 情報処理学会誌， Vol.40,No.6,pp.628-630(1999)
- [50] 萩谷昌己： 生命コンピューティング， 人工知能学会誌， Vol.23, No.3, pp.404-414(2008)
- [51] 久保田直行， 福田敏男： ウイルス進化とファジィ理論， 日本ファジィ学会誌， pp.629-636, vol.10, No.4 (1998)
- [52] 久保田直行,福田敏男：ウイルス進化論に基づく遺伝的アルゴリズムの巡回セールスマンへ問題への応用,計測自動制御学会論文集,Vol.34,No.5,pp.408-414, 1998
- [53] 久保田直行,福田敏男,下島康嗣,『ウイルス進化論に基づく遺伝的アルゴリズム』,11th Fuzzy Symposium,pp.229-230(1995)
- [54] 久保田直行,福田敏男,小島史男,森岡利仁,『ファジィコントローラのための進化的計算とウイルス感染』,14th Fuzzy Symposium,pp.549-552(1998)
- [55] 下島康嗣,久保田直行,福田敏男,『ウイルス進化論に基づく学習型ファジィコントローラ』,日本機械学会論文集,68 巻 608 号,pp.223-230(1995)
- [56] 中谷直司,近藤邦雄,『ウイルス進化論に基づく新しい遺伝的アルゴリズム』,数理モデル化と問題解決,pp.31-36(1998)
- [57] 中谷直司,金杉昭徳,近藤邦雄,『ウイルス進化論に基づく遺伝的アルゴリズム』,情報処理学会論文誌,vol.40,No.5,pp.2346-2355(1999)

- [58] 杉本匡敏, 狩野均, 『ウイルス進化論に基づく GA を用いた人員配置問題の解決』
- [59] 川口貴正, 前原雅尚, 吉江修, 『ウイルス進化論に基づく GA における遺伝子伝達の提案』, 人工知能学会全国大会 (第 13 回), pp.271-273(1999)
- [60] 十河聡・西川郁子: 「ウイルス感染操作を用いた進化的計算の組合せ最適化問題への適用」, MPS シンポジウム, pp.25-32(2001)
- [61] 田村謙次, 犬塚信博, 伊藤英則, 『ウイルス間取込を導入したウイルス進化論型遺伝的アルゴリズム』, 電気関係学会東海支部連合大会講演論文集, pp.297(1999)
- [62] Kenji Tamura, Takeshi Tanigawa, Tsuyoshi Nakamura, Hidenori Itoh: Virus Evolutionary Genetic Algorithm with Immunity rate, APIEMS'2000, pp.274-277 (2000)
- [63] Kenji Tamura, Atsuko Mutoh, Tsuyoshi Nakamura, Hidenori Itoh: Linear Genetic Programming with Virus Individuals, Software Engineering, Artificial Intelligence, Networking And Parallel/Distributed Computing (SNPD'01), 2001.8.22, pp.936-941
- [64] Kenji Tamura, Atsuko Mutoh, Shohei Kato, Hidenori Itoh: Genetic Algorithm Adopting Selective Virus Infection, Iasted International Conference Artificial and Computational Intelligence, pp.289-293(2002)
- [65] 田村 謙次, 武藤 敦子, 中村 剛士, 伊藤 英則: 線形表現を用いたウイルス進化型遺伝的プログラミング, 電気学会論文誌, Vol.126-C, No.7, pp.913-918(2006)
- [66] 平澤宏太郎, 大久保雅文, 片桐広伸, 胡敬垆, 村田純一: 蟻の行動進化における Genetic Network Programming と Genetic Programming の性能比較, 電気学会論文誌 C, Vol.121, No.6, pp.1001-1009 (2001)
- [67] 白川真一, 長尾智晴: Graph Structured Program Evolution による複雑なプログラムの自動生成とその解析, 情報処理学会研究報告 [数理モデル化と問題解決], No.86, pp.21-24 (2007)
- [68] 伊庭斉志, 佐藤泰介, 『システム同定アプローチに基づく遺伝的プログラミング』, 人工知能学会誌, vol.10, No.4, pp.100-600(1994)

