

博士学位請求論文

共生進化の帰納学習への適用に関する研究

A Study on Applications of Symbiotic Evolution  
to Inductive Learning

2006年

大谷紀子

# 目次

第 1 章 序論	1
1.1 研究の背景と目的	1
1.2 論文の構成	4
第 2 章 遺伝的アルゴリズムと共生進化	5
2.1 生物のコンピュータモデル	5
2.2 遺伝的アルゴリズム	7
2.3 共生進化	9
2.3.1 共同共進化	9
2.3.2 共生進化	10
2.3.3 2 集団並行進化による共生進化	11
2.3.4 Hierarchical SANE のアルゴリズム	13
2.4 共生進化がモデル化する生物の行動	15
2.4.1 協働のモデル化	15
2.4.2 対象問題の分類	16
2.5 単純な問題における共生進化の効果	18
2.5.1 OneMax 問題	19
2.5.2 Harik の部分だまし問題	20
2.5.3 Floyd 問題 ~ HS モデル	22
2.5.4 Floyd 問題 ~ MGG モデル	25
2.5.5 2 集団並行進化の効果	32
2.6 共生進化の関連研究	32
2.7 まとめ	33
第 3 章 帰納論理プログラミングにおける予測精度向上	34
3.1 帰納論理プログラミング	34
3.1.1 基本的な枠組み	34
3.1.2 集合被覆アルゴリズム	36
3.2 共生進化に基づく最適仮説探索	37
3.2.1 Progol の学習アルゴリズム	37

3.2.2	帰納論理プログラミングシステム ILP/SE	38
3.3	ILP/SE の設計とアルゴリズム	39
3.3.1	アンサンブル学習	39
3.3.2	atom	40
3.3.3	clause	41
3.3.4	clause の表現型への変換	41
3.3.5	適応度	42
3.3.6	世代交代	43
3.3.7	クラス判定	43
3.3.8	システム構成	44
3.4	評価実験	44
3.4.1	実験方法	45
3.4.2	生成された仮説集合の評価	46
3.4.3	アンサンブル学習の効果	47
3.5	比較研究	52
3.6	まとめ	53
<b>第 4 章</b>	<b>簡素な決定木の生成</b>	<b>54</b>
4.1	決定木	54
4.1.1	分類対象となる事例	54
4.1.2	決定木の構造	55
4.1.3	決定木の生成	55
4.2	共生進化に基づく決定木生成	57
4.2.1	共生進化の適用	57
4.2.2	解の構造に関する問題点	58
4.2.3	遺伝的プログラミング	58
4.2.4	決定木生成システム SESAT	60
4.3	SESAT の設計とアルゴリズム	60
4.3.1	決定木の制約	61
4.3.2	sprig	61
4.3.3	決定木構成子	62
4.3.4	適応度	63
4.3.5	世代交代	65
4.4	評価実験	66
4.4.1	実験方法	66

4.4.2	予測正解率と木の簡素さ . . . . .	68
4.4.3	生成された決定木 . . . . .	70
4.4.4	探索能力と過学習の可能性 . . . . .	75
4.4.5	共生進化の効果 . . . . .	76
4.5	まとめ . . . . .	78
<b>第 5 章</b>	<b>多属性データに適した決定木の生成</b>	<b>79</b>
5.1	ポリマー判別 . . . . .	79
5.2	共生進化に基づくポリマー判別 . . . . .	80
5.2.1	ポリマーの近赤外スペクトルデータ . . . . .	80
5.2.2	ポリマー判別システム TS-SEPT . . . . .	82
5.3	TS-SEPT の設計とアルゴリズム . . . . .	83
5.3.1	sprig . . . . .	83
5.3.2	決定木構成子 . . . . .	84
5.3.3	適応度と世代交代 . . . . .	84
5.3.4	2 段階判別法 . . . . .	86
5.4	評価実験 . . . . .	88
5.4.1	実験方法 . . . . .	88
5.4.2	全事例判別 . . . . .	88
5.4.3	生成された決定木 . . . . .	90
5.4.4	2 段階判別法の効果 . . . . .	92
5.4.5	未知事例判別 . . . . .	92
5.5	まとめ . . . . .	93
<b>第 6 章</b>	<b>考察</b>	<b>94</b>
6.1	問題の難易度との関連性 . . . . .	94
6.2	部分解の設計条件 . . . . .	95
6.2.1	部分解の意味づけ . . . . .	95
6.2.2	部分解個体の組合せの学習 . . . . .	96
<b>第 7 章</b>	<b>結論</b>	<b>98</b>
7.1	研究のまとめ . . . . .	98
7.2	共生進化の展望 . . . . .	99
	謝 辞	101
	参 考 文 献	102

付録 A SESAT の部分解に関する検討	109
A.1 検討事項 . . . . .	109
A.2 実験 . . . . .	109
A.3 結果と考察 . . . . .	110
発表論文	113

## 目次

2.1	遺伝的アルゴリズムにおける遺伝オペレータ	9
2.2	共同共進化における集団と全体解	10
2.3	共生進化における集団と全体解	11
2.4	共生進化における2つの集団	11
2.5	Hierarchical SANE における個体の遺伝子型	13
2.6	Hierarchical SANE における2つの集団	14
2.7	network blueprint の表現型	15
2.8	OneMax 問題における最良適応度の推移	19
2.9	Harik の部分だまし問題における最良適応度の推移	21
2.10	Floyd 問題 (HS モデル) における平均ハミング距離の推移	24
2.11	Floyd 問題 (HS モデル) における最良適応度の推移	26
2.12	Floyd 問題 (HS モデル) における最良適応度減少前後の平均ハミング距離の推移	27
2.13	Floyd 問題 (MGG モデル) における平均ハミング距離の推移	29
2.14	Floyd 問題 (MGG モデル) における最良適応度の推移	30
2.15	Floyd 問題 (MGG モデル) における被参照数ごとの部分解個体数の推移	31
3.1	イルカ判定ルールを生成するための入力	35
3.2	モード宣言とタイプ情報	38
3.3	仮説空間の例	39
3.4	atom の遺伝子型と表現型	40
3.5	clause の遺伝子型	41
3.6	システム構成	44
3.7	正解率の推移 (背景知識 B1)	48
3.8	正解率の推移 (背景知識 B2)	49
3.9	正解率の推移 (背景知識 B3)	50
3.10	正解率の推移 (背景知識 B4)	51
4.1	降水の分類規則を表す決定木	56
4.2	表 4.2 の事例を分類する決定木	57
4.3	1つの属性に関する分岐を表す部分木	57

4.4	遺伝的プログラミングにおける遺伝オペレータ . . . . .	59
4.5	sprig の表現型 . . . . .	62
4.6	sprig の遺伝子型 . . . . .	62
4.7	決定木構成子 . . . . .	63
4.8	全正解事例数が 10 の場合の正解局在率の例 . . . . .	64
4.9	3 種類のアヤメの散布図 . . . . .	71
4.10	データ iris で生成された決定木 . . . . .	72
4.11	データ monks-1 で SESAT により生成された決定木 . . . . .	73
4.12	データ monks-1 で C5.0 により生成された決定木 . . . . .	74
4.13	適応度の推移 . . . . .	77
5.1	近赤外スペクトル . . . . .	81
5.2	sprig の表現型 . . . . .	84
5.3	sprig の遺伝子型 . . . . .	84
5.4	決定木構成子 . . . . .	85
5.5	再現率と適合率による値の変化 . . . . .	86
5.6	2 段階判別 . . . . .	87
5.7	ポリ塩化ビニルを判別するための決定木 . . . . .	91
6.1	Link, chrom の例 . . . . .	96
6.2	sprig の例 . . . . .	96
A.1	PA の場合の決定木構成子 . . . . .	110

# 表 目 次

2.1	音楽演奏の分類 . . . . .	17
2.2	比較実験に使用したパラメータ . . . . .	18
2.3	Harik の部分だまし問題で得られた解 . . . . .	21
2.4	集合 A と集合 B の要素の和の差 (HS モデル) . . . . .	23
2.5	集合 A と集合 B の要素の和の差 (MGG モデル) . . . . .	28
3.1	海獣の性質を表す事例 . . . . .	34
3.2	Mutagenesis の背景知識 . . . . .	45
3.3	ILP/SE の評価実験に用いたパラメータ . . . . .	46
3.4	訓練事例における平均正解率 . . . . .	46
3.5	未知事例単独分類における平均正解率 . . . . .	47
3.6	未知事例多数決分類における平均正解率 . . . . .	52
4.1	降水の種類を表す事例 . . . . .	55
4.2	クラス間の偏りの大きい事例 . . . . .	56
4.3	SESAT の評価実験に用いたデータ . . . . .	67
4.4	SESAT の評価実験に用いたパラメータ . . . . .	68
4.5	テスト事例における平均正解率と平均ノード数 . . . . .	69
4.6	訓練事例における平均適応度 . . . . .	75
5.1	本研究に使用した試料 . . . . .	82
5.2	TS-SEPT の評価実験に用いたパラメータ . . . . .	88
5.3	全事例判別における正解率と平均ノード数 . . . . .	89
5.4	C5.0 における 2 段階判別の正解率とノード数 . . . . .	92
5.5	未知事例判別における平均正解率と平均ノード数 . . . . .	93
6.1	共生進化を適用した問題の分類 . . . . .	94
A.1	テスト事例における平均正解率と平均ノード数 . . . . .	111
A.2	平均学習時間 . . . . .	112



# 第1章 序論

本論文の主題は、帰納学習全般における共生進化の適用可能性について論じる点にある。以下では、共生進化の機械学習分野における位置づけを明らかにした上で、研究の目的を述べる。また、本論文の構成を記す。

## 1.1 研究の背景と目的

人工知能の分野では、人間の知能と知的活動をコンピュータ上で実現することを目標として、さまざまな研究が行なわれてきた。人間の行なう知的活動のうち、とりわけ有用かつ高度といえる知的活動は学習である。人間は、意識的にはもちろんのこと、無意識のうちにも学習を行ない、直面する問題を解決したり、有用な知識を獲得したり、行動を効率化したりすることができる。人間の社会的生活に関わる行動は、すべて学習により獲得されるといっても過言ではない。

一方、コンピュータは人間よりはるかに速いスピードで正確な演算を行なえるが、具体的な処理手順が明確に指示されていることが大前提である。手順が未知で柔軟な発想が必要な学習は、コンピュータにとって決して容易な作業ではない。しかし、人間にとって不可欠な学習機能をコンピュータ上に実現することは、人工知能分野の目指すところでもあり、コンピュータが持つ演算能力と学習機能を組み合わせることで、人間が作業する際の支援や、人間には不可能に近い処理をコンピュータに行なわせることができるかもしれない。このような背景のもと、コンピュータに学習機能を持たせるべく行なわれているのが、機械学習の研究である。

人工知能学辞典 [人工 05] では、機械学習の代表的研究パラダイムとして以下の4種の学習を挙げている。

- 教示による学習 ... 教師から体系的な情報を獲得
- 例からの学習 ... 多くの事例から一般的な概念を獲得
- 類推 ... 過去の経験から類似分野の概念を獲得
- 観察からの学習と発見 ... 教師なしで新たな概念を発見

上記の学習の中で特に活発に研究されているのは、人間も日頃から頻繁に行なっている“例からの学習”である。人間は、いくつかの事例をもとにして、ある事物に関する概括的な意味内容、すなわち一般的な概念を獲得することが多い。学習対象とする事物に共通して含まれ、他の事物には含まれない性質を見出して、当該事物の一般的な性質と判断する。このように、事例に基づく帰納的な

推論により一般的な概念を獲得する学習を帰納学習 (inductive learning) と呼ぶ。与えられた有限個の事例を説明するだけでなく、無限個の未知事例をも視野に入れた一般的な概念を導き出すことで、予測問題の解決も可能となる。

帰納学習に相対する学習に演繹学習 (deductive learning) がある。演繹学習では、一般的な公理が与えられている中で演繹推論を行ない、同領域における推論や学習を効率化するような具体的知識を獲得する。帰納学習のように新しい知識が獲得されるわけではないので、帰納学習の方が演繹学習よりも高次の学習として位置づけられる。

帰納学習に関する研究の歴史は古く、さまざまな知識表現とアルゴリズムによる手法が提案されてきたが、可読性の高い形式で概念を記述する帰納学習としては、決定木生成と帰納論理プログラミングが挙げられる。決定木は分類規則の木構造による表現手法であり、属性と属性値の組からなる事例を入力として受け取り、事例の属するクラスを判定する。Breiman による CART [Breiman 84] や、Quinlan による ID3 [Quinlan 86]、C4.5 [Quinlan 93] など、分岐の多様性や情報量に基づく決定木生成アルゴリズムが主流となっている。C4.5 の後継である C5.0 [C50] はデータマイニングツールとして市販されており、幅広い分野における予測問題に利用されている。一方、帰納論理プログラミングは2クラス分類の規則を生成する手法といえる。入力する事例と出力する規則を一階述語論理により表現し、事例のみではなく背景知識をも利用して推論を行なうため、知識の表現能力と推論能力の両面において、帰納論理プログラミングは決定木よりも優れているといえる。GOLEM [Muggleton 90]、Progol [Muggleton 95]、FOIL [Quinlan 95] などが代表的な帰納論理プログラミングシステムである。

帰納学習では、学習対象を表す事例を包括し、それ以外的事例を包括しないような概念の導出を目標とする。導出すべき概念の候補は、無条件にあらゆる事例を包括する最も一般的な概念から、具体的な個々の事例まで、無数に存在する。これらすべての候補概念を、互いの包括関係が表されるように配置した包摂束を概念空間と呼ぶ。帰納学習の過程は、概念空間の探索として捉えることができる。与えられた事例だけではなく未知の事例をも考慮しながら概念空間を探索することで、目標概念すなわち最適解を獲得する。

一般に概念空間は広大であるため、効率よく目標概念に達する手法が求められる。探索手法としては、最も基本的な盲目的探索である深さ優先探索と幅優先探索のほか、探索コストを考慮したヒューリスティック探索である山登り法、最良優先探索、ビーム探索、A\*アルゴリズムなど、数多くの手法が提案されている。また、金属やガラスの熱処理操作であるアニーリングを模した焼きなまし法 (Simulated Annealing; SA)、探索過程の記録を利用するタブー探索 (Tabu Search; TS)、生物の進化過程を模した遺伝的アルゴリズム (Genetic Algorithm; GA) など、局所解の回避が可能な確率的探索もある。

前述した探索手法のうち、遺伝的アルゴリズムは広大な概念空間を探索できる多点探索手法であり、他の手法と比較して以下の特長を持つ。

- 大域的な探索が可能である．
- 多目的最適化が可能である．
- 実用的な時間で優れた解を発見することができる．

遺伝的アルゴリズムは、上記の特長に加えて、対象とする問題領域を限定せず、問題領域に関する事前知識がなくても適用できるという利点をも兼ね備えているため、実用システムへの応用という点でも有用な探索手法といえる．実際に、実用問題を扱う市販のシステムにも組み込まれており、遺伝的アルゴリズムの応用分野は多岐にわたる．しかし、目的に適合する解を獲得するためには対象問題に適した設計を行なう必要があるため、解を表現するデータ構造や評価に用いる関数に加えて、アルゴリズム自体の拡張も研究されている．

Moriarty らは、動的で複雑な問題を対象とする拡張アルゴリズムとして共生進化を提案し、ニューラルネットワークの学習に適用した [Moriarty 95, Moriarty 96, Moriarty 98]．共生進化は、“複雑な問題は、単純な問題に分割することによって解決が容易になる”という統治分割法の考え方に基づいて考案された手法であり、問題に対する解はいくつかの部分の組合せにより構成されると考える．このとき、最適解に含まれるべき部分解の探索と並行して、最適解となる部分解の組合せをも探索する必要があるため、部分解を個体とする部分解集団と、部分解の組合せを個体とする全体解集団を保持し、両集団を並行して進化させる．両集団は互いに影響を及ぼし合いながら最適解を探索する．

部分解を個体とする場合、集団の多様性が維持されるため、多くの解候補からすばやく効果的な解を獲得できるが、最適解に達するのは難しい．一方、全体解を個体とする場合には、局所解に収束する可能性は高いものの、最適解を含む領域に到達すると確実に収束する．共生進化では、両者を統合することで局所解への収束の回避、および最適解への収束の促進を同時に実現する．また、進化過程で得られた評価の高い部分をその後の進化操作によって壊してしまうという“積み木の崩壊”の問題をも回避する．これらの特長を持つ共生進化の有効性は、ニューラルネットワークだけでなく、ファジールールの学習においても確認されている．

共生進化における“部分解と全体解の並行進化”という発想は、ニューラルネットワークやファジールールに特化されたものではない．解が意味のある部分に分割できる問題では、最適解に含まれるべき部分解と、最適解を構成するための部分解の組合せを同時に考え、解を導き出すのが一般的である．したがって、他の問題領域においても共生進化の効果が発揮されると予想される．

本論文では、先行研究で扱われていない問題を対象として、共生進化の効果を明らかにするとともに、帰納学習全般への共生進化の適用可能性を示唆することを目的とする．学習対象としては、代表的な帰納学習である帰納論理プログラミングと決定木生成を取り上げる．共生進化の根底にある統治分割法は、解の構造が複雑な場合ほど大きな効果が得られるため、一階述語論理や木構造という複雑な解構造を学習対象とすることによって、一般的な遺伝的アルゴリズムと共生進化との差異が明確に現れ、有用な比較検討ができると思う．共生進化に基づく帰納論理プログラミングの最適解探索手法、簡素な決定木の生成手法、およびポリマー判別のための決定木生成手法を提案し、各手法を実

装した帰納学習システム ILP/SE, SESAT, TS-SEPT により動作を確認して, 共生進化の解探索について考察する.

## 1.2 論文の構成

本論文は, 以下に述べる通り, 7つの章から構成される.

第2章は“遺伝的アルゴリズムと共生進化”と題し, 過去に行なわれてきた生物のコンピュータモデルに関する研究を概観した後, 多様な分野に应用されている解探索手法の1つとして遺伝的アルゴリズムを取り上げ, 基本的枠組みを概説する. 数多く提案されている遺伝的アルゴリズムの拡張アルゴリズムのうち, 複数の人間が協働する様子をモデル化した共生進化に焦点をあて, 処理手法について説明する. 共生進化では, 対象問題の解を同型の部分に分割し, 部分解の組合せで目的の全体解を表現する. 良い部分解と良い全体解を並行して探索していく点が共生進化の特徴である. 共生進化の学習対象となる問題を設計上の複雑さという視点から分類し, 単純な問題において共生進化が有効であることを実験により確認する.

第3章は“帰納論理プログラミングにおける予測精度向上”と題し, 共生進化に基づく最適仮説探索および多数決によるアンサンブル学習を組み合わせた手法を提案する. Horn節で表される仮説をリテラルの組合せと見なし, リテラルを部分解, リテラルの組合せを全体解とする共生進化により, 最適仮説探索を行なう. 提案手法を帰納論理プログラミングシステム ILP/SE として実装し, ベンチマークデータ Mutagenesis を用いた評価実験により, 共生進化とアンサンブル学習の効果を示す.

第4章は“簡素な決定木の生成”と題し, 共生進化に基づく決定木生成手法を提案する. 決定木を高さ1の部分木の組合せと見なし, 高さ1の部分木を部分解, 高さ1の部分木を結合した木を全体解とする共生進化により, 予測正解率が高く簡素な決定木を生成する. 提案手法を決定木生成システム SESAT として実装し, UCIリポジトリのベンチマークデータを用いて C5.0 をはじめとするシステムとの比較を行なう.

第5章は“多属性データに適した決定木の生成”と題し, 近赤外スペクトルデータからポリマーを判別するための共生進化に基づく決定木生成手法を提案する. 近赤外スペクトルデータの特徴を踏まえて個体表現と適応度関数を設計する. また, さらなる正解率向上のために2段階判別法を導入する. ポリマー判別システム TS-SEPT により実データを判別するための決定木を生成し, 100%の正解率が得られることを示す.

第6章では, 第3章~第5章で述べた研究をもとに, 共生進化と組み合わせるアンサンブル学習と2段階判別法, および部分解の遺伝子表現を設計するにあたっての留意点を中心に考察する.

最後に, 結論として第7章で本研究の結果をまとめ, 共生進化の展望について述べる.

## 第2章 遺伝的アルゴリズムと共生進化

広大な解空間において多点探索が行なえる遺伝的アルゴリズムは、生物の進化メカニズムをモチーフとしたコンピュータモデルである。ここでは、生物のコンピュータモデルに関する過去の研究と、最適化問題の解探索の一手法として広く利用されている遺伝的アルゴリズムについて概説した後、本論文の主題である共生進化の特徴について説明し、単純な問題に対する実験で共生進化の効果を示す。また、共生進化の関連研究と他分野への適用可能性についても述べる。

### 2.1 生物のコンピュータモデル

哲学の分野では古代ギリシャの時代から“人間は機械である”という概念が提唱されており、Epicurus は人間の身体も精神もすべて原子の運動に由来すると考えた。近世では、動物機械説を唱えた Descartes 以来、人間の思考や行動のメカニズムを機械のアナロジーで説明する試みが盛んになり、La Mettrie は著書である『人間機械論』[ド・ラ・メトリ 32]において「人体は自らゼンマイを巻く機械である」、「脚が歩くための筋肉を持っているように、脳も考えるための筋肉を持っている」と述べ、単なるアナロジーではなく人間と等価な機械を人工的に作ることができると主張した。

人間の持つさまざまな能力と可能な活動のうち、知能および知的活動をコンピュータ上で実現しようと試みる研究が、人工知能 (Artificial Intelligence; AI) の分野であると考えられる。人工知能については、研究が正式に開始された 1956 年のダートマス会議以来さまざまな定義がなされているが、機械における意識の有無の観点から次の 2 つのアプローチに大別される。

- 強い人工知能 (strong AI)

人間と同じメカニズムで問題を解決しようとするアプローチ。機械は意識を持つことができると主張する。

- 弱い人工知能 (weak AI)

人間とメカニズムは異なっても、結果として人間のような知的活動が行なえればよいとするアプローチ。知的に活動できる機械であっても意識を持つことはないと主張する。

現在行なわれている人工知能の研究は後者に相当するものがほとんどであるが、人間あるいは一般的な生物の思考、進化、行動等のメカニズムに着想を得たコンピュータモデルや、人間による認知・心理実験の結果に基づいて考え出された手法は非常に多い。

生物の脳内には多数のニューロンが存在し、ニューロン間の信号のやり取りによってさまざまな情報が処理されている。このメカニズムをコンピュータ上でモデル化したものがニューラルネットワーク

ク (Neural Network; NN) である。1943年に単一ニューロンの数理モデルが提案され [McCulloch 43] , 1951年には Minsky らによって SNARC と呼ばれる最初のニューラルネットコンピュータが製作された。3000本の真空管と B24 爆撃機の自動操縦装置を使って作られた SNARC は、40個のニューロンを持つネットワークをシミュレートした。各ニューロンは多数の近接ニューロンから受けた刺激によって ON と OFF の2状態を切り替える。ニューロンの状態は受けた刺激に対応する命題と事実上等価であり、ニューラルネットワークにより関数の計算や論理結合子の実現が可能であることが示された。また、学習の可能性についても示唆された。

しかし、ニューラルネットワークは安定性 - 可塑性のジレンマ、すなわち既存の知識を保持しながら新しい知識を獲得するという困難な問題を抱えていた。Grossberg は適応共鳴理論 (Adaptive Resonance Theory; ART) を提案し [Grossberg 76b, Grossberg 76a] , 既存の知識に影響を与えることなく安定性と可塑性を切り替えることで、この問題を解決した。適応共鳴理論は生物学のおよび行動科学的データに基づいており、自己組織的な神経パターン認識ネットワークに対する超並列アーキテクチャのモデルとして ART-1[Carpenter 87a] , ART-2[Carpenter 87b] , ART-3[Carpenter 90] を示した。Grossberg は「人間の思考モデルをコンピュータで明らかにしたい、という一念で20年間孤独に研究を続けてきた」[斎野 88] と述べており、ネットワーク中の重みベクトルを“memory trace”という言葉で表現していることから、生物システムが理論の根底にあることがわかる。

Newell らにより提案された一般問題解決器 (General Problem Solver; GPS) では、人間が問題解決をするときの手順を模倣して設計されている [Newell 63] 。初期状態と目標状態の差異が小さくなるようにオペレータを適用し、オペレータが適用できない場合にはいくつかの副問題に分割することで問題を解決していくという枠組みである。彼らの関心事は問題解決の可否よりも推論過程にあり、パズルなどさまざまな問題における人間の問題解決過程が GPS によりシミュレート可能であることが示された。

人間は練習によって作業効率を向上することができる。これは、同じ試行を繰り返すことによって課題に関するパターン知識が獲得されるためと考えられており、学習のチャンキング理論 (chunking theory of learning) と呼ばれている。Rosenbloom らはこの理論を XAPS2 というプロダクションシステムとして実装し、作業効率の向上だけでなく知識獲得の過程をも示すモデルであることを示した [Rosenbloom 82] 。

人間は既存の知識や経験からのアナロジーによって新たな問題を解決することが多い。アナロジーの計算モデルとしては、Winston の ANALOGY[Winston 80] に始まり、Gentner らによる MAC/FAC [Gentner 93] , Holyoak らの ACME[Holyoak 89] , ARCS[Thagard 90] などの類比検索・写像モデルや、Anderson らの ACT-R などの問題解決のアナロジーモデルが提案されている。また、大谷 [大谷 95] は多段階学習におけるアナロジーモデルを提案し、算数の「率」の概念を学習対象として実験を行った結果、コンピュータ上に実装された提案モデルと実際の人間の概念獲得過程が適合することを示している。

蟻は、餌を見つけるとフェロモンと呼ばれる芳香性物質を地面に残しながら餌を巣に運ぶことが知られている。また、他の蟻が過去に残したフェロモンの道筋に沿って移動するという習性がある。フェロモンは時間の経過とともに揮発するが、蟻がフェロモンの道筋をたどる際にはフェロモンを追加しながら移動するため、多くの蟻がたどった道筋は他の場所よりも強いフェロモンを発するようになる。このフェロモンの道筋が複数の蟻の間のコミュニケーション手段となり、他の蟻が見つけた餌の在り処に到達し、迷うことなく巣へと餌を運ぶことができる。この蟻の集団行動のメカニズムに基づいた最適化問題の探索手法がアントコロニーシステム (Ant Colony System; ACS) である [Dorigo 96]。フェロモンの付着量やフェロモンを付着できる蟻を限定するなどの手法が提案され、さまざまな組合せ最適化問題に適用されている。

生物は地球上の動的な環境に適応するための進化を続け、現在の生態系が形成された。動的な環境に適応するという困難な問題を解決すべく世代交代を繰り返し、得られた解が現在の生態系であるともいえる。この生物の進化メカニズムをコンピュータ上に実現し、さまざまな問題に対する解の探索に利用しようとする手法が進化的計算 (evolutionary computation) である。環境への適応度合や個体間の相互作用によって遺伝的操作が加えられ、集団全体が進化するということになり、Darwin の進化論に基づいたプロセスをたどる。個体の表現や遺伝的操作の方法の違いによって多様な計算手法が考えられており、遺伝的アルゴリズム (Genetic Algorithm; GA)、遺伝的プログラミング (Genetic Programming; GP)、進化的戦略 (Evolutionary Strategy; ES)、クラシファイアシステム (Classifier System; CS) などが代表的な手法として挙げられる。この中でも特によく使われているのが遺伝的アルゴリズムであり、多様な分野に応用されている。

以上のように、人間あるいは生物をベースとする人工知能の手法が数多く提案されている。本論文では、構造が不明で広大な解空間を持つ問題における探索が可能といわれ、さまざまな分野で実用化されている遺伝的アルゴリズムを帰納学習における解探索手法として取り上げる。特に、複数生物による協働作業をモデル化した共生進化に注目する。

## 2.2 遺伝的アルゴリズム

遺伝的アルゴリズムは 2.1 節で取り上げた進化的計算の一手法であり、確率的解探索アルゴリズムといえる [Holland 75, Goldberg 89]。与えられた問題に対する解候補を遺伝子列で表現し、交叉、突然変異などの遺伝操作によって新たな解候補を生成することで、目的に適合した解を探索していく。適応能力の高いものほど多くの子孫を残すことができるという自然淘汰説に従うため、集団内の個体として保持された複数の解候補のうち、適応度 (fitness value) の高い個体の形質が次世代の個体へと受け継がれるよう、遺伝オペレータの適用対象となる親個体を選択する。大まかな処理の流れは以下に示す通りである。

Step 1 【初期化】複数の個体をランダムに生成し、初期世代の集団とする。

Step 2 【評価】集団内の個体の適応度を算出する。

Step 3 【世代交代】次世代の個体を生成する。

Step 4 終了条件が満たされるまで、Step 2~3 を繰り返す。

Step 2 の評価の段階では、解としての適切さの度合を測るために定義された適応度関数 (fitness function) により、各個体の適応度を算出する。通常、良い個体ほど高い適応度を持つように定義されるが、対象となる問題によっては適応度の低さで個体の良さを表現し、適応度の低い個体の形質が次世代に受け継がれるよう設計される場合もある。

Step 3 は次の3段階からなる。

Step 3-1 《複製選択》子個体を生成する親個体を選択する。

Step 3-2 《生殖》子個体を生成する。

Step 3-3 《生存選択》次世代に残す個体を選択する。

Step 3-1 の複製選択では子孫を残すことのできる親個体、すなわち遺伝オペレータの操作対象となる個体を選択し、Step 3-3 の生存選択では Step 3-2 で次世代に生き残るべき個体を選択する。主な選択手法を以下に示す。

Selection 1 ランダム選択

Selection 2 ランク選択

Selection 3 トーナメント選択

Selection 4 ルーレット選択

Selection 1 は適応度を無視して選択する手法であり、Selection 2~4 は適応度の高い個体ほど子孫を残し、次世代に生き残る確率が高くなるように選択する手法である。Selection 2,3 は適応度を序数的に使用し、Selection 4 は適応度の大きさに応じて選択確率を変動させる。適応度の良い個体はそのまま次世代に継承されるよう、エリート保存戦略をとる場合もある。

Step 3-2 の生殖の段階では、遺伝子列のうちランダムに選択された部位に対して、遺伝オペレータが定められた確率で適用される。遺伝オペレータとしては図 2.1 に示すような交叉、逆位、突然変異などが用いられる。交叉では2つの親個体の部分を入れ替えて新たな2つの子個体を生成する。逆位は親個体の一部分の遺伝子の順序を入れ換える操作であり、突然変異ではある遺伝子の値をランダムに変更する。

遺伝的アルゴリズムを効果的に動作させるには、適応度関数、世代交代モデル、生殖方法を適切に定めるほか、遺伝子型 (genotype) の設計、および遺伝子型から表現型 (phenotype) への変換手法が重要である。遺伝子型とは解を表す個体の遺伝子列表現であり、表現型とは動作時に現れる形質のことで、遺伝子型と環境によって決定される。遺伝オペレータは遺伝子型に対して適用され、適応度は表現型から算出される。

遺伝子列のうち、意味のある部分列をスキーマ (schema) と呼び、定義長が短く、次数が低く、かつ適応度の高いスキーマを積み木 (building blocks) と呼ぶ。ここで、定義長はスキーマに含まれる遺



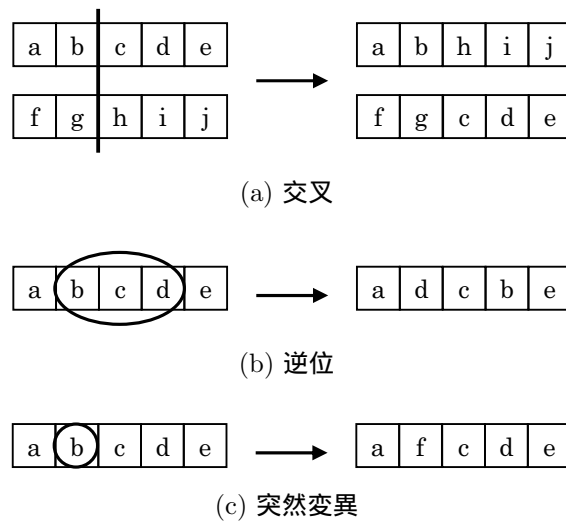


図 2.1: 遺伝的アルゴリズムにおける遺伝オペレータ

伝子の数を表し、次数は値が固定された遺伝子の数を表す。積み木は有用な部分解であるといえ、世代の進行に伴って積み木を含む個体の数は増大するという積み木仮説 (building block hypothesis) が提唱されている。この積み木を活用することで、遺伝的アルゴリズムの探索能力は改善されると考えられている。

遺伝的アルゴリズムは、構造が不明確で広大な解空間における最適解探索が可能であり、最適性の定義があいまいな問題にも有効であるため、多様な分野において広く利用されている。また、プログラム生成や概念形成など構造を持つ解を探索するための遺伝的プログラミング、多目的最適化問題のための遺伝的アルゴリズム、異種個体の相互作用により適応度の定義が困難な問題に対処する共生進化 (coevolution) など、さまざまな拡張アルゴリズムが提案されている。

## 2.3 共生進化

複雑で大規模な問題を単純で小規模な問題に分割することができるならば、分割した問題に対する解を求め、得られた解から最終的な解を導くことで、大規模な問題をそのままの形で解くよりも容易に解決することができる。この統治分割法 (Divide-and-Conquer) に基づいて考案された遺伝的アルゴリズムが共同共生進化である。以下では、共同共生進化およびその一手法である共生進化について詳細を説明する。

### 2.3.1 共同共生進化

共同共生進化 (cooperative coevolution) は、複雑で動的な問題を扱うことを目的として提案された遺伝的アルゴリズムである [Smith 93, Horn 94, Potter 94, Paredis 96, Whitehead 96]。統治分割法に基づき、問題に対する解がいくつかの部分の組合せにより構成されると考える。各部分に対する解

の候補を個体として保持し、各個体を協力的な相互作用によって進化させることで、最適解を構成するために必要な部分を獲得する。以下では、解の一部を部分解、部分解の組合せによって形成される完全な解を全体解と呼ぶ。スキーマを個体として有用な積み木を生成し、複数の積み木の組合せで解を形成する手法と表現することもできる。共同共進化における集団と全体解の一例を図 2.2 に示す。この例では、各集団において最良適応度を持つ個体の組合せを全体解としている。

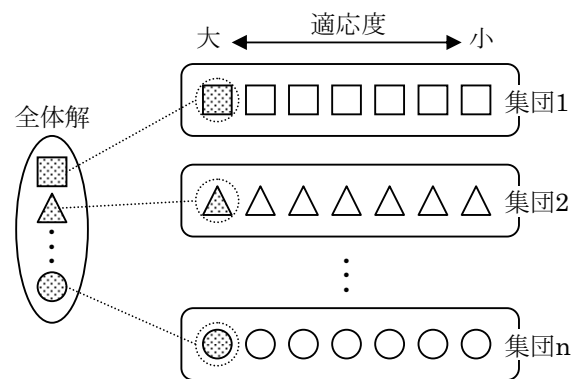


図 2.2: 共同共進化における集団と全体解

各個体は2つの目標を持って進化する。第1の目標は解の一部を最適化することであり、第2の目標は他の部分を最適化する個体と協働して解全体を最適化することである。前者の対象となる部分は個体ごとに異なるため、異なる目標を持った複数の探索が並行して行なわれることになる。また、後者のために個体の適応度は全体解の評価に基づいて決定され、各個体は互いに影響を及ぼし合って進化する。したがって、全体解を個体とする一般的な遺伝的アルゴリズムよりも緻密な探索が行なえるものと期待される。

共同共進化では、解の分割、個体の進化、個体の相互作用、適応度共有 (fitness sharing) に関する設計が動作結果を大きく左右するため、各種の問題に対してさまざまな手法が提案されている。

### 2.3.2 共生進化

Moriarty らは、ニューラルネットワークを学習するための共同共進化の一手法として共生進化 (symbiotic evolution) を提案した [Moriarty 95, Moriarty 96]。共生進化では、全体解を構成する部分解はすべて同型の遺伝子列で表現されるものとし、各部分に対する解の候補を1つの集団で進化させる。問題に対する解を同型のスキーマの組合せで表現し、定義長と次数が一定のスキーマを個体として、複数の異なる積み木を1集団で生成する手法と表現することもできる。例えば、ニューラルネットワークでは、複数の隠れ層のニューロンを組み合わせることでネットワークが表現されるので、隠れ層のニューロンが部分解、それらの組合せからなるネットワークが全体解となる。共生進化における集団と全体解は図 2.3 のように図示される。

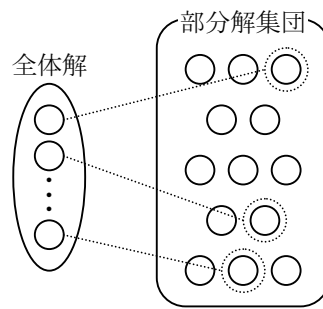


図 2.3: 共生進化における集団と全体解

図 2.2 に示した共同共生進化と異なり，解の異なる部分を表す個体を 1 つの集団で保持するため，異なる目標を持った複数の探索が 1 つの集団内で並行して行なわれることになる．通常の進化と比較して，個体同士の協調性が増進され，個々の役割に応じた特異性が確立していく分化の現象が促進される．その結果，進化が進んだときに集団が 1 つの個体に収束する可能性が低くなる．したがって，集団内の多様性が維持されることにより効果的な探索が実現される．ニューラルネットワークの学習システム SANE(Symbiotic, Adaptive Neuro-Evolution) により，共生進化の有効性が示されている．

### 2.3.3 2 集団並行進化による共生進化

個体の適応度は全体解の評価に基づいて決定されるため，全体解を構成する個体の組合せが各個体の適応度を大きく左右する．有用な部分解であっても，適切な組合せに含まれていなければ，有用な部分解が持つ効力を発揮することはできない．したがって，有用な部分解の探索と並行して，有用な部分解の組合せをも探索する必要がある．以上の考えから，部分解を個体とする部分解集団と，部分解の組合せを個体とする全体解集団を保持し，両集団を並行して進化させる手法が提案された [Moriarty 98]．部分解の遺伝子型は部分解の形質決定に必要な情報をすべて盛り込むように設計し，全体解の遺伝子型は部分解集団の個体へのポインタ列により表現する．共生進化の全体解集団と部分解集団の例を図 2.4 に示す．

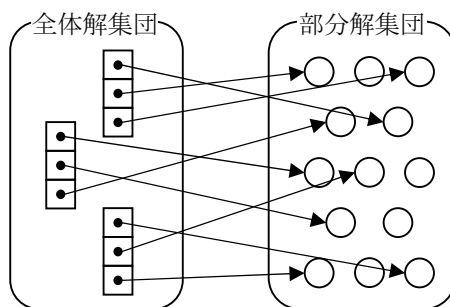


図 2.4: 共生進化における 2 つの集団

一般的な遺伝的アルゴリズムと同様に、初期化、評価、世代交代という3つの処理によって解を探索していくが、各段階においてそれぞれ部分解集団と全体解集団に対する操作を行なう。全体解集団の個体は部分解集団の個体の組合せで表現されるため、初期化と世代交代の段階では全体解集団への操作に先立って部分解集団の処理を行なう。一方、部分解集団の個体の適応度は全体解集団の個体の適応度により決まるため、評価の段階では全体解集団の個体の適応度を求めてから部分解集団の個体の適応度を決定する。全体の流れは以下の通りである。

#### Step 1 【初期化】

- (a) 部分解を表す個体をランダムに生成し、初期世代の部分解集団とする。
- (b) 部分解集団の個体をランダムに組み合わせて、初期世代の全体解集団を生成する。

#### Step 2 【評価】

- (a) 全体解集団内の個体の適応度を算出する。
- (b) 全体解の適応度をもとに、部分解集団内の個体の適応度を決定する。

#### Step 3 【世代交代】

- (a) 部分解集団の次世代の個体を生成する。
- (b) 全体解集団の次世代の個体を生成する。

Step 4 終了条件が満たされるまで、Step 2~3を繰り返す。

Step 1の初期化では、最初に部分解集団をランダムに生成した後、部分解集団の個体をランダムに組み合わせて全体解集団を生成する。複数の全体解個体からポイントが張られる部分解個体もあり、いずれの全体解個体からもポイントが張られない部分解個体もある。

