

広島市立大学審査博士学位論文

複数の木構造を持つ遺伝的手法を用いた
自律エージェントの学習

2018年9月

広島市立大学大学院 情報科学研究科
博士後期課程 情報科学専攻

伊藤 崇

要旨

近年，お掃除ロボットのように，自ら考え行動する自律的なエージェントが普及してきている．エージェント学習は，人工知能の研究分野の1つであり，目的を達成するためにエージェントの判断と行動の規則を自動生成する手法に関する研究である．生物進化のメカニズムを模倣する進化的計算は，エージェント学習にとって有効であることが知られている．本研究では，進化的計算の1つである遺伝的プログラミング (GP) を使用する．

GP の個体は，1つの木構造でエージェントの行動規則を表現するため，複雑な行動規則を表現するために木の深さを深くする必要がある．しかしながら，そのような木構造には，1度の遺伝操作によって優れた行動規則が容易に破壊されてしまうという問題がある．そこでこの問題を改善するために，調整ノード付き GP (GP_{CN}) が提案されている． GP_{CN} の個体は複数の木構造により構成される．また，各木は，エージェントが木を参照する順序を示す識別番号と木を繰り返し参照する回数を示すプロセス数 P を持つ． GP_{CN} の個体内の複数の木構造は，それぞれが独立しているため，木ごとに異なる行動規則を表現することができる．したがって， GP_{CN} は，GP において木構造が深くなる原因となる複雑な行動規則を個体内の複数の木に分割することができるため，複雑な行動規則を表現するために木の深さが深くなるという問題を解決することができる．

しかしながら， GP_{CN} は個体が複数の木を持つため，進化の速度が遅くなるという問題があった．本論文では， GP_{CN} の性能を向上させるために，個体内の複数の木の参照順に着目した木の順序入れ替え手法を提案する． GP_{CN} では，エージェントが木を参照する順序は，識別番号によって管理されている．しかしながら，はじめから各木が適切な順序に配置されているとは限らない．木の順序入れ替え手法では，個体内の木をその個体内の他の木と交換することによって，エージェントが各木を参照する順序を並べ替えることができる．実験では，先行研究で提案されている手法と木の順序入れ替え手法を GP_{CN} に適用し，ベンチマーク問題を用いて比較を行った．本論文のベンチマーク問題は，ごみ拾い問題と Santa Fe Trail 問題である．実験結果から，ごみ拾い問題では，提案手法が従来手法よりも優れた性能を示すことが確認された．また，提案手法は，2つのベンチマーク

問題において1000世代を過ぎても進化が続いており、局所解に陥りやすいという問題を改善することにも有効であることが示された。

また GP_{CN} は、木の数 M と各木のプロセス数 P をあらかじめ設定しなければならず、これらの最適値はベンチマークの問題によって異なることがわかっている。そこで、先行研究では、プロセス数 P を少しずつ更新し、進化の中で問題に適したプロセス数 P を獲得するプロセス数 P の更新手法が提案されていた。しかしながら、プロセス数 P の更新手法にはプロセス数 P の取りうる範囲が広がるほど効率が悪くなるという問題がある。本論文では、この問題を解決するために、プロセス数 P の継承手法を提案する。プロセス数 P の継承手法では、遺伝操作の1つである交叉が実行される際に、2つの親個体のうち高い適合度を有する親個体のプロセス数 P を生成される2つの子個体に継承する。したがって、優良な個体のプロセス数 P を次世代個体に継承できるため、問題に適したプロセス数 P の値を獲得できる。しかしながら、プロセス数 P の継承手法は1つの親個体のプロセス数 P を2つの子個体に継承するため、個体集団中のプロセス数 P の多様性が急速に失われてしまう可能性がある。そこでこの問題を回避するために、プロセス数 P の突然変異を提案する。プロセス数 P の突然変異は、遺伝操作の1つである突然変異の操作のうち、ランダム木との交換が行われた際にプロセス数 P をランダムに変化させる。したがって、本手法によって個体集団中に新たなプロセス数 P が発生するため、プロセス数 P の多様性を維持することができる。