- [69] 徳井直生, 伊庭斉志, 石塚満: 汎用 Linear GP システムの実装と評価,
- [70] 徳井直生, 伊庭斉志: Empirical and Statistical Analysis of Genetic Programming with Linear Genome, 1999 IEEE International Conference on Systems, Man and Cybernetics(SMC99), IEEE Press (1999)
- [71] 伊庭斉志: 遺伝的プログラミングと進化論的手法, 人工知能学会誌, pp.251-258, vol.15, No.2, 2000
- [72] N.Tokui, S.Iba: Empirical and Statistical Analysis of Genetic Programming with Linear Genome, 1999 IEEE International Conference on Systems, Man and Cybernetics(SMC99), IEEE Press (1999)
- [73] Mike J.Keith, Martin C.Martin: Genetic Programming in C++:Implementation Issues, Advances in Genetic Programming, pp.285-310 (1994)
- [74] Paul Holmes: Peter J.Barclay: Functional Languages on Linear Chromosomes, GENETIC PROGRAMMING 1996, pp.427 (1996)
- [75] 苗村高義・橋山智訓・大熊繁:「GP における階層モジュール構造の獲得とその段階的進化」, 14th Fuzzy Symposium, pp.541-544 (1998)
- [76] ハリヨノ スラメト, 久保洋, 『GP を用いた論理回路の自動設計』, 情報処理学会第 58 回全国大会 (1999)
- [77] 森直樹: 遺伝的プログラミングにおける多様性評価手法と個体表現の簡約化, システム/制御/情報, Vol52, No.10, pp.368-373(2008)
- [78] W.M.Spears: Crossover or Mutation?, Proc. Foundations of Genetic Algorithms Workshop2, pp.221-237(1993)
- [79] W.M.Spears: Adapting Crossover in Evolutionary Algorithms, Proc. 4th Annual Evolutionary Programming Conference, pp.369-384(1995)
- [80] T.Jones: Crossover, Macromutation, and Populationbased Search, Proc. 6th International Conference on Genetic Algorithms, pp.73-80(1995)

- [81] R.Soraya: The Distributional Biases of Crossover Operators, Proc. Genetic and Evolutionary Computation Conference, pp.549-556(1999)
- [82] 八田浩一, 松田憲治, 若林真一, 小出哲士:「遺伝的アルゴリズムにおける交差手法の適応的選択の一手法」, 電子情報通信学会論文誌 D-I, pp900-909, Vol.8, No.7(1998)
- [83] 馬野元秀, 吉里友宏, 増井敏克, 宇野裕之, 瀬田和久, 岡田真:「複数の交叉演算による遺伝的アルゴリズムの改良」, 第19回ファジィシステムシンポジウム予稿集, pp.489-492(2003)
- [84] 姫野雅子・姫野龍太郎:「多峰性問題の Deterministic Crowding 法適用における初期世代の交叉と突然変異の影響」, 情報処理学会論文誌, pp54-62, Vol.43, No.1 (2002)

研究業績

学術論文

1. 田村 謙次, 武藤 敦子, 中村 剛士, 伊藤 英則 : 線形表現を用いたウイルス進化型遺伝的プログラミング, 電気学会論文誌C, Vol.126, No.7, pp.913-918, 2006.
2. 田村 謙次, 鳥居 隆史, 武藤 敦子, 中村 剛士, 加藤 昇平, 伊藤 英則 : ウイルス進化遺伝的アルゴリズムにおける感染手法による個体進化の相違に関する一考察, 日本知能情報ファジィ学会誌, Vol.20, No.5, pp.791-799, 2008.

国際会議

1. Kenji Tamura, Takeshi Tanikawa, Tsuyoshi Nakamura and Hidenori Itoh : Virus Evolutionary Genetic Algorithm with Immunity rate, Proceeding of the Third Asia-Pacific Conference on Industrial Engineering and Management Systems(APIEMS 2000), pp.274-277, 2000.
2. Kenji Tamura, Atsuko Mutoh, Tsuyoshi Nakamura and Hidenori Itoh : Linear Genetic Programming with Virus Individuals, Proceedings of the ACIS 2nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing(SNPD'01), pp.936-941, 2001.
3. Kenji Tamura, Atsuko Mutoh, Shohei Kato and Hidenori Itoh : Genetic Algorithm Adopting Selective Virus Infection, Iasted International Conference, Artificial and Computational Intelligence, pp.289-293, 2002.

その他

1. 田村謙次, 犬塚信博, 伊藤英則: ウイルス間取込を導入したウイルス進化型遺伝的アルゴリズム, 電気関係学会東海支部連合大会, pp.297, 1999.
2. 田村謙次, 鳥居隆司: 共進化による局所探索手法を用いた分布推定アルゴリズム, 情報処理学会第 69 回全国大会, pp.99-100, 2007.
3. 田村謙次, 鳥居隆司: 推定値の類似度を考慮した共進化による分布推定アルゴリズム, 情報処理学会第 6 回情報科学技術フォーラム, A001, 2007.

授賞

1. ACIS SCHOLARSHIP AWARD (SNPD'01) , 2001