Step 2の評価の段階では、全体解の表現型をもとに適応度を算出する。全体解の表現型は、遺伝子型のポイントが指している部分解の個体を参照しながら生成する。全体解個体の適応度は、問題の解としての適切さを表すように定義された適応度関数により算出する。部分解個体の適応度は、当該部分解個体にポイントが張っている全体解個体の適応度から算出する。

Step 3の世代交代では、部分解個体は次世代の部分解個体により置き換えられる。全体解集団の進化により変化しなかった全体解個体でも、ポイントが指している部分解個体に変化した場合には、全体解個体の動作も変化する。

最適化問題の解探索を遺伝的アルゴリズムによって行なう場合、得られる解の良さに加え、解を得るまでにかかった世代交代数が評価指標となる。有用な解を短時間で求められる手法が望ましい。遺伝的アルゴリズムの探索能力を向上させる一手法に積み木の活用があるが、積み木は交叉や突然変異によって崩壊される可能性がある点を考慮しなければならない。

部分解を個体とする進化では、集団の多様性が維持されるため、多くの解候補からすばやく“効果的な解”を獲得することができる。しかし、全体解自体を洗練することはできないので、“最適な解”

に達するのは難しい。すなわち、解が多数あるような問題には有効であるが、厳密な最適解を求めるような問題には不適とされている。

一方、全体解を個体とする進化は、部分解の場合と違って探索の目標が 1 つであるため、集団の多様性の維持が困難である。したがって、“効果的な解”を得るまでに非常に時間がかかり、進化の早い段階で局所解に収束する可能性も高い。しかし、“最適な解”を含む領域に達すると確実に収束するという特長を持っている。

本節で述べた 2 集団並行進化による共生進化では、部分解の進化と全体解の進化を統合することによって、両者の長所を生かし、短所を補完する効果が期待される。部分解を保持することで集団の多様性を維持し、より効果的な解をより速く獲得する。集団の多様性の維持は局所解への収束の回避にもつながる。また、全体解を保持することで最適解への収束を促進する。さらに、部分解の組合せで表現された全体解を交叉や突然変異によって進化させることで、積み木を壊さずに新たな解候補を生成することができる。

Moriarty らは 2 集団並行進化による共生進化を SANE の拡張システム Hierarchical SANE として実現し、ロボットアームの制御タスクに関する評価実験を行なった。実験の結果、Hierarchical SANE の有効性は SANE よりも高いことが報告されている。以降、“共生進化”は本節で説明した 2 集団並行進化による共生進化を指すものとする。

### 2.3.4 Hierarchical SANE のアルゴリズム

Hierarchical SANE では 3 層フィードフォワードネットワークを学習対象とする。入力層と出力層を結ぶ複数の隠れ層のニューロンを組み合わせることでネットワークが形成されるので、隠れ層のニューロンを部分解とし、部分解の組合せからなるネットワークを全体解としている。部分解を表す個体を neuron、全体解を表す個体を network blueprint と呼ぶ。

neuron と network blueprint の遺伝子型の例を図 2.5 に示す。neuron の奇数番目のフィールドはニューロンの結合先を表すラベルであり、偶数番目のフィールドはニューロン結合の重みを表している。ラベルフィールドは 8 ビット、重みフィールドは 16 ビットのバイナリ遺伝子からなる。network blueprint の遺伝子は neuron へのポインタである。

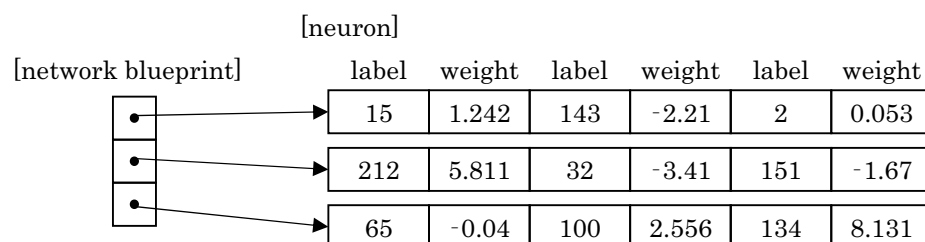


図 2.5: Hierarchical SANE における個体の遺伝子型

遺伝子の値をランダムに設定した neuron 個体，および neuron 個体をランダムに組み合わせた network blueprint 個体を複数個生成し，初期世代の neuron 集団と network blueprint 集団とする．両集団の例を図 2.6 に示す．

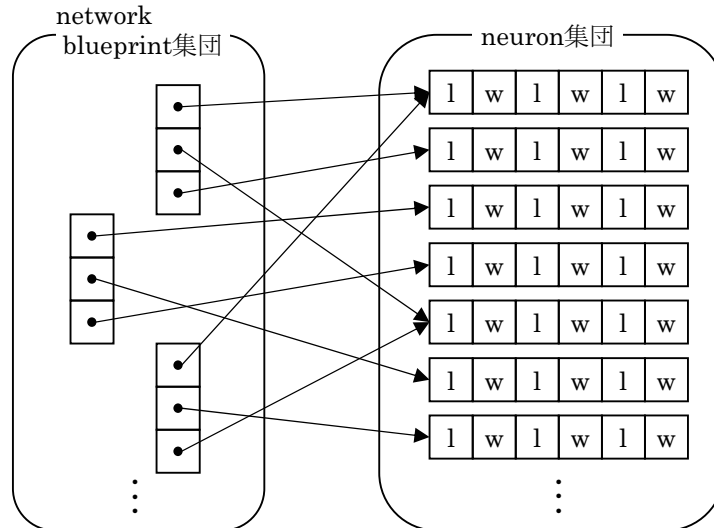


図 2.6: Hierarchical SANE における 2 つの集団

個体を評価する際には，network blueprint のポインタが指している neuron を次の規則により隠れ層ニューロンに変換し，すべてのニューロンを組み合わせる表現型であるネットワークを生成する．

- ラベルが 127 より大きければ出力層，127 以下であれば入力層に結合する．
- 結合先のニューロンは，遺伝子の値を入力層もしくは出力層のニューロンの数で割ったときの剰余により決定する．
- 各ラベルの次にある遺伝子の値を結合時の重みとする．

上記の規則に従うと，図 2.5 の network blueprint の遺伝子型は図 2.7 に示すような表現型のネットワークに変換される．適応度関数によりネットワークが評価され，当該 network blueprint の適応度が算出される．neuron の適応度は，当該 neuron にポインタを張っている network blueprint のうち，適応度の高い 5 個の個体の適応度の和とする．network blueprint の適応度の平均により neuron の適応度を設定する方法もあるが，他の neuron との組合せが悪いために評価が低くなった network blueprint の影響を軽減するために，評価の高い個体の適応度の和を採用する方が効果的であるとされている [Moriarty 98] ．

neuron 集団と network blueprint 集団では，以下の手順で世代交代を行ない，次世代の個体を生成する．以降，この世代交代モデルを“HS モデル”と呼ぶことにする．

Step 3-1 《複製選択》上位 25%のある個体に対して，上位 25%の別の個体をランダムに選択し，親個体とする．

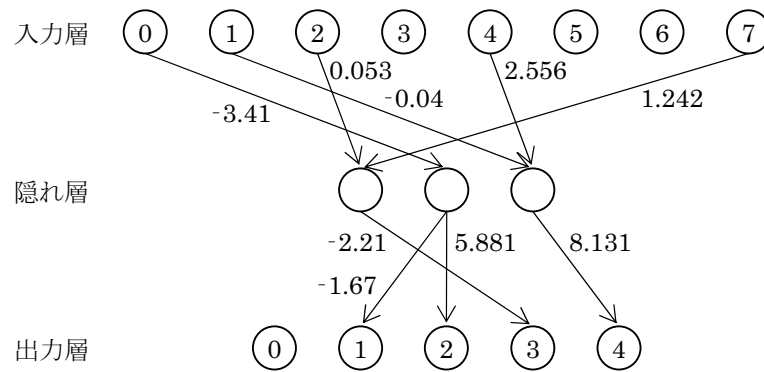


図 2.7: network blueprint の表現型

Step 3-2 《生殖》1点交叉により生成された2つの新しい個体の一方と、親個体の一方のコピーを子個体とする。子個体に対して、ある確率で突然変異を発生させる。

Step 3-3 《生存選択》評価の低い2つの個体と2つの子個体を置き換える。

Step 3-4 上位25%のすべての個体に関して、Step 3-1~3-3を繰り返す。

Step 3-2の突然変異において、neuron集団では遺伝子型のビットを反転させる。network blueprint集団では、次の2種類の突然変異を発生させる。

Mutation 1 別のneuronへのポイントに置き換える。

Mutation 2 50%程度の確率で、neuron集団の生殖において親個体だったneuronへのポイントを、当該neuronから生成された子個体へのポイントに置き換える。

Step 3-1~3-4は非常に強い収束力を持つ戦略であるが、Mutation 2により集団の多様性が維持される。また、交叉で生成されたneuronへのポイントにより新たなneuronに関する探索を行なうことができ、親個体のコピーへのポイントによりneuron集団における有害な突然変異から迅速に復旧することができる。

## 2.4 共生進化がモデル化する生物の行動

一般的な遺伝的アルゴリズムは生物の染色体が環境に適応できるよう進化していく様子をモデル化している。ここでは、共生進化がモデル化している生物の行動について述べるとともに、共生進化の適用対象問題の特徴と設計の難易度によって分類し、列挙する。

### 2.4.1 協働のモデル化

適応度の定義が困難な問題に対して有効な遺伝的アルゴリズムに共生進化がある。生物学上での共生進化は、異種生物が共存するために互いに影響を与え合いながら進化を続けることを意味する。共生

化に基づく遺伝的アルゴリズムでは、2種の異なる生物が互いに害を与え合う“競争”，両者がともに利益を受ける“協調”，片方のみが利益を受ける“寄生”をモデル化したメカニズムによって、ソーティングネットワークの生成 [Hillis 92] やマルチエージェント学習 [Iba 97] など、さまざまな問題において効果を発揮している。

これに対し共生進化は、ある目的を達成するために複数の人が協働する状況をモデル化したアルゴリズムといえよう。目的を効率的に達成する、あるいは高い成果を上げるためには、目的達成に貢献する能力を備えたグループを育成しなければならない。例えば、駅伝競走や体操の団体競技などが具体例として挙げられる。競技で好成績をおさめるためには、選手個人の能力を高めると同時に効果的なチーム編成を考える必要がある。最も足の速い選手が属する駅伝チームでも、他のメンバーが速くなければ優勝できないように、能力の高い選手がいるだけでなくチーム編成も適切でなければならない。また、各選手が役割を分担するような競技の場合には、選手の配置も重要である。同じメンバーで駅伝チームを編成しても、上り坂が得意な選手に上り坂のある区間を走らせる場合と、そうでない場合とでは、明らかに前者の方が良い成績がおさめられるはずである。

競技と共生進化の各要素は以下のように対応付けられる。

チーム	全体解
選手	部分解
チーム編成案の集合	全体解集団
候補選手群	部分解集団
競技の成績	全体解の適応度
個人の評価	部分解の適応度

部分解集団の進化が個人の能力の向上、全体解集団の進化が強いチーム編成の模索に相当する。部分解集団の進化では、各選手は能力の高い選手の影響を受けながら成長する。全体解集団の進化では、既存の案をもとに、さらに強いチーム編成を作り出す。チーム編成が変化していなくても、選手が成長することによってチーム全体の能力が変化することもある。選手は日々成長していくことから、両集団の並行進化が功を奏することは容易に理解できる。

また、2.3.3節において部分解の適応度は全体解の適応度に基づいて決定されると説明したが、“優勝したチームの選手”というように、属しているチームの成績が各選手の評価に大きく影響することからも、この部分解の適応度の定義は妥当であると考えられる。

## 2.4.2 対象問題の分類

2.4.1節ではスポーツ競技を例に共生進化のモデルを説明したが、他にも共生進化の対象となる場面は多く見受けられる。ここでは、音楽を演奏する際のグループに着目し、実装方法の違いによる対象問題の分類を試みる。



共生進化の適用対象となる問題は、部分解と全体解の設計上の観点から分類される。部分解の各遺伝子が表現型に変換されたときの意味に着目すると、次の2種類に分類される。

カテゴリ P1 部分解の遺伝子間に依存関係がない。

カテゴリ P2 部分解の遺伝子間に依存関係がある。

カテゴリ P1 は各遺伝子が独立した意味を持つ場合であり、カテゴリ P2 は他の遺伝子の値によって意味が変動する場合である。

また、全体解の表現により次の3種類に分類される。

カテゴリ W1 全体解は部分解の単なる組合せで表現される。

カテゴリ W2 全体解は部分解の1次元配列で表現される。

カテゴリ W3 全体解は部分解同士を複雑に結合させた構造で表現される。

それぞれ  $P1 < P2$ ,  $W1 < W2 < W3$  の順で部分解と全体解の設計の難易度が増すといえ、カテゴリ P1, P2 とカテゴリ W1 ~ W3 の組合せにより6種類のカテゴリが考えられる。音楽演奏を例とした分類を表 2.1 に示す。

表 2.1: 音楽演奏の分類

	カテゴリ W1	カテゴリ W2	カテゴリ W3
カテゴリ P1	斉唱	重唱	合唱
カテゴリ P2	斉奏	重奏	管弦楽

“歌を歌う”ことがカテゴリ P1 の例として挙げられる。部分解の各遺伝子では歌唱能力を表現すればよい。斉唱は複数の人が同じ旋律を歌うもので、各メンバーは“歌を上手に歌う”という同じ能力の獲得を目指し、メンバーの組合せさえ決まれば演奏の良し悪しも決まるため、カテゴリ P1-W1 といえる。重唱は複数のパートをそれぞれ一人ずつ受け持って歌う形態である。各メンバーの“歌を上手に歌う”という目標は斉唱と同じだが、配置されるパートによって歌う旋律が異なり、適切なメンバー配置が必要となるため、カテゴリ P1-W2 となる。さらに、複数のパートを複数のメンバーで受け持って歌う形態が合唱で、同じ旋律を歌う同パートのメンバーでもパートリーダー等の役割分担が必要であり、メンバー構成は構造的といえるため、カテゴリ P1-W3 に該当する。

歌ではなく楽器を演奏することに焦点を当てると、個人が身につけるべき能力は演奏対象となる楽器によって異なるため、個人の演奏能力を表す遺伝子表現には演奏対象楽器をも含めなければならない。同じ遺伝子座に値の等しい遺伝子が存在しても、対象楽器が異なれば、その遺伝子の意味する能力の内容も異なるはずである。例えば、対象楽器がピアノの場合は鍵盤上での指の動きを表している遺伝子が、シンバルの場合は腕の振り方を表す、という具合である。したがって、楽器演奏は部分解の遺伝子間に依存関係があるため、カテゴリ P2 に属するといえる。

同じ旋律を複数の楽器で演奏する斉奏では、“バイオリンとフルート”，“琴と尺八”など，メンバーの組合せさえ決まれば演奏としては成立するため，カテゴリ P2-W1 といえる．それに対し，重奏のように必要とされる楽器が既定の演奏では，各楽器の奏者を適切に組み合わせなければならないので，カテゴリ P2-W2 となる．さらに，管弦楽は弦楽器，管楽器，打楽器を複数メンバーで受け持って演奏する形態で，合唱同様にメンバー構成は構造的であるため，カテゴリ P2-W3 に相当する．

2.3.4 節で説明した Hierarchical SANE では，部分解 neuron をニューロンの結合先を特定するラベルフィールドと，それぞれに対応する重みフィールドから構成し，遺伝子間に依存関係のある遺伝子表現を用いている．また，全体解 network blueprint は単なる neuron の組合せで表現される．したがって，Hierarchical SANE で扱っているニューラルネットワークはカテゴリ P2-W1 の問題といえ，このカテゴリの問題に関しては共生進化の効果が確認されていると考える．

以降の節では，共生進化の効果が示されたカテゴリ P2-W1 の問題よりも単純なカテゴリ P1-W1 およびカテゴリ P1-W2 の問題について実験を行ない，共生進化の効果を検証する．

## 2.5 単純な問題における共生進化の効果

共生進化の対象問題は，2.4.2 節で述べた通り，設計上の観点から 6 つのカテゴリに分類できる．ここで，全体解を個体とする 1 集団のみを進化させる遺伝的アルゴリズムと，2 集団並行進化による共生進化に関して，設計の比較的容易なカテゴリに属する単純な問題を用いて比較実験を行ない，共生進化によって効果的な解探索が実現されることを確認する．以降，1 集団のみを進化させる遺伝的アルゴリズムを“1 集団 GA”を呼ぶことにする．

対象問題としては，カテゴリ P1-W1 に相当する OneMax 問題と Harik の部分だまし問題 (Harik's partially deceptive problem)[Harik 99]，およびカテゴリ P1-W2 に相当する Floyd 問題 (Floyd's partition problem)[Knuth 96] を取り上げる．

各実験には表 2.2 のパラメータを使用する．ここで  $L_w$  は共生進化における全体解の遺伝子長を表す．1 集団 GA により動作するシステムと共生進化により動作するシステムが同条件で探索できるよう，初期世代の集団としてまったく同じ解候補を持つようにした．このため，部分解集団の個体数を  $I_w \times L_w$  とし，初期世代の全体解集団の各個体はそれぞれ異なる部分解に対してポイントを張るようになる．

表 2.2: 比較実験に使用したパラメータ

パラメータ名	値
全体解集団の個体数 $I_w$	100
部分解集団の個体数 $I_p$	$I_w \times L_w$
突然変異確率 $p_m$	0.001
世代交代回数 $G$	500

初期世代の部分解集団の各個体は全体解からの被参照回数が1回となり、Hierarchical SANEのように最良5個体の適応度の和を部分解の適応度とするのは困難であるため、当該部分解にポインタを張っている全体解のうち、最良のものの適応度を部分解の適応度とする。

### 2.5.1 OneMax 問題

共生進化と1集団GAの比較実験として、最も基本的な最適化問題であるOneMax問題を最初に取り上げる。OneMax問題は以下のように記述される。

#### OneMax 問題

長さ  $N$  のビット列において、値が1のビット数を適応度とすると、適応度の最も高くなるビット列を求めよ。

本実験では  $N = 400$  のOneMax問題を扱うこととする。1集団GAにより動作するシステムでは、長さ400のビット列によって遺伝子列を表現する。共生進化により動作するシステムでは、長さ10のビット列を部分解の遺伝子列とし、 $L_w = 40$ 個の部分解へのポインタ列を全体解とする。部分解の遺伝子間に依存関係がなく、全体解は部分解の単なる組合せで表現されるため、最も単純なカテゴリP1-W1の問題といえる。両システムの世代交代はHSモデルにより行なう。

実験の結果、両システムにおいて適応度400の最適解が得られたが、1集団GAが第384世代で最適解に達したのに対し、共生進化では第48世代という早い段階で最適解を得た。集団内の最良個体の適応度の推移を図2.8に示す。グラフからも共生進化の方が急激に適応度が増加していることを容易に読み取ることができ、共生進化により迅速な学習が行なえたことがわかる。

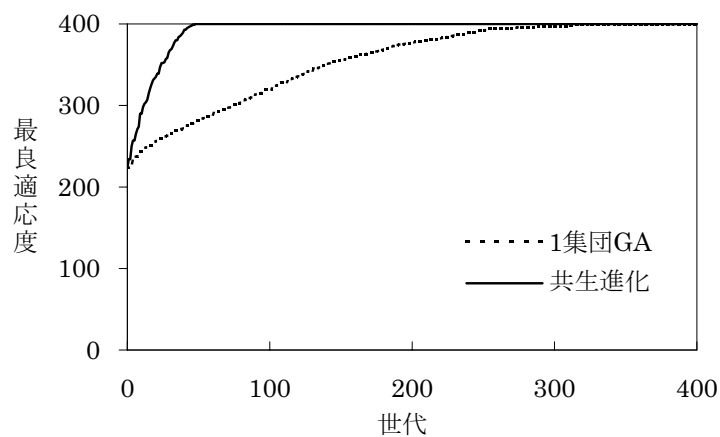


図 2.8: OneMax 問題における最良適応度の推移

### 2.5.2 Harik の部分だまし問題

適応度の高い解がハミング距離の離れたところにあるような問題をだまし問題 (deceptive problem) と呼ぶ。だまし問題では積み木の成長が妨げられ、最適解探索が困難である。ここでは、OneMax 問題を拡張した Harik の部分だまし問題を取り上げ、共生進化の振る舞いを確認する。Harik の部分だまし問題は以下のように記述される。

#### Harik の部分だまし問題

長さ  $M$  のビット列  $B_i$  において、値が 1 のビット数を  $one(B_i)$  と表すとき、 $B_i$  の評価値  $value(B_i)$  を次のように定める。

$$value(B_i) = \begin{cases} one(B_i) & one(B_i) \neq 0 \\ M + 1 & otherwise \end{cases}$$

長さ  $M$  のビット列が  $N$  個連続して並んでいるビット列  $BS = B_1 B_2 \cdots B_N$  の適応度  $fit(BS)$  を次のように定めるとき、適応度の最も高くなる  $BS$  を求めよ。

$$fit(BS) = \sum_{i=1}^N value(B_i)$$

すべてのビットが 0 であるビット列が最適解であるが、次に良い解はすべてのビットが 1 であるビット列であり、これは最適解とのハミング距離が最も大きい解であるため、最適解への到達が難しいというだまし問題である。OneMax 問題と同様の遺伝子列にて表現できるため、単純なカテゴリ P1-W1 の問題といえるが、だまし問題であるという点で OneMax 問題よりも解探索は困難である。音楽演奏の例で表現するならば、“斉唱は上手に歌えるメンバーが多い方が良いが、各メンバーがそれぞれ型にはまらない個性的な歌い方をすると芸術的にも素晴らしい演奏となるので、全員が上手に歌うよりも評価が高くなる” という言い方ができよう。

全ビット長が前節の OneMax 問題と等しくなるように  $M = 4$ ,  $N = 100$  とした Harik の部分だまし問題に対し、共生進化により動作するシステムでは長さ 4 のビット列  $B_i$  を部分解、 $L_w = 100$  個の部分解へのポイント列を全体解として実験を行なった。実験の結果、表 2.3 に示すような解が得られた。共生進化ではすべてのビットが 0 の最適解が得られたのに対し、1 集団 GA では 0000 と 1111 が混在する結果となった。

第 300 世代までの最良適応度の推移を図 2.9 に示す。共生進化では第 91 世代で適応度 500 の最適解が得られたが、1 集団 GA では第 253 世代で適応度が 426 となり、以後第 500 世代まで変化は見られなかった。共生進化によるシステムのみが最適解を得たことはもちろんだが、収束に至るまでの最良適応度が急速に向上していることから、共生進化の優位性が示されたといえる。

共生進化で初期集団の部分解個体をランダムに生成する際、0000 という部分解を生成する確率は  $\frac{1}{16}$  であり、部分解集団の個体数  $I_p$  が  $I_w \times L_w = 10000$  個であることから、初期集団に 0000 という部分解個体が含まれていることは明らかである。部分解個体 0000 を含んだ全体解が他より高い適応

表 2.3: Harik の部分だまし問題で得られた解

1 集団 GA	0000, 1111, 0000, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 0000, 1111, 1111, 0000, 1111, 1111, 0000, 1111, 0000, 1111, 1111, 1111, 1111, 1111, 1111, 0000, 1111, 1111, 1111, 1111, 0000, 0000, 1111, 1111, 0000, 0000, 1111, 1111, 0000, 1111, 1111, 0000, 1111, 1111, 1111, 1111, 1111, 1111, 0000, 1111, 1111, 0000, 1111, 1111, 0000, 1111, 1111, 1111, 1111, 1111, 1111, 0000, 1111, 0000, 0000, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 0000, 1111, 0000, 1111, 1111, 1111, 0000, 0000, 0000, 1111, 1111, 1111, 1111, 1111, 1111, 1111, 0000, 0000
共生進化	0000, 0000

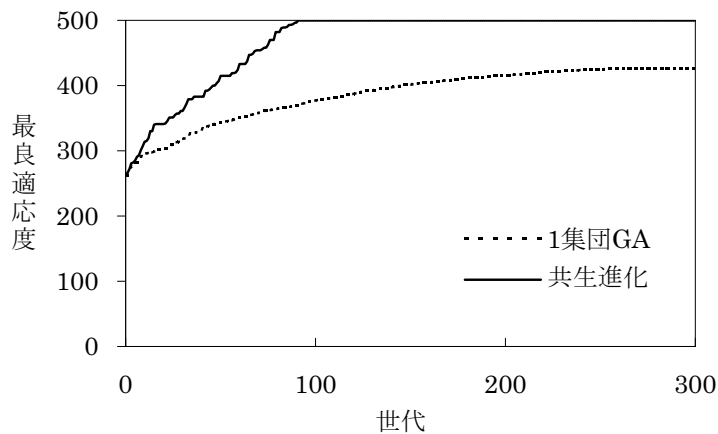


図 2.9: Harik の部分だまし問題における最良適応度の推移

度を持つ確率も当然高くなるため、部分解個体 0000 は高く評価され、部分解集団内に広まっていく。したがって、この実験において最適解が見つかるのは当然といえる。

次に、 $M$  を 1 ずつ増加させ、 $M \times N \approx 400$  となるように  $N$  を設定して実験したところ、 $I_p > 2^M$  である  $M = 11$  までは共生進化で確実に最適解が得られた。 $M = 12$  では、共生進化ですべてのビットが 1 の解が得られたが、1 集団 GA では 2 つの  $B_i$  がすべて 0 のビット列、残りはすべて 1 のビット列が解となった。このことから、すべてのビットが 0 である部分解個体が生成されない場合には、共生進化でもだまし問題の最適解は得られないが、少なくとも局所解には確実に収束し、1 集団 GA と比較して効果的な解探索が行なえるといえる。

### 2.5.3 Floyd 問題 ~ HS モデル

本節および次節では、カテゴリ P1-W2 の問題として次の Floyd 問題を取り上げ、共生進化の効果を詳細に渡って検証する。

#### Floyd 問題

$N$  個の数  $\sqrt{1}, \sqrt{2}, \dots, \sqrt{N}$  を集合 A と集合 B にわけるとき、各集合の要素の和が最も近くなるような集合 A と集合 B を求めよ。

Floyd 問題は本来、 $N = 50$  のときに 10 秒という制限時間内で解を求めることが条件となっているが、本実験では 1 集団 GA と共生進化の動作比較を目的としているため、実行時間については言及しない。また、 $N = 50$  より複雑な  $N = 160$ ,  $N = 256$ ,  $N = 320$  の 3 つの場合について解探索を行なうこととする。

共生進化では、部分解の評価は属している全体解によって決まるため、全体解における組合せが悪いために部分解の評価が低くなることもあり、学習の早い段階で最適解に含まれるべき部分解個体が消滅するという問題点が考えられる。例えば、前節の Harik のだまし問題で 0000 という部分解個体が全体解個体に参照されていても、その他の参照先の部分解個体がすべて 0001 であれば、適応度は  $5 + 1 \times 99 = 104$  となり、適応度の期待値  $\sum_{k=1}^5 (k \times \frac{1}{5}) \times 100 = 300$  よりも低くなる。このような状況では、最適解に含まれるべき部分解個体 0000 の適応度が低くなり、進化により死滅してしまう。部分解の働きが全体解内の位置に依存する Floyd 問題においては、組合せは良くても並び方が悪いために評価が下がることもあるため、良い部分解の早期消滅の問題はさらに影響が大きいと考えられる。

1 集団 GA と共生進化の比較に加え、良部分解の早期消滅が共生進化に及ぼす影響を調べるため、第 20 世代までは部分解集団は進化させずに全体解集団のみを進化させ、第 21 世代以降に 2 集団並行進化を行なう共生進化においても、Floyd 問題の実験を行なうこととする。以降では、この共生進化を“共生進化  $\beta$ ”と記す。

1 集団 GA により動作するシステムでは、長さ  $N$  のビット列によって遺伝子列を表現する。  $i$  番目の遺伝子は整数  $i$  がいずれの集合に属するかを表しており、  $i = 1$  のとき集合 A、  $i = 0$  のとき集合 B に属すると判断する。共生進化および共生進化  $\beta$  により動作するシステムでは、32 ビットの遺伝子列を部分解とし、  $L_w = \frac{N}{32}$  個の部分解へのポインタ列を全体解とする。部分解の各遺伝子是对応する整数が集合 A に入るか否かを表しており、遺伝子間に依存関係はないが、全体解の何番目の要素として参照されるかによって対応する整数が異なるため、Floyd 問題はカテゴリ P1-W2 の問題であるといえる。

3 つのシステムの世代交代は HS モデルにより行なう。適応度関数には集合 A と集合 B の要素の和の差を用い、適応度が低い個体ほど良いと判断する。

表 2.2 のパラメータを用いて実験を行なった結果、各システムの  $N = 160, 256, 320$  における集合 A と集合 B の要素の和の差は表 2.4 に示す通りとなった。  $N = 160$  の場合に共生進化  $\beta$  で共生進化よりも若干適応度の低い、すなわち良い解が得られているが、  $N = 256, 320$  の場合には共生進化で他よりも桁違いに良い解が得られている。したがって、全般的には共生進化によって効果的な解探索が行なわれているといえる。

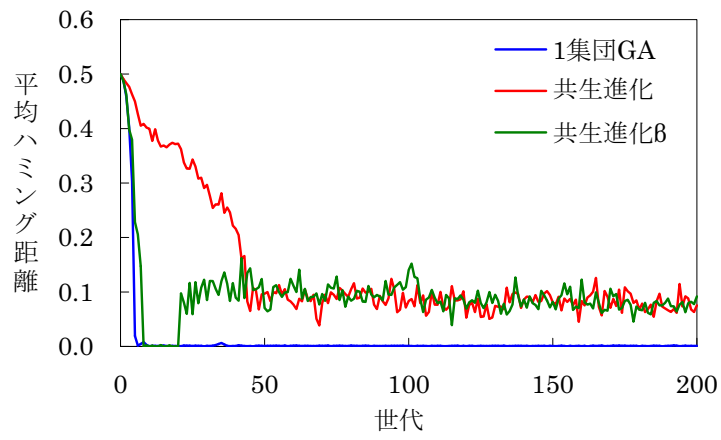
表 2.4: 集合 A と集合 B の要素の和の差 (HS モデル)

	$N = 160$	$N = 256$	$N = 320$
1 集団 GA	0.19265	0.12408	0.06254
共生進化	0.00294	0.00052	0.00205
共生進化 $\beta$	0.00123	0.02945	0.01653

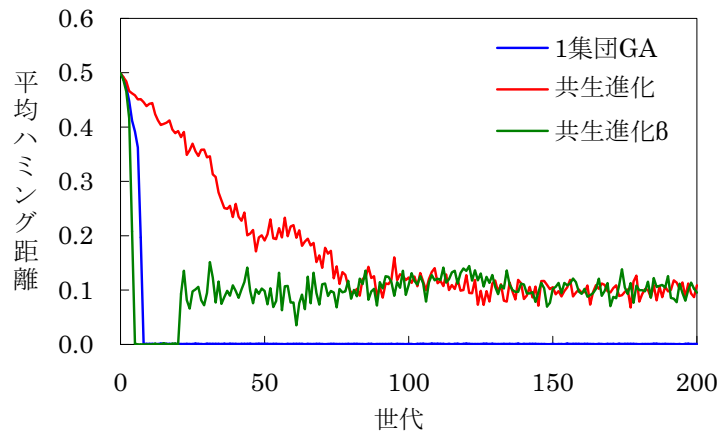
次に、集団の多様性と適応度の推移に着目する。集団の多様性は、集団内の個体の異なり具合によって表現されるため、各個体間の平均ハミング距離  $\Phi$  により表現する。平均ハミング距離は式 (2.1) で定義される。

$$\Phi = \frac{2}{I_w(I_w - 1)N} \sum_{i=1}^{I_w} \sum_{j=i+1}^{I_w} \text{ham}(i, j) \quad (2.1)$$

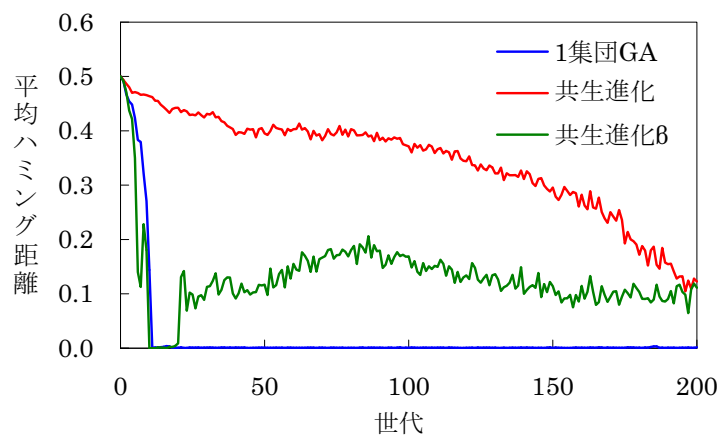
ここで  $\text{ham}(i, j)$  は  $i$  番目の個体と  $j$  番目の個体のハミング距離を表す。世代ごとの平均ハミング距離の推移を図 2.10 に示す。1 集団 GA では、いずれの  $N$  においても急激に多様性が失われている。  $N = 160$  では第 6 世代、  $N = 256$  では第 8 世代、  $N = 320$  では第 11 世代で平均ハミング距離が 0.01 未満となり、以後はほとんど変化が見られない。一方、共生進化では多様性が維持されており、図 2.10(a) および (b) では、平均ハミング距離がある程度減少した後でも一定の多様性が保たれている様子がうかがえる。最も興味深い推移を見せたのが共生進化  $\beta$  である。全体解集団のみで進化を行なっている間は 1 集団 GA と同じように平均ハミング距離が急激に低下したが、部分解集団での進化を始めた途端、平均ハミング距離が向上し、共生進化と同程度の値に落ち着いている。この結果から、2 集団並行進化は集団の多様性維持に非常に効果的であるといえる。



(a)  $N = 160$



(b)  $N = 256$



(c)  $N = 320$

図 2.10: Floyd 問題 (HS モデル) における平均ハミング距離の推移



最良適応度の推移を図 2.11 に示す。1 集団 GA では、 $N = 160$  の第 4 世代、 $N = 320$  の第 2 世代で大きく最良適応度が減少しており、 $N = 160$  では第 32 世代、 $N = 320$  では第 12 世代と第 182 世代でさらに減少している。図 2.10 と照らし合わせると、最初の大きな変化は多様性が残っている間の変化であり、平均ハミング距離が 0.01 未満になった後で 2, 3 回目の変化が起こっているといえる。

共生進化では、 $N = 160$  で第 3, 6, 12, 28 世代、 $N = 256$  で第 15, 22, 43 世代、 $N = 320$  で第 2, 11, 17, 33, 35 世代と、比較的早い段階で最良適応度が減少している。共生進化  $\beta$  で大きく最良適応度が減少したのは、 $N = 160$  で第 25 世代、 $N = 256$  で第 76 世代、 $N = 320$  で第 16, 23 世代であり、1 集団 GA や共生進化に比べて遅い世代となっている。良い部分解の早期消滅を避けるために、学習初期で全体解集団のみを進化させる手法は、逆に全体としての進化を妨げているといえよう。

$N = 256$  の場合には、初期世代で他と比べて低い適応度が得られた。1 集団 GA では第 200 世代までに初期世代より良い個体を生成することはできなかったが、共生進化と共生進化  $\beta$  ではさらに良い個体が早期に得られている。

最良適応度が減少した前後の平均ハミング距離の推移を拡大して示したグラフを図 2.12 に示す。グラフ中の黒点が適応度が減少した世代を表している。(a)~(c) は集団の多様性が失われた後での 1 集団 GA の推移を示しているが、適応度減少後、平均ハミング距離は若干増加するものの、すぐに減少している。類似した個体が集団を占めている状態で、突発的に良い個体が生成され、一時的に良い個体の形質が集団内に広まることで集団の多様性が増すが、すぐに当該個体へと集団が収束しているものと考えられる。

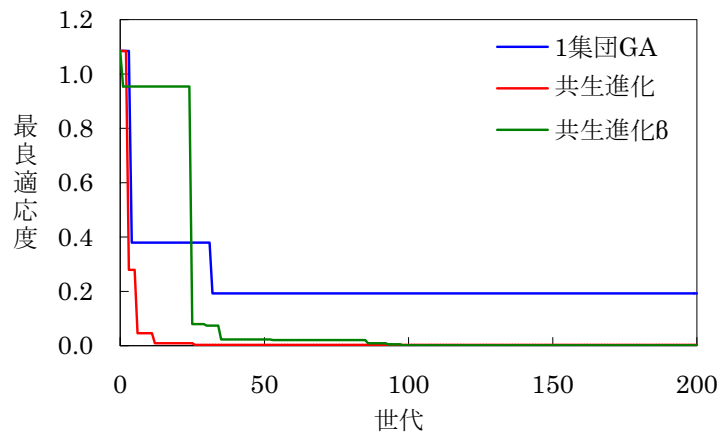
(d) は集団の多様性が残っている間の推移を示している。1 集団 GA と共生進化の両者において、(a)~(c) とは異なり、最良適応度減少後に平均ハミング距離が増加していない。1 集団 GA については、最良適応度減少によって急激に平均ハミング距離も減少している。多様性が維持されている状態で良い個体が生成されても、その後で集団の多様性が増すことはないが、1 集団 GA では急激に多様性が失われると考えられる。

#### 2.5.4 Floyd 問題 ~ MGG モデル

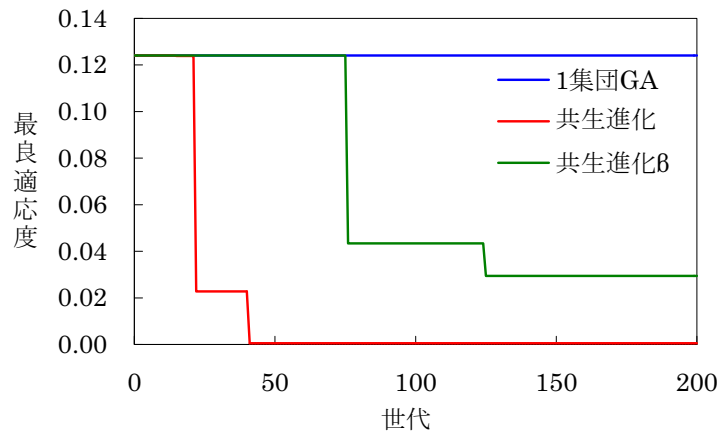
2.5.3 節では、共生進化のために設計された HS モデルにより世代交代を行なったため、1 集団 GA において集団の多様性が急激に低下したものと考えられる。ここでは、同程度の多様性がある場合における 1 集団 GA と共生進化との動作の違いを調べるため、MGG (Minimal Generation Gap) モデル [佐藤 97] により同様の実験を試みる。MGG モデルは、局所解収束の回避と進化的停滞の抑制を意図して考案された世代交代モデルであり、集団の多様性の維持に効果的であることが示されている。MGG モデルの処理手順は以下の通りである。

Step 3-1 《複製選択》ランダムに異なる 2 つの個体を選択し、親個体とする。

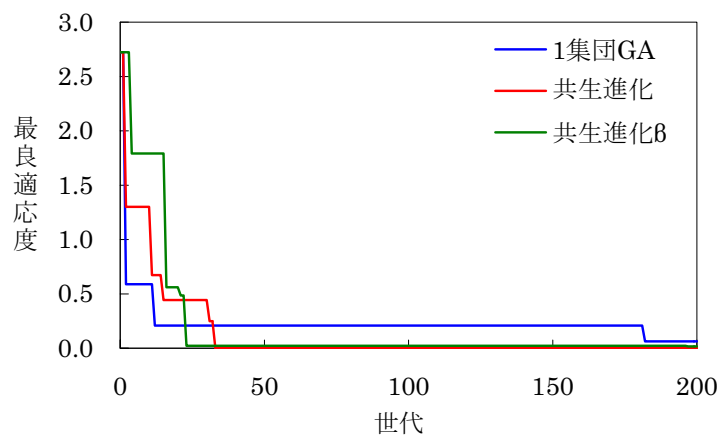
Step 3-2 《生殖》親個体から複数の子個体を生成し、親個体と子個体をあわせて家族とする。