さらに本論文では、先行研究において人手で決定していた個体の木の数 M を進化の過程で自動的に決定するための手法を提案する。木の数 M の自動決定は、初期個体集団に異なる木の数 M を持つ個体を生成し、適合度を用いた自然淘汰によって行う。我々はできるだけ少ない木の数 M に決定しシンプルな個体としたいが、従来の適合度では木の数 M の多い個体が生き残りやすくなっていた。木の数 M の多い個体は、複数の行動規則を持つために問題に柔軟に対応することができる反面、必要のない木を多く含んでしまう可能性がある。そこで、無駄のない木の数 M に決定するために評価関数を木の数 M を考慮するように変更する。個体の複雑性と性能を考慮した評価関数として Akaike's Information Criterion (AIC) や Minimum Description Length (MDL) が提案されている。しかしながら、本研究で用いる GP_{CN} は複数の木構造を持ち、単純に木の数やノード数を当てはめるこ

とができないため、経験則に基づいて直感的に評価関数を変更した。一方、評価関数の変更により木の数 M が少ない数に決定されるようになったが、自然淘汰による収束が急速に行われるため、様々な木の数 M を持つ個体の探索が十分にできていない。我々はこの問題を解決するために個体の突然変異を提案する。個体の突然変異は、木の数 M を増やす操作と木の数 M を減らす操作からなる。木の数 M を減らす操作のために、エージェント問題のための新しいセマンティクスを提案する。これは関数同定問題に対して有効なセマンティクスは見つかったが、エージェント問題に応用できなかったためである。また、木の数を減らす操作の1つとしてセマンティクスを用いた木の削除を提案する。セマンティクスを用いた木の削除は、隣接した木々のセマンティクスを比較し類似した木の一方を削除することで、個体の行動規則全体に影響を与えず木の数 M を減らすことができる操作である。プロセス数 P と木の数 M を自動的に決定する手法の有効性を確認するために、提案手法を GP_{CN} に適用し、プロセス数 P と木の数 M を自動的に決定する実験を行った。Santa Fe Trail 問題では、自動決定で得られたプロセス数 P と木の数 M は人手で決定した値と同様の値であった。ごみ拾い問題では、自動決定で得られたプロセス数 P と木の数 M は人手で決定した値と異なっていた。そこで、自動決定で得られた値と人手によって決定された値を GP_{CN} に適用し、ごみ拾い問題を用いて実験を行った。実験結果から、自動決定により得られたプロセス数 P と木の数 M は、人手で決定した値よりも適した値であることが確認された。

Abstract

In recent years, autonomous agents that think and act by oneself like a vacuum cleaning robot are popular. Agent learning is one of active fields in artificial intelligence, and is research on a method for generating automatically agent decisions and action rules to reach a goal. Evolutionary learning, which imitates the mechanism of biological evolution, is known to be valid for agent learning methods. In this study, we use genetic programming (GP), which is one of evolutionary learning.

The individual of GP is comprised of a single tree structure representing agent behavior. In GP, the depth of generated trees tends to be deep to obtain complex action rules because one individual has only one tree. Deep trees with complex action rules have a shortcoming that there is a possibility that excellent action rules might be destroyed by a single genetic operation. In our study, genetic programming with control nodes (GP_{CN}) has been proposed to improve the shortcoming. An individual in GP_{CN} comprises multiple trees. Each tree has the following two numbers: the identification number that indicates the order in which an agent refers to a tree, and the number of processes P that indicates the number of repetition by which an agent carries out the action designated by the leaf node in a tree. In GP_{CN} , an individual has multiple trees, and each tree represents different an action rule. Therefore, GP_{CN} can solve the shortcoming that a single tree becomes the deep tree to obtain complex action rules, because in GP_{CN} , complex action rules represented by a single tree in GP can be divided into multiple simple trees. However, GP_{CN} had problems that the rate of evolution becomes slow and it falls easily to local optima because an individual has multiple trees. In the previous work, to solve these problems, the method generating individuals by using the conditional probabilities and the cross-cultural island model to promote diversity and improve performance have been proposed. In addition, the island model that combines the conditional probability with two islands in the cross-cultural island model has been proposed. By employing these methods, the improvement in the performance of GP_{CN} has been proved.

In this paper, to improve the performance of GP_{CN} , a new method that focuses on

the reference order of multiple trees in an individual is proposed. In GP_{CN} , the order in which an agent refers to a tree is controlled by the identification number. However, the trees might not be in proper order. The new method can rearrange the order in which an agent refers to a tree by exchanging a tree in an individual with another tree in the individual. The previous methods and the proposed method are applied to GP_{CN} , and then are compared by using the benchmark problems. In this paper, the benchmark problems indicate a garbage collection problem and the Santa Fe Trail problem. From the experimental results, we ascertained that in the garbage collection problem the proposed method showed better performance than the previous methods. In addition, the proposed method is effective to overcoming the local optima problem because evolution has continued after 1000 generations in both benchmark problems.

In GP_{CN} , the identification number and the number P have to be preset, and these optimal values might be different depending on benchmark problems. Thus, in previous work, a method for updating the number P to obtain P suitable to the tree in evolution has been proposed. However, there is a problem that efficiency becomes worse as the range of P becomes wider. In order to solve the problem, a method to inherit the number P is proposed. When crossover is applied to two parent individuals, value of P of the parent individual with the higher fitness value in the two parent individuals is set to P of both two child individuals. The authors consider that with the method, the number P can be converged to an optimal value, because the number P of excellent individuals can be inherited to descendants. However, a value of P might quickly spread out among individuals, which might cause to lose the diversity of P and to hinder P from converging a value because the number P in a parent individual is inherited to two child individuals. Therefore, to avoid the problem, a method using mutation for the number P is proposed. The method randomizes the value of P when mutation for a tree in genetic operations is carried out. When the population whose values of P are being lead into local optima, the method can prevent it by setting a random value to P .

Additionally, in this paper, a method for determining automatically the number of trees M in individuals that had been manually determined in previous work is proposed.

In the automatic determination of the number of trees M , individuals with the number of various trees are generated to initial population, and a suitable value for the benchmark problem is determined by the natural selection. We hope to decide the number of trees M on a small number and obtain a simple individual. However in the previous fitness function, a lot of individuals with the large number of trees M survive a lot. The individuals with the large number of trees M can flexibly cope with benchmark problem because it has multiple action rules. On the other hand, it may contain a lot of unnecessary trees. Thus, to decide the number of trees M such as not to contain wasted trees, we change the fitness function to the fitness function that takes the number of trees into consideration. As the fitness function considering complexity and performance of individuals, Akaike's Information Criterion (AIC) and Minimum Description Length (MDL) have been proposed. However, we can not easily apply those fitness functions to GP_{CN} because GP_{CN} using in this study have multiple trees. Thus, we intuitively changed the fitness function based on heuristics. The small number of trees M is decided by changing the fitness function. On the other hand, there is problem that the number of various trees M can not be fully searched because the population rapidly converges by natural selection. In order to solve this problem, we propose the mutation of individuals. The mutation of individuals is composed of the manipulation that increases the number of trees M and the manipulation that decreases the number of trees M . For the manipulation decreasing the number of trees M , we propose new semantics for the agent problem. Though effective semantics for the function identification problem has been found, effective semantics for the agent problem is not found. Therefore, we propose a method for deletion of trees employing the semantics as one way of the manipulation decreasing the number of trees M . The method for deletion of trees employing the semantics is a manipulation that can decrease the number of trees M without affecting the action rules overall by deleting the neighboring similar trees. To confirm effectiveness of the methods for determining automatically the value of P and the number of trees M , we applied these methods to GP_{CN} , and then made the experiments for determining automatically these P & M . In Santa Fe Trail problem, P & M obtained by automatic determination were similar values to manual determination.

In the garbage collection problem, P & M obtained by automatic determination were different from manual determination. We applied these P & M to GP_{CN} , conducted the experiment employing the garbage collection problem. From the experimental results, we ascertained that P & M obtained by automatic determination were suitable values than manual determination.

目次

第 1 章	まえがき	1
1.1	研究背景と目的	1
1.2	本論文の構成	5
第 2 章	遺伝的プログラミング (GP)	7
2.1	GP のアルゴリズム	7
2.2	GP の遺伝操作	8
第 3 章	調整ノード付き GP (GP_{CN})	13
3.1	GP_{CN} のアルゴリズム	13
3.2	GP_{CN} の遺伝操作	14
第 4 章	エージェントシミュレーション (ベンチマーク問題)	21
4.1	概要	21
4.2	ごみ拾い問題	21
4.3	Santa Fe Trail 問題	24
第 5 章	GP_{CN} の性能を向上させるための研究	27
5.1	概要	27
5.2	先行研究	27
5.2.1	条件付き確率を用いた個体生成手法	27