(a)  $N = 160$

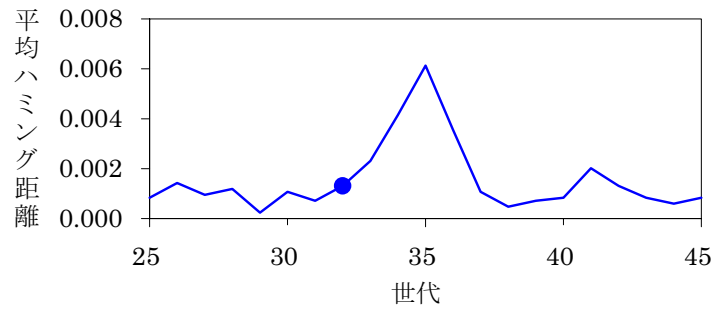


(b)  $N = 256$

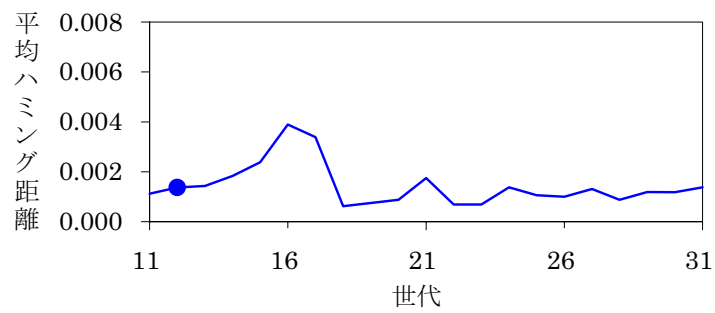


(c)  $N = 320$

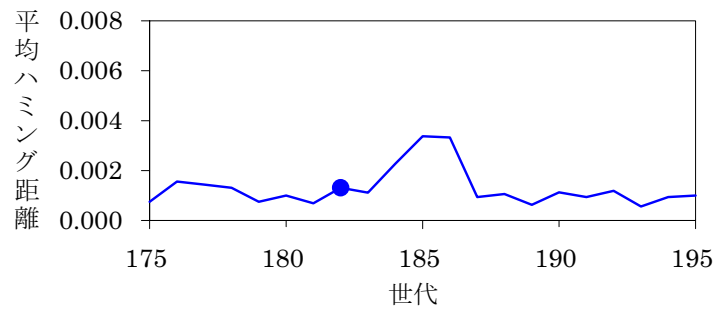
図 2.11: Floyd 問題 (HS モデル) における最良適応度の推移



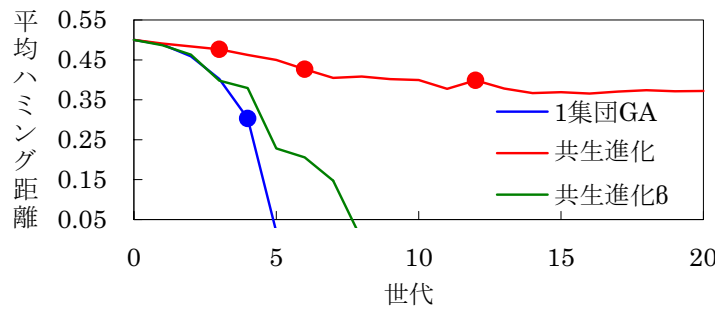
(a)  $N = 160$  (1 集団 GA)



(b)  $N = 320$  (1 集団 GA)



(c)  $N = 320$  (1 集団 GA)



(d)  $N = 160$

図 2.12: Floyd 問題 (HS モデル) における最良適応度減少前後の平均ハミング距離の推移

Step 3-3 《生存選択》家族のうち，最良 1 個体およびルーレット選択により選ばれた 1 個体を親個体と置き換える．

本実験では，1 点交叉と確率  $p_m$  での突然変異により生成された 2 つの個体，および親個体にそれぞれ確率  $p_m$  で突然変異を発生させた 2 つの個体の計 4 つの個体を子個体として生成する．共生進化により動作するシステムでは，全体解集団において MGG モデル，部分解集団において HS モデルにより世代交代を行なう．

表 2.2 のパラメータを用い，初期世代の集団は 2.5.3 節と同様の解候補を持つようにして，実験を行なった．実験の結果，集合 A と集合 B の要素の和の差は表 2.5 のようになった．1 集団 GA の  $N = 160$  を除く場合において表 2.4 以上の値となっており，MGG モデルによる進化では HS モデルよりも良い解は得られないが，3 つのシステムを比較すると，共生進化により動作するシステムにおいて最も良い解が得られている．

表 2.5: 集合 A と集合 B の要素の和の差 (MGG モデル)

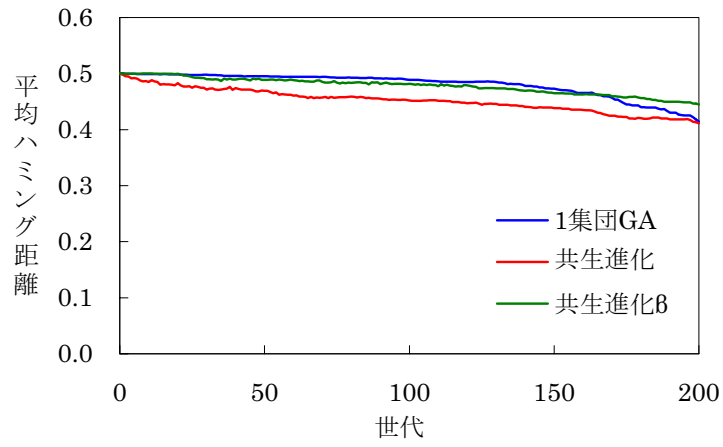
	$N = 160$	$N = 256$	$N = 320$
1 集団 GA	0.03933	0.12408	0.16230
共生進化	0.01305	0.06059	0.11939
共生進化 $\beta$	0.02271	0.11006	0.30937

平均ハミング距離と最良適応度の推移をそれぞれ図 2.13，図 2.14 に示す．MGG モデルを採用したことにより，すべてのシステムにおいて同程度の高い多様性が維持されていることがわかる．図 2.11 に比べて遅い世代においても最良適応度が頻繁に減少しているのは，多様性の高さが原因であると考えられる．

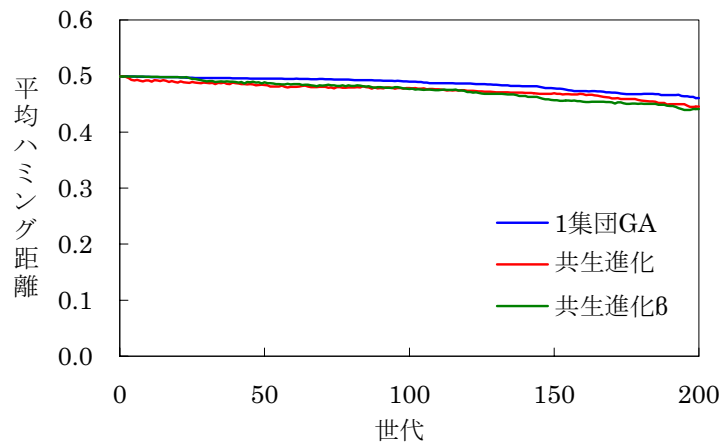
$N = 160$  の最良適応度の変化に関して，1 集団 GA では第 8，13 世代と 2 回の減少が見られるが，共生進化では第 1 世代で大きく減少したのみで，以後は変化が見られなかった．また， $N = 320$  の最良適応度は，1 集団 GA で第 8，137，159，173 世代，共生進化で第 1，3，29，30，36，48，75 世代，共生進化  $\beta$  で第 24，28 世代と段階的に減少した． $N = 256$  の 1 集団 GA では 2.5.3 節と同様に初期世代から最良適応度の変化はなかったが，共生進化では第 6，39，40，102 世代，共生進化  $\beta$  では第 110 世代において，前の世代よりも良い適応度を得た．以上の結果から，集団内の個体が同程度の多様性を持つ場合にも，共生進化は 1 集団 GA や共生進化  $\beta$  よりも迅速に有用な解に到達したといえる．

図 2.15 は，共生進化により動作するシステムにおいて，各部分解個体における全体解個体からの被参照数の推移を示したグラフである．進化の終盤では 6 個の全体解個体から参照される部分解個体も見られたが，その個数は少数であるため，図 2.15 には被参照が 3 個以下の結果のみを表示した．

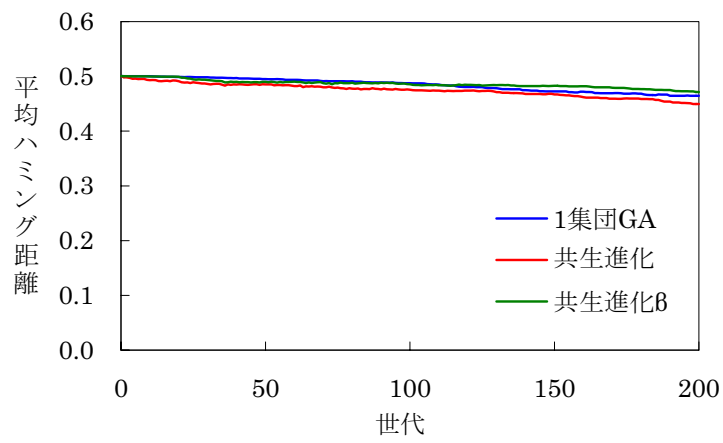
被参照数の推移は  $N$  の値に依らず，同様の傾向があることがグラフからわかる．初期世代ではすべての部分解個体が 1 つの全体解個体からポイントを張られているが，進化が進むに従って 1 つの



(a)  $N = 160$

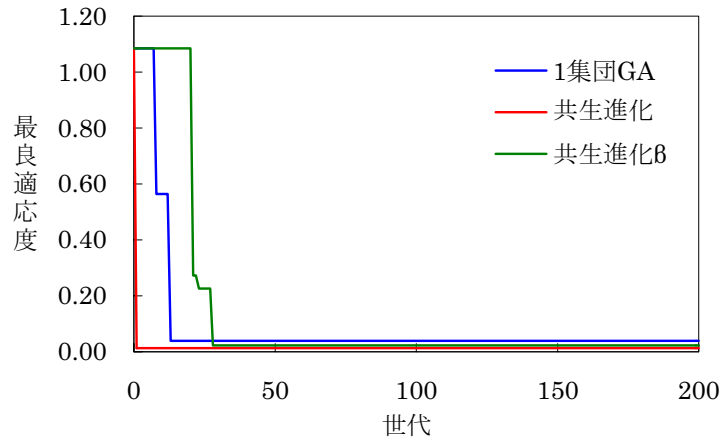


(b)  $N = 256$

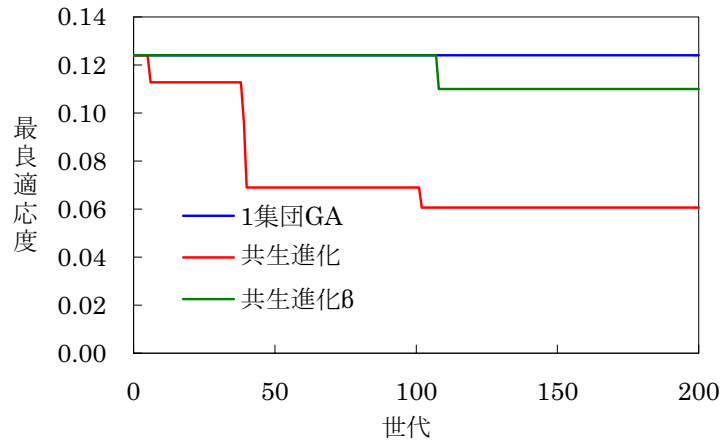


(c)  $N = 320$

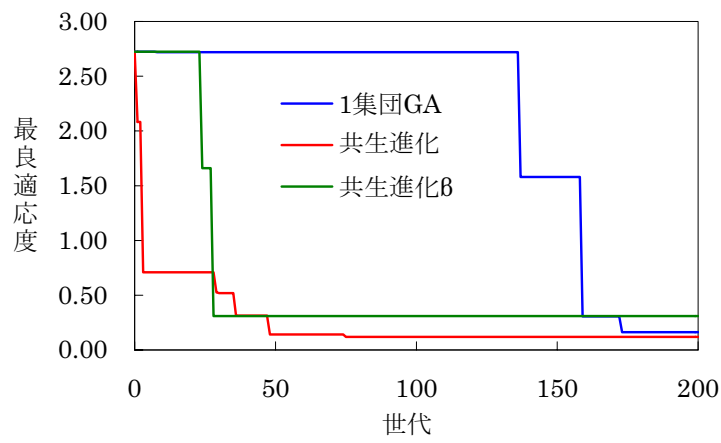
図 2.13: Floyd 問題 (MGG モデル) における平均ハミング距離の推移



(a)  $N = 160$

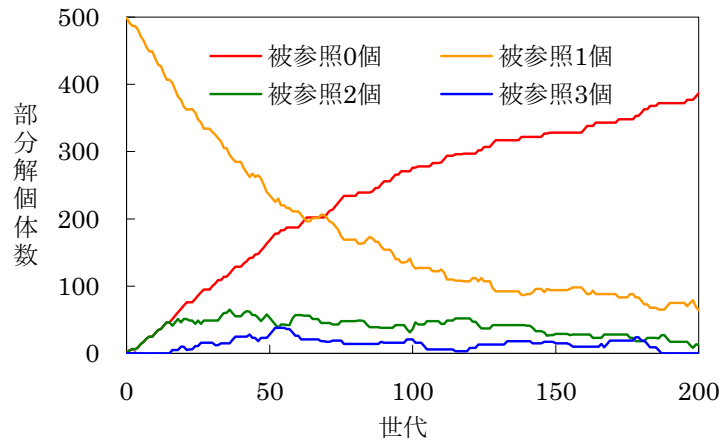


(b)  $N = 256$

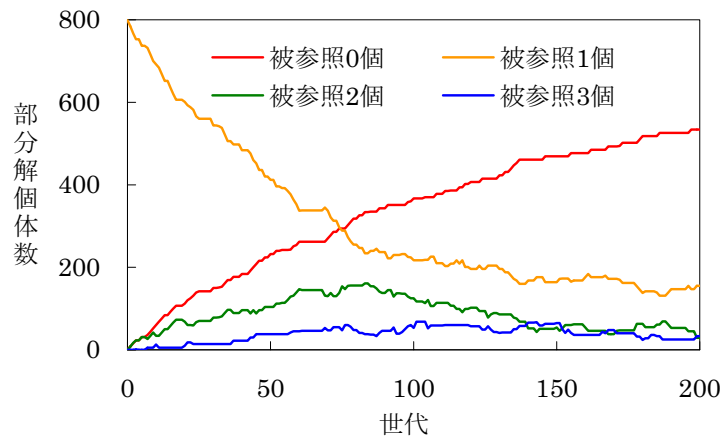


(c)  $N = 320$

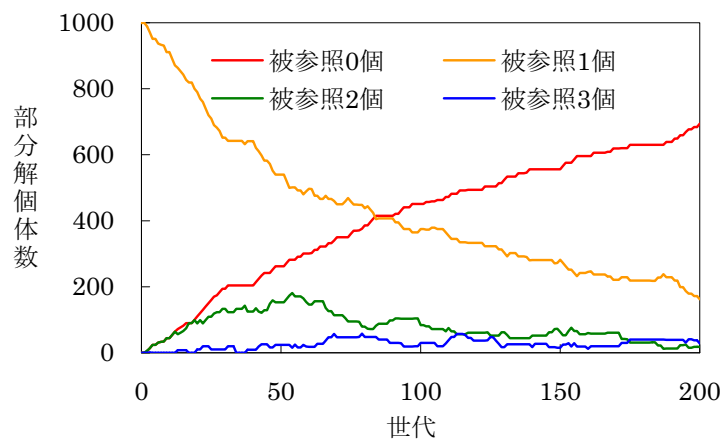
図 2.14: Floyd 問題 (MGG モデル) における最良適応度の推移



(a)  $N = 160$



(b)  $N = 256$



(c)  $N = 320$

図 2.15: Floyd 問題 (MGG モデル) における被参照数ごとの部分解個体数の推移

全体解個体から参照される部分解個体は減っていき、逆にどの全体解個体からも参照されない部分解個体が増えていく。また、2つ以上の全体解個体から参照される部分解個体も見られるようになる。

しかし、被参照数が0の個体が増えても、一部の部分解個体のみが参照されるのではなく、ある程度の個数の部分解個体がいずれかの全体解から参照されている。同一の遺伝子列を持つ部分解個体が多く存在すると、全体解の表現型としては同じものが生成されるが、全体解の表現型は部分解の進化により変化するため、異なる部分解へのポインタ列からなる全体解個体は、進化の過程で変化する可能性が高いと考えられる。したがって、被参照数の推移からも、共生進化における多様性維持の様子が示されたといえる。

### 2.5.5 2集団並行進化の効果

2.5.1節～2.5.4節における実験から、全体解を個体とする1集団を進化させるよりも、全体解と部分解を個体とする2集団を並行して進化させる共生進化の方が、有用な解を早期に獲得できることが示された。これは、部分解を個体とする進化の『“効果的な解”のすばやい獲得』という特長と、全体解を個体とする進化の『“最適な解”への確実な収束』という特長の両者が統合された結果であり、共生進化の意図する効果が実現されたものと考えられる。

2.3節で述べた通り、共生進化は統治分割法に基づいた手法である。対象とする問題が小規模な問題に分割できる場合には、部分解とそれらの組合せをそれぞれ集団の個体とし、2集団を並行して進化させる共生進化が効果的であるといえる。すなわち、求めるべき解の部分、およびそれらの組合せの学習をバランスよく行なうことが必要な問題に対して、共生進化の適用可能性が期待される。

## 2.6 共生進化の関連研究

SANEの拡張システムとして、カオス制御のためのニューラルネットワーク学習システム [Weeks 97] や、一般的な系列を予測できる回帰ニューラルネットワークの学習システム [Wierstra 05] などが構築され、それぞれ望ましい成果が得られている。また、共生進化の考えを利用して異なるタイプのファジールールを1集団で進化させ、ファジー制御システムのルールベースを生成する研究もなされている [Juang 00, Mahfouf 01, Jamei 04]。Zhaoは共生進化の一般形である社会モデルを提案し、パターン認識エージェントの設計に適用している [Zhao 98, Zhao 00]。

上述以外にも、不確かな環境におけるニューラルネットコントローラ的设计 [Sebald 94]、協調自動化エージェントの行動戦略生成 [Puppala 98]、免疫組織を模倣したパターン分類システム [Hunt 96] など、さまざまな研究において共生進化の考えが用いられている。



## 2.7 まとめ

本章では、遺伝的アルゴリズムの一手法である共生進化、特に全体解集団と部分解集団という 2 集団を並行して進化させる共生進化に着目し、特徴と基本動作、および効果について述べた。2.6 節で述べたように、共生進化の考え方はニューラルネットワークをはじめとするいくつかの分野に適用され、高い成果が得られている。2.5 節では、非常に単純な構造の遺伝子列で解が表現される場合について、共生進化が効果的に動作することを示した。これは、共生進化が最適解探索手法として基本的な有効性を備えていることを示唆するものと考えられ、既存研究では扱われていないさまざまな分野への適用可能性が期待される。

以下の章では、カテゴリ P2-W2 の問題として帰納論理プログラミング、カテゴリ P2-W3 の問題として決定木生成を取り上げ、それぞれの学習に共生進化を適用する手法について述べるとともに、本章で示した問題より解構造が複雑な問題においても共生進化が効果的であることを示す。

## 第3章 帰納論理プログラミングにおける予測精度向上

帰納論理プログラミングシステムにより生成される仮説は、学習に用いた訓練事例だけでなく、未知事例をも正しく説明することが望まれる。未知事例の分類における予測精度の向上を目的として、共生進化に基づく帰納論理プログラミングの手法を提案する。帰納論理プログラミングの最適仮説探索において共生進化が有効であることを評価実験により示す。

### 3.1 帰納論理プログラミング

帰納論理プログラミング (Inductive Logic Programming; ILP) は、帰納推論と論理プログラミングを統合した機械学習の手法であり、背景知識の利用と一階述語論理による知識表現を特徴とする [古川 97]。以下では、帰納論理プログラミングの基本的な枠組みと一般的な学習アルゴリズムについて説明する。

#### 3.1.1 基本的な枠組み

帰納論理プログラミングは分類問題を解決するための枠組みである。与えられた複数の事例から、あるカテゴリに属するものの性質を示すルールを帰納推論により生成し、生成されたルールによって新しい事例が当該カテゴリに属するか否かを判定する。海獣の性質を表す事例を表 3.1 に示す。この例では、各動物が口腔に持つもの、成体における体長、クジラへの分類の可否、イルカへの分類の可否が示されている。ただし、体長は 4m 未満のものを小型、4m 以上のものを大型と表記している。また、表 3.1 の他に次の知識が与えられているものとする。

『口腔に歯を持つクジラはハクジラである』

表 3.1: 海獣の性質を表す事例

海獣名	口腔	体長	クジラ	イルカ
シロナガスクジラ (blue whale)	ヒゲ	大型	yes	no
マッコウクジラ (sperm whale)	歯	大型	yes	no
マイルカ (common dolphin)	歯	小型	yes	yes
ゴマフアザラシ (spotted seal)	歯	小型	no	no
コククジラ (gray whale)	ヒゲ	大型	yes	?
スナメリ (finless porpoise)	歯	小型	yes	?

表 3.1 の事例と上記の知識を用いて、イルカへの分類の可否を判定するルールを帰納論理プログラミングにより生成するとき、入力は図 3.1 に示すような正事例 (positive example)、負事例 (negative example)、背景知識 (background knowledge) となる。帰納論理プログラミングにおける入力と出力は、すべて述語論理を用いて表現されるが、図 3.1 では Prolog と同様の Horn 節により表現されている。各述語の意味は以下の通りである。

```
dolphin(A)           ... 「A はイルカである」
oral(A, B)           ... 「A は口腔に B を持つ」
size(A, large/small) ... 「A は大型/小型である」
whale(A, yes/no)     ... 「A はクジラである/ない」
toothed_whale(A, yes/no) ... 「A はハクジラである/ない」
```

正事例：

```
dolphin(common_dolphin).
```

負事例：

```
dolphin(blue_whale).
```

```
dolphin(sperm_whale).
```

```
dolphin(spotted_seal).
```

背景知識：

```
oral(blue_whale, whalebone).
```

```
oral(sperm_whale, tooth).
```

```
oral(common_dolphin, tooth).
```

```
oral(spotted_seal, tooth).
```

```
oral(gray_whale, whalebone).
```

```
oral(finless_porpoise, tooth).
```

```
size(blue_whale, large).
```

```
size(sperm_whale, large).
```

```
size(common_dolphin, small).
```

```
size(spotted_seal, small).
```

```
size(gray_whale, large).
```

```
size(finless_porpoise, small).
```

```
whale(blue_whale, yes).
```

```
whale(sperm_whale, yes).
```

```
whale(common_dolphin, yes).
```

```
whale(spotted_seal, no).
```

```
whale(gray_whale, yes).
```

```
whale(finless_porpoise, yes).
```

```
toothed_whale(X, yes) :- whale(X, yes), oral(X, tooth).
```

図 3.1: イルカ判定ルールを生成するための入力

「イルカである」を目標概念として帰納推論を行なう場合、「マイルカはイルカである」という目標概念として正しい事例を正事例、「マッコウクジラはイルカではない」など目標概念として正しくない事例を負事例とする。また、目標概念に関連する知識を背景知識とする。背景知識としては、「シロナガスクジラは口腔にヒゲを持つ」、「ゴマフアザラシはクジラではない」というような事実だけでなく、「口腔に歯を持つクジラはハクジラである」というルールをも与えることができる。

正事例を被覆し、かつ負事例を被覆しないことを目標として、背景知識をもとに帰納推論を行なうと、以下のような仮説が生成される。この仮説は「小型のハクジラはイルカである」ことを意味する。

$$\text{dolphin}(X) \text{ :- } \text{toothed\_whale}(X, \text{yes}), \text{size}(X, \text{small}). \quad (3.1)$$

負事例は、「口腔に歯を持つものはイルカである」、「クジラはイルカである」というような過度に一般化された仮説の生成を防止する役割を担っている。また、ルール形式の背景知識は、表現の拡張に有効である。ハクジラに関するルールと同等の知識を事実形式で表すためには、考え得るすべてのハクジラに関する事例を列挙しなければならない。ハクジラに関するルールの記述がない場合には、「口腔に歯を持つ小型のクジラはイルカである」という仮説が生成される。この仮説は式(3.2)のように表現されるが、式(3.1)の方がより適切な表現であるといえる。

$$\text{dolphin}(X) \text{ :- } \text{whale}(X, \text{yes}), \text{oral}(X, \text{tooth}), \text{size}(X, \text{small}). \quad (3.2)$$

式(3.1)の仮説と図3.1の背景知識を用いることにより、次のようにコククジラとスナメリに関する分類が決定される。コククジラはイルカではなく、スナメリはイルカである。

```
?- dolphin(gray_whale).
no
?- dolphin(finless_porpoise).
yes
```

以上のように、仮説の生成は目標概念としての正誤が明らかな事例をもとに行なうが、生成された仮説の良否は未知事例における説明の正確さにより決定される。したがって、帰納論理プログラミングにおいて予測精度の向上は重要な研究目標といえる。

### 3.1.2 集合被覆アルゴリズム

正事例集合を  $E^+$ 、負事例集合を  $E^-$ 、背景知識を  $BK$  と表すと、帰納論理プログラミングは次の式を満たす仮説集合  $H$  を導出する処理である。

$$\begin{cases} H \cup BK \models E^+ \\ H \cup BK \cup E^- \not\models \\ H = \{h_1, h_2, \dots, h_n\} \end{cases} \quad (3.3)$$

一般的に、仮説集合  $H$  の要素である仮説  $h_1 \sim h_n$  は、集合被覆アルゴリズムに従って生成される。集合被覆アルゴリズムを以下に示す。

- Step 1  $H := \emptyset, E := E^+, B := BK, i := 1$   
 Step 2 候補仮説束内を探索し、最適仮説  $h_i$  を導出する。  
 Step 3  $H := H \cup h_i, B := B \cup h_i$   
 Step 4  $E' := \{e' | e' \in E \wedge B \models e'\}$   
 Step 5  $E := E \setminus E', i := i + 1$   
 Step 6  $E = \emptyset$  となるまで Step 2~5 を繰り返す。

Step 2 の候補仮説束内の探索手法としては、大別してボトムアップアプローチとトップダウンアプローチがある。前者では、相対最小汎化 (relative least general generalization) に基づいて事例を一般化していくことによって目的の仮説を求める。後者では、ボディが空の節のボディにリテラルを追加し、節を精密化していくことによって目的の仮説を生成する。これまでに開発された代表的な帰納論理プログラミングシステムのうち、GOLEM [Muggleton 90] ではボトムアップアプローチ、FOIL [Quinlan 95] ではトップダウンアプローチが採られている。また、Progol [Muggleton 95] や GKS [Mizoguchi 95] のように両アプローチを併用するシステムもある。

## 3.2 共生進化に基づく最適仮説探索

トップダウンアプローチにより最適仮説探索を行なう代表的なシステムとして FOIL と Progol が挙げられるが、入力が基底アトムに限定されている FOIL に対し、Progol は任意の Horn 節により背景知識が記述できるため、種々の問題への適用が可能である。以下では、Progol の学習アルゴリズムの概要を説明した上で、最適仮説探索を共生進化により行なう手法について検討する。

### 3.2.1 Progol の学習アルゴリズム

Progol では、3.1.2 節に記した集合被覆アルゴリズムに基づいて学習を行なうが、Step 2 の最適仮説探索では、最弱仮説 (Most Specific Hypothesis; MSH) をボトム、最も一般的な単位節をトップとする概念ラティスを探索する。Progol における最適仮説探索の手順を以下に記す。

- Step 2-1 1つの正事例  $e_i \in E$  と背景知識  $B$  から、逆伴意法によって最弱仮説  $m_i$  を求める。  
 Step 2-2 最弱仮説  $m_i$  をボトムとする仮説空間を探索し、最適仮説  $h_i$  を導出する。

探索空間を制限するため、モード宣言とタイプ情報という2つの言語バイアスが与えられる。Step 2-1 と Step 2-2 で生成されるすべての仮説、すなわち仮説空間に含まれる仮説のリテラルは、モード宣言とタイプ情報による制約を満たしている。両者により、仮説中の各述語の位置、および各項の取りうる領域と入出力モードが指定される。

入出力モードとしては、仮説を構成するリテラルの各項に +, -, # のいずれかが割り当てられる。+モードと -モードは項がそれぞれ入力引数と出力引数であることを表し、#モードは定数項の

まま汎化の対象とならないことを表す。ボディの+モードの項には、既出していない変数が入ることはない。ここで、当該変数を含むリテラルよりも左側のリテラルに同名の変数が含まれる場合、その変数は「既出している」という。

モード宣言とタイプ情報の記述例を図3.2に示す。modehは仮説のヘッド、modebは仮説のボディのリテラルに関する記述である。第2引数の括弧の前の文字列east, has\_car, shape, load, wheelsがリテラルの述語名を表し、括弧内で各項のモード+, -とタイプtrain, car, shape, intを指定する。第1引数は、第2引数のパターンに合う解を求めるべき個数、すなわちバックトラック回数を表す。“\*”は解の個数に制限を加えないことを意味する。

```
:-modeh(1,east(+train)).
:-modeb(*,has_car(+train,-car)).
:-modeb(1,shape(+car,-shape)).
:-modeb(1,load(+car,-shape,-int)).
:-modeb(1,wheels(+car,-int)).
```

図 3.2: モード宣言とタイプ情報

Step 2-2では、最小記述長原理に基づくA\*-likeアルゴリズムに従って、仮説空間を探索し、最適仮説を発見する。仮説空間は、トップの仮説から順に精密化演算子を適用して特殊化された仮説により構成された概念ラティスである。仮説空間の例を図3.3に示す。精密化演算子とは、モード宣言とタイプ情報による制約を満たしながら、次の2つの機能を果たすものである。

機能1 異なる変数同士が同じ値をとるようにする。

機能2 仮説のボディに新たなリテラルを追加する。

上記2機能によって、元の仮説に包摂される仮説が逐次生成され、構築される概念ラティスは包摂束となる。図3.3の太線は機能1による特殊化を表し、普通線は機能2による特殊化を表す。最弱仮説を仮説空間のボトムとするため、機能1の代入制約は最弱仮説よりも特殊にならない範囲で与えられ、機能2で新たに追加されるリテラルは最弱仮説に含まれるリテラル、あるいはそれを一般化したものに限られる。したがって、生成される仮説は、最弱仮説に含まれるリテラルのうちのいくつかを組み合わせ、代入制約を緩和したものといえる。

訓練事例におけるノイズに対処するため、探索中の仮説の被覆事例数における負事例の割合が制限されている。負事例の被覆割合が制限を越えていない最適仮説 $h_i$ の集合を $H$ とする。

### 3.2.2 帰納論理プログラミングシステム ILP/SE

Progolでは、3.2.1節で述べた通り、最弱仮説に含まれるリテラルを適度に一般化し、必要なリテラルのみを組み合わせた節が最適仮説となる。したがって、最適仮説探索には部分と全体をバランスよく最適化する共生進化が有効であると考えられる。Progolの最適仮説探索を共生進化により行な

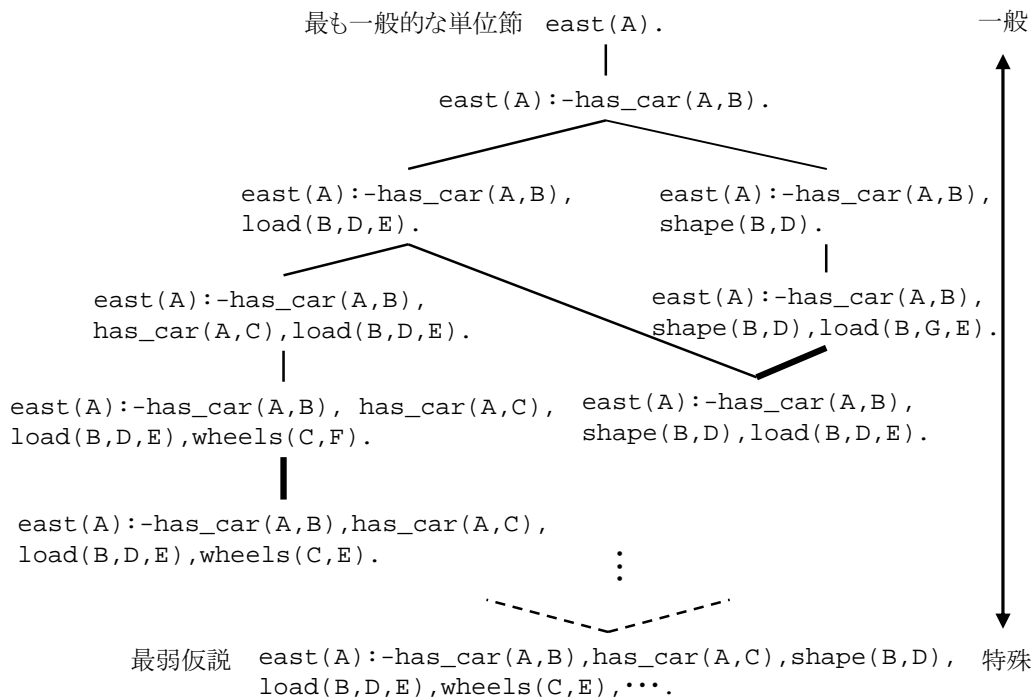


図 3.3: 仮説空間の例

帰納論理プログラミングシステム ILP/SE(Inductive Logic Programming / Symbiotic Evolution) を構築する。ILP/SE では、仮説を全体解、リテラルを部分解として、3.2.1 節の Step 2-2 を共生進化により行なう。部分解と全体解の集団の進化が、それぞれ精密化演算子の機能 1 と機能 2 による特殊化の度合の学習に相当する。

ILP/SE の設計とアルゴリズムの詳細については次節にて説明する。

### 3.3 ILP/SE の設計とアルゴリズム

帰納論理プログラミングシステム ILP/SE における最適仮説探索手法の詳細について説明する。ILP/SE の最適仮説探索では、逆伴意法により生成された最弱仮説と、モード宣言およびタイプ情報を利用する。

以降、最弱仮説に含まれるリテラルの数を  $N_l$  で表し、最弱仮説を次のように表現する。

$$L_0 :- L_1, L_2, L_3, \dots, L_{N_l-1}.$$

最弱仮説中の各リテラルの添字  $0 \sim N_l-1$  をリテラル ID と呼ぶ。

#### 3.3.1 アンサンブル学習

学習精度を向上させる手法にアンサンブル学習がある。複数の分類器の結果を統合することにより、個々の分類器を上回る精度が得られるという考えに基づいており、別名コミッティ学習と

も呼ばれる。アンサンブルの生成に関しては、バギング (bagging)[Breiman 96] やブースティング (boosting)[Freund 96] をはじめとするさまざまな方法が提案されている。

ILP/SE では、予測精度を向上するためにアンサンブル学習を採用する。共生進化により最適仮説を探索する集合被覆アルゴリズムを  $R$  回繰り返し、得られた  $R$  個の仮説集合により未知事例を分類する。アンサンブル学習の効果を得るには、個々の分類器の動作が互いに異なることが重要であるため、多様な解が生成されるよう共生進化に関する設計を行なう。

### 3.3.2 atom

ILP/SE では、最弱仮説に含まれるリテラル、あるいはそれを一般化したリテラルを部分解とし、atom と呼ぶ。最弱仮説に含まれるリテラルの項数の最大値を  $N_t$  とするとき、atom を表す遺伝子列は 1 個の述語遺伝子と  $N_t$  個の項遺伝子からなる。述語遺伝子には  $0 \sim N_t - 1$  のリテラル ID、項遺伝子には 0 または 1 の値が割り当てられる。述語遺伝子は最弱仮説中の対応するリテラルを表しており、以降ではこれを最弱リテラルと呼ぶことにする。項遺伝子が 0 の場合、最弱リテラルの対応する項を一般化することを表し、1 の場合は最弱リテラルと同じ変数もしくは定数が入ることを表す。最弱リテラルの項数が  $N_t$  個に満たない場合、atom を表現型に変換するときに余剰分の項遺伝子は用いられない。

項遺伝子の値は、最弱仮説の対応する項のモードにより制限される。+ モードまたは - モードのときにはいずれの値も取ることができるが、# モードの場合は必ず 1 となるようにする。この制限によって、最弱リテラルより特殊なリテラルの生成が回避される。

図 3.2 のモード宣言と次の最弱仮説が与えられたときの atom の例を図 3.4 に示す。図中の“?” は最弱リテラルの当該項を一般化することを表している。

```
east(A) :- has_car(A, B), shape(B, C), load(B, C, D).
```

atom 集団の個体数  $I_a$  は  $N_t$  の  $m$  倍とする。ただし、 $m$  は任意の自然数とする。述語遺伝子が  $0 \sim N_t$  である atom をそれぞれ  $m$  個ずつ生成し、前述の条件を満たす範囲内でランダムに項遺伝子を決して、初期の atom 集団を形成する。

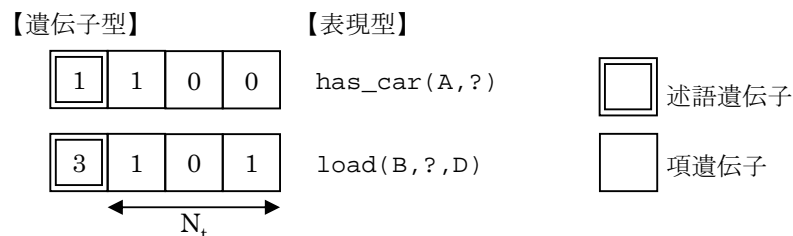


図 3.4: atom の遺伝子型と表現型



### 3.3.3 clause

ILP/SE における全体解を clause と呼ぶ。clause の遺伝子型は長さ  $N_l$  の atom へのポインタ列で表現される。 $i$  番目の要素としては、述語遺伝子の値が  $i$  の atom へのポインタ、あるいは空ポインタが入る。clause の例を図 3.5 に示す。

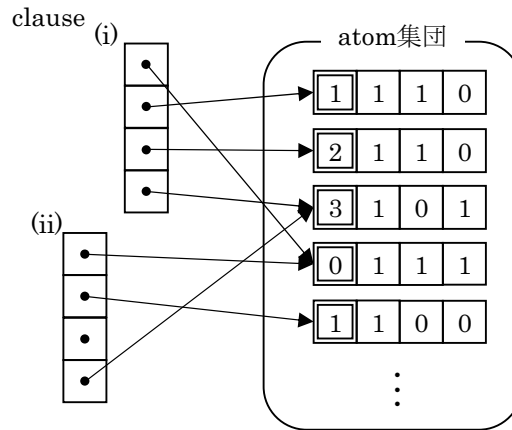


図 3.5: clause の遺伝子型

clause 集団の個体数  $I_c$  は最弱仮説のリテラル数  $N_l$  以上とし、ランダムに atom へのポインタを設定して、初期の clause 集団の個体とする。ただし、すべての clause の任意の  $i$  番目の要素において、少なくとも 1 つは空ポインタでない要素が存在するようにする。

ランダムに clause を生成する際は、次の手順に従う。

Step 1 clause のすべての遺伝子を空ポインタとする。

Step 2 atom 集団から atom を 1 つ、ランダムに選択する。

Step 3 選択した atom の述語遺伝子の値が  $i$  で、clause の  $i$  番目の遺伝子が空ポインタのとき、選択した atom へのポインタを  $i$  番目の遺伝子とする。

Step 4 Step 2~3 を  $N_l$  回繰り返す。

繰返しが終了した時点で、0 番目の遺伝子、つまりヘッドを表す遺伝子が空ポインタの場合は、最弱仮説のヘッドと同じ atom へのポインタを入れ、ヘッドのない不正な節の生成を避ける。

### 3.3.4 clause の表現型への変換

clause の遺伝子が指している atom の遺伝子を参照して、対応する節を表す表現型を生成する。このとき、次の 3 つの制約条件を課し、条件を満たさない場合はそのリテラルを含まないものとする。同時に clause の対応する遺伝子は空ポインタに書き換えられる。

条件 1 ボディの + モードの項には既出していない変数が出現しない。

条件2 ボディにのみ2回以上出現する変数の種類は  $N_{vmax}$  以下とする。

条件3 生成される節のリテラル数は  $N_{lmax}$  個以下である。

条件1は、ボディの+モードの項に関して不正な節の生成を回避するための必須条件である。条件2は評価時間の増大を防ぐために与えられた制約であり、被覆事例数を求める際のバックトラック発生を抑制する。Progol同様、生成される節のリテラル数に制限を加えているのが条件3である。clauseの表現型への変換は、リテラル数が上限値  $N_{lmax}$  に達したところで終了し、参照されなかった遺伝子は空ポインタに書き換えられる。

atomの参照は、clauseの各遺伝子に与えられた優先度の高い順に行なう。各遺伝子を下記のように2つのグループに分け、Aグループには負の乱数を、Bグループには正の乱数を優先度として与える。

Aグループ ヘッドの+モードの項ではない変数が、+モードの項となっているatomを指す遺伝子  
Bグループ Aグループに属さない遺伝子

atomの参照順序を設定することで、条件1による遺伝子の空ポインタへの書き換えや、条件2によるリテラルIDの大きいatomの非参照を抑制する。

atomの項遺伝子の値が1のときは最弱リテラルの該当する変数または定数を項の値とし、0のときは対応する項のモードによって次のように値を決める。

- +モード ... 既出変数のうち、最弱仮説で同変数だった項に現れる変数をランダムに選択。既出変数がなければ当該リテラルを削除。
- -モード ... 最弱仮説で同変数だった項に現れている変数と新しい変数から、いずれか1つをランダムに選択。

図3.5の(i)と(ii)はそれぞれ次のように変換される。(i)の遺伝子型から生成される節は5通りであるが、(ii)の遺伝子型では1通りのみとなる。

- (i) `east(A) :- has_car(A, B), shape(B, C), load(B, C, D).`  
`east(A) :- has_car(A, B), shape(B, C), load(B, E, D).`  
`east(A) :- has_car(A, B), shape(B, C).`  
`east(A) :- has_car(A, B), load(B, E, D).`  
`east(A) :- has_car(A, B).`
- (ii) `east(A).`

### 3.3.5 適応度

atomとclauseは適応度関数により評価され、集団内での優劣が判定される。clauseの評価は、表現型の節が被覆する事例数により決まる。clause  $c$  が被覆する正事例数と負事例数をそれぞれ  $p(c)$ ,

$n(c)$  と表すと,  $c$  の適応度  $cfit(c)$  は次式により求められる.

$$cfit(c) = p(c) - \alpha \cdot n(c) \quad (3.4)$$

$\alpha$  は負事例の被覆を認める度合を表すパラメータであり, 1 以上の値を指定する. 適応度の値が等しい場合は表現型の節のリテラル数が少ない clause を優位とし, 適応度とリテラル数がともに等しい場合は前世代における優位度に従って, 各 clause 間の優位関係を一意に決定する.

atom の適応度は, 当該 atom を参照している clause のうち, 最も適応度の高い clause の適応度とする. atom  $a$  を参照している clause の集合を  $C_a$  とすると,  $a$  の適応度  $afit(a)$  は次式で定義される.

$$afit(a) = \max_{c_i \in C_a} (cfit(c_i)) \quad (3.5)$$

### 3.3.6 世代交代

両集団の世代交代は HS モデルにより行なう. 突然変異は全遺伝子に対して確率  $p_m$  で発生させる. 述語遺伝子では  $0 \sim N_{l-1}$  の値をランダムに選んで置き換え, 項遺伝子では値を反転する. 交叉および突然変異によりモード宣言とタイプ情報による制約を満たさなくなった場合は, 矛盾のないように項遺伝子の値を変更する.

clause の進化では, 参照している atom の述語遺伝子が atom 集団の進化により変化している可能性があるため, 新しい atom に合わせて clause の遺伝子を修正してから交叉と突然変異を行なう. 突然変異の発生確率は atom と同様に  $p_m$  とする.

atom 集団と clause 集団の進化は, それぞれ図 3.3 の太線と普通線によるアークをたどることに相当する. 1 世代の進化で複数のアークをたどり, 空間内の仮説を評価する. 世代を重ねるごとに探索領域は評価の高い節の周辺へと絞られる.  $N_g$  世代にわたって clause 集団の最良個体に変化しない場合, その時点における最良個体の表現型を最適仮説として仮説集合  $H$  の要素とする.