5.2.2	異文化型島モデル	30
5.2.3	条件付き確率を用いた個体生成手法を導入した島モデル	32
5.3	Santa Fe Trail 問題	36
5.4	評価実験	38
5.4.1	実験設定	38
5.4.2	性能評価	40
5.5	まとめ	43
第 6 章 GP _{CN} の性能を向上させるための研究		45
6.1	概要	45
6.2	プロセス数 P を自動決定するため手法	45
6.2.1	プロセス数 P の更新手法	45
6.2.2	提案手法	46
A.	終端ノードのみの木を淘汰するための評価関数	46
B.	プロセス数 P の継承手法	47
C.	プロセス数 P の突然変異	47
6.2.3	終端ノードのみの木を淘汰するための評価関数の変更	50
6.2.4	プロセス数 P を自動決定する手法の実験	54
6.2.5	まとめ	57
6.3	木の数 M を自動決定するための手法	58
6.3.1	提案手法	58
A.	木の数 M を考慮した評価関数	58
B.	個体の突然変異	58
C.	エージェント問題のためのセマンティクス	64
6.3.2	木の数 M を考慮した適合度の評価実験	66
6.3.3	木の数 M を自動決定する手法の実験	68
6.3.4	自動決定により得られたパラメータの評価	72

6.3.5	まとめ	76
第7章	むすび	77
	謝辞	79
	参考文献	81
	研究業績	85

第 1 章 まえがき

1.1 研究背景と目的

人間の知能を計算機で実現させることを目指す人工知能分野において、目的を達成するためにエージェントの判断と行動の規則を自動生成する手法の研究が行われてきた。近年、お掃除ロボやソフトバンクが販売している Pepper のように、自分で状況を判断し行動する人工知能を搭載した自律エージェントが我々の身の周りに普及してきている。その代表的な手法として、強化学習 [1, 2, 3] と進化的計算がある。生物進化のメカニズムを模倣する進化的計算は、エージェントのための最適な行動規則を獲得する手法として有効であることが知られている。代表的な手法として、遺伝的アルゴリズム (GA) [6], 遺伝的プログラミング (GP) [5, 6], 遺伝的ネットワークプログラミング (GNP) [7] があり、本研究では GP を用いてエージェント学習の研究を行う。

現世代の個体集団において、その環境下で適合度の高い個体を親として子個体を生成し、次世代の個体集団を形成する。各個体は、1つの木構造で表現される。葉ノードはエージェントの行動、それ以外のノードは知覚情報によるエージェントの分岐に相当する。次世代個体の生成には、交叉、突然変異、逆位と呼ばれる遺伝操作が用いられ、エリート保存によって優良個体の一部が次世代個体に受け継がれる。

この GP の性能を向上させるために、交叉や突然変異といった遺伝操作を用いず、複数の親個体から木構造の断片をランダムにサンプリングし結合させることで次世代個体を生成する手法 [8] や、有用な木構造を利用する手法として、個体集団の中に頻繁に出現する部分木を抽出し個体生成に用いる頻出部分木という手法 [9, 10] とそれを島モデルに導入した手法 [11] が提案されている。セマンティックスを取り入れた研究として、セマンティックスによる部分木の類似度を用いた破壊的操作の起こりにくい交叉 [12] や多様性を考慮した親個体の選択手法 [13], セマンティックバックプロパゲーションを用いた部分木の突然変異と交叉 [14] が提案さ

れている。また、ブロート現象を回避するために、遺伝操作の1つである交叉を排除し個体間の距離を目的関数に加えた多目的手法を用いた GP[15] や、親ノードだけでなく子ノードの依存関係も考慮した近傍探索に基づく多段階探索交叉 [16] が提案されている。さらに、ヒューマノイドロボットを制御するグラフ構造を獲得するために GP を用いた研究として二分決定グラフのための交叉が提案され [17], 画像分類の分野においては深層学習をヒントに2つの GP を重ねた階層的な特徴構築 [18] が行われている。

GP ではエージェントの行動規則が1つの木で処理されるため、複雑な行動規則を獲得するためには木の深さを深くする必要がある。しかしながら、木の深さを深くすると可読性は下がり、1度の遺伝操作によって優良な行動規則が破壊されてしまう可能性がある。これを解決する手法として、我々の研究では調整ノード付き GP (GP_{CN}) を提案している [19, 20]。 GP_{CN} では1個体が複数の木を持ち、各木には参照順を表す識別番号と行動回数を表すプロセス数 P が設定されている。各木は互いに独立な行動規則を持っているため、木の深さによる問題を解決することができる。しかしながら、複数の木を持つために適合度の上がり方が遅くなる。

そこで、我々は GP_{CN} の性能を向上させるための研究を行った。先行研究において、条件付き確率を用いた個体生成手法の提案 [20, 21] と GP の性能向上のために提案されている異文化型島モデル [22] の GP_{CN} への応用を行った [21, 23, 24]。条件付き確率を用いた個体生成手法では、優良個体から抽出した連結ノード間の条件付き確率を用いた操作と遺伝操作により次世代個体が生成される。これにより、多様性を維持しながら次世代個体に優良個体の行動規則を継承することができる。また、異文化型島モデルでは、母集団を2つの島に分ける。一方の島は、適合度重視の島と呼び、適合度を上げることに重点を置く。もう一方の島は、多様性重視の島と呼び、多様性の維持に重点を置く。また、島モデルの特徴である移住によりそれぞれの島の個体の一部を交換することで、それぞれの島の欠点を補い局所解に陥ることを防ぐことができる。さらに、有効性の示された2つの手法を組み合わせた条件付き確率を用いた個体生成を導入した島モデルを提案し [25, 26], 島モデルの島の働きを活性化させ GP_{CN} の性能を向上させることに成功した。条件付き確率を用いた個体生成を導入した島モデルでは、適合度重視の島

に条件付き確率を用いた個体生成を導入し、適合度の向上を助ける。条件付き確率表は適合度重視の島の優良個体を用いて作成する。多様性重視の島では、あまり選出されないノードが選出されやすいように、適合度重視の島で作成された条件付き確率を変換した多様性を考慮した条件付き確率を用いて個体生成を行い、多様性の維持を助ける。

本論文では、 GP_{CN} の個体が複数の木を持ち、それらが識別番号順に参照されることに着目し、木の順序をランダムに入れ替えることによって進化を促進させる木の順序入れ替え手法を提案している [26, 27]。木の順序入れ替え手法では、順序を入れ替えることで個体内の行動規則が適当な順に並ぶことを期待している。

また、これまでの研究では、個体の木の数は木の数を変更して複数回予備実験を行い、最も適合度が高くなったものを選択し使用していた。しかしながら、問題ごとに異なる木の数を最適化するために毎回予備実験を行うのは非効率的である。また、各木のプロセス数 P の値には問題ごとに設定されているエージェントの最大行動回数を個体の木の数で割ったものを使用していたが、全ての木が1度しか参照されないように設定する必要はない。そこで、進化の過程でそれらのパラメータを自動的に決定する手法の研究も行った。先行研究において、プロセス数 P の値を増減させ、世代の進み具合によって増減幅を小さくし収束させるプロセス数 P の更新手法が提案されている [23, 24]。しかしながら、この手法は1度の操作で個体内の1つの木のプロセス数 P の値しか変更しないため、木の数が多くなると効率が悪くなる。また、個体の木の数が少ない場合にも、プロセス数 P の探索範囲が広がるため効率が悪くなってしまう。

そこで、本研究ではプロセス数 P の継承手法を提案している [28, 29]。プロセス数 P の継承手法では、遺伝操作によって子個体を生成する際に、優良個体のプロセス数 P の値を子個体に引き継がせる手法である。このように子個体に引き継いでいくことで、優良な親個体のプロセス P の値が個体集団中に拡散し、より効率的にプロセス P が収束すると考えられる。また、探索が十分に行われる前に個体集団中のプロセス数 P の値が収束することを防ぐために、プロセス数 P の突然変異を提案している [28, 29]。プロセス数 P の突然変異は、決められた範囲から値をランダムに選択し、その値を対象として選択された木に設定する手法である。これらの手法を用いて、プロセス数 P の自動的な決定を行った。さらに、進化を促

進しプロセス数 P の収束を促すために、評価関数の変更も行った [28, 29]. GP_{CN} は個体が複数の木を持つために、通常の GP では進化の過程で淘汰されていく終端ノードのみで構成された木が生き残ってしまっていた. 終端ノードのみの木が遺伝操作で選択された場合、遺伝操作は行われないため進化が妨害されてしまう. そこで、終端ノードのみの木を持つ個体の評価を下げることで問題の解決を行った.

さらに、本研究では、これまで人手で決定していた個体の木の数 M を、進化の過程で自動的に決定するための手法を提案する [30, 31, 32]. 木の数 M の自動決定は、初期個体集団に異なる木の数 M を持つ個体を生成し、適合度を用いた自然淘汰によって行う. 我々はできるだけ少ない木の数 M に決定しシンプルな個体としたいが、従来の適合度では木の数 M の多い個体が生き残りやすくなっていた. 木の数 M の多い個体は、複数の行動規則を持つために問題に柔軟に対応することができる反面、必要のない木を多く含んでしまう可能性がある. そこで、無駄のない木の数 M に決定するために評価関数を木の数 M を考慮するように変更する [30, 31, 32]. 個体の複雑性と性能を考慮した評価関数として Akaike's Information Criterion (AIC) や Minimum Description Length (MDL) [33] が提案されている. しかしながら、本研究で用いる GP_{CN} は複数の木構造を持ち、単純に木の数やノード数を当てはめることができないため、経験則に基づいて直感的に評価関数を変更した.