訓練事例におけるノイズに対処するため, Progol が探索中の仮説の被覆事例数における負事例の割合を制限しているが, ILP/SE では被覆負事例数が上限値  $T_{neg}$  を超える仮説  $h_i$  を仮説集合  $H$  に加えないようにする.  $H$  が空集合になる可能性もあり得るが, 式 (3.4) で定義した通り, clause の適応度において被覆負事例数を考慮するため,  $H$  が空集合になる可能性は非常に低いと考えられる.

### 3.3.7 クラス判定

前節までに記した共生進化に基づく最適仮説探索の手法を用いて, 同一の訓練事例による仮説集合生成を  $R$  回繰り返す. 得られた  $R$  個の仮説集合  $H_1 \sim H_R$  により, 未知事例のクラス判定を行なう. 未知事例  $e$  を被覆する仮説が  $p(e)$  個であるとき,  $e$  のクラス  $class(e)$  は次式に従って決定される.

$$class(e) = \begin{cases} \text{positive} & \frac{R}{2} \leq p(e) \\ \text{negative} & \text{otherwise} \end{cases} \quad (3.6)$$

出力されたクラスが positive の場合，入力された事例は目標概念を満たすことになり，negative が出力された場合には，事例は目標概念を満たさないことを表す．

### 3.3.8 システム構成

ILP/SE は，最適仮説探索を行なう探索モジュールと，最弱仮説を生成し，最適仮説探索を制御する制御モジュール，未知事例を positive と negative に分類するクラス判定モジュールから構成される．ILP/SE のシステム構成を図 3.6 に示す．探索モジュールは C 言語，制御モジュールは Sicstus Prolog により実装され，C インタフェース部を介して動作する．制御モジュールで生成された最弱仮説が探索モジュールに渡され，それに基づいて clause 集団と atom 集団が形成されて，進化が進んでいく．clause を評価するときには，各 clause の表現型が被覆する事例数を制御モジュールに問い合わせる．clause 集団が収束したときの最良個体の表現型を，最適仮説  $h_i$  として制御モジュールに返す．すべての訓練事例が最適仮説  $h_1 \sim h_n$  により被覆されるまで最弱仮説生成と最適仮説探索を行ない，仮説集合  $H_j = \{h_1, h_2, \dots, h_n\}$  を出力する．以上の動作を  $R$  回繰り返して得られた  $R$  個の仮説集合  $H_1 \sim H_R$  により，クラス判定モジュールが未知事例のクラスを決定する．

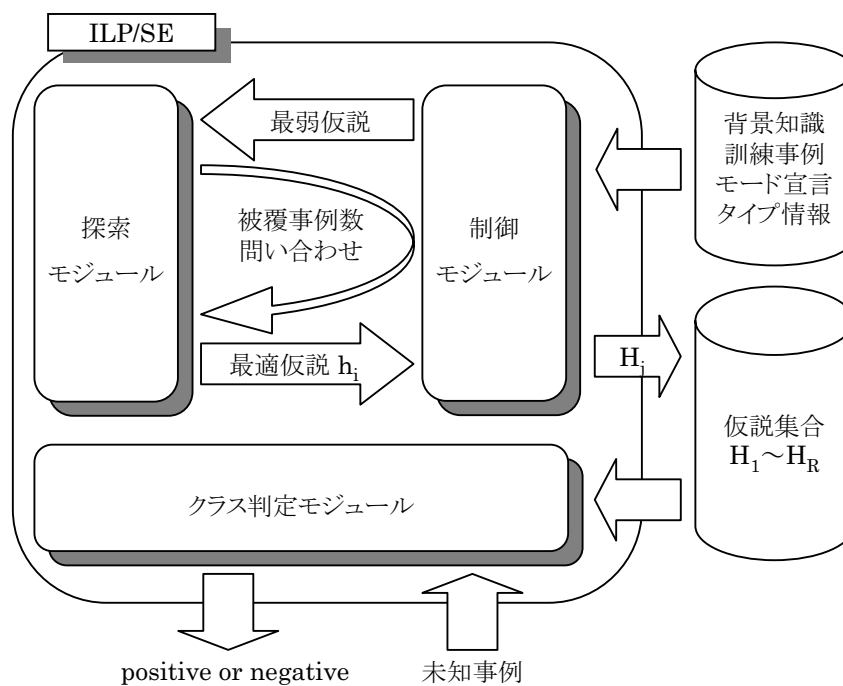


図 3.6: システム構成

## 3.4 評価実験

共生進化に基づく最適仮説探索の有効性を確かめるため，ILP/SE，および 2 つの比較対照システムにおいて評価実験を行なった．以下では，実験方法と結果を記すとともに，考察を述べる．

### 3.4.1 実験方法

ILP/SE と比較対照するための2つのシステムを以下に示す。ILP/E との比較により2集団並行進化の有効性を確認し、既存システムとの差異を示すために Aleph+Bagging との比較も行なう。

ILP/E 1 集団 GA に基づいて最適仮説を探索するシステム。clause を個体とする集団のみを保有する。clause の遺伝子型は、ILP/SE における atom を  $N_i$  個連結した遺伝子列により表現する。このとき、atom の述語遺伝子は0または1の値を取り、0のときには最弱リテラルあるいは最弱リテラルを一般化したリテラルが節に含まれないことを表し、1のときには含まれることを表す。すなわち、述語遺伝子の値が0であることは、ILP/SE における clause の遺伝子の値が空ポインタであることに相当する。最適仮説探索を1集団 GA で行なう以外は ILP/SE と同様の処理を行なう。

Aleph+Bagging 帰納論理プログラミングのプロトタイプシステム Aleph(A Learning Engine for Proposing Hypotheses)[Ale] とバギングを組み合わせたシステム。 $m$  個の事例から  $m$  回の復元無作為抽出により取り出した事例を訓練事例とし、Aleph により仮説集合を生成する。1組の事例に対して仮説集合生成を20回繰り返し、多数決によりテスト事例のクラス判定を行なう。

各システムにおいて、帰納論理プログラミングのベンチマークデータとして最も頻繁に用いられている Mutagenesis[Debnath 91] を分類対象事例とした。Mutagenesis はニトロ芳香族化合物の突然変異性に関する化合物データであるが、230 個のデータのうち重回帰分析式で判別可能な188 個を使用した。125 個が突然変異について高活性、63 個が低活性の化合物であり、高活性化合物の性質に関するルール導出が課題である。背景知識としては、表 3.2 に示す B1 ~ B4 の4種類が与えられている。

表 3.2: Mutagenesis の背景知識

背景知識	内容	節の数
B1	atm, bond, 部分電荷の値 (=)	10137
B2	B1 + 部分電荷の値の範囲 ( $\leq, \geq$ )	10139
B3	B2 + lumo, logp	10512
B4	B3 + 構造に関する定義	11948

実験は、188 個の事例のうち、10 分の9 を訓練事例、10 分の1 をテスト事例とする 10-fold クロスバリデーションにて行ない、各事例による試行を10回繰り返したときの平均値により評価する。仮説集合生成の繰り返し回数  $R$  は20回とし、各回において得られた仮説集合により訓練事例とテスト事例を分類したときの正解率、および複数回の試行で得られた仮説集合により多数決でテスト事例を分類したときの正解率を調査する。以下では、1つの仮説集合により分類する処理を“単独分類”、複数の仮説集合により多数決で分類する処理を“多数決分類”と呼ぶ。

ILP/SE と ILP/E で設定したパラメータを表 3.3 に示す． $p_m$  は [Moriarty 96] に従って設定し， $N_{lmax}$  と  $T_{neg}$  は [Srinivasan 95] と同様に設定した． $\alpha$  と  $N_g$  は全データを訓練事例とテスト事例の両方として用いる予備実験の結果から決定した．予備実験で  $\alpha$  を 2~8 で変化させたとき， $\alpha=5$  で平均正解率が最大となった．また， $N_g=100$  として同一の個体が継続して最良個体となる世代数を調べた結果， $N_g \neq 100$  である世代数の平均は 2.28 世代で，98%以上が 10 世代未満であった．

表 3.3: ILP/SE の評価実験に用いたパラメータ

パラメータ	値
clause の適応度の係数 $\alpha$	5
突然変異確率 $p_m$	0.01
atom 集団の個体数 $I_a$	$N_l \times 3$
clause 集団の個体数 $I_c$	$\max(N_l, 30)$
収束条件 $N_g$	10
リテラル数の上限値 $N_{lmax}$	4
変数の種類数の上限値 $N_{vmax}$	2
被覆負事例数の上限値 $T_{neg}$	5

### 3.4.2 生成された仮説集合の評価

各システムの訓練事例とテスト事例における単独分類の結果を示す．すなわち，ILP/SE と ILP/E における 1 回の仮説集合生成，および Aleph の性能評価である．

訓練事例における単独分類の平均正解率を表 3.4 に示す．括弧内の数字は標準偏差を表す．ILP/SE と ILP/E の結果を比較すると，いずれの背景知識においても ILP/SE で高い正解率が得られており，共生進化により適応度の高い解が探索できることがわかる．また，背景知識によって多少の差はあるものの，ILP/SE では Aleph+Bagging と同程度の正解率が得られており，既存システムに劣らない探索能力を備えているといえる．

表 3.4: 訓練事例における平均正解率

背景知識	ILP/SE	ILP/E	Aleph+Bagging
B1	0.82 (0.01)	0.77 (0.01)	0.84 (0.01)
B2	0.87 (0.01)	0.84 (0.01)	0.86 (0.01)
B3	0.88 (0.01)	0.84 (0.01)	0.88 (0.01)
B4	0.90 (0.01)	0.88 (0.01)	0.88 (0.01)

(括弧内は標準偏差)

テスト事例における単独分類の平均正解率を表 3.5 に示す．訓練事例の場合と比較して正解率が低だけでなく，試行ごとのばらつきも大きくなっているが，システム間の優劣は訓練事例の正解率と同じ傾向であるといえる．

表 3.5: 未知事例単独分類における平均正解率

背景知識	ILP/SE	ILP/E	Aleph+Bagging
B1	0.72 (0.02)	0.69 (0.03)	0.74 (0.02)
B2	0.79 (0.02)	0.77 (0.02)	0.78 (0.02)
B3	0.80 (0.02)	0.78 (0.02)	0.81 (0.02)
B4	0.84 (0.02)	0.82 (0.02)	0.81 (0.03)

(括弧内は標準偏差)

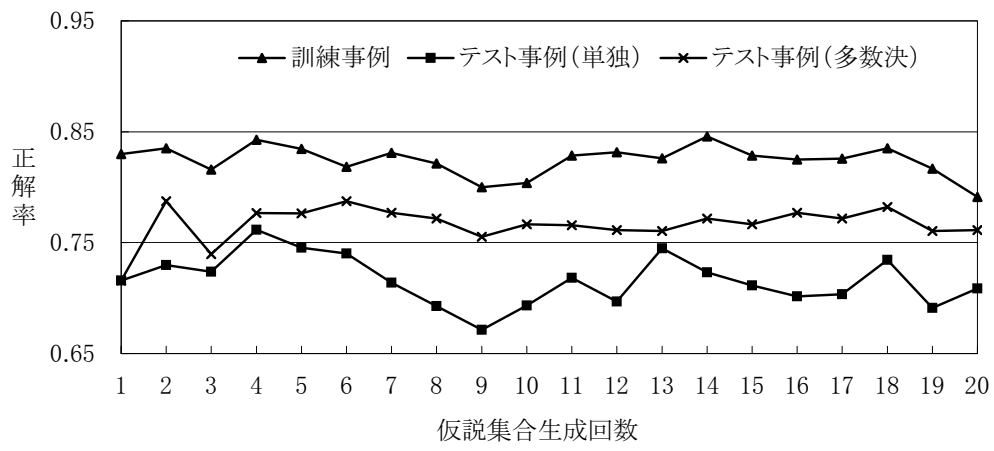
### 3.4.3 アンサンブル学習の効果

各システムにおけるアンサンブル学習の効果を示す。20回の仮説集合生成における正解率の推移を図3.7～図3.10に示す。各折れ線は、 $r$ 回目の仮説集合生成で得られた仮説集合  $H_r$  により単独分類を行なったときの訓練事例における正解率とテスト事例における正解率、および  $r$ 回の仮説集合生成で得られた  $r$ 個の仮説集合  $H_1 \sim H_r$  を用いて多数決分類を行なったときの正解率の推移を表している。

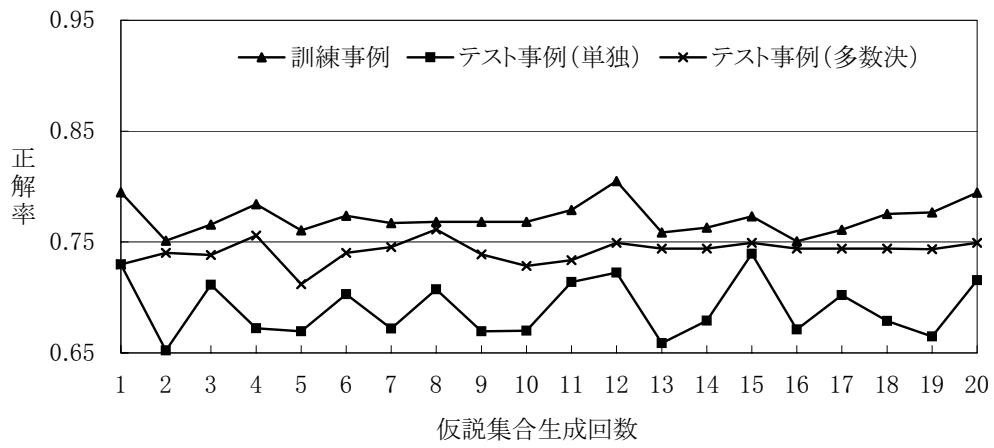
テスト事例における単独分類と多数決分類の2つの折れ線を比較すると、いずれのシステムにおいても、多数決分類では単独分類よりも安定して高い正解率が得られていることがわかる。Aleph+Baggingにおいては2つの折れ線に明確な優劣は見られないが、ILP/SEとILP/Eでは多数決分類が単独分類を明らかに上回っており、B2～B4では訓練事例の場合に近い正解率が得られている。したがって、多様な解を生成するように設計した遺伝的アルゴリズムに基づくILP/SEとILP/Eにおいて、多数決を行なうアンサンブル学習は効果的であるといえる。

図3.7(b)の2回目、図3.8(a)の2回目、図3.9(a)の2回目など、前回よりも単独分類の正解率が低い仮説集合が得られているにもかかわらず、多数決分類の正解率が向上している箇所が多く見られる。単独分類の平均正解率が大きく変化しても、多数決を行なうことでその変化による影響は抑えられ、正解率の極端な低下が回避されるものと考えられる。

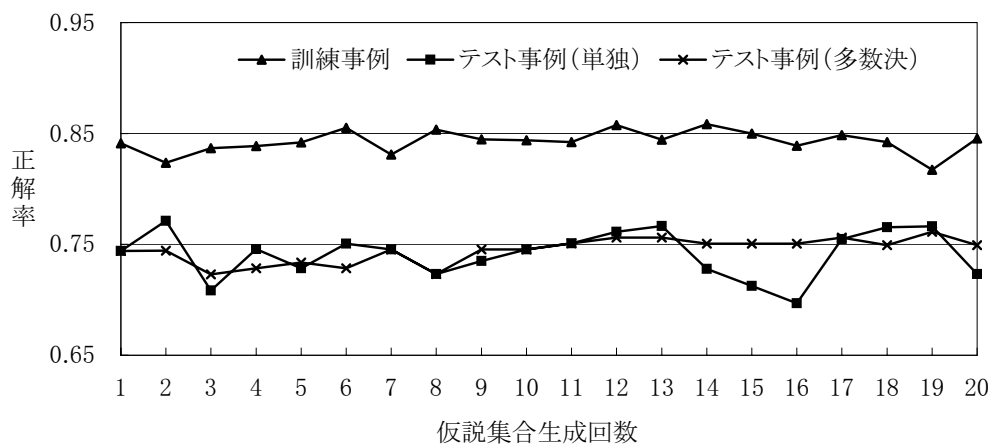
各背景知識での正解率がほぼ安定している  $R = 14 \sim 20$  の平均正解率と Progol の平均正解率を表3.6に示す。ILP/SEでは、ILP/EとAleph+Baggingよりも安定して高い正解率が得られている。また、いずれの背景知識を用いた場合にも、ILP/SEではProgolを越える正解率が得られており、特にB2、B3における正解率はProgolを大きく上回った。この結果、Mutagenesisにおける未知事例分類の予測精度に関して、ILP/SEはいずれのシステムよりも優位であることが示された。ただし、本実験では変数の種類数を2に制限しているため、深い変数連鎖を必要とする問題での性能については言及できない。



(a) ILP/SE



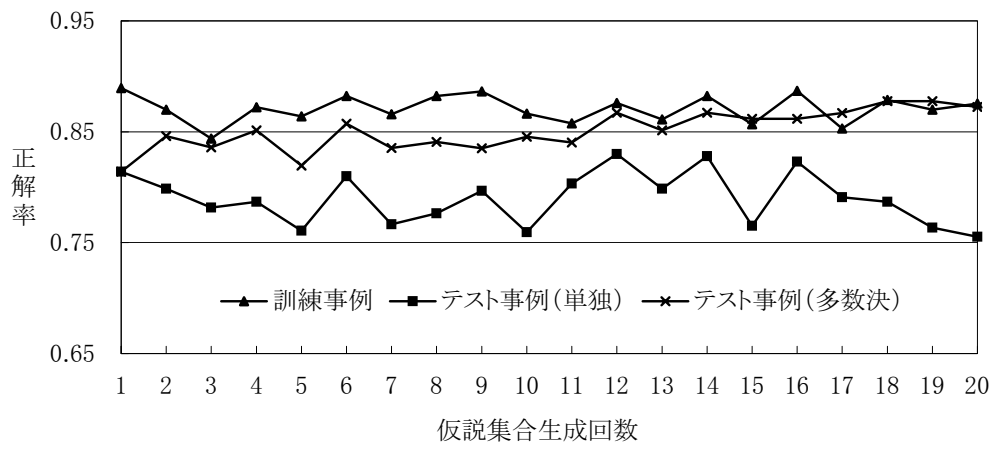
(b) ILP/E



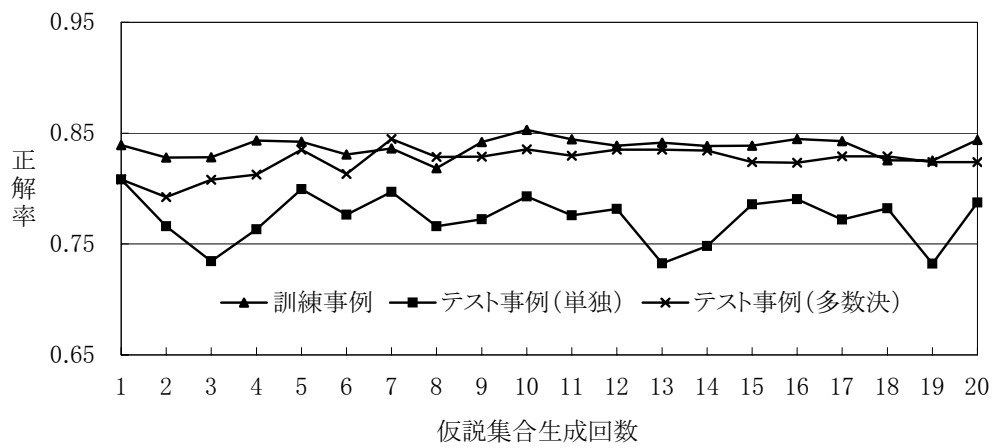
(c) Aleph+Bagging

図 3.7: 正解率の推移 (背景知識 B1)

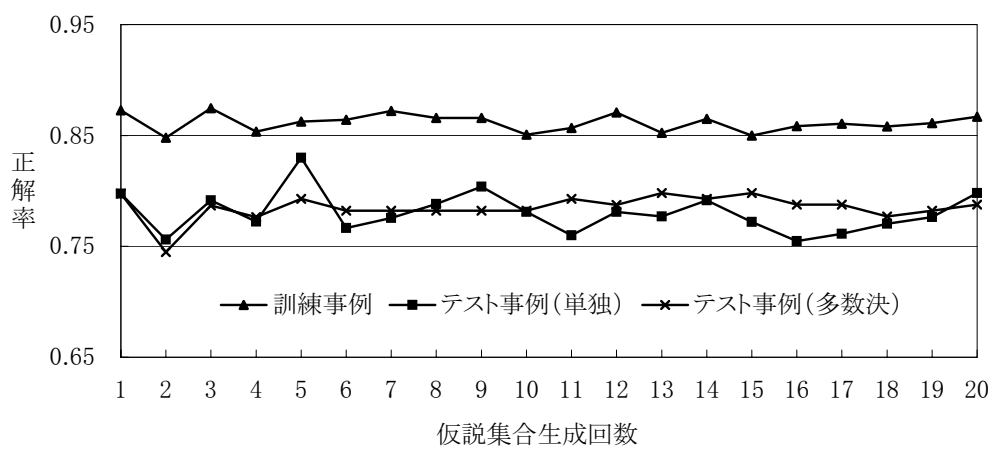




(a) ILP/SE

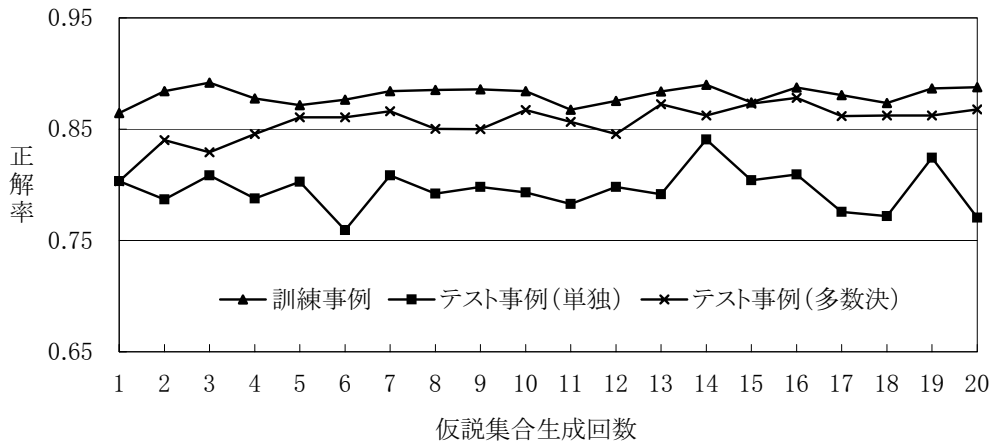


(b) ILP/E

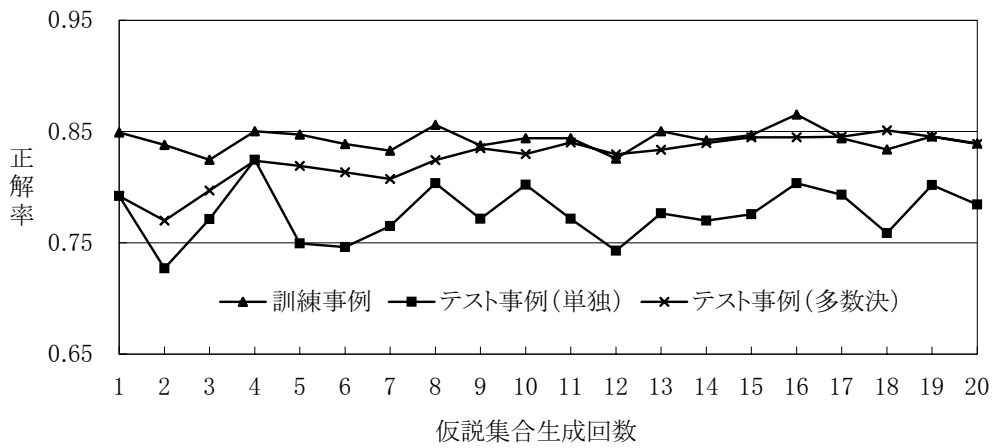


(c) Aleph+Bagging

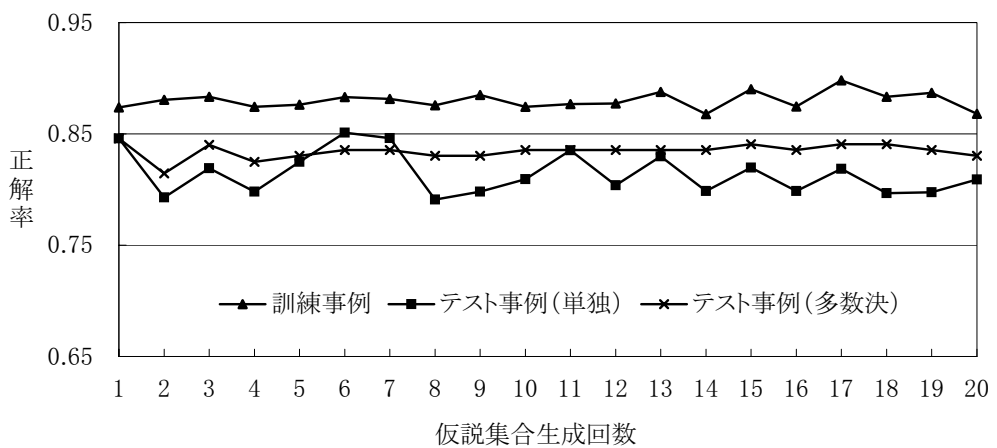
図 3.8: 正解率の推移 (背景知識 B2)



(a) ILP/SE

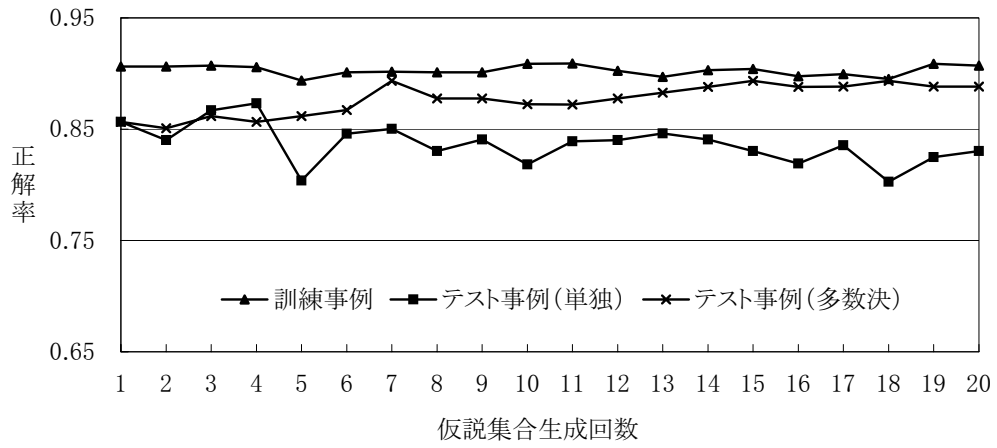


(b) ILP/E

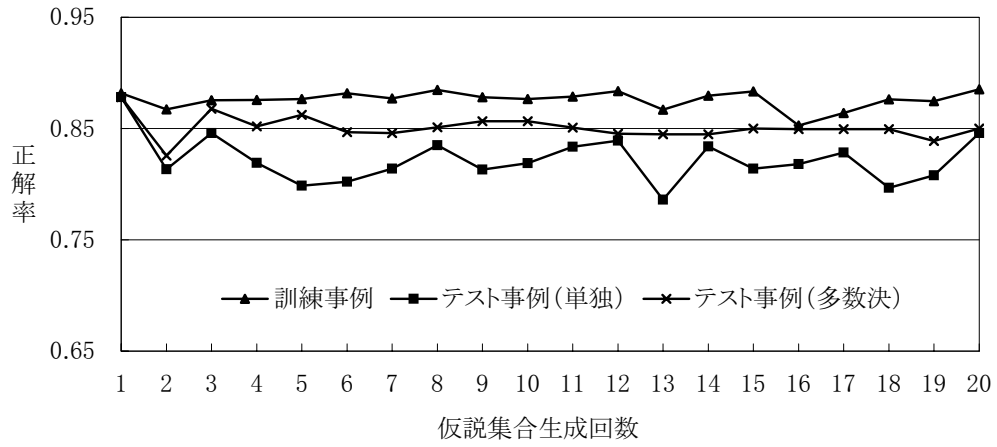


(c) Aleph+Bagging

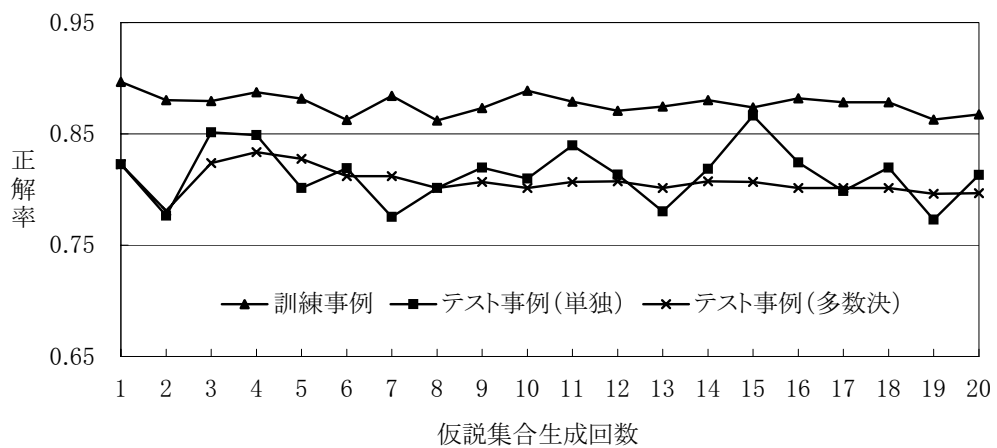
図 3.9: 正解率の推移 (背景知識 B3)



(a) ILP/SE



(b) ILP/E



(c) Aleph+Bagging

図 3.10: 正解率の推移 (背景知識 B4)

表 3.6: 未知事例多数決分類における平均正解率

背景知識	ILP/SE	ILP/E	Aleph+Bagging	Progol
B1	0.77 (0.07)	0.75 (0.07)	0.75 (0.14)	0.76 (0.03)
B2	0.87 (0.06)	0.83 (0.07)	0.79 (0.11)	0.81 (0.03)
B3	0.87 (0.06)	0.84 (0.07)	0.84 (0.10)	0.83 (0.03)
B4	0.89 (0.04)	0.85 (0.07)	0.80 (0.18)	0.88 (0.02)

(括弧内は標準偏差)

### 3.5 比較研究

概念学習に遺伝的アルゴリズムを適用するいくつかの手法がこれまでに提案されてきた。概念記述が一階述語論理による場合は、学習すべきルールがどのようなリテラルを含み、どの引数同士が等値関係にあるかなど、ルールの形式を既定するパターンが未知であるため、個体の表現方法に工夫を要する。REGAL[Giordana 96]では、ユーザがパターン (template と呼ぶ) を指定する。template はリテラルの連言であり、各リテラルには変数の取り得るすべての値を要素とする集合が引数に加えられている。引数間の等値関係は template により固定されており、各変数が取るべき値のみを GA により学習する。

REGAL の後継である G-NET[Anglano 98]では、G-layer と Supervisor という2つのレベルで共進化を行なう。G-layer では概念の部分定義 (1つの Horn 節) を GA により学習し、Supervisor では G-layer で生成された Horn 節の選言である概念を評価して、その結果を G-layer の学習にフィードバックする。さまざまな問題において高正解率が報告されているが、Mutagenesis では B3 の背景知識で 91% の正解率が得られている。

G-NET と ILP/SE はともに共同共進化を採用しているが、各個体の設計方法や進化の流れは異なっている。G-NET では template をユーザが指定しなければならず、引数の等値関係は既定のものとしてされているのに対し、ILP/SE では逆伴意法により導出された最弱仮説をパターンとし、引数の等値関係を含むルールを GA により学習できる点で優れているといえる。

REGAL と G-NET がユーザ指定のパターンに基づくのに対して、SIAO1[Augier 95] は seed として選んだ1つの正事例を満たすルールを一般化することで、目標概念を学習する。遺伝子操作オペレータの適用により、述語の一般化や変数間の等値関係をも学習できる。Tamaddoni-Nezhad らは、Progol の逆伴意法と GA による最適仮説探索とを統合するためのフレームワークを提案している [Tamaddoni-Nezhad 01]。節をバイナリ行列で表現し、タスク依存の遺伝子操作オペレータを定義することで、効率的な探索が可能となっている。ここで  $N$  個の変数間等値関係は  $N \times N$  の対称行列の上三角部分で表現される。Braud らのシステムでは、各変数の等値関係と数値制約を等値クラスとしてコード化し、集合操作に基づく遺伝子操作オペレータにより学習を行なう [Braud 01]。 $N$  個の変数間等値関係は長さ  $N$  の個体で表される。上記3つのシステムでは、簡易な問題による予備実験では良い結果が得られているが、実用データへの適用には至っていない。

複数仮説の多数決によるクラス判定法は、反復学習により得られた複数の分類器を用いて未知事例进行分类するという点でアンサンブル学習と類似している。一般に、信頼度評価によるブースティング [Quinlan 96] では、学習結果が巨大なルール集合となり、理解が困難という欠点がある。C<sup>2</sup>RIB [Hoche 01] ではルール集合に制限を加え、学習結果の理解性の増大と学習時間の短縮を実現しており、Mutagenesis の B4 の背景知識で 88% の正解率が得られている。C<sup>2</sup>RIB では、すべての訓練事例がいずれかの節によって正しく分類されるように訓練事例の重みを変化させながら節を 1 つずつ学習し、得られた節の信頼度による重みつき投票で未知事例进行分类する。これに対し ILP/SE では、学習時に訓練事例の設定や分類の際の重みが不要であり、同一の訓練事例に関する学習を反復し、単純な多数決を行なうだけで、89% の正解率が得られている。また、確率的サンプリングの手法を用いたシステム STILL [Sebag 00] も実現されており、Mutagenesis の B3 の背景知識で ILP/SE と同じ 87% の正解率が得られている。

### 3.6 まとめ

本章では、未知事例における予測精度の向上を目的として、共生進化に基づく帰納論理プログラミングの未知事例分類法を提案した。精密化演算子の 2 つの機能を共生進化の 2 つの集団の進化により実現し、最適仮説を探索した。また、従来の探索では得られない解の多様性を未知事例への適合可能性と捉え、複数回の仮説集合生成で得られた複数の仮説集合により未知事例を識別することで、予測精度の向上を目指した。提案手法を帰納論理プログラミングシステム ILP/SE として実現し、Mutagenesis のデータを用いて実験を行なった結果、Progol を上回る正解率が得られた。この結果は、帰納論理プログラミングの学習方法として共生進化が有効であると同時に、多数決によるクラス判定が未知事例の分類に有効であることを示唆するものと考えられる。

共生進化に基づく帰納論理プログラミングの未知事例分類法は、A\*-like アルゴリズム等の逐次探索を行なう場合と比較して、データ量の増加に伴なう探索負荷の増加量が少ないため、大量データを対象とするデータマイニングへの効果が期待される。

以上より、2.4.2 節で示したカテゴリ P2-W2 の問題に対しても、共生進化の適用可能性が示されたと考えられる。

## 第4章 簡素な決定木の生成

決定木では事例を正しく分類することが望まれる。同時に、記憶容量や分類処理の速度、分類規則の解釈の容易さの点から、木の簡素さも重要な評価指標となる。未知事例を正確に分類でき、かつできるだけ簡素な決定木の生成を目的として、共生進化に基づく新しい決定木生成法を提案する。提案手法の有効性を示すための評価実験を行ない、共生進化の決定木生成への適用可能性について論じる。

### 4.1 決定木

決定木は分類規則の木構造による表現技法である。以下では、決定木により分類対象となる事例と決定木の構造を説明し、生成する際に満たされるべき性質について述べる。

#### 4.1.1 分類対象となる事例

決定木の分類対象となる事例は、複数の属性とそれらに相当する属性値、およびその事例が属するクラスからなる。決定木は属するクラスが既知の訓練事例を用いて生成され、未知事例のクラスは得られた決定木を走査することによって判定される。降水の種類を判定するための事例を表 4.1 に示す。この例では、「状態」「直径」「固い」が属性、「氷粒」「4mm」「true」などが各属性に対する属性値、「氷霰」「雪霰」「雹」などが各事例の属するクラスである。分類規則は、属性、属性値、およびクラスを用いた条件文の形で表される。条件文の前件は1つの属性に関する分類条件の連言であり、後件は属するクラスとなる。例えば、「状態が氷粒であり、かつ直径が5mm未満であれば雹である」という分類規則は、以下の式で表される。

$$(\text{状態} = \text{氷粒}) \wedge (\text{直径} \geq 5\text{mm}) \rightarrow (\text{雹}) \quad (4.1)$$

属性は以下の3種類に大別される。《 》内に属性値の例を示す。

バイナリ属性：ある命題の真偽を示す。《「true」「false」》

カテゴリカル属性：カテゴリを表す値が属性値として与えられる。《「氷粒」「氷結晶」「水滴」》

数値属性：ある閾値により定められる数値範囲が属性値となる。《「5mm未満」「5mm以上」》

バイナリ属性による分類条件は、「固い場合」「固くない場合」のように、属性の真偽により表現すればよい。また、カテゴリカル属性による分類条件には、「状態が氷粒の場合」「状態が水滴の場合」

表 4.1: 降水の種類を表す事例

	属性			クラス
	状態	直径	固い	
訓練事例	氷粒	4mm	true	氷霰
	氷粒	3mm	false	雪霰
	氷粒	7mm	true	雹
	氷結晶	2mm	false	雪
	水滴	1mm	false	雨
未知事例	氷粒	8mm	true	?
	水滴	3mm	false	?