また、評価関数の変更により木の数 M が少ない数に決定されるようになったが、自然淘汰による収束が急速に行われるため、様々な木の数 M を持つ個体の探索が十分にできていない. そこで、様々な木の数を探索するために個体の突然変異を提案する [30, 31, 32]. 個体の突然変異は木を追加する操作と木を削除する操作から成り、木を削除する操作の1つとしてセマンティクスを用いた木の削除がある. GP_{CN} は識別番号の小さい順に木を参照するため、隣り合っている木が同じような行動規則を持つならば、どちらかを削除しても個体の行動にはほぼ影響がない. そこで、個体の性質を変えることなく木を削除する方法としてセマンティクスを用いた木の削除を考えた. さらに、新たにエージェント問題のためのセマンティクスを提案する. これは、関数同定問題のためのセマンティクスについては有効なものが見つかったが、エージェント問題に応用できなかったため

ある.

実験では, ベンチマーク問題として広く用いられているごみ拾い問題 [34] と Santa Fe Trail 問題 [5, 35] に各手法を適用し, 性能を評価する. 前者は比較的容易な問題で, 後者は難しい問題となっている.

1.2 本論文の構成

本論文は以下のような構成となっている. 2章では本研究で使用している複数木構造を持つ調整ノード付き GP_{CN} の基となった遺伝的プログラミング (GP) について説明する. そして, 3章では調整ノード付き GP_{CN} について説明し. 4章で GP_{CN} のエージェントシミュレーションに用いたごみ拾い問題と Santa Fe Trail 問題について述べる. 5章は GP_{CN} の向上を目的としたこれまでの研究をまとめた章となっており, 先行研究において提案されている条件付き確率による個体生成手法, 異文化型島モデル及び2つの手法を組み合わせた条件付確率を用いた個体生成手法を導入した異文化型島モデルや, 本研究の提案手法である木の順序入れ替え手法について説明し, ベンチマーク問題を用いた実験結果によってその有効性について議論している. 6章は, GP_{CN} におけるパラメータの自動決定に関する研究についてまとめた章であり, プロセス数 P と木の数 M を自動決定するための提案手法を説明し, ベンチマーク問題を用いた実験結果によって議論している. 最後に, 7章でまとめと今後の課題を述べる.

第2章 遺伝的プログラミング (GP)

2.1 GP のアルゴリズム

進化的計算とは、生物進化のメカニズムを模倣して提案された最適化手法であり、エージェントのための最適な行動規則を獲得する手法として有効であることが知られている。その代表的な手法として、遺伝的アルゴリズム (GA) [6], 遺伝的プログラミング (GP) [5, 6], 遺伝的ネットワークプログラミング (GNP) [7] の研究が盛んに行われている。我々は、この中の GP に焦点を当てて研究を行っている。GP は GA の拡張手法であり、GA において配列で表現されていた個体を GP では木構造で表現する。そのため、GA では表現できなかった数式やプログラムのコードなど、構造を持ったデータの表現が GP では可能である。GP の個体例を図 2.1 に示す。各個体は、1 つの木構造で表現される。葉ノードは終端ノードと呼ばれエージェントの行動を示し、それ以外のノードは非終端ノードと呼ばれ知覚情報によるエージェントの分岐を示す。エージェントは根ノードから参照を開始し、非終端ノードの判定に基づいて木構造を辿り、終端ノードでエージェントが実行すべき行動を決定する。エージェントが終端ノードを参照すると、エージェ

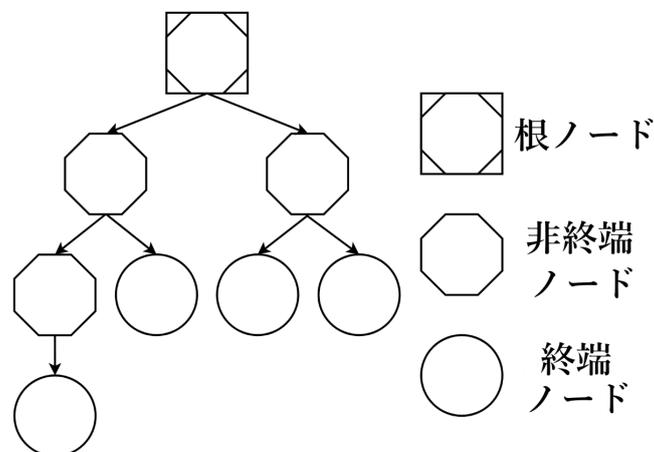


図 2.1 GP の個体例

ントは根ノードから再びノードの参照を行う。

GPの進化の流れを図2.2に示す。はじめに、個体集団の初期生成を行う。次に、生成した各個体の適合度を求める。ここで、終了条件が満たされていないならば、個体集団の再生及び遺伝操作を行い、次世代の個体集団を生成する。なお、終了条件は「世代数が実行時に与える最大世代数（ここでは N_g ）になる」とする。遺伝操作には、交叉、突然変異、逆位の3つを用いる。

2.2 GPの遺伝操作

GPでは、次世代個体を生成するために3つの遺伝操作を用いる。これから、それぞれの操作について説明する。

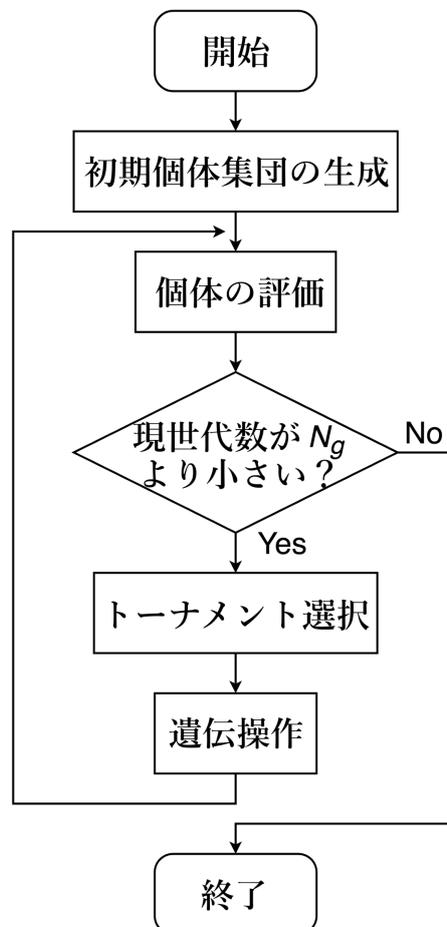


図 2.2 GPの進化の流れ

- 交叉

交叉とは、選択された2つの親個体のそれぞれの木の中の部分木を交換する操作である。交叉の例を図2.3に示す。まず、それぞれの親個体の全ノードからランダムに交叉するノードを選択する。図2.3の例では、親個体1は実線枠内の非終端ノードAが選択され、親個体2は破線枠内の非終端ノードBが選択されている。次に、選択されたノードを根ノードとする部分木を交換し、子個体を生成する。