などのように、訓練事例に現れる属性値のうちいずれかを用いて表現すればよい。しかし、数値属性の分類条件では、隣り合う2つのアークによる分岐で有効となる閾値を選定し、各アークのラベルを決定しなければならない。すなわち、分類条件に含まれる属性値自体をも学習する必要があるため、バイナリ属性やカテゴリカル属性に比べて数値属性を扱う決定木は学習が難しい。

本研究では、決定木で分類する事例の属性はすべて数値属性として扱い、決定木生成の際には分類条件の閾値をも同時に学習する。事例中のバイナリ属性では、true を 1, false を 0 という数値と見なし、カテゴリカル属性では各属性値に 1 から順に整数の番号を割り当てることで、すべての属性を同様に扱う。low, middle, high など順序付けが可能なカテゴリカル属性の場合は、番号の大小と属性値の順序が一致するようにする。

#### 4.1.2 決定木の構造

決定木の非終端ノードには属性、アークのラベルには属性値に関する条件、終端ノードには分類結果となるクラスが割り当てられる。1つの非終端ノードとその下に結合しているアークの組によって、「状態が氷粒の場合」というような1つの分類条件を表現する。事例の各属性値に従って根ノードから終端ノードまで決定木をたどることで、その事例の属するクラスが判定される。根ノードから終端ノードまでの経路が1つの分類規則を表し、非終端ノードとアークによる分類条件の連言が前件、終端ノードが後件となる。

降水の分類規則を表した決定木の例を図 4.1 に示す。「状態」「直径」「雹」の順でノードをたどる経路が式 (4.1) の分類規則を表している。また、この決定木に従うと、表 4.1 の未知事例のクラスはそれぞれ雹、雨と決定される。

#### 4.1.3 決定木の生成

決定木を生成する際には、訓練事例をできるだけ正確に分類できるように、木の形状および各ノードとアークラベルへの値の割当てを決定する。しかし、得られる決定木では、訓練事例だけでなく、決定木生成には用いられなかった未知事例をも正しく分類できることが望ましい。

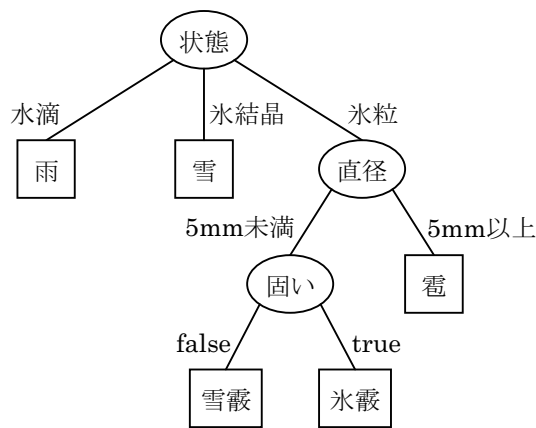


図 4.1: 降水の分類規則を表す決定木

これまでに決定木生成に関するさまざまなアルゴリズムが提案されており、代表的なシステムとしては CART[Breiman 84], ID3[Quinlan 86], C4.5[Quinlan 93] などが挙げられる。これらのシステムでは、分類誤り率による枝刈りやブースティングアルゴリズム等の利用により、未知事例分類において高い正解率を得ている。しかし、正解率の高い決定木は複雑でノード数が多いという傾向が一般に知られている。

例えば、表 4.2 に示すようなクラス間の偏りが大きい事例进行分类する際、図 4.2(a) のような終端ノードのみからなる決定木で正解率 82.5% が得られるが、他にも図 4.2(b) ~ (d) のような決定木も候補として考えられ、正解率 100% の決定木が最もノード数が多くなっている。

表 4.2: クラス間の偏りの大きい事例

属性			クラス
A	B	C	
0	0	0	$\alpha$
0	0	1	$\alpha$
0	1	0	$\alpha$
0	1	1	$\alpha$
1	0	0	$\alpha$
1	0	1	$\alpha$
1	1	0	$\alpha$
1	1	1	$\beta$

複雑な決定木は、記憶に必要な容量が大きく、分類処理速度が遅いことに加え、分類規則の解釈が困難であるという短所を持つ。特に規則の解釈に関しては、人間が分類規則を扱うにあたって重要視すべき点であり、木の簡素さは決定木の評価において重要な指標といえる。高正解率を保持しつつ木を簡素にすることは、決定木生成での一目標となっている。



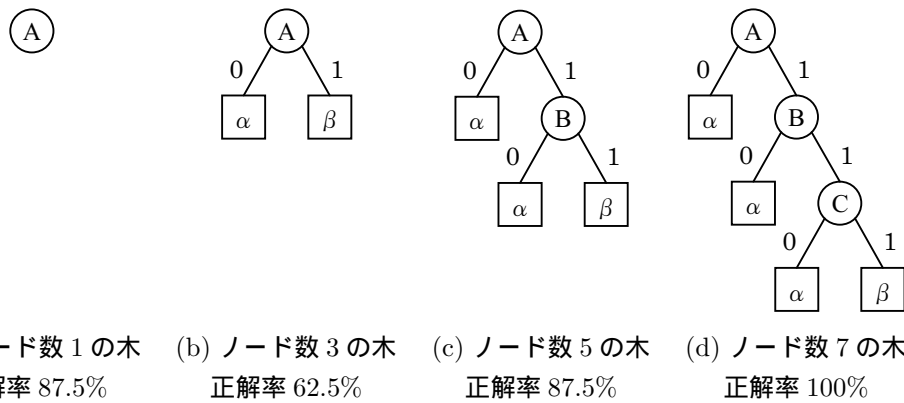


図 4.2: 表 4.2 の事例を分類する決定木

## 4.2 共生進化に基づく決定木生成

決定木は高さが 1 である部分木をいくつか組み合わせたものといえる。解に含まれるべき部分木、および部分木の組合せを同時に学習することで、有用な決定木が生成される。これは共生進化の適用可能な問題の特徴と一致しており、2.5.5 節で述べた効果を勘案すると、共生進化が正解率の高い簡素な決定木の生成に効果を発揮するものと考えられる。以下、共生進化に基づく決定木の生成手法に関しての検討内容を述べる。

### 4.2.1 共生進化の適用

決定木における高さ 1 の部分木は、1 つの属性に関する分岐を表す。例えば、図 4.3 に示すような高さ 1 の部分木によって、「直径が 5mm 未満の事例と 5mm 以上の事例を分ける」という分岐が表現される。図 4.3 の三角で示されたノードには、他の部分木、あるいは分類クラスの割り当てられた終端ノードが接続される。

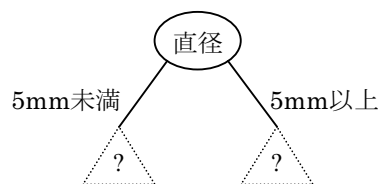


図 4.3: 1 つの属性に関する分岐を表す部分木

決定木を生成するにあたって学習対象となるのは以下の 5 項目である。

- 項目 1 非終端ノードに割り当てる属性
- 項目 2 非終端ノードが持つ子ノードの数
- 項目 3 非終端ノードが持つ各子ノードの種類

項目4 非終端ノードから各子ノードへ分岐するための閾値となる属性値

項目5 上記4つが定まった非終端ノードの上下の結合関係

項目1~4は高さ1の部分木の構成に関する項目であり、項目5は高さ1の部分木の組み合わせ方に関する項目である。目的に適合した決定木を生成するためには、相互関係の強い各項目を同時に考慮する必要がある。すなわち、決定木生成は求めるべき解の部分、およびそれらの組合せをバランスよく行なうべき問題であるといえ、これは2.5.5節で述べた共生進化の適用可能な問題の特徴に合致する。したがって、共生進化における2集団でそれぞれ高さ1の部分木とそれらの組合せを学習することによって、効果的な決定木生成が実現されると考えられる。

#### 4.2.2 解の構造に関する問題点

決定木生成に共生進化を適用するにあたって問題となるのが、決定木の構造に関する点である。Moriartyらが共生進化を適用したニューラルネットワークでは、入力層からの情報が隠れ層にある複数のニューロンを介して出力層に伝達される。隠れ層の各ニューロンの動作は互いに独立であるため、隠れ層を構成するニューロンが決まるとネットワーク全体の動作は一意に定まる。Hierarchical SANEでは全体解を隠れ層のニューロンの組合せで表現しているが、その組み合わせ方には制約がないため、遺伝操作により構造上不備のあるネットワークが生成されることはない。

一方、決定木は複数の部分木を組み合わせたものであるが、決定木に含まれる部分木が指定されても、決定木の構造は一意に定まらない。各部分木の結合関係により決定木の構造は変化する。部分木の数に過不足が生じたり、冗長で無意味な決定木が生成される場合もある。したがって、決定木生成に共生進化を適用する際には、部分解の単なる組合せで表現された全体解を一般的な遺伝操作で進化させることは望ましくない。部分解の結合関係を保持した形で全体解を表現し、解候補として適切な構造を持った決定木を保持することが必要となる。

#### 4.2.3 遺伝的プログラミング

グラフ構造や木構造などの構造的表現を扱うために拡張された遺伝的アルゴリズムが遺伝的プログラミング (Genetic Programming; GP) である [Koza 92]。遺伝的プログラミングの基本的概念や処理の流れは遺伝的アルゴリズムと同様であるが、解候補を表す個体が構造を持っている点が最大の特徴といえる。

遺伝的プログラミングでは、木構造の個体を遺伝操作で進化させるため、図2.1に示した各遺伝オペレータの代わりに、図4.4のような交叉、逆位、突然変異の遺伝オペレータを適用する。交叉では2つの親個体の部分木を交換し、逆位では親個体中の2つの部分木を入れ換える。突然変異では木のノードをランダムに変更する。

解を固定長の遺伝子列で表現する一般的な遺伝的アルゴリズムでは、2つの同じ親個体から交叉により生成される子個体は親個体とまったく同じ個体となる。しかし、遺伝的プログラミングでは、交

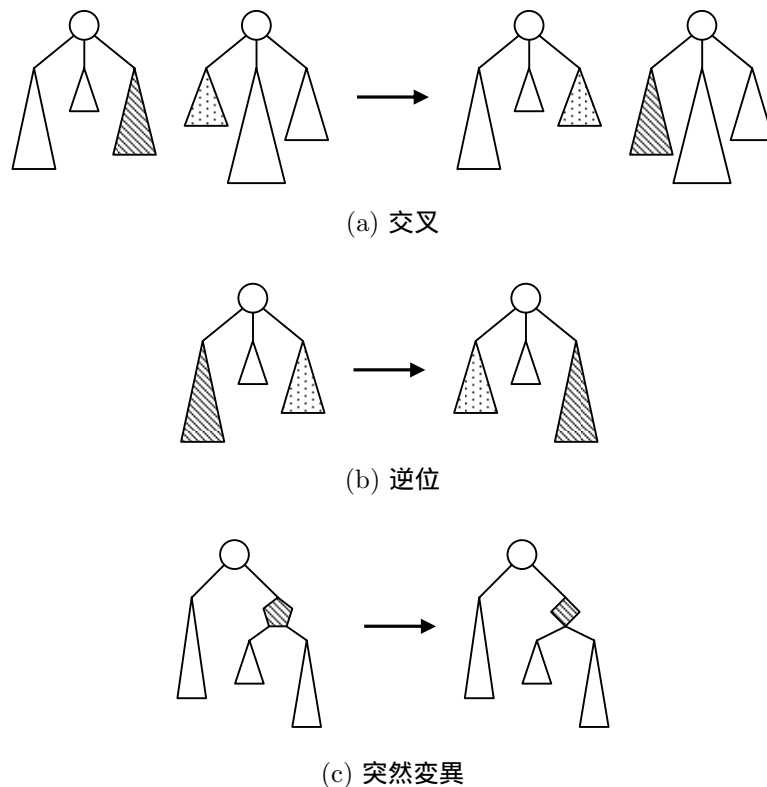


図 4.4: 遺伝的プログラミングにおける遺伝オペレータ

又における遺伝子列の切断箇所を2つの親個体で異なる位置に指定できるため、生成される子個体が親個体と同じ個体になる確率は低い。このため、遺伝的プログラミングでは集団の多様性が比較的維持されるといえる。

一方で、学習過程で木のノード数が急速に増大するブloat現象 (bloat phenomenon) が発生しやすいという問題もある [Angeline 98]。ブloat現象の主な原因はイントロンの成長にある。生物学におけるイントロンとは、DNA 中で遺伝情報を持たない部分のことであり、遺伝子が働くときには切り捨てられる。これを受けて、遺伝的プログラミングの解において動作に無関係な部分をイントロンと見なし、実行されても正解率に関与しない冗長な部分を意味論的イントロン (semantic intron)、実行されない無意味な部分を構文的イントロン (syntactic intron) と呼んでいる。イントロンは、破壊的な交叉を回避したり、新規個体生成のきっかけを作るなど、最適解探索に有益な場合もある。しかし、実行時間、解の複雑さ、過学習の点を踏まえると、木の複雑化は可能な限り回避すべきであり、ブloat現象の主要因であるイントロンの生成を抑制する必要がある。このため、生成される木の高さやノード数に制限を加える、適応度に木のサイズを反映する、木の成長を抑えるように遺伝オペレータを適用するなど、ブloat現象を回避するための様々な手法が提案されている。

適応度の工夫により木の複雑化を抑える手法として、MDL (Minimum Description Length) 基準を用いた適応度に基づく手法が提案されている [Iba 94]。木の記述長と誤差の記述長の和で表される

MDL 値は、以下の式により算出される [Quinlan 89, Iba 94] .

$$MDL = TreeCodingLength + ExceptionCodingLength \quad (4.2)$$

$$TreeCodingLength = (n_f + n_t) + n_t \log_2 T_s + n_f \log_2 F_s \quad (4.3)$$

$$ExceptionCodingLength = \sum_{x \in Terminals} L(n_x, w_x, n_x) \quad (4.4)$$

$$L(n, k, b) = \log_2(b+1) + \log_2\left(\binom{n}{k}\right) \quad (4.5)$$

ここで、 $n_f$  は非終端ノードの数、 $n_t$  は終端ノードの数、 $T_s$  はクラス数、 $F_s$  は属性数を表す。また、式 (4.4) はすべての終端ノードに関する  $L(n_x, w_x, n_x)$  の和を求めるものであり、 $n_x$  はノード  $x$  到達した事例数、 $w_x$  はノード  $x$  到達した  $n_x$  個の事例のうち誤分類した事例の数を表す。式 (4.5) で定義される  $L(n, k, b)$  は  $n$  ビットの文字列を伝達するために必要なコストである。

[Iba 94] では、以下に示す手順によって各個体の適応度を算出し、6 マルチプレクサ問題に対する決定木生成などでの効果が確認されていることから、MDL 基準は簡素な決定木の生成に有効な評価指標であるといえる。

- Step 1 第  $t$  世代の個体を生成し、各個体の MDL の値を求める。
- Step 2 すべての個体の MDL 値のうち、最も大きい値を  $Worst(t)$  とする。
- Step 3  $Worst(t), Worst(t-1), \dots, Worst(t - W_{size})$  のうち、最も大きい値を  $W_{mdl}$  とする。
- Step 4 各個体の適応度を  $W_{mdl} - mdl_t(i)$  とする。

#### 4.2.4 決定木生成システム SESAT

正解率の高い簡素な決定木を共生進化に基づいて生成するシステム SESAT(Symbiotic Evolution for Simple and Accurate Trees) を構築する。SESAT では、高さ 1 の部分木を部分解、部分木の組合せで表現される決定木を全体解として、両者をそれぞれ個体とする 2 集団を並行進化させる。4.2.2 節での問題点を解決するために、全体解は単なる部分解の組合せではなく、部分解を参照して作られる木構造とし、遺伝的プログラミングの遺伝オペレータにより進化させる。

SESAT の設計とアルゴリズムの詳細については次節にて説明する。

### 4.3 SESAT の設計とアルゴリズム

決定木生成システム SESAT における決定木生成方法の詳細について説明する。4.2.1 節に挙げた学習対象項目のうち、項目 3 と項目 5 に関しては、遺伝子表現や集団の構成によって部分解で学習する範囲が変動する。この影響について検討した結果を付録 A に記す。以下に記す SESAT のアルゴリズムは、この検討結果をもとに決定されたものである。

### 4.3.1 決定木の制約

決定木では、隣り合う同一の部分木が意味論的イントロン、どの事例も到達しない部分木が構文的イントロンにあたる。ここで、隣り合う部分木とは、共に同一のノードを親ノードとし、かつ隣り合っているノードを根ノードとする部分木を指す。木の形も割り当てられている値も等しい部分木が2つ並んでおり、両者の根ノードが同じノードを親ノードとしている場合、この2つの部分木は隣り合う同一の部分木という。SESATでは、複雑な木の生成を避けるため、解候補として生成される木に対して次の4つの制約を課す。

- 制約1 木の高さは  $H$  以下である。
- 制約2 非終端ノードは  $2 \sim M$  個の子ノードを持つ。
- 制約3 隣り合う部分木は必ず異なる。
- 制約4 全ノードにいずれかの訓練事例が到達する。

$H$  および  $M$  の値を目的に応じて設定することで、複雑すぎる木の生成を避ける。制約3が意味論的イントロン、制約4が構文的イントロンへの対処である。SESATの学習過程で保持する決定木は、すべて上記4制約を満たすものとする。

### 4.3.2 sprig

SESATでは、高さ1の部分木を部分解とし、sprigと呼ぶ。sprigの例を図4.5に示す。根ノードを属性ノード、葉ノードをクラスノードと呼ぶ。事例に出現する属性数が  $A$ 、クラス数が  $C$  であるとき、属性ノードには  $1 \sim A$  の属性番号、クラスノードには  $1 \sim C$  のクラス番号もしくは0が割り当てられる。sprigが決定木に組み込まれる際、0が割り当てられているクラスノードは非終端ノードとなり、別のsprigの属性ノードが接続される。

各アークは  $1 \sim M$  の属性値番号によりラベル付けされる。事例の属性値に従って属性ノードからクラスノードへと走査するときは、アーク間に設定された閾値と属性値の大小関係に従ってたどるアークを選択する。閾値としては、訓練事例の属性値の範囲を  $1 \sim M$  に対応付け、ラベル平均に相当する値を設定する。属性ノードが示す属性の訓練事例における最小値を  $v_{min}$ 、最大値を  $v_{max}$  とし、隣り合う2つのアークのラベルをそれぞれ  $l_a, l_b$  とすると、閾値  $Th$  は次の式で算出される。

$$Th = (v_{max} - v_{min}) \times \frac{l_a + l_b}{2(M-1)} + v_{min} \quad (4.6)$$

例えば、子ノード数の最大値  $M$  が5、訓練事例における属性値の範囲が属性1で5~9、属性2で2~18であるとき、図4.5(i)の閾値はラベル平均3に相当する7、図4.5(ii)の閾値はラベル平均2.5、4.5に相当する8、16となる。事例の属性値に従って属性ノードからクラスノードへと走査するときは、属性値が閾値未満の場合には左のアークをたどり、閾値以上の場合には右のアークをたどる。

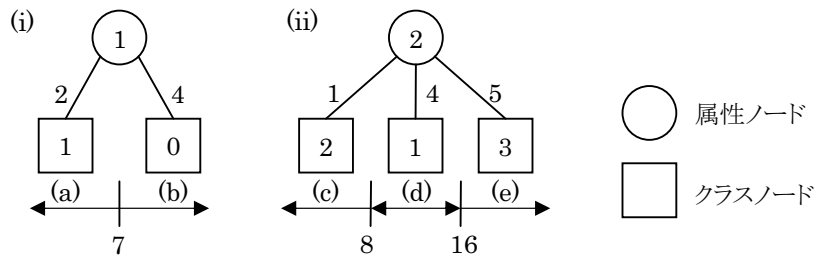


図 4.5: sprig の表現型

sprig を表す遺伝子列は、1 個の属性遺伝子と  $M$  個のクラス遺伝子からなる。図 4.5 の sprig を表す遺伝子列の例を図 4.6 に示す。属性遺伝子が属性ノードに対応し、クラス遺伝子がクラスノードに対応する。クラス遺伝子が  $-1$  のとき、sprig は対応するノードを持たない。図 4.6 のクラス遺伝子 (a) ~ (e) が、それぞれ図 4.5 のクラスノード (a) ~ (e) に対応する。 $-1$  以外のクラス遺伝子の位置を表す遺伝子座  $1 \sim M$  がアークのラベルとなる。

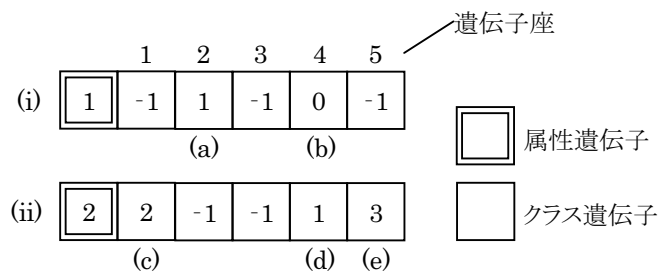


図 4.6: sprig の遺伝子型

sprig の遺伝子列を新たに生成する場合は、制約 2 および制約 3 を満たす範囲で各遺伝子をランダムに設定する。 $I_s$  個の sprig を生成し、初期集団とする。

### 4.3.3 決定木構成子

SESAT における全体解を決定木構成子と呼ぶ。決定木構成子は sprig へのポインタと解候補決定木から構成される。解候補決定木とは、各ポインタが指している sprig を参照して形成された決定木である。決定木構成子の例を図 4.7 に示す。

各ポインタは解候補決定木の非終端ノードを始点、sprig 集団の個体を終点とし、解候補決定木の各部分が生成時に参照した sprig を表している。決定木構成子を生成する際には以下の手順に従う。

Step 1 sprig 集団から無作為に選択した sprig の値に基づいて部分木を生成し、解候補決定木とする。非終端ノードから sprig にポインタを張る。

Step 2 下位の部分木が不定の非終端ノードに対して、無作為に選んだ sprig による部分木を付加

し、非終端ノードから sprig にポインタを張る．レベルが  $H$  の非終端ノードはクラス未定終端ノードとする．

Step 3 すべての非終端ノードについて下位の部分木が定まるまで、Step 2 を繰り返す．

Step 4 すべての訓練事例で解候補決定木を走査する．

Step 5 クラス未定終端ノードのクラスを決定する．

Step 6 訓練事例の到達しない部分木を削除する．

Step 7 隣り合う同一部分木を統合する．

Step 8 子ノードの数が1つである非終端ノードを削除し、当該ノード以下の部分木を接続する．

クラス未定終端ノードとは、制約 1 を満たすために暫定的に作られた終端ノードであり、Step 5 において訓練事例における正解率が最も高くなるようにクラスが割り当てられる．これにより、同一の sprig を参照する非終端ノードでも子ノードの種類が異なる可能性が生じる．Step 6 と Step 7 では、それぞれ制約 4 および制約 3 が満たされるように決定木構成子を変形する．子ノードが1つとなって制約 2 が満たされなくなった場合は Step 8 で修正する．

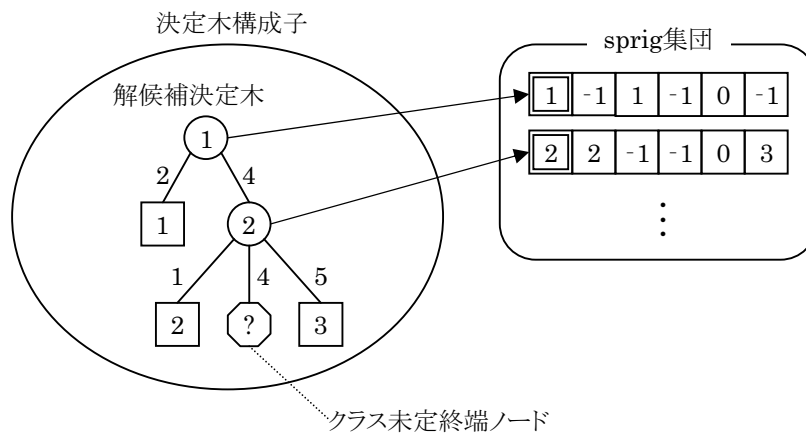


図 4.7: 決定木構成子

決定木構成子集団の個体数として指定された  $I_t$  個の決定木構成子を生成し、初期集団とする．複数の決定木構成子から参照される sprig や、どの決定木構成子からも参照されない sprig が存在することもある．

#### 4.3.4 適応度

C4.5 では、正解率を高める指標として情報利得比を、過学習を回避するための指標として分類誤り率を採用している．情報利得比に基づいて決定木を生成した後、各ノードにおける分類誤り率により枝刈りを行なう．一方、SESAT における決定木構成子は解候補決定木の生成後に評価するため、決定木構成子の適応度では、生成後の決定木の正解率を計ると同時に、木全体としてどの程度過学習

が起きているかを見極める必要がある。また、子個体が生成されるたびに適応度を算出するため、処理時間の面から各データでの1回の走査結果だけで算出できる適応度が望ましい。

4.2.3節で述べたMDL基準では、木の記述長と誤差の記述長の和により、両者のトレードオフを評価している。誤差の記述長が小さいほど正解率が高く、木の記述長が小さいほど決定木は簡素である。複雑な木では過学習が起きている可能性が高いことを考慮すると、木の記述長が小さくなるようにすることで結果的に過学習を回避できると考え得るが、過学習に関して直接的に配慮した指標とはいえない。

SESATでは、特に過学習という観点から個体を評価し、結果的に簡素な決定木が生成される、という手法を試みる。多くの訓練事例を正しく分類するように、訓練事例に特化した決定木を生成した場合、各訓練事例が互いに異なる終端ノードで正解と判断されるような決定木になることが予想される。各訓練事例間に矛盾がなく、決定木の表現能力が十分であれば、このような決定木により100%の正解率で訓練事例の分類が可能となる。したがって、終端ノードごとの正解数の散らばりが小さいほど、過学習の可能性が低いと考えられる。また、正解事例数が多い終端ノードほど、根ノードから終端ノードまでの経路で多数の事例の特徴を表しているといえ、予測正解率の向上に貢献するといえる。

全正解事例数が10個である図4.8(a)~(d)の決定木を例に挙げる。各終端ノードでの正解事例数が図中に記された数であったとき、(a)~(c)の木は同形であるが、(a)よりも(b)、(b)よりも(c)の方が、多くの事例を説明できる経路が含まれており、(c)が最も大きな特徴を表現できている。また、(c)と(d)では、両者とも8個の事例を説明する経路が含まれているが、(c)よりも(d)の方が正解数の散らばりが小さく、望ましい決定木であるといえる。

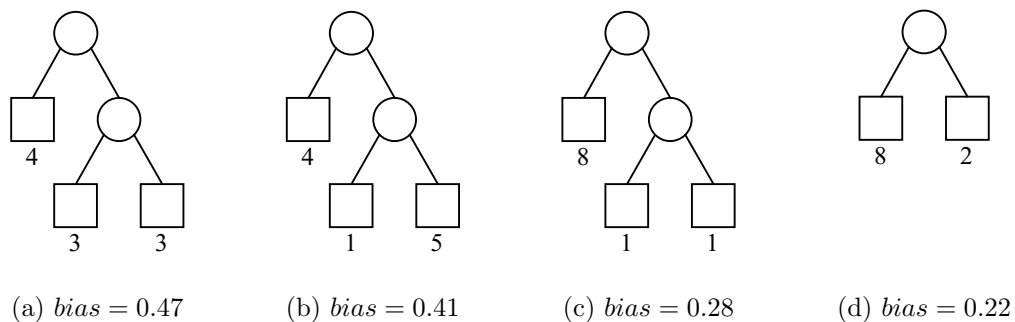


図 4.8: 全正解事例数が10の場合の正解局在率の例

以上の考察より、正解数の散らばりを表す正解局在率をエントロピーをもとに定義し、正解率が高く正解局在率が低いときに高い値を示す適応度によって、決定木構成子を評価する。

決定木構成子  $T$  の正解局在率  $bias(T)$  は次式に従って算出する。ここで、全正解事例数を  $c$ 、 $n$  個



の終端ノードにおける正解事例数をそれぞれ  $c_1 \sim c_n$  とする．

$$bias(T) = \begin{cases} -\frac{\sum_{i=1}^n \frac{c_i}{c} \log_2 \frac{c_i}{c}}{-\log_2 \frac{1}{c}} & (c \neq 0) \\ 1.0 & (c = 0) \end{cases} \quad (4.7)$$

式 (4.7) は、正解事例が最も散らばっているときを基準として、 $T$  の正解数がどの程度散らばっているかを表している．すべての正解事例が1つの終端ノードに到達したときに0、互いに異なる終端ノードに到達したときに1となり、終端ノードごとの正解事例数の散らばりが大きいほど大きな値を取る．図 4.8(a) ~ (d) の決定木の正解局在率はそれぞれ 0.47, 0.41, 0.28, 0.22 となり、前述した意図が数値として表現されていることがわかる．

訓練事例の正解率  $acc(T)$  と正解局在率  $bias(T)$  から、決定木構成子の適応度  $tfit(T)$  を次式により算出する． $\alpha$  は正解局在率を考慮する度合を示す定数である．

$$tfit(T) = acc(T) \cdot (1 - \alpha \cdot bias(T)) \quad (4.8)$$

sprig は、当該 sprig にポイントを張っている部分木により評価する．sprig  $S$  を参照する部分木のうち、最も適応度の高い決定木構成子に属する部分木を  $S$  の最良部分木と呼ぶ．最良部分木の属する決定木構成子の適応度を  $S$  の適応度とする．制約を満たすための変形により、sprig の表現型とポイント元の部分木が一致しない場合があるが、sprig の適応度算出の際に、sprig が最良部分木を表現するよう sprig の遺伝子に変更を加える．これにより、適応度の高い決定木構成子に参照される sprig が高い評価を受け、その性質が集団内に広まる可能性が高くなる．

#### 4.3.5 世代交代

sprig 集団の世代交代は HS モデルにより行なう．突然変異は全遺伝子に対して確率  $p_m$  で発生させる．取り得る値の中から1つの値をランダムに選択し、新たな遺伝子の値とする．交叉および突然変異により終端ノード数が2個未満になった場合は、ランダムに位置番号と遺伝子の値を設定し、制約 2 を満たすようにする．子個体が制約 3 を満たさない場合は、問題箇所の遺伝子を他の遺伝子に置き換える．

決定木構成子集団の世代交代モデルとしては MGG モデルを採用する．子個体としては、以下の4種類の木  $C_1 \sim C_4$  を解候補決定木とする決定木構成子を生成する．ここで、親個体として選択した決定木構成子の解候補決定木を  $P_1, P_2$  とし、 $P_1$  と  $P_2$  からランダムに選択したノードを  $n_1, n_2$  とする．

$C_1$ :  $P_1$  の  $n_1$  以下の部分木を  $P_2$  の  $n_2$  以下の部分木で置き換え、確率  $p_m$  で突然変異オペレータを適用した木

$C_2$ :  $P_2$  の  $n_2$  以下の部分木を  $P_1$  の  $n_1$  以下の部分木で置き換え, 確率  $p_m$  で突然変異オペレータを適用した木

$C_3$ : ポインタが指している sprig の遺伝子の値に従って  $P_1$  を作り直し, 確率  $p_m$  で突然変異オペレータを適用した木

$C_4$ : ポインタが指している sprig の遺伝子の値に従って  $P_2$  を作り直し, 確率  $p_m$  で突然変異オペレータを適用した木

$C_1, C_2$  は 1 点交叉により生成された木であり,  $C_3, C_4$  は sprig の進化を反映させた木である.  $C_3, C_4$  に対する変更処理は根ノードから順に行なう. 子ノードの減少, あるいは非終端ノードから終端ノードへの変更が発生した場合には不要な部分木を削除する. 子ノードの増加, あるいは終端ノードから非終端ノードへの変更が発生した場合には新たに部分木を追加する.

突然変異は全非終端ノードに対して確率  $p_m$  で発生させる. 当該ノードのポインタが異なる sprig を指すように変更し, 以下の部分木を作り変える.

sprig 集団と決定木構成子集団をランダムに生成した後, 評価と世代交代を  $G$  回繰り返す, 最良個体の解候補決定木を SESAT の出力解とする.

## 4.4 評価実験

共生進化が最適木の探索に有効であることを示すため, SESAT, および 4 つの比較対照システムにおいて評価実験を行なった. 以下では, 実験方法と結果を記すとともに, 結果に対する考察を述べる.

### 4.4.1 実験方法

SESAT と比較対照するための 4 つのシステムを以下に示す. SESAT-MDL との比較により SESAT における適応度定義の効果を確認し, ESAT との比較により 2 集団並行進化の有効性を示す. また, ランダムサーチ, および情報量に基づく既存システムとの差異を示すため, RSAT と C5.0 との比較をも行なう.

**SESAT-MDL** SESAT の過学習の抑制を MDL 基準を用いて行なうシステム. 式 (4.8) の  $bias(T)$  を式 (4.2) の  $MDL$  に置き換え, 4.2.3 節に示した [Iba 94] と同様の手順で適応度を算出する.  $W_{size}$  は 30 とする.

**ESAT** 1 集団 GP に基づいて解を探索するシステム. SESAT から sprig に関する処理を除き, 解候補決定木を個体とする集団のみを保有する. 初期化の段階では, SESAT が生成し得る木の範囲内でランダムに各値を決定して, 初期集団の解候補決定木を生成する. 世代交代の段階では  $C_1$  と  $C_2$  の子を生成して進化させる.

RSAT ランダムに決定木を生成するシステム．ESAT の初期集団形成時と同様，ランダムに候補決定木を生成し，式 (4.8) により適応度を算出するルーチンを繰り返す．適応度が最大となる決定木を最終出力とする．

C5.0 代表的な決定木生成システム C4.5[Quinlan 93] の後継システム．各パラメータはデフォルトの設定を使用する．

各システムにおいて，機械学習の分野で広くベンチマークデータとして用いられている UCI 機械学習リポジトリのデータ [UCI] の分類対象事例とした．使用した 12 種類のデータの事例数，属性数，クラス数を表 4.3 に示す．

表 4.3: SESAT の評価実験に用いたデータ

データ名	事例数	属性数	クラス数
australian	690	14	2
balance-scale	624	4	3
breast-cancer	683	9	2
bupa	345	6	2
glass	214	9	6
heart-c	297	13	2
iris	150	4	3
monks-1	124	6	2
monks-2	170	6	2
monks-3	123	6	2
pima-indians	768	8	2
post-operative	87	8	3

実験は，表 4.3 に示した事例数のうち，10 分の 9 を訓練事例，10 分の 1 をテスト事例とする 10-fold クロスバリデーションとし，訓練事例における適応度の推移，テスト事例における正解率と決定木のノード数を調査する．SESAT，SESAT-MDL，ESAT，RSAT では，乱数による動作の変動の影響を抑えるため，各事例による決定木生成を 10 回繰り返したときの平均値により評価する．

SESAT，SESAT-MDL，ESAT で設定したパラメータを表 4.4 に示す． $M$  と  $H$  に関しては，C5.0 により生成された決定木に基づいて設定した．australian のあるデータで生成された決定木は 14 個の子ノードを持つ非終端ノードを含んでいるが，それ以外のデータで生成されたすべての決定木の非終端ノードの子ノード数は 4 以下となった．また，pima-indians では高さが 11 の決定木が 2 つ，13 の決定木が 1 つ生成されたが，それ以外のデータで生成されたすべての決定木の高さは 10 以下となった．この結果により，一部の例外を除いて， $H$  と  $M$  は C5.0 で生成された決定木よりもノード数が多い決定木が生成可能となるようにした．

その他のパラメータには，表 4.3 のデータの一部による予備実験で安定して良い結果が得られた値を用いた．表 4.4 に示した値を大幅に変えない限り，予測正解率や木のノード数は大きく変化しない

表 4.4: SESAT の評価実験に用いたパラメータ

パラメータ	値
決定木構成子の適応度の係数 $\alpha$	0.2
突然変異確率 $p_m$	0.01
sprig 集団の個体数 $I_s$	400
決定木構成子集団の個体数 $I_t$	1000
世代交代回数 $G$	50000
木の高さの上限値 $H$	10
sprig のクラスノード数の上限値 $M$	5

ことが予備実験の結果により示されている．RSAT では世代交代を行わないため，解候補決定木の生成・評価ルーチンは，SESAT における決定木構成子の選択回数と同じ回数繰り返した．

#### 4.4.2 予測正解率と木の簡素さ

各システムのテスト事例における平均正解率と，生成された決定木の平均ノード数を表 4.5 に示す．平均正解率はデータごとに多少のばらつきがあるが，正解率の母平均に関して検定を行なったところ，正解率の母平均は有意水準 0.05 で RSAT，C5.0 よりも SESAT の方が高いという結果が得られた．SESAT と SESAT-MDL，ESAT の正解率の間には有意差は見られなかった．平均ノード数については，RSAT，SESAT，SESAT-MDL，ESAT，C5.0 の順で少なく，各母平均間に有意水準 0.05 で差があるという検定結果が得られた．

SESAT で生成される決定木では，ラベル平均に対応した属性値を閾値とする範囲のみがアークをたどる条件となる．一方，C5.0 による決定木では，個々の訓練事例の属性値から求めた値を分岐の閾値としたり，属性値の等値関係を分岐の条件としたりすることもできる．このように SESAT と C5.0 とでは生成可能な決定木が異なるが，検定結果によると予測正解率の母平均は SESAT の方が若干上回っている．したがって，SESAT における決定木の表現力は予測正解率の向上を特に抑制するものではないと考える．

生成された決定木のノード数については，SESAT が C5.0 の平均 70%程度となっている．SESAT では他の進化的計算と同様，処理に多少の時間を要し，学習時間は C5.0 と比べて大幅に劣る．しかし，本研究で目標とする予測正解率と木の簡素さの両面において C5.0 を上回る決定木が生成されたことから，決定木生成における共生進化の適用可能性が示唆されたものと考えられる．

SESAT では，SESAT-MDL と同程度の予測正解率を持ち，若干ではあるが簡素な決定木が生成された．両システムの平均ノード数の大小関係はデータにより異なり，その差は大きいとはいえない．しかし，過学習回避への着目によって簡素な決定木を生成することをねらった正解局在率が，MDL 基準と同等に簡素な決定木を生成できることが示されたといえる．

表 4.5: テスト事例における平均正解率と平均ノード数

データ	SESAT		SESAT-MDL		ESAT		RSAT		C5.0	
	正解率 [%]	ノード数	正解率 [%]	ノード数	正解率 [%]	ノード数	正解率 [%]	ノード数	正解率 [%]	ノード数
australian	84.9 ( 4.1)	16.4 (3.3)	85.2 ( 4.4)	18.2 (2.8)	85.1 ( 4.0)	14.8 ( 5.8)	85.2 ( 3.8)	7.1 (2.5)	84.5 ( 5.1)	27.0 ( 9.9)
balance-scale	78.7 ( 4.9)	26.5 (5.1)	79.5 ( 5.6)	31.0 (5.0)	80.1 ( 5.6)	39.6 ( 6.2)	68.8 ( 7.3)	9.7 (6.6)	77.3 ( 5.0)	84.0 ( 9.9)
breast-cancer	95.9 ( 2.3)	16.1 (2.9)	96.1 ( 2.7)	16.2 (2.1)	95.8 ( 2.4)	19.7 ( 5.0)	95.2 ( 2.4)	7.8 (3.6)	96.3 ( 2.3)	18.2 ( 4.9)
bupa	63.1 ( 8.0)	18.4 (3.8)	61.8 ( 8.3)	20.2 (4.0)	62.3 ( 8.3)	24.9 ( 7.7)	61.7 ( 8.5)	9.8 (4.1)	64.1 ( 7.0)	50.2 (10.3)
glass	63.1 (11.6)	20.0 (3.0)	62.0 (11.7)	17.3 (2.7)	64.2 (11.7)	31.9 ( 5.4)	54.4 (12.4)	15.0 (5.6)	69.5 (11.7)	47.4 ( 6.1)
heart-c	77.4 ( 7.6)	21.6 (3.5)	78.1 ( 6.6)	20.4 (3.0)	77.3 ( 8.2)	31.9 ( 6.7)	71.4 ( 9.6)	9.4 (4.2)	77.6 (10.3)	36.8 ( 6.0)
iris	95.6 ( 5.1)	6.9 (1.7)	95.2 ( 4.7)	6.0 (0.0)	94.7 ( 6.5)	6.3 ( 0.9)	92.4 ( 8.8)	5.0 (1.5)	92.7 ( 6.6)	8.0 ( 1.4)
monks-1	99.6 ( 2.5)	14.3 (0.8)	99.4 ( 4.1)	13.0 (0.1)	98.9 ( 3.5)	14.5 ( 1.0)	76.3 (11.3)	11.0 (4.4)	80.8 (11.1)	21.1 ( 5.1)
monks-2	67.6 (12.8)	21.5 (4.8)	69.8 (11.3)	22.6 (4.5)	64.5 (13.5)	25.0 (10.3)	63.4 (12.7)	9.8 (4.7)	56.5 (11.8)	21.1 (13.8)
monks-3	92.5 ( 5.2)	7.0 (2.9)	90.3 ( 6.0)	10.2 (4.2)	92.9 ( 5.5)	6.1 ( 2.5)	92.3 ( 6.8)	6.5 (3.7)	93.3 ( 5.3)	12.0 ( 0.0)
pima-indians	73.3 ( 6.4)	20.3 (4.1)	73.8 ( 7.8)	21.7 (3.5)	73.1 ( 6.7)	26.7 ( 6.2)	73.6 ( 6.1)	6.9 (2.9)	74.2 ( 7.6)	50.2 (13.4)
post-operative	72.3 (17.7)	9.8 (2.3)	67.5 (17.4)	16.0 (3.1)	74.1 (18.7)	9.5 ( 1.7)	68.6 (18.2)	7.2 (2.3)	68.8 (17.9)	4.6 ( 3.9)
全平均	80.3	16.6	79.9	17.7	80.3	20.9	75.3	8.8	78.0	31.7

(括弧内は標準偏差)

### 4.4.3 生成された決定木

実験に用いた 12 種類のデータのうち、C5.0 よりも SESAT において正解率が高く、簡素な決定木が得られた iris と monks-1 を取り上げ、両システムで生成された決定木を比較する。

#### データ iris での決定木

iris は、がく片の長さや幅、および花弁の長さや幅からアヤメを 3 つに分類するためのデータであり、4 つの属性値は連続値を取る。属性番号とクラス番号の表す意味を以下に示す。

属性 1 : がく片の長さ (cm)

属性 2 : がく片の幅 (cm)

属性 3 : 花弁の長さ (cm)

属性 4 : 花弁の幅 (cm)

クラス 1 : Iris Setosa

クラス 2 : Iris Versicolour

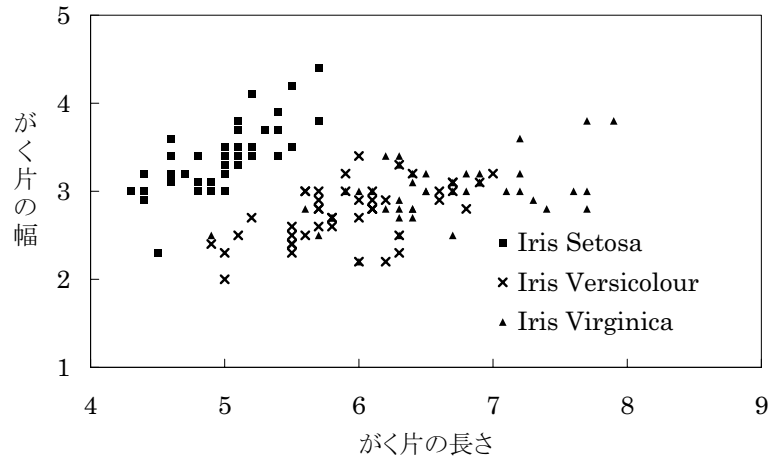
クラス 3 : Iris Virginica

実験で用いた 10 種類の訓練事例集合のうち 1 つにおいて、がく片の長さや幅、および花弁の長さや幅による 3 種類のアヤメの分布を図示すると、図 4.9 のようになる。この散布図から、Iris Setosa の特徴は他と区別しやすいが、Iris Versicolour と Iris Virginica には同じような値を持つ事例が多く、分類は難しいことがわかる。また、がく片と花弁では、花弁による分類の方が比較的容易であるといえる。

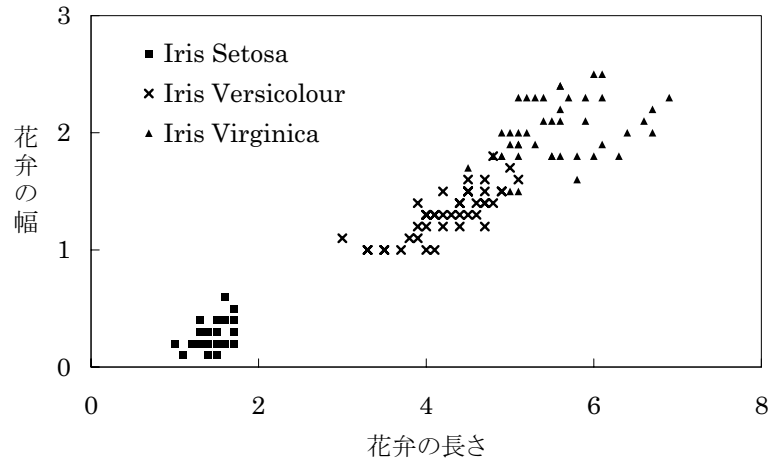
この訓練事例を用いて SESAT により生成された決定木が図 4.10(a)、C5.0 により生成された決定木が図 4.10(b) である。図 4.10、および後掲する図 4.11、図 4.12 において、終端ノードの横の“ $n/m$ ”は、当該終端ノードにたどり着いた  $m$  個の事例のうち、 $n$  個が正解であったことを示す。

両者とも、分類の容易なクラス 1 の Iris Setosa を最初に分別し、残りの事例を花弁の長さや幅を用いて分類する規則となっている。SESAT で生成された決定木 (a) では、まず花弁の幅で Iris Setosa と Iris Virginica であると考えられるものを分別し、次に花弁の長さで Iris Versicolour と例外的な Iris Virginica を分別している。これに対し、C5.0 で生成された決定木 (b) では、花弁の長さや幅を交互に用いて確実に分別できるものを順に除外していき、最後に値が近い事例を分別している。終端ノードに着目すると、両者とも各種類のアヤメを表すノードが 1 つずつあるが、(a) ではその他に例外的なノードが 1 つであるのに対し、(b) には例外的なノードが 2 つある。以上より、SESAT で生成された決定木の方が、各種類のアヤメの大まかな特徴を表現できていると考えられる。

訓練事例に対する正解率は SESAT で 97%、C5.0 で 97.8% となり、C5.0 の方が正しく分類できた事例が 1 事例多くになっている。しかし、テスト事例に対する正解率は SESAT で 100%、C5.0 で



(a) がく片の長さ と 幅



(b) 花弁の長さ と 幅

図 4.9: 3 種類のアヤメの散布図

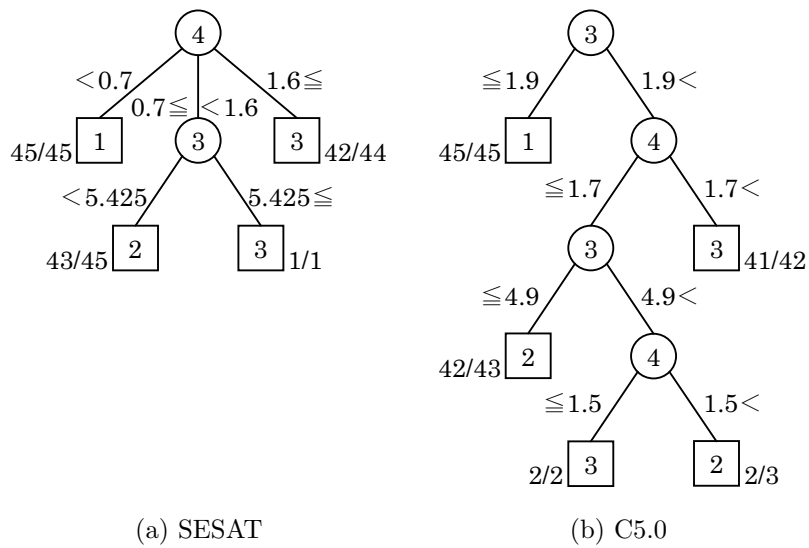


図 4.10: データ iris で生成された決定木

86.7%となった。これは、前述の通り訓練事例に特化しない大まかな特徴を決定木で表現できていることが要因と考えられる。

データ monks-1 での決定木

monks-1 は属性値として離散値を扱うデータであり、6 つの属性はそれぞれ以下の整数値を取る。

属性 1 : 1, 2, 3

属性 2 : 1, 2, 3

属性 3 : 1, 2

属性 4 : 1, 2, 3

属性 5 : 1, 2, 3, 4

属性 6 : 1, 2

クラスは以下の規則に従って割り振られている。ただし、ノイズデータは含まれていない。

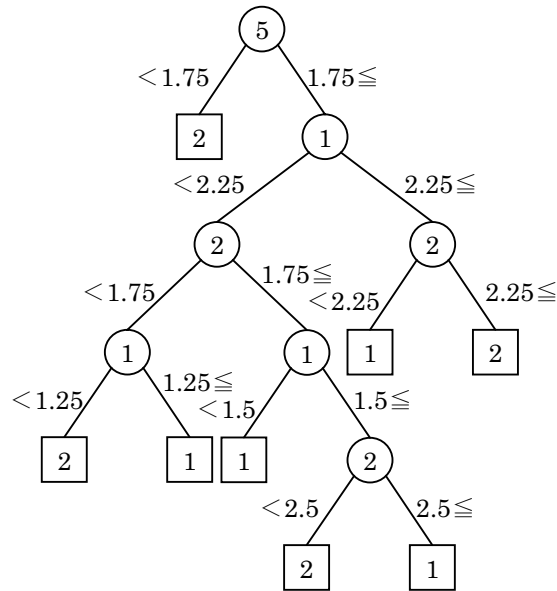
クラス 1 : 下記以外の場合

クラス 2 : 属性 1 と属性 2 の値が等しいか、属性 5 の値が 1 の場合

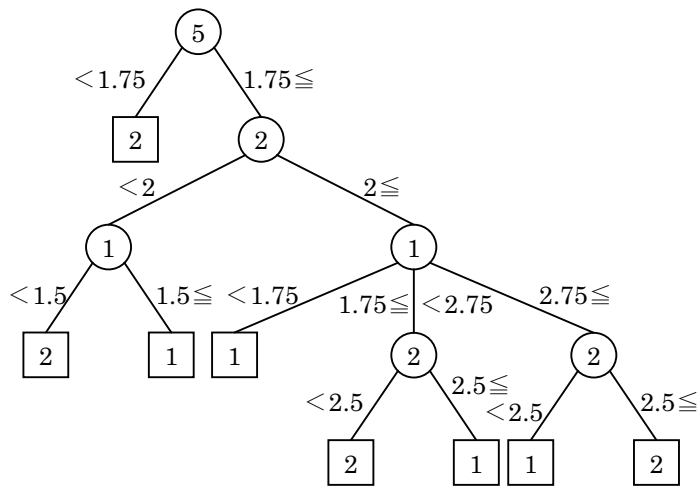
SESAT では、図 4.11(a) および (b) に示す 2 種類の決定木が主に生成された。いずれも“属性 5 の値が 1”というクラス 2 の条件を満たす事例を最初に分別し、その後で“属性 1 と属性 2 の値が等しい”を表すような規則になっている。ノード数は (a) で 15、(b) で 14 となったが、正解率は訓練事例とテスト事例の両方において 100%を得ている。

一方、C5.0 では図 4.12 のような決定木が生成された。デフォルトの設定で C5.0 を動作させると、入力データの種類を自動判別し、属性値が離散値であると判断した場合にはアークに各属性値を割



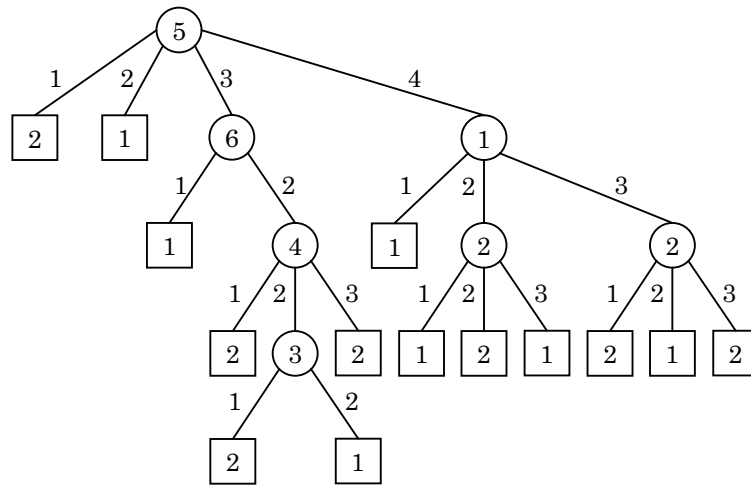


(a) ノード数 15 の決定木

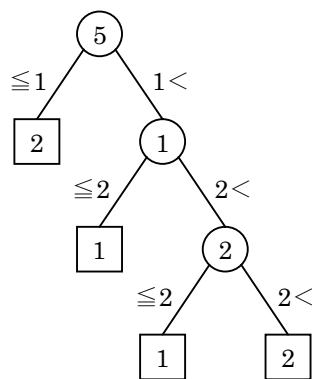


(b) ノード数 14 の決定木

図 4.11: データ monks-1 で SESAT により生成された決定木



(a) 離散値扱いにした場合



(b) 連続値扱いにした場合

図 4.12: データ monks-1 で C5.0 により生成された決定木

り振った形の決定木が生成され、SESAT で生成されるような閾値で分岐する決定木は生成されない。比較のため、属性値が連続値であると指定した場合についても決定木の生成を試みた。デフォルトの設定で生成された決定木が図 4.12(a)、属性値は離散値であると指定した場合の決定木が図 4.12(b) である。

(a) では、訓練事例に対する正解率が 88.4%、テスト事例に対する正解率が 58.33% となり、ノード数も 21 と非常に複雑な決定木が得られた。(b) ではノード数が 7 個の簡素な決定木が生成されたが、訓練事例に対する正解率は 80.4%、テスト事例に対する正解率は 75.0% となり、データの意図する規則を表現することはできなかった。両者とも“属性 5 の値が 1”という条件は表現できているものの、“属性 1 と属性 2 の値が等しい”という条件は、(a) で属性 5 の値が 4 でかつ属性 1 の値が 2 の場合にのみ、(b) で属性 1 と属性 2 の値が 3 の場合にのみ表現されるにとどまった。

#### 4.4.4 探索能力と過学習の可能性

同じ適応度関数を用いている SESAT、ESAT、RSAT において、訓練事例で学習を行なった際の平均適応度を表 4.6 に示す。SESAT と ESAT の平均適応度は、どのデータにおいても近い値となったが、RSAT はいずれのデータでも他より低い適応度となった。遺伝的アルゴリズムによる探索のランダムサーチに対する優位性はしばしば議論されるところだが、この結果は遺伝的プログラミングがランダムサーチよりも優れた探索能力を持つことを裏付けるものといえる。

表 4.6: 訓練事例における平均適応度

データ	SESAT	ESAT	RSAT
australian	0.850 (0.005)	0.847 (0.006)	0.840 (0.004)
balance-scale	0.774 (0.005)	0.787 (0.006)	0.682 (0.009)
breast-cancer	0.949 (0.002)	0.949 (0.002)	0.929 (0.004)
bupa	0.665 (0.010)	0.667 (0.011)	0.630 (0.010)
glass	0.665 (0.016)	0.687 (0.018)	0.544 (0.021)
heart-c	0.818 (0.006)	0.827 (0.008)	0.765 (0.012)
iris	0.928 (0.004)	0.927 (0.004)	0.919 (0.005)
monks-1	0.918 (0.008)	0.914 (0.016)	0.759 (0.020)
monks-2	0.724 (0.022)	0.719 (0.036)	0.664 (0.014)
monks-3	0.897 (0.007)	0.896 (0.006)	0.886 (0.019)
pima-indians	0.751 (0.007)	0.751 (0.008)	0.733 (0.007)
post-operative	0.751 (0.018)	0.751 (0.018)	0.739 (0.018)
全平均	0.807	0.810	0.758

(括弧内は標準偏差)

次に、過学習の可能性について考える。SESAT と ESAT では、RSAT より訓練事例における適応度とテスト事例分類の正解率がともに高かったが、ノード数については RSAT の方が簡素な決定木を生成した。式 (4.8) で定義された適応度が過学習回避に貢献しないとすると、訓練事例における

適応度が高くなるように学習を行なっても、テスト事例における正解率は低くなるはずである。また、一般にノード数の多い決定木は過学習の可能性が高い点を考慮すると、他よりノード数の少ない決定木を生成した RSAT は、過学習の可能性が低いと予測できる。ところが、SESAT と ESAT は、RSAT よりノード数の多い木を生成したにもかかわらず、テスト事例における正解率が高くなった。訓練事例における適応度も高いことから、式 (4.8) によって過学習回避に有効な適応度が定義できたと考えられる。

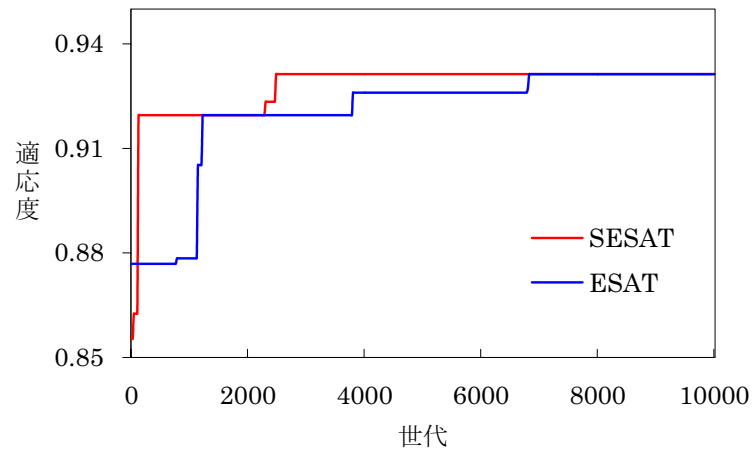
#### 4.4.5 共生進化の効果

SESAT と ESAT により生成された決定木を比較すると、予測正解率は同程度であるが、SESAT の方が簡素な決定木となっている。一般に、決定木の交叉では有効な積み木を取り入れられる反面、決定木のサイズが大きくなるという問題があるが、SESAT では解候補決定木に sprig の遺伝子を反映するという子個体の生成も行なっている。適応度の高い sprig の遺伝子を解候補決定木に反映することで、ノード数を増加させることなく、有効な解の一部である積み木を取り入れることが可能となる。部分解と全体解を並行して進化させる共生進化の特長が、SESAT により生成された決定木の簡素さに表れたと考えられる。

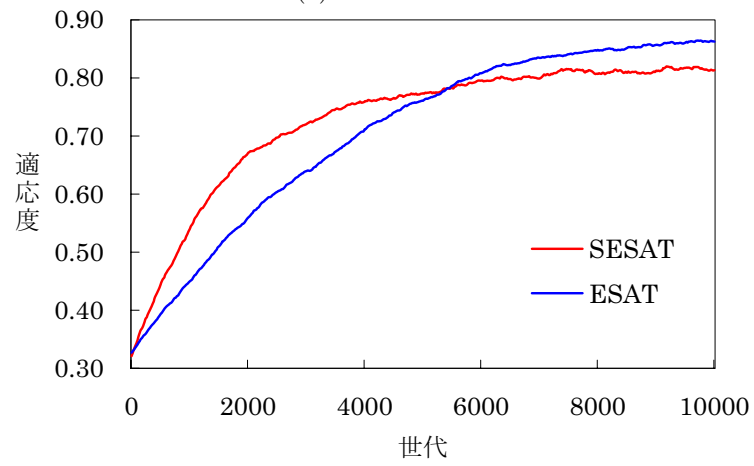
SESAT と ESAT におけるデータ iris の最良個体の適応度推移を図 4.13(a) に示す。SESAT では第 2480 世代、ESAT では第 6820 世代に最良適応度が 0.931 となり、両システムともそれ以降は変化が見られなかった。(a) のグラフには図 2.14 と同様の傾向が示されており、より早い世代において良い解に到達するという共生進化の特長が、遺伝的プログラミングに適用した場合にも現れている。

図 4.13(b), (c) は、SESAT と ESAT におけるデータ iris の全個体の平均適応度推移である。(b) が初期世代から第 10000 世代、(c) が第 20000 世代から第 50000 世代の推移を示している。(b) のグラフから、SESAT の方が学習初期において集団全体が急速に適応度の高い個体へと移行することがわかる。これは最良適応度の推移にも起因すると考えられるが、適応度 0.931 の最良個体が生成されたときの平均適応度が SESAT で 0.697、ESAT で 0.831 と大きく異なっており、平均適応度の大小関係が逆転している点は興味深い。SESAT が集団内に適応度の低い個体を含んだ状態で最良個体に到達したのに対し、ESAT は集団全体の適応度が高くなってから最良個体に到達したといえる。学習後期では、(c) のグラフが示すように ESAT の平均適応度が大きく変化していないのに対し、SESAT では徐々に平均適応度が高くなっており、第 50000 世代では同程度の平均適応度となっている。

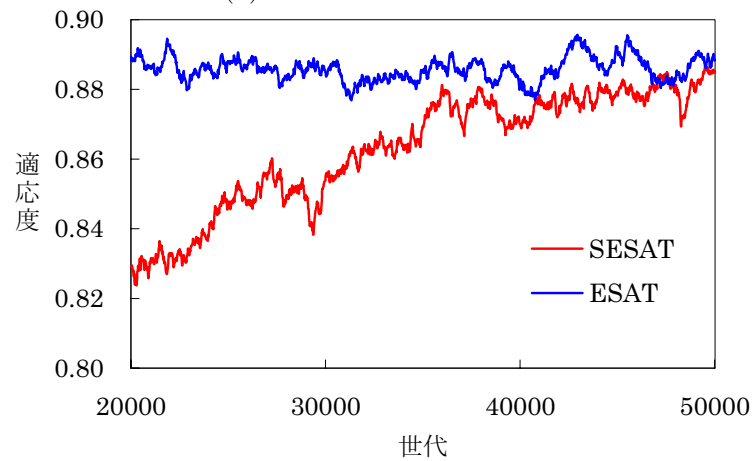
以上をまとめると、共生進化を適用しない遺伝的プログラミングの平均適応度は学習の初期段階で緩やかに上昇し、後期では大きく変化しないが、共生進化の平均適応度は学習初期で急速に上昇し、学習後期でも緩やかな上昇を続けるという特徴が見受けられる。したがって、共生進化に基づく決定木生成は早期に良い解を見つけれられるという効率的な学習能力と、評価の高い解を見つけた以降も急速に収束することなく徐々に学習を進めるといった持続的な学習能力を兼ね備えているといえる。これは、部分解を個体とする進化の「すばやく“効果的な解”を獲得する」という特徴と、全体解を



(a) 最良適応度



(b) 平均適応度《学習初期》



(c) 平均適応度《学習後期》

図 4.13: 適応度の推移

個体とする進化の「“最適な解”に確実に収束する」という特徴を表しており、2集団並行進化の効果であると考えられる。

## 4.5 まとめ

本章では、予測正解率が高く簡素な決定木の生成を目的として、共生進化による新しい決定木構築法を提案した。高さ1の部分木、および部分木の組合せにより構成される決定木をそれぞれ個体とする2集団を保持し、正解率と正解局在率により算出した適応度に従って2集団を並行して進化させる方法である。提案手法による決定木生成システム SESAT を構築し、UCI リポジトリのデータを用いて評価実験を行なった結果、SESAT は予測正解率が高く、簡素な決定木を生成することができた。

以上より、2.4.2 節で示したカテゴリ P2-W3 の問題に対しても、共生進化の適用可能性が示されたと考えられる。

## 第5章 多属性データに適した決定木の生成

決定木は分類基準が明確に表記できることから、さまざまな実用データに関する分類規則を表現するために使用されている。ポリマーに近赤外線を照射して得られたスペクトルデータを用いて、ポリマーの種類を正確に判別するための決定木生成手法を提案する。

### 5.1 ポリマー判別

近年、資源枯渇と環境破壊の問題が深刻化する中で、プラスチックに代表されるポリマー製品のリサイクルの必要性が高まっている。リサイクルのための処理方法はポリマーの種類ごとに異なるため、リサイクルに先立ってポリマーの種類を判別しなければならない。特に、ポリ塩化ビニルにはダイオキシン発生の原因であるクロルが含まれているため、他のポリマーと分別して処理することが不可欠である。ポリマーの種類を一度に同定可能な手法が理想的ではあるが、処理の性格上、高い正解率が求められるため、特殊な処理を行なうべき種類のポリマーを正確に選り分けることが重要といえる。

また、リサイクルコストを考慮すると、判別手法は簡便であることが望ましい。近赤外スペクトルは、赤外スペクトルと比較して透過率が高く、異物の付着や凹凸、着色などの表面状態から受ける影響が少ない。すなわち、近赤外スペクトルの波形を測定する際に、測定対象の汚れを落としたり、変形したりといった前処理を必要としない。したがって、近赤外スペクトルは非破壊的かつ迅速なポリマー判別に有用であるといえ、低コストによるリサイクルの実現が可能なデータとして注目されている。

近赤外スペクトルを用いたポリマー判別法として、ニューラルネットワークや1/fゆらぎなどに基づく手法が提案されている [松本 99, 小野寺 99]。しかし、両手法では判別を正確に行なうことのみを目標としており、判別処理の透明性に欠ける。ポリマーの特徴を表す波長とそのときの吸光度が具体的な数値で明示されると、スペクトル全体ではなく特定の波長の吸光度のみを測定すればよいので、安価な判別装置が実現できる。また、化学分野の研究者からは『ポリマーの判別規則を分析化学ハンドブックに掲載し、有機化学分野の研究に生かしたい』という要望もあり、化学分野への貢献という観点からも、判別規則の明示は重要である。

第4章で学習対象とした決定木では、各クラスの特徴である分類基準を明確に示すことができるため、ポリマー判別規則の表現に適しているといえる。波長を属性、吸光度を属性値、ポリマーの種類をクラスとする決定木でポリマーの判別規則を表現すると、ポリマーの特徴を表す波長と吸光度

の明示が可能となる。しかし、これまでに代表的な決定木生成システム CART[Breiman 84] によるポリマー判別のための決定木生成が試みられているが、試料の特徴が十分に反映された決定木は得られていない [貝原 03]。

第 4 章では、共生進化に基づく決定木生成手法が簡素な決定木の生成に有効であることを確認した。しかし、近赤外スペクトルデータは第 4 章の評価実験に用いたベンチマークデータと比較して属性数が非常に多いという特徴がある。近赤外スペクトルデータを用いて正確なポリマー判別を行なうためには、多属性データに適した手法により決定木を生成する必要がある。

## 5.2 共生進化に基づくポリマー判別

研究には実際に試料から測定した実データを使用した。以下では、研究に用いたポリマーの近赤外スペクトルデータについて説明し、第 4 章の評価実験に用いたベンチマークデータとの相違点を示す。その上で、ポリマーを正確に判別するための決定木を生成するにあたり、強化すべき機能を列挙する。

### 5.2.1 ポリマーの近赤外スペクトルデータ

本研究では、表 5.1 に示す 18 種類 258 試料のポリマーから得られた近赤外スペクトルを用いて決定木を生成する。近赤外分光装置を用いて各試料の近赤外波長領域 1200 ~ 2400nm の吸光度を 0.5nm おきに測定し、得られた 2401 個の吸光度  $x_1, x_2, \dots, x_{2401}$  を正規化して使用する。測定時のノイズレベルは  $0.0016 \pm 0.0022$  であり、測定値とノイズの標準偏差の比である SN 比は 260 と十分大きい。

正規化データ  $\vec{x}'$  は測定データ  $\vec{x} = (x_1, x_2, \dots, x_{2401})$  を用いて次式により求められる。

$$\vec{x}' = (x'_1, x'_2, \dots, x'_{2401}) = \frac{\vec{x}}{|\vec{x}|} \quad (5.1)$$

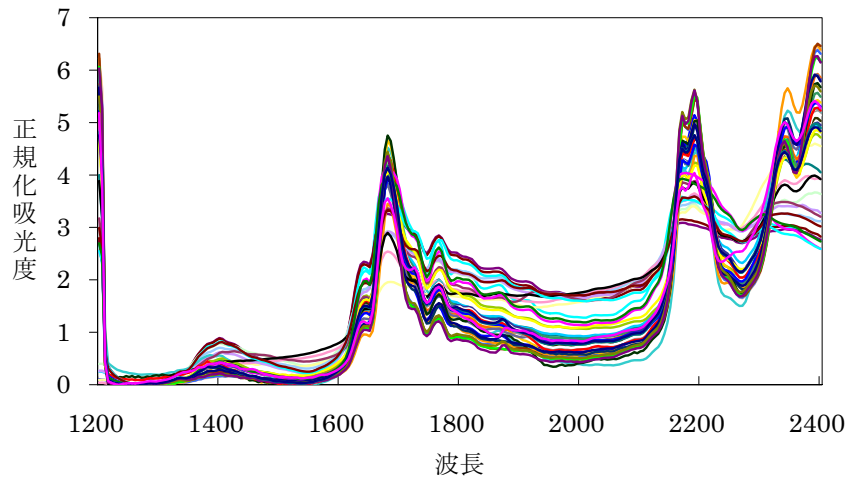
例として、ポリスチレン、ABS 樹脂、ポリ塩化ビニルの正規化後の近赤外スペクトルを図 5.1 に示す。ポリマーの種類ごとにある程度の波形の特徴は見られるが、同種類のポリマーであっても波形が類似している試料ばかりではない。ABS 樹脂のように波形がまったく異なる試料が存在する場合もある。これらの試料において注目すべき波長を見定め、正確に分別する作業が非常に困難であることは一目瞭然である。

近赤外スペクトルを用いた決定木では、1200 ~ 2400nm の 2401 種類の波長を属性、各属性に対応する正規化吸光度  $x'_1, x'_2, \dots, x'_{2401}$  を属性値、ポリマーの種類をクラスとして試料を分類することになる。近赤外スペクトルのデータ  $dat_{pol}$  には、第 4 章の評価実験で用いられた UCI 機械学習リポジトリ [UCI] のデータ  $dat_{uci}$  と比較して、以下のような相違点がある。

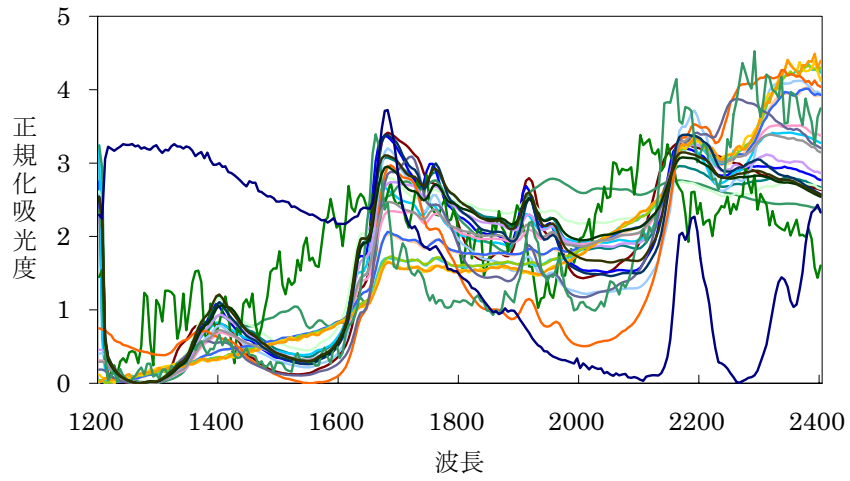
相違点 1  $dat_{uci}$  は属性数が 4 ~ 14 個であるのに対し、 $dat_{pol}$  は 2401 個と非常に多い。

相違点 2  $dat_{uci}$  の属性は年齢、性別、血圧、痛みの種類など互いに異なる意味を持つが、 $dat_{pol}$  の属性は波長という連続値を一定間隔で抽出した値を表す。

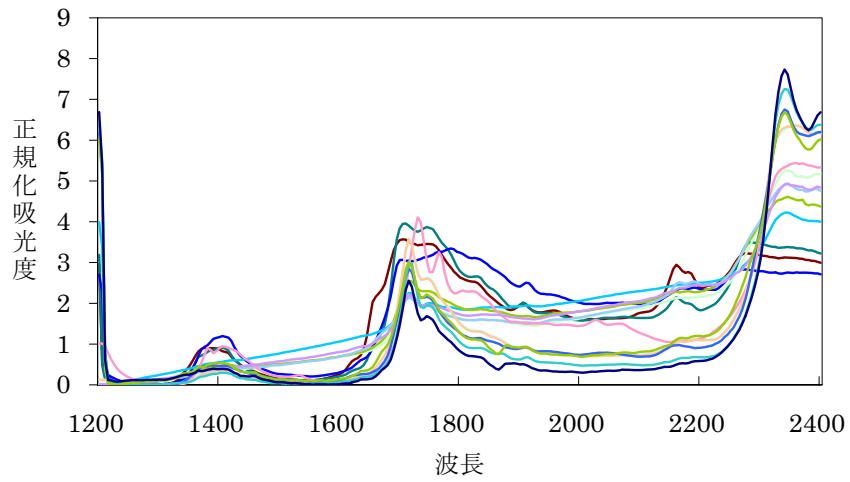




(a) ポリスチレン



(b) ABS 樹脂



(c) ポリ塩化ビニル

図 5.1: 近赤外スペクトル

表 5.1: 本研究に使用した試料

ポリマー名	略称	試料数
ポリスチレン	PS	44
ポリエチレン	PE	43
ABS 樹脂	ABS	24
ポリプロピレン	PP	31
ポリカーボネート	PC	13
ポリ塩化ビニル	PVC	15
ポリオキシメチレン	POM	18
ポリカーボネート/ABS 樹脂共重合体	PC/ABS	11
アクリロニトリル/スチレン共重合体	AS	13
ポリメタクリル酸メチル	PMMA	8
ポリエチレンテレフタレート	PET	11
ポリブチレンテレフタレート	PBT	3
尿素樹脂	UF	3
メラミン	MF	3
ポリフェノール	PF	3
ナイロン 66	PA66	6
ナイロン 6	PA6	6
セルロース	CEL	3

相違点 3  $dat_{uci}$  の属性値には連続値だけでなく整数値やカテゴリを表す値もあるが、 $dat_{pol}$  のすべての属性値は吸光度という同一対象を表す連続値である。

相違点 1 から、決定木の各ノードで属性を決める際、2401 種類の中から 1 つを選択することになるので、属性に関して詳細な記述が可能な遺伝子型が必要とされる。また、相違点 2 と相違点 3 より、属性を表す番号が変わっても、属性は波長、属性値は吸光度を表すことに変わりはなく、高さ 1 の部分木は常に『波長が  $x$  のときの吸光度が  $y$  であるときに左子ノードと右子ノードに分岐する』ということの意味する。したがって、分岐の基準となる属性、分岐の閾値、子ノードの種類をそれぞれ独立に変化することができ、詳細な記述が可能な遺伝子型により解を表現する必要がある。

### 5.2.2 ポリマー判別システム TS-SEPT

5.1 節で述べたように、リサイクルのコスト、および判別規則の理解しやすさを鑑みると、生成される決定木は簡素であることが望ましい。簡素な決定木によりポリマーを正確に判別する手段として、第 4 章で提案した SESAT の利用が考えられる。

しかし、SESAT では属性を 1 つの遺伝子で表現しているため、交叉によって属性を表す遺伝子の値が変化することはなく、親個体の属性は子個体の属性として引き継がれる。また、子ノードの種類を表す遺伝子座により分岐の閾値が決まるため、子ノードの種類と閾値を独立に変化させることはできない。さらに、分岐の閾値は非常に限られており、1 つの非終端ノードが持つ子ノード数の上限

値を  $M$  とすると、分岐の閾値は  $2M - 3$  通りしかない。以上より、前節で述べた特徴を持つ近赤外スペクトルデータを扱う場合、SESAT では多様な解候補を生成できず、ポリマーの正確な判別は難しいと考えられる。

近赤外スペクトルデータの扱いに適した遺伝子表現によりポリマーを判別するシステム TS-SEPT (Two Step Symbiotic Evolution for Polymer discriminative Trees) を提案する。TS-SEPT では、特定の種類のポリマーとそれ以外に分別する 2 分木を生成する。2 クラス分類問題とすることで試料数の偏りが問題となるが、適応度関数の定義によって試料が少数のクラスに配慮する。また、図 5.1 に見られるようなさまざまな波形を持つ試料を正確に判別するために、2 段階判別法を導入する。

TS-SEPT の設計とアルゴリズムの詳細については次節にて説明する。

### 5.3 TS-SEPT の設計とアルゴリズム

ポリマー判別システム TS-SEPT における決定木生成方法の詳細について説明する。TS-SEPT は前章で提案した SESAT を 5.2.1 節で挙げた問題点を解消するための部分解 sprig の構造、および事例数の偏りが大きい場合に対処できる適応度の定義が提案の主眼点である。特定の種類のポリマーをクラス 2、それ以外のポリマーをクラス 1 として判別を行なう。

#### 5.3.1 sprig

TS-SEPT においても、SESAT と同様に高さ 1 の部分木 sprig を部分解とする。TS-SEPT における sprig の例を図 5.2 に示す。事例に  $A$  種類の属性に対する値が含まれているとき、属性ノードには  $1 \sim A$  の属性番号、左右 2 つのクラスノードにはそれぞれ  $0 \sim 2$  のクラス番号が割り当てられる。sprig が決定木に組み込まれる際、クラス番号が 0 のクラスノードは非終端ノードとなり、別の sprig の属性ノードが接続される。クラス番号が 1, 2 のクラスノードはそれぞれクラス 1, 2 を表す終端ノードとなる。

sprig には閾値  $Th$  が与えられる。事例の属性値に従って属性ノードからクラスノードへと走査するとき、属性値が閾値以下の場合には左部分木へ、それ以外の場合には右部分木へと進む。

sprig の遺伝子型は、図 5.3 に示すような 32 ビットの 0 と 1 の並びで表現する。第  $i$  ビット目の値を  $g_i$ 、属性値の最大値を  $v_{max}$ 、最小値を  $v_{min}$  と表すと、属性番号  $At$ 、閾値  $Th$ 、左子ノードのクラス番号  $Cl_l$ 、右子ノードのクラス番号  $Cl_r$  は次式で求められる。ここで  $\%$  は整数除算の剰余を求めるとする。

$$At = \sum_{i=1}^{18} 2^{18-i} g_i \% A + 1 \quad (5.2)$$

$$Th = \frac{v_{max} - v_{min}}{2^{11} - 1} \sum_{i=19}^{28} 2^{28-i} g_i + v_{min} \quad (5.3)$$

$$Cl_l = g_{29}(g_{30} + 1) \quad (5.4)$$

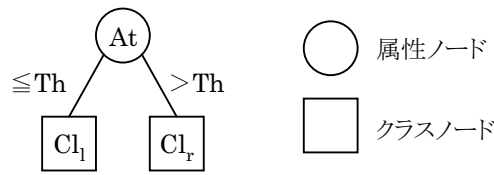


図 5.2: sprig の表現型

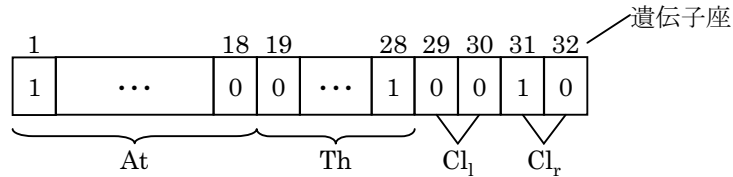


図 5.3: sprig の遺伝子型

$$Cl_r = g_{31}(g_{32} + 1) \tag{5.5}$$

最初の第 1~18 ビットで表される数の属性数による剰余が属性番号である．第 19~28 ビットは，事例に出現する属性値の範囲のどこに閾値を設定するかを表している．第 29, 31 ビットが 0 のときはそれぞれ左右の子ノードが非終端ノードであることを表し，1 のときは終端ノードであることを表す．子ノードが終端ノードのとき，第 30, 32 ビットに 1 を加えた値がクラス番号となる．

sprig の遺伝子型を以上のように定義することで， $2^{10}$  通りの閾値が表現できるようになる．ランダムに設定した  $I_s$  個のビット列を初期集団の個体とし，1 点交叉と突然変異により進化させる．

### 5.3.2 決定木構成子

sprig の組合せで表現される全体解が決定木構成子である．決定木構成子は SESAT と同様，図 5.4 に示すような sprig へのポインタと解候補決定木から構成され，解候補決定木は sprig 集団から選択した sprig のノードとアークを参照して形成される．ただし，解候補決定木は 2 分木となる．

指定された  $I_c$  個の決定木構成子をランダムに生成し，初期集団とする．1 点交叉と進化後の sprig 遺伝子の反映，および突然変異により進化する．

### 5.3.3 適応度と世代交代

過学習を回避しつつ正解率の高い決定木を生成するため，C4.5 では情報利得比と分類誤り率を，SESAT では正解率と正解局在率を用いて決定木を評価している．ポリマー判別における適応度としては，決定木の生成後に評価できる点で後者が適しているが，2 クラス間の事例数の偏りが大きいいため，正解率による評価は望ましくない．1 つの事例が不正解から正解に変わったときの正解率の変化

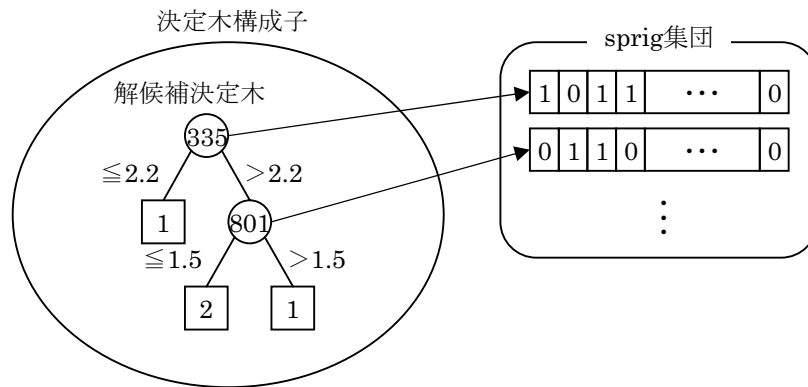


図 5.4: 決定木構成子

量は，その事例のクラスに依らず一定なので，単に事例数の多い方のクラスと判定するだけで高正解率が得られ，局所解に陥りやすいからである．

上述の問題を解決するため，正解率の代わりにクラス 2 の事例の再現率と適合率を用いて決定木構成子の適応度を算出する．再現率はクラス 2 の事例をクラス 2 として判断できた割合を表し，適合率はクラス 2 と判断した事例のうちクラス 2 の事例が占める割合を表す．決定木構成子  $T$  でクラス 2 と判定された事例の数を  $f_2(T)$ ，クラス 2 の事例の数を  $r_2$ ，決定木構成子  $T$  でクラス 2 と判定されたクラス 2 の事例の数を  $rf_2(T)$  とすると，決定木構成子  $T$  の再現率  $rec(T)$ ，適合率  $pre(T)$ ，および適応度  $tfrit(T)$  は次式で算出される．ただし， $bias(T)$  は式 (4.7) により定義される正解局在率である．

$$rec(T) = \frac{rf_2(T)}{r_2} \tag{5.6}$$

$$pre(T) = \frac{rf_2(T)}{f_2(T)} \tag{5.7}$$

$$tfrit(T) = rec(T) \cdot pre(T) \cdot (1 - \alpha \cdot bias(T)) \tag{5.8}$$

再現率と適合率の両者を向上させるため，式 (5.8) では再現率と適合率の積を用いている．情報検索の分野では，再現率と適合率を 1 つの尺度として表現するために，F-measure[Van Rijsbergen 79]を用いることが多い．F-measure の値  $F$  は次の式で算出される．

$$F = \frac{2 \cdot rec(T) \cdot pre(T)}{rec(T) + pre(T)} \tag{5.9}$$

再現率と適合率を 0.1 から 1.0 まで変化させたとき，F-measure，および再現率と適合率の積は図 5.5 のように変化する．いずれも再現率と適合率の変化により同様の増減を示していることがわかる．しかし，再現率と適合率の積では変化率がほぼ一様であるのに対し，F-measure では再現率と適合率の値が小さいときほど大きな変化率となっている．式 (4.8) と同様，分類の正確さの度合いに依らず適応度を一様に変化させるため，TS-SEPT の適応度には再現率と適合率の積を採用する．

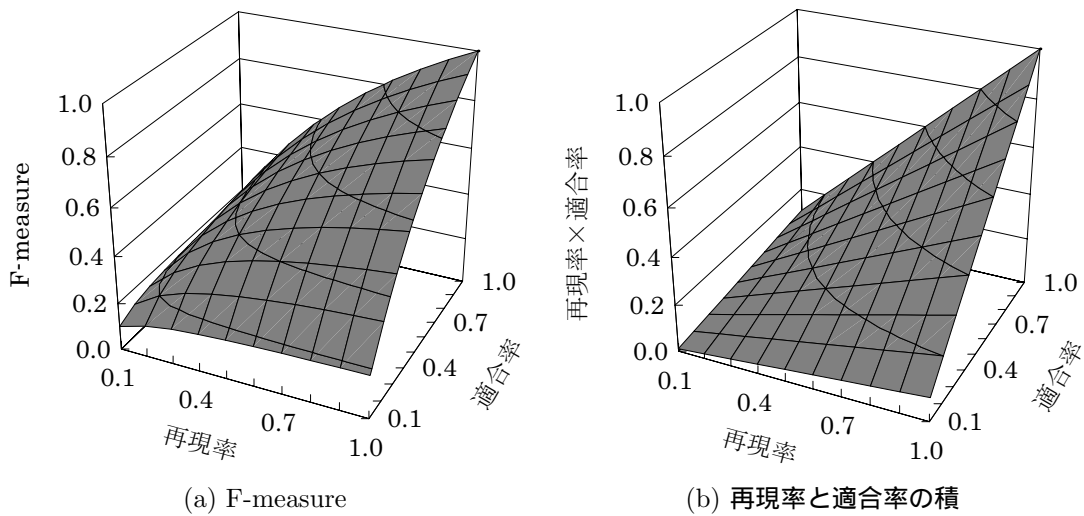


図 5.5: 再現率と適合率による値の変化

sprig の適応度は, SESAT と同様に最良部分木の属する決定木構成子の適応度とする. 式 (5.8) の適応度に基づいて個体を進化させると, 再現率と適合率が共に高い個体が次世代へと受け継がれる. したがって, クラスの事例数の差が大きい場合にも, 少数データの特徴を反映した決定木を生成することができる.

SESAT と同様に, sprig 集団の世代交代は HS モデル, 決定木構成子集団の世代交代は MGG モデルにより行なう. 初期集団の生成後, 評価と世代交代を  $G$  回繰り返す, 決定木構成子集団の最良個体の解候補決定木を出力解とする.

### 5.3.4 2 段階判別法

前節までに説明した手法にて各ポリマーを判別するための決定木を生成し, 各事例の判別結果を分析したところ, 数回の反復試行で特定の事例が毎回誤判別されていた. 誤判別される事例には両クラスの事例が含まれていることから, 異なる特徴を持つ同種ポリマーの試料や, 特徴が似ている異種ポリマーの試料が存在するといえる. 学習は各クラスの多数を占める事例, すなわち代表的な特徴を持つ事例を正解に導くよう行なわれるため, 少数の事例が持つ特徴を表現できていないと考えられる.

各クラスを代表する特徴, およびそれ以外の特徴をそれぞれ別の決定木で表し, 高正解率の判別を実現するため, 2 段階で判別を行なう方法を導入した. 決定木生成の段階では, まず最初に 1 つの決定木で正しく判別できる試料と誤判別される試料に全試料を分け, 両者を分離するための決定木  $T_{all}$  を生成する. 次に, 先の決定木により分離された 2 つの試料群を特定の種類のポリマーとその他に判別するような決定木  $T_A, T_B$  を生成する. 決定木生成手順は以下の通りである.

Step 1 全試料を特定のポリマーとそれ以外に判別する決定木を生成し, 生成された決定木で各試料が正しく判定できるか, 誤判定されるかを調べる. すべての試料が正しく判定された場合は処理を終了する.

Step 2 Step 1 で誤判定された試料を B グループの候補とする．既に候補がある場合は追加する．  
B グループの候補以外の試料を A グループの候補とする．

Step 3 A, B グループの候補をそれぞれクラス 1, 2 として, 両者を判別する決定木  $T_{all}$  を生成する． $T_{all}$  でクラス 1 と判定された試料を A グループ, クラス 2 と判定された試料を B グループとする．

Step 4 A グループの試料を特定の種類のポリマーとそれ以外に判別する決定木  $T_A$  を生成する．

Step 5 B グループの試料を特定の種類のポリマーとそれ以外に判別する決定木  $T_B$  を生成する．

Step 6  $T_A$  および  $T_B$  の正解率がともに 100%である場合は処理を終了する．100%でない場合は Step 1 に戻り, Step 1~5 を最高 5 回繰り返す．

Step 1 で正解率が 100%の決定木が生成された場合には,  $T_{all}, T_A, T_B$  は生成しない．Step 3 ではできるだけ適応度の高い決定木が生成されるよう学習を行なうが,  $T_{all}$  の正解率は必ずしも 100%であるとは限らない．各グループの候補と, Step 3 で決定したグループが一致しない場合もある．グループにいずれか一方のクラスの試料しか含まれない場合には判別の必要がないため, Step 4 または Step 5 を省略する．

Step 1 を繰り返すことで誤判定の可能性がある試料が列挙される．Step 1, 2 を繰り返す予備実験を行なったところ, 6 回目以降は新たな B グループの候補は見つからなかったため, Step 1~5 の繰り返し回数を最高 5 回とした．5 回繰り返しても 100%の正解率が得られなかった場合には, 最高正解率を得た  $T_{all}, T_A, T_B$  を最終出力とする．

Step 3~5 では 5.3.1 節~5.3.3 節に記した手法により決定木を生成する． $T_{all}$  と  $T_A, T_B$  は分別意図が異なるが, 事例に振られたクラス番号に沿った分別を目標に学習するため, 同手法により決定木を生成することとした．

$T_{all}, T_A, T_B$  の 3 つの決定木が生成された場合, 最初に全試料を  $T_{all}$  で A グループと B グループに分離し, 次に A グループの試料は  $T_A$ , B グループの試料は  $T_B$  で判別する．2 段階判別の処理の流れは図 5.6 のように図示される．

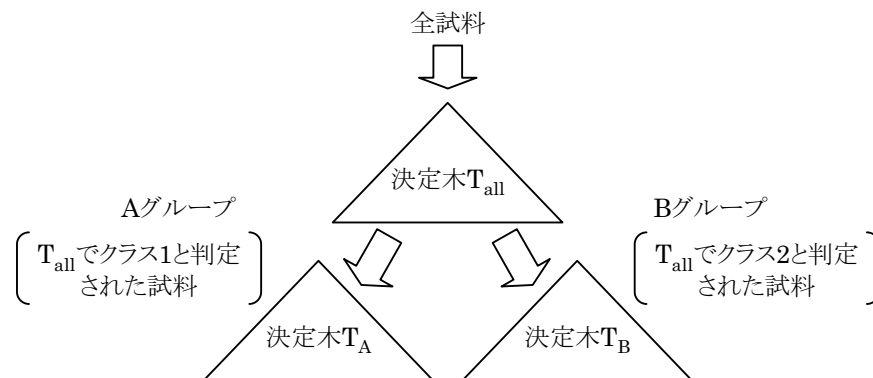


図 5.6: 2 段階判別

## 5.4 評価実験

ポリマー判別を目的とした決定木生成においても共生進化が有効であることを確認するため、TS-SEPT、および3つの比較対照システムにおいて評価実験を行なった。以下では、実験方法と結果を記すとともに、結果に対する考察を述べる。

### 5.4.1 実験方法

TS-SEPTと比較対照するための3つのシステムを以下に示す。SEPTとの比較により2段階判別法の効果を確認し、SESATとの比較により5.3.1節~5.3.3節に記した手法の有効性を示す。また、共生進化を用いない既存のシステムとの差異を示すために、C5.0との比較をも行なう。

SEPT 2段階判別を行わず5.3.1節~5.3.3節に記した手法で1つの決定木を生成するシステム。

SESAT 第4章における決定木生成システム。

C5.0 代表的な決定木生成システム C4.5[Quinlan 93]の後継システム。各パラメータはデフォルトの設定を使用する。

評価実験には表5.1に記したデータを用いた。また、各システムで設定したパラメータの値を表5.2に示す。 $\alpha$ 、 $p_m$ 、 $G$ は第4章における評価実験と同様に設定し、 $H$ はC5.0で生成された木の高さの最高値とした。 $I_s$ と $I_t$ は表5.1のデータの一部による予備実験で安定して良い結果が得られた値を用いた。

表 5.2: TS-SEPT の評価実験に用いたパラメータ

パラメータ	値
決定木構成子の適応度の係数 $\alpha$	0.2
突然変異確率 $p_m$	0.01
sprig 集団の個体数 $I_s$	800
決定木構成子集団の個体数 $I_t$	800
世代交代回数 $G$	50000
決定木の高さの上限値 $H$	5

### 5.4.2 全事例判別

全事例を用いて決定木生成を行なったときの正解率と生成された決定木のノード数を表5.3に示す。TS-SEPT、SEPT、およびSESATについては、10回の試行の最高正解率、平均正解率、平均ノード数を記す。また、TS-SEPTのTSの欄には、10回の試行のうち2段階判別用の決定木  $T_{all}$  が生成された回数を記す。



表 5.3: 全事例判別における正解率と平均ノード数

データ	TS-SEPT						SEPT				SESAT			C5.0	
	TS	正解率 [%]		ノード数			正解率 [%]		ノード数	正解率 [%]		ノード数	正解率 [%]	ノード数	
		最高	平均	$T_{all}$	$T_A$	$T_B$	最高	平均		最高	平均				
PS	9	100.0	99.8	8.8	8.3	2.3	99.6	98.9	11.8	98.8	97.0	7.9	100.0	13	
PE	8	100.0	100.0	4.0	12.3	1.0	99.6	99.6	11.6	99.6	96.3	10.1	99.2	11	
ABS	10	100.0	100.0	14.2	10.6	6.0	97.3	96.7	15.0	94.2	85.2	3.7	97.3	17	
PP	5	100.0	100.0	5.0	11.0	1.0	100.0	99.5	10.6	99.6	95.6	9.1	96.9	13	
PC	1	100.0	100.0	5.0	11.0	1.0	100.0	100.0	13.4	97.7	96.9	4.6	99.6	9	
PVC	10	100.0	100.0	7.6	12.4	1.0	99.6	99.3	13.0	96.5	95.7	2.4	99.2	13	
POM	0	100.0	100.0	-	-	-	100.0	100.0	7.0	99.6	99.6	6.6	99.2	9	
PC/ABS	10	100.0	100.0	7.6	9.4	1.0	99.6	99.3	9.6	96.5	90.7	4.6	98.1	5	
AS	4	100.0	100.0	3.5	11.0	1.5	100.0	99.7	13.4	95.0	94.1	1.4	98.8	11	
PMMA	3	100.0	100.0	4.3	7.0	1.0	100.0	100.0	9.2	99.6	98.1	4.4	99.6	5	
PET	10	100.0	100.0	7.8	6.4	1.0	99.6	99.2	7.8	99.2	98.6	5.8	99.6	9	
PBT	0	100.0	100.0	-	-	-	100.0	100.0	7.6	98.8	98.8	1.0	100.0	5	
UF	0	100.0	100.0	-	-	-	100.0	100.0	5.0	100.0	99.0	2.0	99.2	5	
MF	0	100.0	100.0	-	-	-	100.0	100.0	3.0	100.0	99.9	5.6	100.0	5	
PF	0	100.0	100.0	-	-	-	100.0	100.0	5.0	100.0	100.0	4.4	99.2	5	
PA66	2	100.0	100.0	5.0	12.0	1.0	100.0	99.9	10.8	97.7	97.7	1.4	99.6	7	
PA6	0	100.0	100.0	-	-	-	100.0	99.9	10.0	99.6	99.3	7.6	98.8	7	
CEL	0	100.0	100.0	-	-	-	100.0	100.0	5.0	100.0	99.1	2.0	100.0	5	
平均	-	100.0	99.9	-	-	-	99.7	99.6	9.4	98.5	96.8	4.7	99.1	8.6	

この結果、TS-SEPT と SEPT の正解率は、すべての種類のポリマーにおいて SESAT を上回り、ポリマーの種類による正解率の変動は SESAT よりも少なかった。したがって、提案手法の sprig の構造、および決定木構成子の適応度が、ポリマーの種類に依らず高い正解率を得ることに寄与していると考えられる。

TS-SEPT では、PS 以外の 17 種類のポリマーについてすべての決定木が正解率 100% となり、PS でも 10 回の試行で作られた決定木のうち、7 つの決定木で正解率 100% が得られた。この結果は、2 段階判別法の正解率向上に対する有効性を示唆するものといえる。

C5.0 の結果と比較すると、SEPT と TS-SEPT で生成した決定木はノード数がやや多いものの、ほとんどのポリマーに関して正解率が C5.0 を上回った。学習時間の面では、TS-SEPT が 15~300 秒を要するのに対し、C5.0 は同一環境で数秒のうちに処理を終える。しかし、TS-SEPT の学習時間や木のノード数は、リサイクルのためのポリマー判別という利用目的で問題になる程度ではなく、高正解率の決定木を重視する場合には TS-SEPT が有効といえる。

### 5.4.3 生成された決定木

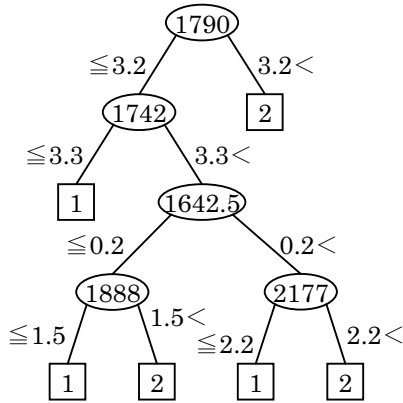
各システムによって生成されたポリ塩化ビニルを判別するための決定木を図 5.7(a)~(d) に示す。ただし、分岐の閾値は小数第 2 位もしくは第 3 位を四捨五入した値を記している。判別の正解率は TS-SEPT で 100%、SEPT で 98.8%、SESAT で 96.5%、C5.0 で 99.2% となっている。

TS-SEPT では、 $T_B$  が終端ノードのみからなる木となった。SESAT では、属性 2325、閾値 5.9、左子ノードはクラス 1 の終端ノード、右子ノードはクラス 2 の終端ノードという決定木が生成された。SEPT と C5.0 により生成された決定木には、SESAT の決定木と非常に類似した木が部分木として含まれている。(b) の属性 2327、閾値 6.1 の非終端ノード、および (d) の属性 2321.5、閾値 5.5 の非終端ノードを根ノードとする部分木である。また、(a) の  $T_A$  では、属性が 2325.5、閾値が 5.9 で、右子ノードがクラス 2 の終端ノードという非終端ノードが根ノードとなっている。SESAT では非常に大雑把な特徴を捉えて簡素な決定木が生成されたが、その他のシステムでは、SESAT で表現された特徴に加えてさらなる特徴をも盛り込んだ決定木が生成されているといえる。

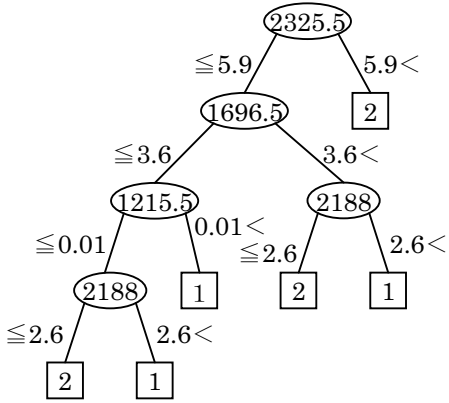
(a) の  $T_A$ 、(b)、(d) の最左部分木は、それぞれ属性が 2188、2157.5、2190.5、閾値が 2.6、2.5、2.9 と非常に近い値で、左子ノードがクラス 2 の終端ノード、右子ノードがクラス 1 の終端ノードとなっており、(c) 以外に加えられた特徴にも共通点が見られる。さらに、(a) の  $T_A$  と (d) では、最左部分木の 1 つ上のレベルの非終端ノードにおいて、属性が 1215.5、1214.5、閾値が 0.01、0.01 と類似している。

以上より、各システムにおいてポリ塩化ビニルの特徴を捉えられているが、TS-SEPT では、例外的なポリ塩化ビニルの試料を分けることで正解率 100% の決定木が生成できたといえる。

【 $T_{all}$ 】



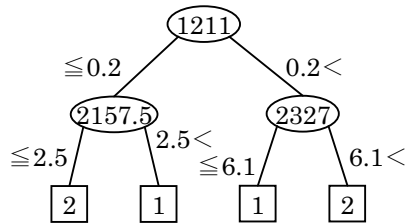
【 $T_A$ 】



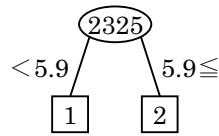
【 $T_B$ 】



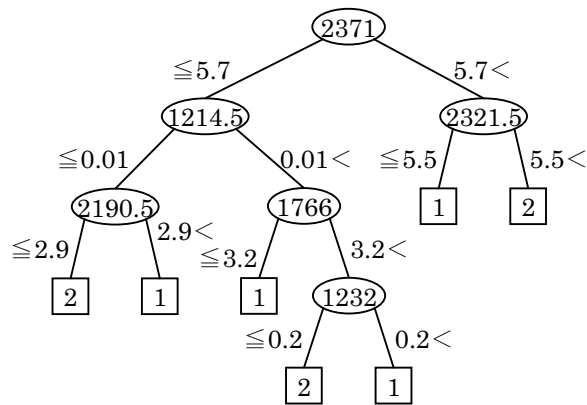
(a) TS-SEPT



(b) SEPT



(c) SESAT



(d) C5.0

図 5.7: ポリ塩化ビニルを判別するための決定木

#### 5.4.4 2 段階判別法の効果

ABS, PVC, PC/ABS, PET の全事例判別において, SEPT では 100%を達成できなかったが, TS-SEPT では全試行で 2 段階判別用の決定木  $T_{all}$  の生成により 100%の正解率が得られた. このことから, 2 段階判別法は特にこの 4 種のポリマーの判別に有効であったといえる.

決定木生成に C5.0 を用いた場合の 2 段階判別法の効果を調べるため, 上記 4 種類の試料について 5.3.4 節に示した手順に従って決定木生成を C5.0 で行ない,  $T_{all}$ ,  $T_A$ ,  $T_B$  の生成を試みたところ, 表 5.4 に示すような結果が得られた.

ABS と PC/ABS では正解率が 1%程度向上したが, PVC と PET では  $T_{all}$  がクラス 1 を表す根ノードのみからなる決定木となり,  $T_A$  は表 5.3 と同様の決定木となった. 2 段階判別法の効果は適切な  $T_{all}$  により発揮されるものであり,  $T_{all}$  の学習に有用な B グループ候補が必要となる. C5.0 の PVC と PET では 1 つの決定木で 99%以上の正解率が得られたため,  $T_{all}$  の学習に十分な B グループ候補が抽出できなかったと考えられる. 一方 TS-SEPT では, SEPT で 99%以上の正解率が得られたポリマーに関しても, 乱数処理を含む繰返しによって B グループ候補が作られ, 100%の正解率を得ている. したがって, 2 段階判別法では, 1 つの決定木による分別が困難なデータに対して複数の正解率の高い解を生成できる手法との組合せが効果的といえる.

表 5.4: C5.0 における 2 段階判別の正解率とノード数

データ	正解率 [%]	ノード数		
		$T_{all}$	$T_A$	$T_B$
ABS	98.8	9	15	3
PVC	99.2	1	13	-
PC/ABS	99.2	9	5	3
PET	99.6	1	9	-

#### 5.4.5 未知事例判別

未知事例判別の正解率を調査するため, 試料数が 30 以上の 3 種類のポリマーについて, 試料の 10 分の 9 を訓練事例, 10 分の 1 をテスト事例とする 10-fold クロスバリデーションを行なった. 各訓練事例に対して 10 回ずつ決定木を生成したときの平均ノード数と, テスト事例の平均正解率を表 5.5 に示す. 正解率欄の括弧内の数字は, 各テスト事例での平均正解率の標準偏差を表している.

この結果, どのポリマーにおいても TS-SEPT, SEPT, SESAT の順に高い正解率が得られ, 提案手法は未知事例判別においても有効であることが示された. TS-SEPT では, ポリマーの種類による正解率の変動, およびテスト事例による正解率の変動も他システムより少なく, ポリマーの種類や事例に依存しない安定した正解率を得るために有効であることが確認された.

各データの特徴を調べるため, 全事例を特定の種類のポリマーとそれ以外の 2 クラスに分け, 同ク

表 5.5: 未知事例判別における平均正解率と平均ノード数

データ	TS-SEPT				SEPT		SESAT		C5.0	
	正解率 [%]	ノード数			正解率 [%]	ノード数	正解率 [%]	ノード数	正解率 [%]	ノード数
		$T_{all}$	$T_A$	$T_B$						
PS	96.3 (1.8)	10.0	7.9	1.7	95.4 (2.0)	10.2	94.1 (2.4)	7.3	95.8 (2.8)	11.0
PE	96.4 (2.1)	4.4	12.1	1.1	96.0 (2.0)	12.3	94.7 (3.7)	10.8	98.4 (2.7)	9.8
PP	97.3 (2.5)	5.0	9.8	1.1	96.9 (2.9)	11.2	93.0 (2.4)	7.4	92.7 (5.8)	12.0

(括弧内は標準偏差)

クラスに属する事例のすべての組合せ，および異クラスに属する事例のすべての組合せに関して，2つのデータの内積を算出した．PS，PE，PPの同クラスにおける内積の分散はそれぞれ0.082，0.091，0.081，異クラスでは0.042，0.067，0.057となり，両者においてPEは他よりも内積の分散が大きかった．共生進化に基づくシステムではPEの正解率がC5.0よりも低いことを考慮すると，多様な事例を含む種類の判別が提案手法の弱点であると予想される．

## 5.5 まとめ

本章では，ポリマーを正確に判別するための共生進化に基づく決定木生成手法を提案した．近赤外スペクトルデータの特徴を踏まえた遺伝子表現，クラス間の事例数の偏りを考慮した適応度関数，代表的な事例とは異なる少数派の事例の特徴をも学習するための2段階判別法により，詳細な判別ルールの獲得が可能となった．分割したデータに特化したルールを生成する手法は，探索的データ解析やアンサンブル学習などの分野で提案されてきたが，本手法ではどのルールを適用するかを判別するためのルール $T_{all}$ が存在する点で既存研究と特長を異にする．

実データを用いた実験により，両手法が正確なポリマー判別に有効であることを確認した．特に，これまで100%の正解率が得られていなかった全事例判別において，最高正解率100%を得られたことが最大の成果といえる．本研究で得られた決定木は，プラスチックのリサイクル処理において実用化が可能と考えられ，資源有効活用の一助となることが期待される．

## 第6章 考察

共生進化に基づく帰納論理プログラミングと決定木生成に関する研究をふまえ、共生進化と組み合わせるアンサンブル学習と2段階判別法、および部分解の遺伝子表現を設計するにあたっての留意点を中心に考察する。

### 6.1 問題の難易度との関連性

第3章～第5章で扱った帰納論理プログラミングと決定木生成のほか、ガウス性雑音重畳画像のエッジ強調を目的とした多数決機械の学習に共生進化を適用し、既存手法よりも良好な結果が得られている [Otani 06]。多数決機械とは、奇数個の線形識別機械を組み合わせたパターン識別器であり、各線形識別機械の判別結果を統合することで非線形パターンの識別を目指す。線形識別機械を部分解、線形識別機械の組合せである多数決機械を全体解とする共生進化により、エッジ部分を検出する多数決機械を学習する。エッジ部分を正しく検出することで、雑音を強調することなくエッジのみを強調することができる。

第2章～第5章で扱った学習対象と多数決機械、および Moriarty らが学習対象としたニューラルネットワークにおける共生進化は、2.4.2 節で示した設計上の難易度による分類基準に従うと、表 6.1 のように分類できる。

表 6.1: 共生進化を適用した問題の分類

	カテゴリ W1	カテゴリ W2	カテゴリ W3
カテゴリ P1	OneMax 問題 Harik の部分だまし問題	Floyd 問題	-
カテゴリ P2	ニューラルネットワーク 多数決機械	帰納論理プログラミング	決定木

表 6.1 のカテゴリ P1-W1, P1-W2, P2-W1 に示した問題は、解構造が比較的単純であり、解探索の目標を適応度関数として明確に表現できるため、共生進化による解探索のみで良好な解を得た。しかし、第3章～第5章で取り上げた帰納論理プログラミングの予測精度向上、簡素な決定木の生成、および多属性データを扱うための決定木の生成という問題では、解構造の複雑さと制約条件の厳しさによる影響が大きいいため、単純な設計の共生進化による解探索のみでは、目標とするような解の獲得は困難であった。

より高い精度を得るため、帰納論理プログラミングシステム ILP/SE では、共生進化に基づく解探索に加えて予測精度向上のために多数決によるアンサンブル学習を行なった。また、決定木生成システム SESAT では予測正解率と木の簡素さの両者を考慮するための適応度関数を定義し、ポリマー判別システム TS-SEPT では2段階判別法を取り入れることで、よりよい解を導出することができた。第3章と第5章では、アンサンブル学習や2段階判別法を既存手法と組み合わせたシステムでの結果をも示しているが、いずれも ILP/SE や TS-SEPT ほどの効果は得られていない。共生進化による解探索を補完するための手法であるが、共生進化による解探索と組み合わせてこそ効果を発揮するものと考えられる。

## 6.2 部分解の設計条件

一般に、遺伝的アルゴリズムに基づく処理結果は、個体の遺伝子表現や進化戦略、適応度関数などの設計に大きく依存するため、より高い精度を得るためには、対象とする問題や処理の目的に応じた設計を考案しなければならない。なかでも個体の遺伝子表現は重要であり、本論文で扱った帰納論理プログラミングと決定木生成においては、解構造が比較的複雑であるため、遺伝子表現に関するさまざまな検討を行なった。SESAT と TS-SEPT は両者とも決定木生成システムであるが、異なる目標を達成するために検討を繰り返した結果、部分解の遺伝子表現は異なったものとなった。

各システムにおける部分解の遺伝子表現を検討した結果、共生進化の特徴でもある部分解の遺伝子表現の設計にあたっては、次の2点に留意すべきであるとの考えに至った。

部分解条件1 意味づけの明確な構造を部分解とする。

部分解条件2 部分解の単純な組合せにより全体解を表現する。

以下では、2つの部分解条件について考察する。

### 6.2.1 部分解の意味づけ

Harik の部分だまし問題のように、適応度が解の一部分の評価値に基づいて決定される場合には、適応度決定のもととなる当該部分を部分解とするのは当然である。また、リテラルの組合せからなる Horn 節や、線形識別機械の組合せからなる多数決機械、隠れ層のニューロンの組合せからなるニューラルネットワークのように、求めるべき解を構成する基本単位が明らかな場合には、基本単位を部分解とするのが最も妥当であるといえよう。しかし、決定木のように解の分割方法が明確でない場合には、“意味づけの明確な構造”を部分解として共生進化による学習を行なうと良いと考えられる。

決定木を生成するための共生進化において、オスカルら [オスカル 97] と中山ら [中山 00] は、1本のアークとそれに連結する2つのノードの組を部分解として定義し、それぞれ Link, chrom と呼んでいる。一方、本論文における決定木生成システム SESAT および TS-SEPT では、高さ1の部分木を部分解 sprig として学習を行なう。Link と chrom は1つの属性に関する1つの分岐を表すのに対

し, sprig は 1 つの属性に関するすべての分岐を表す. Link, chrom の例を図 6.1, sprig の例を図 6.2 に示す. 図 6.1(a) は「属性  $A_p$  の値が  $Th$  前後<sup>1</sup>の事例を選び分けた後, 属性  $A_c$  でさらに選び分ける」ことを意味しており, 図 6.1(b) は「属性  $A_p$  の値が  $Th$  前後の事例をクラス  $C$  と判定する」ことを意味する. また, 図 6.2 の sprig は「属性  $A$  の値が  $Th$  未満の事例はさらに他の属性で分類し,  $Th$  以上の事例はクラス  $C$  と判定する」という意味を持つ.

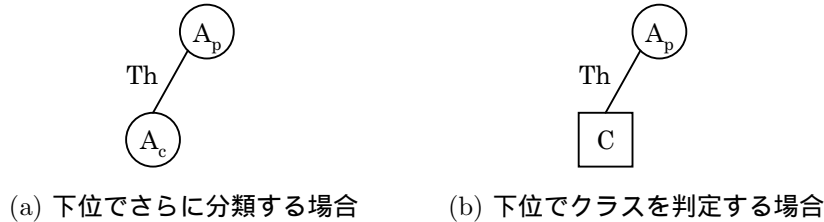


図 6.1: Link, chrom の例

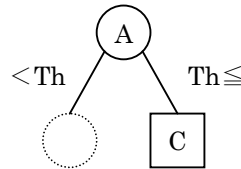


図 6.2: sprig の例

Link と chrom では, 2 つの属性に関して分類の基準とする順序を表現できるが, 単独で特定の属性に関する複数の分岐を表すことはできず, 他の個体との連結の仕方に依存する部分が多い. したがって, sprig の方が単体での意味づけが明確であるといえよう. 両者の学習結果を比較すると, sprig を部分解とする場合の方が正解率の高い簡素な決定木を得ている. 前掲したりテラルや線形識別機械なども当然ながら意味づけは明確であり, 部分解とすべき構造を決定する上で重要な指標といえる.

共生進化は, 2.4.1 節で述べた通り, 複数の人が協働する状況をモデル化したものである. グループとして目的を達成するためには, 各人の働きや役割分担が明確でなければ, 効果的なグループ編成を考えることは難しい. この点からも, 部分解個体の意味づけは重要であるといえる. 以上より, 意味づけの明確な構造を部分解とすることで, 共生進化の効果を最大限に引き出すことができると考える.

## 6.2.2 部分解個体の組合せの学習

付録 A の CY における sprig や, 前述の Link, chrom では, 下位に接続される非終端ノードの種類を指定しているため, 接続可能な部分解個体が制限される. 組合せの制限された部分解の学習で

<sup>1</sup>chrom は sprig と同様, アークに振られたラベルの平均値を分岐の閾値とするが, Link では属性値の上限と下限を遺伝子内に保持する.



は、部分解個体の組合せに関する学習の一部を部分解集団で行なうことになる。一方、付録 A の CN における sprig, すなわち SESAT と TS-SEPT の sprig では、下位に接続される非終端ノードの種類を指定しておらず、他のいずれの部分解個体とも接続できる構造となっている。部分解の組合せに制限がない場合は、部分解個体の組合せに関してはすべて全体解集団で学習することになる。

付録 A における実験の結果、CY より CN の方が正解率と簡素さの両面で優れている決定木を生成することが示された。また、カテゴリ P2-W3 以外の各問題において、部分解個体の組合せに関する学習をすべて全体解集団で行なって良好な結果を得ていることから、部分解の設計にあたっては、部分解個体の組合せに関して部分解集団で学習しないような表現を考案すべきと考える。

生物モデルの観点から見ると、組合せを制限された部分解は「この人と一緒じゃないとイヤ！」と自ら主張するわがままなメンバーに例えられるだろう。同じグループに属するメンバーを限定されると、候補として考えられるグループ編成も限られ、候補以外のグループを編成するためには、わがままなメンバーの意識改革を行なわなければならない。誰とでも素直に組めるメンバーの方が、より柔軟にグループ編成を考えることができ、結果的に強いグループを作り上げられると思われる。したがって、生物モデルの観点からも、部分解条件 2 は部分解の設計において重要であると考えられる。

## 第7章 結論

共生進化の応用として行なった帰納論理プログラミングと決定木生成に関する研究をまとめ、共生進化の展望について述べる。

### 7.1 研究のまとめ

本論文では、帰納学習における解探索手法として、遺伝的アルゴリズムの一手法である共生進化に注目し、共生進化に基づく帰納論理プログラミングおよび決定木生成の手法を提案した。各手法を実装した3つのシステムにおいて学習を行ない、既存手法の学習結果と比較したところ、いずれにおいても提案手法で得られた解の方が目的に適合することが示された。本論文で示した評価実験の結果のみから、共生進化が帰納学習全般における探索手法として有効であると断言することはできないものの、Horn節や決定木という構造の複雑な解を持つ問題を対象として良好な結果が得られていることから、多様な問題への適用可能性は十分に高いと考える。

遺伝的アルゴリズムを適用する際には、学習時間が問題となる。世代交代を繰り返すことで最適解を探索していくため、他の手法と比較して学習に長い時間を要するのが一般的であり、リアルタイムに解を出力することが求められないような問題にのみ適用される。本研究における評価実験では、一般的な遺伝的アルゴリズムよりも共生進化が解探索手法として有効であるという結果が得られたことから、既に遺伝的アルゴリズムが適用されている問題に共生進化を適用することで、さらに目的に適合する解が得られるものと考えられる。

また、遺伝的アルゴリズムは、個体の遺伝子表現や進化戦略、適応度関数などの設計を、対象問題に応じて適切に行なわなければならないという問題も抱えている。共生進化でも、この問題点は引き継いでいるものの、本研究を通して以下のような解表現の設計指針が得られた。

- 意味づけの明確な構造を部分解とする。
- 部分解の単純な組合せにより全体解を表現する。

この設計指針に従うことによって、以前に共生進化が適用されたことのない問題においても、共生進化を容易に適用することができる。

多属性データに適した決定木の生成手法に関しては、ベンチマークデータではなく、実データにより有効性を検証した。実データを用いたシステムの検証はしばしば望まれるところであり、ポリマー判別システム TS-SEPT が実データで他システムよりも良好な解を導出したことで、提案手法の有効

性が明確に示されたものとする。また、本研究は“正解率の高いポリマーの判別規則を決定木により表現したい”という化学者からの依頼を受けて始めたものであるが、最終成果として得られた決定木により、化学者の要望に応えることができた点も大きな成果である。

## 7.2 共生進化の展望

共生進化は統治分割法に基づいていることから、決定木よりもさらに複雑な構造を対象とする問題に適用することで、大きな効果が得られると考えられる。決定木よりも複雑な構造には、例えば一般的なグラフ構造や隠れマルコフモデルなどがある。DNA 配列群に含まれる複合モチーフを表現するための隠れマルコフモデルを遺伝的アルゴリズムにより生成する研究がなされている [矢田 99, Won 04, Won 05]。DNA 配列には、タンパク質が DNA に結合するために必要な配列断片である要素モチーフが複数存在する。複数の要素モチーフの並びと出現位置のパターンが複合モチーフである。すなわち、複合モチーフは要素モチーフの組合せであり、複合モチーフを抽出する隠れマルコフモデルは、要素モチーフを抽出する隠れマルコフモデルの組合せとして表現される。したがって、要素モチーフ用の隠れマルコフモデルを部分解、複合モチーフ用の隠れマルコフモデルを全体解として、共生進化を適用することができる。

また、決定木よりも複雑な構造としては、複数の 2 クラス分別用決定木を順に使用して事例のクラスを判定するような決定木列も挙げられる。決定木列の学習では、個々の決定木生成に加え、複数の決定木を並べる順序にも共生進化が適用できるだろう。第 5 章で説明したポリマー判別システム TS-SEPT では、特定の種類のポリマーとそれ以外を分別する決定木を生成したが、各ポリマーの種類を判別するためには、すべての決定木を使用して 1 つずつ順に種類を特定していくことになる。このとき、種類を特定する順序によって精度が変化する可能性がある。また、順序を適切に設定すると、前の段階で特定された種類の試料は判別対象とならないため、より簡素で正解率の高い決定木が生成できるかもしれない。決定木の生成と並行して決定木を並べる順序をも学習することで、各ポリマーの種類を判別するという目的が達成される。生成される決定木とその順序は互いに影響を及ぼしあうため、両者を並行して学習する必要がある。これは共生進化が適用できる問題の特徴に合致する。したがって、高さ 1 の部分木を部分解、その組合せを全体解とする共生進化により決定木を生成し、これと並行して決定木を部分解、その並び順を全体解とする共生進化により決定木列を生成するというように、3 つの集団を並行進化させる共生進化が考えられる。

さらに、共生進化は複数人が協働する状況をモデル化したものであり、6.2 節で述べたように部分解の設計には部分解の意味づけが重要であることから、部分解とすべき基本単位が明らかで、部分解同士がより“協働”すべき状況に適用することで、共生進化の効果がより明確に現れるものとする。例として、RoboCup[Rob] のシミュレーションリーグにおけるコーチエージェントの采配決定に適用することが挙げられる。各選手を部分解、チームを全体解として共生進化を行ない、各選手への指示とチーム編成を決定することで、効果的な采配の実現が期待される。

ポリマー判別システム TS-SEPT では、正解率 100%の判別を実現するために 2 段階判別法を導入した。共生進化に基づく決定木生成と組み合わせることで、2 段階判別法の効力は発揮されることが評価実験により示された。しかし、“互いに異なる特徴を持つ事例をあらかじめ分けておき、別の基準でクラスを判別する”という 2 段階判別法は、決定木以外にも適用可能である。例えば、6.1 節で挙げた多数決機械への適用が考えられる。多数決機械では、複数の線形識別機械の相互作用により、非線形パターンを識別する。しかし、エッジ検出のための入力パターンは多様であり、判別が極めて困難なパターンも数多く存在する。特徴の異なる入力パターンを分別するための多数決機械を導入し、2 段階判別法によりクラスを判別することで、より正確なエッジ検出が期待される。

昨今では、さまざまな処理がコンピュータを用いて行なわれており、インターネットとパーソナルコンピュータの爆発的な普及もあいまって、多種多様なデータが電子化された状態で保存されている。これらのデータの中には、帰納学習の対象となる事例を含むものも多く存在する。膨大な事例から有用な概念や規則が獲得される可能性も十分に高いため、コンピュータで行なう帰納学習に対して、新たな需要が発生することも大いに予想される。このような状況下で、帰納学習における解探索手法の提案を主題とした本論文における成果が、今後の帰納学習の研究および実践の一助となることを期待する。

## 謝 辞

本論文をまとめるにあたり，終始あたたかいご指導とご助言をいただきました東京大学大学院情報理工学系研究科の石塚満教授に深く感謝の意を表します．また，東京大学大学院新領域創成科学研究科の伊庭齊志教授には，研究方法と成果に関する貴重な示唆と詳細なご助言をいただきました．ここにお礼を申し上げます．さらに，審査過程において有益なご助言をいただきました東京大学大学院新領域創成科学研究科の近山隆教授，東京大学大学院情報理工学系研究科の坂井修一教授，田中久美子助教授に深く感謝いたします．

本研究を通して数多くのご指導とご助言をいただきました大阪大学産業科学研究所の沼尾正行教授，東京工業大学大学院情報理工学研究科の佐藤泰介教授，東京工業大学大学院総合理工学研究科の新田克己教授，研究に際してさまざまな形でご協力をいただきました東京理科大学理工学部の大和田勇人教授，一関工業高等専門学校物質化学工学科の貝原巳樹雄教授に深く感謝の意を表します．

武蔵工業大学環境情報学部の高田達雄元学部長，および増井忠幸学部長には，学位取得にあたって数々の励ましの言葉をいただきました．ここに感謝の意を表します．

筆者が東京工業大学の学生であった頃から現在まで，研究はもちろん，研究以外の面でも親身にご指導いただきました東京工業大学名誉教授，武蔵工業大学環境情報学部教授の志村正道先生に心より感謝いたします．

最後に，日頃から精神的な支えとなり，陰ながら応援してくれた両親に感謝いたします．

## 参考文献

- [Ale] <http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>
- [Angeline 98] Angeline, P. : Subtree crossover causes bloat, in *Proceedings of the 3rd Genetic Programming Conference*, pp. 745–752 (1998)
- [Anglano 98] Anglano, C. , Giordana, A. , Bello, G.L. , and Saitta, L. : An Experimental Evaluation of Coevolutionary Concept Learning, in *Proceedings of the 15th International Conference on Machine Learning*, pp. 19–27 (1998)
- [Augier 95] Augier, S. , Venturini, G. , and Kodratoff, Y. : Learning first order logic rules with a genetic algorithm, in *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, pp. 21–26 (1995)
- [Braud 01] Braud, A. and Vrain, C. : A Genetic Algorithm for Propositionalization, in *Proceedings of the 11th International Conference on Inductive Logic Programming, Lecture Notes in Artificial Intelligence, No.2157*, pp. 27–40 (2001)
- [Breiman 84] Breiman, L. , Friedman, J.H. , Olshen, R.A. , and Stone, C.J. : *Classification and Regression Trees*, Wadsworth & Brooks (1984)
- [Breiman 96] Breiman, L. : Bagging Predictors, *Machine Learning*, Vol. 24, No. 2, pp. 123–140 (1996)
- [C50] <http://rulequest.com/>
- [Carpenter 87a] Carpenter, G.A. and Grossberg, S. : ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns, *Applied Optics*, Vol. 26, No. 23, pp. 4919–4930 (1987)
- [Carpenter 87b] Carpenter, G.A. and Grossberg, S. : A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine, *Computer Vision, Graphics, and Image Processing*, Vol. 37, pp. 54–115 (1987)
- [Carpenter 90] Carpenter, G.A. and Grossberg, S. : ART-3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures, *Neural Networks*, Vol. 3, pp. 129–152 (1990)

- [Debnath 91] Debnath, A.K. , Compadre, Lopez de R.L. , Debnath, G. , Schusterman, A.J. , and Hansch, C. : Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro compounds. Correlation with molecular orbital energies and hydrophobicity, *Journal of Medicinal Chemistry*, Vol. 34, No. 2, pp. 786–797 (1991)
- [Dorigo 96] Dorigo, M. , Maniezzo, V. , and Colorni, A. : The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 26, No. 1, pp. 29–41 (1996)
- [Freund 96] Freund, Y. and Shapire, R.E. : Experiments with a New Boosting Algorithm, in *Proceedings of the 13th International Conference on Machine Learning*, pp. 148–156 (1996)
- [古川 97] 古川 康一 : 帰納論理プログラミング - チュートリアル - , 人工知能学会誌, Vol. 12, No. 5, pp. 655–664 (1997)
- [Gentner 93] Gentner, D. , Rattermann, M.J. , and Forbus, K.D. : The Roles of Similarity in Transfer: Separating Retrievability from Inferential Soundness, *Cognitive Psychology*, Vol. 25, No. 4, pp. 524–575 (1993)
- [Giordana 96] Giordana, A. , Neri, F. , and Saitta, L. : Search-Intensive Concept Induction, *Evolutionary Computation*, Vol. 3, No. 4, pp. 375–416 (1996)
- [Goldberg 89] Goldberg, D.E : *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989)
- [Grossberg 76a] Grossberg, S. : Adaptive Pattern Classification and Universal Recoding: I. Parallel Development and Coding of Neural Feature Detectors, *Biological Cybernetics*, Vol. 23, No. 3, pp. 121–134 (1976)
- [Grossberg 76b] Grossberg, S. : Adaptive Pattern Classification and Universal Recoding: II. Feedback, Expectation, Olfaction, Illusions, *Biological Cybernetics*, Vol. 23, No. 4, pp. 187–202 (1976)
- [Harik 99] Harik, G. : Linkage Learning via Probabilistic Modeling in the ECGA, IlliGAL Report 99010, Illinois Genetic Algorithms Laboratory, University of Illinois (1999)
- [Hillis 92] Hillis, W.D. : Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure, in *Artificial Life II: Proceedings of the Workshop on Artificial Life*, pp. 313–324 (1992)
- [Hoche 01] Hoche, S. and Wrobel, S. : Relational Learning Using Constrained Confidence-Rated Boosting, in *Proceedings of the 11th International Conference on Inductive Logic Programming, Lecture Notes in Artificial Intelligence, No.2157*, pp. 51–64 (2001)

- [Holland 75] Holland, J.H. : *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975)
- [Holyoak 89] Holyoak, K.J. and Thagard, P. : Analogical Mapping by Constraint Satisfaction, *Cognitive Science*, Vol. 13, No. 3, pp. 295–355 (1989)
- [Horn 94] Horn, J. , Goldberg, D.E. , and Deb, K. : Implicit Niching in a Learning Classifier System: Nature’s Way, *Evolutionary Computation*, Vol. 2, No. 1, pp. 37–66 (1994)
- [Hunt 96] Hunt, J.E. and Cooke, D.E. : Learning Using an Artificial Immune System, *Journal of Network and Computer Applications*, Vol. 19, pp. 189–212 (1996)
- [Iba 94] Iba, H. , Garis, H. , and Sato, T. : Genetic Programming Using a Minimum Description Length Principle, in Kinnear, K.E. ed., *Advances in Genetic Programming*, Chapter 12, pp. 265–284, MIT Press (1994)
- [Iba 97] Iba, H. , Nozoe, T. , and Ueda, K. : Evolving Communicating Agents on Genetic Programming, in *Proceedings of the 4th IEEE International Conference on Evolutionary Computation*, pp. 297–302 (1997)
- [Jamei 04] Jamei, M. , Mahfouf, M , and Linkens, D.A. : Elicitation and Fine-tuning of Fuzzy Control Rules Using Symbiotic Evolution, *Fuzzy Sets and Systems*, Vol. 147, pp. 57–74 (2004)
- [人工 05] 人工知能学会 (編): 人工知能学辞典, 第 5 章, pp. 270–272, 共立出版 (2005)
- [Juang 00] Juang, C.F. , Lin, J.Y. , and Lin, C.T. : Genetic Reinforcement Learning through Symbiotic Evolution for Fuzzy Controller Design, *IEEE Transactions on Systems, man, and cybernetics-Part B: Cybernetics*, Vol. 30, No. 2, pp. 290–302 (2000)
- [貝原 03] 貝原 巳樹雄, 樋口 陽子 : 決定木を用いたポリマーの判別, 日本化学会第 26 回情報化学討論会要旨集, J08 (2003)
- [Knuth 96] Knuth, D.E. : Are Toy Problems Useful?, in *Selected Papers on Computer Science*, CSLI Lecture Notes Number 59, Chapter 10, pp. 169–183, CSLI Publications (1996)
- [Koza 92] Koza, J.R. : *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press (1992)
- [ド・ラ・メトリ 32] ド・ラ・メトリ (杉捷夫訳) : 人間機械論, 岩波書店 (1932)
- [Mahfouf 01] Mahfouf, M , Jamei, M. , and Linkens, D.A. : Rule-Base Generation via Symbiotic Evolution for a Mamdani-Type Fuzzy Control System, in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 396–399 (2001)



- [松本 99] 松本 高利, 田辺 和俊, 佐伯 和光, 天野 敏男, 上坂 博亨: 近赤外分光測定とニューラルネットワーク解析を組み合わせたプラスチック廃棄物の非破壊判別, *分析化学*, Vol. 48, No. 5, pp. 483–489 (1999)
- [McCulloch 43] McCulloch, W. and Pitts, W. : A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133 (1943)
- [Mizoguchi 95] Mizoguchi, F. and Ohwada, H. : Constrained Relative Least General Generalization for Inducing Constraint Logic Programs, *New Generation Computing*, Vol. 13, pp. 335–368 (1995)
- [Moriarty 95] Moriarty, D.E. and Miikkulainen, R. : Efficient Learning from Delayed Rewards through Symbiotic Evolution, in *Proceedings of the 12th International Conference on Machine Learning*, pp. 396–404 (1995)
- [Moriarty 96] Moriarty, D.E. and Miikkulainen, R. : Efficient Reinforcement Learning through Symbiotic Evolution, *Machine Learning*, Vol. 22, pp. 11–32 (1996)
- [Moriarty 98] Moriarty, D.E. and Miikkulainen, R. : Hierarchical Evolution of Neural Networks, in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 428–433 (1998)
- [Muggleton 90] Muggleton, S. and Feng, C. : Efficient Induction of Logic Programs, in *Proceedings of the 1st Conference on Algorithmic Learning Theory*, pp. 368–381 (1990)
- [Muggleton 95] Muggleton, S. : Inverse Entailment and Progol, *New Generation Computing*, Vol. 13, pp. 245–286 (1995)
- [中山 00] 中山 泰夫, 大谷 紀子, 志村 正道: 共生進化に基づく決定木の最適化に関する研究, 第 14 回人工知能学会全国大会論文集, pp. 327–328 (2000)
- [Newell 63] Newell, A. and Simon, H.A. : GPS: A Program that Simulates Human Thought, in Feigenbaum, E.A. and Feldman, J. eds., *Computers and Thought: a Collection of Articles*, McGraw Hill (1963)
- [小野寺 99] 小野寺 光永, 長嶋 雲兵, 加藤 澄恵, 細矢 治夫, 後藤 成志, 天野 敏男, 田辺 和俊, 上坂 博亨: スペクトル揺らぎを利用したプラスチックの近赤外反射スペクトルの特徴抽出, *化学ソフトウェア学会論文誌*, Vol. 5, No. 3, pp. 93–102 (1999)
- [オスカル 97] オスカル オルテガ, 沼尾 正行, 志村 正道: 共存進化における決定木の学習に関する研究, 第 11 回人工知能学会全国大会論文集, pp. 171–174 (1997)
- [大谷 95] 大谷 紀子: 算数における抽象概念の獲得過程, 修士論文, 東京工業大学 (1995)

- [Otani 06] Otani, N. , Kimura, T. , and Nitta, K. : Image Enhancement by Committee Machine with Symbiotic Evolution, 第19回回路とシステム軽井沢ワークショップ論文集, pp. 171–176 (2006)
- [Paredis 96] Paredis, J. : Coevolutionary Computation, *Artificial Life*, Vol. 2, No. 4, pp. 355–375 (1996)
- [Potter 94] Potter, M.A. and De Jong, K.A. : A Cooperative Coevolutionary Approach to Function Optimization, in *Proceedings of the 3rd Parallel Problem Solving From Nature*, pp. 249–257 (1994)
- [Puppala 98] Puppala, N , Sen, S. , and Gordin, M : Shared Memory based Cooperative Coevolution, in *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, pp. 570–574 (1998)
- [Quinlan 86] Quinlan, J.R. : Induction of Decision Trees, *Machine Learning*, Vol. 1, No. 1, pp. 139–159 (1986)
- [Quinlan 89] Quinlan, J.R. and Rivest, R.L. : Inferring Decision Trees Using the Minimum Descriptin Length Principle, *Information and Computation*, Vol. 80, pp. 227–248 (1989)
- [Quinlan 93] Quinlan, J.R. : *C4.5: Programs for Machine Learning*, Morgan Kaufmann (1993)
- [Quinlan 95] Quinlan, J.R. and Camereon-Jones, R.M. : Induction of Logic Programs: FOIL and Related Systems, *New Generation Computing*, Vol. 13, pp. 287–312 (1995)
- [Quinlan 96] Quinlan, J.R. : Boosting first-order learning, in *Proceedings of the 7th International Workshop on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence, No.1160*, pp. 143–155 (1996)
- [Rob] <http://www.robocup.org/>
- [Rosenbloom 82] Rosenbloom, P.S. and Newell, A. : Learning by Chunking: Summary of a Task and a Model, in *Proceedings of National Conference on Artificial Intelligence. American Association for Artificial Intelligence*, pp. 255–257 (1982)
- [斎野 88] 斎野 亨 : 勃興する“ニューロ・エンジニアリング”, ニュ - ロコンピュ - タ最前線 : 応用システムから L S I チップまで, 日経 B P 社 (1988)
- [佐藤 97] 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, Vol. 12, No. 5, pp. 734–744 (1997)
- [Sebag 00] Sebag, M. and Rouveirol, C. : Resource-bounded Relational Reasoning: Induction and Deduction through Stochastic Matching, *Machine Learning*, Vol. 38, pp. 41–62 (2000)

- [Sebald 94] Sebald, A.V. and Schlenzig, J. : Minimax Design of Neural Net Controllers for Highly Uncertain Plants, *IEEE Transactions on Neural Networks*, Vol. 5, No. 1, pp. 73–82 (1994)
- [Smith 93] Smith, R.E. , Forrest, S. , and Perelson, A.S. : Searching for Diverse, Cooperative Populations with Genetic Algorithms, *Evolutionary Computation*, Vol. 1, No. 2, pp. 127–149 (1993)
- [Srinivasan 95] Srinivasan, A. , Muggleton, S.H. , and King, R.D. : Comparing the use of background knowledge by two Inductive Logic Programming systems, in *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pp. 199–230 (1995)
- [Tamaddoni-Nezhad 01] Tamaddoni-Nezhad, A. and Muggleton, S. : Using Genetic Algorithms for Learning Clauses in First-Order Logic, in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 639–646 (2001)
- [Thagard 90] Thagard, P. , Holyoak, K.J. , Nelson, G. , and Gochfeld, D. : Analog Retrieval by Constraint Satisfaction, *Artificial Intelligence*, Vol. 46, No. 3, pp. 259–310 (1990)
- [UCI] <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- [Van Rijsbergen 79] Van Rijsbergen, C.J. : *Information Retrieval, second edition*, Butterworths (1979)
- [Weeks 97] Weeks, E.R. and Burgess, J.M. : Evolving Artificial Neural Networks to Control Chaotic Systems, *Physical Review E*, Vol. 56, No. 2, pp. 1531–1540 (1997)
- [Whitehead 96] Whitehead, B.A. and Choate, T.D. : Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction, *IEEE Transactions on Neural Networks*, Vol. 7, No. 4, pp. 869–880 (1996)
- [Wierstra 05] Wierstra, D. , Gomez, F.J. , and Schmidhuber, J. : Modeling Systems with Internal State Using Evolino, in *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 1795–1802 (2005)
- [Winston 80] Winston, P.H. : Learning and Reasoning by Analogy, *Communications of the ACM*, Vol. 23, No. 12, pp. 689–703 (1980)
- [Won 04] Won, K.J. , Prugel-Bennett, A. , and Krogh, A. : Training HMM Structure with Genetic Algorithm for Biological Sequence Analysis, *Bioinformatics*, Vol. 20, No. 18, pp. 3613–3619 (2004)

- [Won 05] Won, K.J. , Hamelryck, T. , Prugel-Bennett, A. , and Krogh, A. : Evolving Hidden Markov Models for Protein Secondary Structure Prediction, in *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 33–40 (2005)
- [矢田 99] 矢田 哲士, 十時 泰, 浅井 潔, 石川 幹人 : DNA 配列の複合モチーフを表現する隠れマルコフモデルの生成, *情報処理学会論文誌*, Vol. 40, No. 2, pp. 750–767 (1999)
- [Zhao 98] Zhao, Q.F. : A General Framework for Cooperative Co-evolutionary Algorithms: A Society Model, in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 57–62 (1998)
- [Zhao 00] Zhao, Q.F. : Cooperative Co-evolution of Pattern Recognition Agents, in *Proceedings of the 4th Annual Meeting of Japan Association for Evolutionary Economics* (2000)

## 付録 A SESAT の部分解に関する検討

SESAT のアルゴリズムを決定するにあたり，部分解の遺伝子表現や集団の構成に関して行なった検討について記す．

### A.1 検討事項

全体解集団と部分解集団で学習対象とする項目は 4.2.1 節に挙げた通りであるが，sprig の遺伝子表現や集団構成を変化させると，両集団の学習対象項目も変化する．ここでは，次の 2 点に着目し，その影響を検討することとする．

検討事項 1 sprig のクラス遺伝子で下位に接続される非終端ノードの属性を指定するか否か．

検討事項 2 1 つの sprig 集団に各属性の sprig を混在させるか，あるいは属性ごとに sprig 集団を用意するか．

検討事項 1 に関して，下位に接続される非終端ノードの属性を指定する場合は，4.2.1 節の項目 3 に関して「どの属性の sprig を接続するか」をも学習対象とするため，全体解集団では「指定された属性の sprig のうち，どれを接続するか」のみを学習する．一方，指定しない場合は属性に関する制約なしに項目 5 を学習することになり，両集団の学習対象項目が異なる．

検討事項 2 に関して，属性ごとに sprig 集団を形成すると，項目 1 が学習対象ではなくなり，各 sprig 集団では当該属性に適した項目 2~4 を学習すればよいことになる．

### A.2 実験

前節に挙げた 2 つの検討事項について，それぞれいずれの設計が適切かを判断するための実験を行なう．

検討事項 1 について，sprig のクラス遺伝子で属性を指定する場合を CY，指定しない場合を CN と呼び，非終端ノードとなるクラスノードにはそれぞれ次の値を入れることにする．

CY :  $-1 \sim -A$

CN : 0

CY では符号を反転した値が属性番号となり，当該属性の sprig が接続されることを表す．

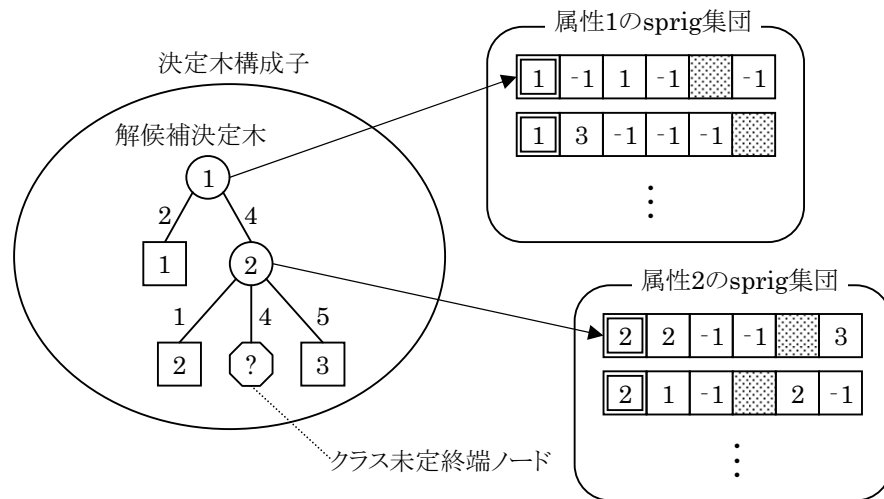


図 A.1: PA の場合の決定木構成子

検討事項 2 について，sprig 集団を 1 つとする場合を P1，属性数分用意する場合を PA とすると，sprig 集団の数は次のようになる．

P1：sprig 集団数 = 1

PA：sprig 集団数 = A

決定木構成子は，P1 の場合に図 4.7，PA の場合に図 A.1 のように図示される．P1 と PA で sprig 集団の個体総数を等しくするために，PA の各 sprig 集団の個体数を  $N_{sp}/A$  個とする．

CY と CN，P1 と PA の各組合せについて，4.4 節と同様の実験を行ない，各集団の学習対象項目の違いによる影響を比較する．決定木構成子の生成手順における Step 2 では，CY と CN，P1 と PA の組合せにより，次のように sprig を選択する．

CN-P1：任意の sprig を選択する．

CY-P1：指定された属性の sprig を選択する．

CN-PA：任意の sprig 集団から，任意の sprig を選択する．

CY-PA：指定された属性の sprig 集団から，任意の sprig を選択する．

### A.3 結果と考察

表 A.1 はテスト事例における平均正解率と生成された決定木の平均ノード数である．平均正解率を全データの平均で比較すると，最大で 0.7% の差が見られた．一方，平均ノード数の全平均は CN-P1，CY-P1，CN-PA，CY-PA の順で小さくなり，CN-P1 のノード数は CY-PA の 81% 程度となった．

学習に要した平均時間の一部と全平均を表 A.2 に示す．各システムは同一のマシン環境において実行されており，同数の sprig と決定木構成子を保持している．この結果，CY-P1 と CY-PA で学習

表 A.1: テスト事例における平均正解率と平均ノード数

データ	CN-P1		CY-P1		CN-PA		CY-PA	
	正解率 [%]	ノード数	正解率 [%]	ノード数	正解率 [%]	ノード数	正解率 [%]	ノード数
australian	84.9 ( 4.1)	16.4 (3.3)	84.7 ( 4.2)	18.2 (3.7)	84.9 ( 4.0)	16.1 (3.6)	84.8 ( 3.6)	18.8 (6.6)
balance-scale	78.7 ( 4.9)	26.5 (5.1)	78.4 ( 5.1)	27.0 (4.6)	78.8 ( 5.1)	28.5 (5.8)	78.6 ( 4.9)	30.9 (5.8)
breast-cancer	95.9 ( 2.3)	16.1 (2.9)	95.8 ( 2.2)	17.4 (3.1)	95.9 ( 2.0)	18.2 (3.3)	95.7 ( 2.1)	18.2 (4.2)
bupa	63.1 ( 8.0)	18.4 (3.8)	63.0 ( 9.2)	19.2 (4.3)	62.9 ( 9.4)	20.9 (4.0)	62.1 ( 8.7)	23.5 (4.5)
glass	63.1 (11.6)	20.0 (3.0)	63.3 (10.5)	21.3 (2.9)	64.1 (11.2)	22.2 (3.5)	62.7 (13.9)	26.4 (4.0)
heart-c	77.4 ( 7.6)	21.6 (3.5)	76.3 ( 8.0)	24.8 (4.3)	77.2 ( 7.6)	25.8 (4.3)	77.3 ( 7.9)	34.3 (6.3)
iris	95.6 ( 5.1)	6.9 (1.7)	94.4 ( 7.3)	7.2 (1.8)	94.6 ( 7.1)	7.4 (2.0)	94.1 ( 7.7)	7.0 (1.8)
monks-1	99.6 ( 2.5)	14.3 (0.8)	100.0 ( 0.0)	14.4 (0.6)	99.5 ( 2.3)	14.2 (1.0)	98.6 ( 4.6)	14.3 (1.5)
monks-2	67.6 (12.8)	21.5 (4.8)	67.1 (12.1)	21.9 (6.2)	65.7 (13.1)	22.4 (5.2)	66.3 (12.7)	25.1 (5.8)
monks-3	92.5 ( 5.2)	7.0 (2.9)	92.0 ( 5.3)	7.4 (3.6)	90.9 ( 5.8)	8.0 (4.1)	91.6 ( 5.2)	7.5 (3.9)
pima-indians	73.3 ( 6.4)	20.3 (4.1)	73.1 ( 6.6)	21.9 (3.7)	72.6 ( 6.7)	23.0 (4.1)	73.1 ( 6.6)	28.8 (5.8)
post-operative	72.3 (17.7)	9.8 (2.3)	72.1 (17.4)	9.9 (2.5)	70.9 (17.3)	10.0 (2.4)	70.6 (16.8)	10.9 (3.7)
全平均	80.3	16.6	80.0	17.5	79.8	18.1	79.6	20.5

(括弧内は標準偏差)

表 A.2: 平均学習時間

データ	CN-P1	CY-P1	CN-PA	CY-PA
australian	15.9	29.7	26.5	29.2
balance-scale	30.9	31.2	31.6	32.5
breast-cancer	15.8	27.7	26.5	27.6
bupa	11.2	19.1	17.7	18.8
glass	17.6	18.1	16.3	18.1
heart-c	19.3	20.6	17.0	21.0
iris	13.6	13.6	11.4	11.1
monks-1	8.9	15.2	13.2	13.4
monks-2	9.3	16.1	14.4	15.0
monks-3	7.7	13.2	11.3	10.9
pima-indians	15.8	29.6	27.8	32.9
post-operative	7.4	13.2	11.3	11.3
全平均	14.5	20.6	18.7	20.2

(単位は秒)

時間が長く、CN-P1 は CY-P1 の 70%程度の時間で学習が行なえることが示された。

最初に CY と CN の違いに着目する。表 A.1 の全平均では、CN と CY の予測正解率の差は 0.25% であり、CN のノード数は CY の約 91%となっている。CY では、CN における決定木構成子集団の学習対象項目の一部を sprig 集団に移行している。sprig 集団で決定木の構造をより細部まで学習し、それが決定木構成子集団で学習される結合関係を制限しているため、CY は CN ほど簡素な決定木を生成できなかったものと思われる。

次に P1 と PA の結果を比較すると、予測正解率の差は 0.45% であり、P1 のノード数は PA の約 88%となっている。PA では、各 sprig 集団における学習対象項目は P1 よりも少なく、遺伝操作は同じ属性遺伝子を持つ個体同士に対して施されるので、特定の属性に関する項目 (b) ~ (d) をより緻密に学習できる。しかし、属性遺伝子の異なる個体を共存させる P1 に比べて sprig 集団内の個体の多様性が少なく、その結果決定木構成子集団の多様性も減少するため、PA による決定木は P1 と比べてノード数が多いと考えられる。

学習時間に関しては、CY と CN の違いによる影響が大きい。CY ではクラスノードに接続可能な sprig が限られているため、解候補決定木の生成や遺伝操作に時間がかかったと推測される。CN と P1 を組み合わせることで学習時間がより短縮され、CY-P1 の 70%程度の時間で学習が行なえることから、CN-P1 は学習時間短縮に効果的といえる。

以上より、学習対象項目は訓練適応度と予測正解率に大きな影響を与えないが、簡素さと学習時間を左右する要因と考えられる。予測正解率と簡素さだけでなく学習時間も考慮すると、SESAT では CN-P1 が最も有用である。



## 発表論文

### 研究論文

1. 大谷 紀子, 河辺 信吾, 沼尾 正行 : 複数の情報媒体を用いた学習の過程 - プログラミング言語 Prolog を題材として, 認知科学 (日本認知科学会論文誌), Vol. 2, No. 1, pp. 72-85 (1995)
2. 伊藤 史朗, 大谷 紀子, 上田 隆也, 池田 裕治 : 属性オントロジーの抽出と統合を用いた実空間と情報空間のナビゲーションシステム, 人工知能学会論文誌, Vol. 14, No. 6, pp. 1001-1009 (1999)
3. 大谷 紀子, 大和田 勇人 : 共生進化に基づく帰納論理プログラミングの予測精度の向上, 人工知能学会論文誌, Vol. 17, No. 4, pp. 431-438 (2002)
4. 大谷 紀子, 志村 正道 : 共生進化に基づく簡素な決定木の生成, 人工知能学会論文誌, Vol. 19, No. 5, pp. 399-404 (2004)
5. 大谷 紀子, 木村 誠聡, 志村 正道 : 多数決機械を用いた雑音重畳画像の強調, 電子情報通信学会和文論文誌, Vol. J87-A, No. 10, pp. 1351-1357 (2004)
6. 大谷 紀子, 貝原 巳樹雄, 志村 正道 : ポリマー判別のための 2 段階判別決定木, 人工知能学会論文誌, Vol. 21, No. 3, pp. 295-300 (2006)

### 国際会議

1. Otani,N. , Itoh,F. , Shibata,S. , Ueda,T. and Ikeda,Y. : An Information Retrieval System based on Personal Viewpoints for Everyday Use, in *Proceedings of KES'98*, pp. 397-404 (1998)

### ワークショップ (査読あり)

1. 大谷 紀子, 木村 誠聡, 志村 正道 : 多数決機械を用いた雑音重畳画像の強調, 電子情報通信学会第 17 回回路とシステム軽井沢ワークショップ論文集, pp. 159-164 (2004)
2. Otani,N. , Kimura,T. and Nitta,K. : Image Enhancement by Committee Machine with Symbiotic Evolution, 電子情報通信学会第 19 回回路とシステム軽井沢ワークショップ論文集, pp. 171-176 (2006)
3. 大谷 紀子, 木村 誠聡, 新田 克己 : 多数決機械を用いたインパルス性雑音の検出法, 電子情報通信学会第 19 回回路とシステム軽井沢ワークショップ論文集, pp. 183-188 (2006)

## 研究会

1. 大谷 紀子, 沼尾 正行: プログラミング言語の学習過程におけるイメージの役割, 日本認知科学会リプレゼンテーション&インタフェース研究分科会資料, pp. 1-9 (1993)
2. 伊藤 史朗, 大谷 紀子, 上田 隆也, 池田 裕治: インターネットカメラを用いた実空間と情報空間の双方向ナビゲーションシステム, 人工知能学会第 43 回知識ベースシステム研究会資料, SIG-KBS-9803, pp. 61-68 (1999)
3. 清水 由美子, 大谷 紀子, 赤間 啓之: 同義語の意味の違いを測る - 「インターネット」と「ネット」に対する認知実験を例に -, 人工知能学会第 37 回音声・言語理解と対話処理研究会資料, SIG-SLUD-A203-01, pp. 1-6 (2003)
4. 大谷 紀子, 志村 正道: 決定木生成のための共生進化における部分解の検討, 電子情報通信学会技術研究報告, Vol. 103, No. 725, pp. 101-106 (2004)

## 全国大会

1. 大谷 紀子, 沼尾 正行: 人間の学習はイメージにより加速されるか? - プログラミング言語 Prolog の学習過程を題材として -, 日本認知科学会第 10 回大会予稿集, pp. 198-199 (1993)
2. 大谷 紀子, 沼尾 正行: 算数の抽象概念獲得におけるアナロジー, 日本認知科学会第 12 回大会予稿集, pp. 190-191 (1995)
3. 大谷 紀子, 伊藤 史朗, 柴田 昇吾, 上田 隆也, 池田 裕治: フロー情報収集・活用のための知的検索システム Fit(3) 類似度判定, 情報処理学会第 53 回全国大会予稿集, 2T-10 (1996)
4. 伊藤 史朗, 大谷 紀子, 柴田 昇吾, 上田 隆也, 池田 裕治: フロー情報収集・活用のための知的検索システム Fit(2) 処理方式, 情報処理学会第 53 回全国大会予稿集, 2T-9 (1996)
5. 上田 隆也, 大谷 紀子, 伊藤 史朗, 柴田 昇吾, 池田 裕治: フロー情報収集・活用のための知的検索システム Fit(1) コンセプト, 情報処理学会第 53 回全国大会予稿集, 2T-8 (1996)
6. 柴田 昇吾, 大谷 紀子, 伊藤 史朗, 上田 隆也, 池田 裕治: 知的検索システム Fit での重複記事の統合, 情報処理学会第 55 回全国大会予稿集, 5N-7 (1997)
7. 大谷 紀子, 伊藤 史朗, 柴田 昇吾, 上田 隆也, 池田 裕治: 知的検索システム Fit での類似度判定の改良, 情報処理学会第 55 回全国大会予稿集, 5N-6 (1997)
8. 中山 泰夫, 大谷 紀子, 志村 正道: 共生進化に基づく決定木の最適化に関する研究, 人工知能学会第 14 回全国大会論文集, pp. 327-328 (2000)
9. 大谷 紀子, 志村 正道: コミュニティ指向共生進化による決定木の生成, 人工知能学会第 15 回全国大会論文集, 3D2-01 (2001)
10. 大谷 紀子, 大和田 勇人: 帰納論理プログラミングにおけるコミッティ学習法とその精度評価, 人工知能学会第 16 回全国大会論文集, 1A4-06 (2002)

11. 大谷 紀子, 木村 誠聡, 志村 正道 : 共生進化に基づく雑音重畳画像の強調, 人工知能学会第 18 回全国大会論文集, 1H3-04 (2004)
12. 大谷 紀子, 貝原 巳樹雄, 佐伯 和光, 志村 正道 : ポリマー判別のための共生進化に基づく決定木生成, 人工知能学会第 19 回全国大会論文集, 2F1-03 (2005)

#### その他

1. 沼尾 正行, 大谷 紀子 : 感性への計算的アプローチ, 日本認知科学会テクニカルレポート, No. 22 (1993)
2. Otani,N. and Numao,M. : Experimental prolog textbooks, *Titech Technical Report (Dept. of CS)*, TR94-0003 (1994)
3. 大谷 紀子, 大和田 勇人 : 共生進化に基づく帰納論理プログラミングの予測精度の向上, 東京理科大学情報メディアセンターテクニカルレポート, AIST-TR-01-026 (2001)
4. 大谷 紀子 : 情報検索におけるベクトル空間モデルの応用, 武蔵工業大学環境情報学部紀要, Vol. 5, pp. 99-109 (2004)