しかし、選択された木がどちらも終端ノードを根ノードとする場合は、交

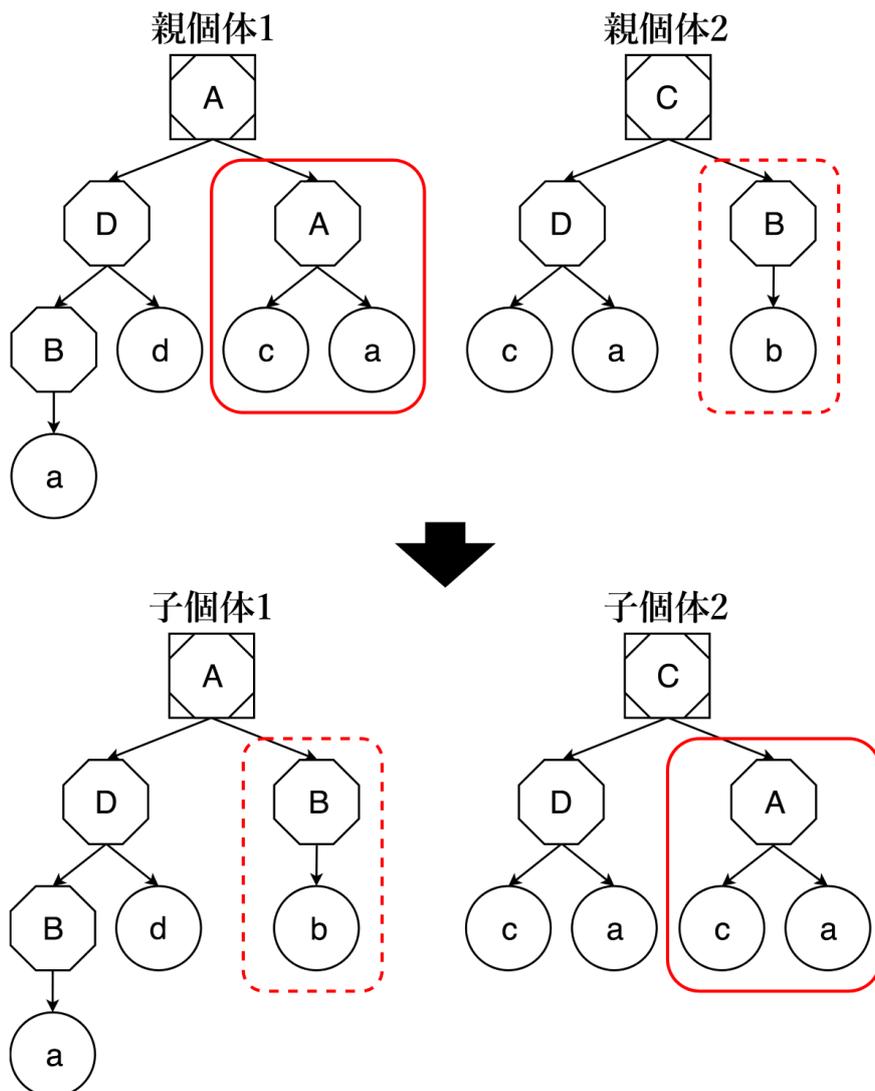


図 2.3 GP における交叉の例

又は行わない。

- 突然変異

突然変異とは、選択された親個体の木の全ノードからランダムに1つのノードを選択し、突然変異木率によってノードの種類の変更，または選択されたノードを根ノードとする部分木とランダムに生成された部分木の入れ替えを行う操作である。図2.4と図2.5に突然変異のノードの種類の変更の例を示す。図2.4は終端ノードが選択された際のノードの種類の変更の例である。例では、選択された終端ノードbが終端ノードaに変更されている。図2.5は非終端ノードが選択された際のノードの種類の変更の例である。非終端ノードの種類の変更では、エッジの数が増える可能性がある。例では、エッジの数が増える場合を示している。エッジの数が増える場合には、

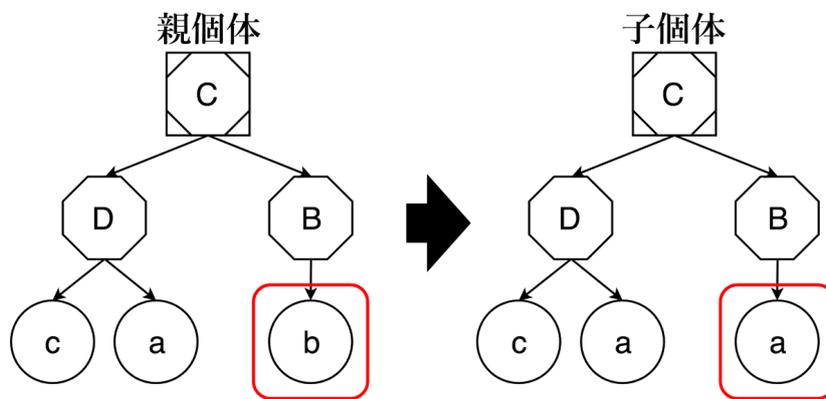


図 2.4 GP における終端ノードが選択された場合のノードの種類の変更の例

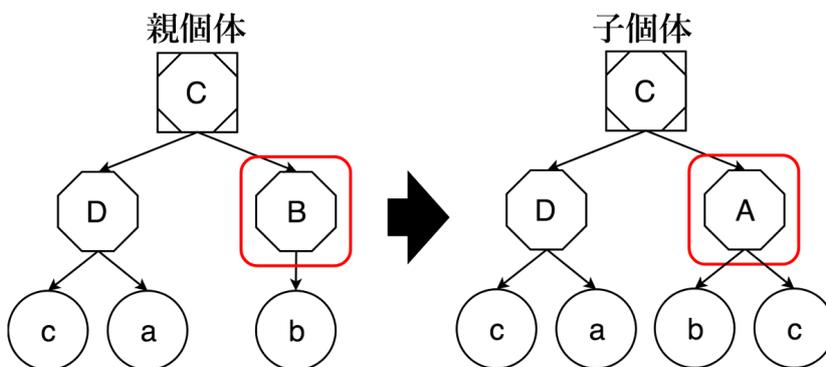


図 2.5 GP における非終端ノードが選択された場合のノードの種類の変更の例

例のように増えたエッジの先にランダムに生成した部分木を新たに接続する．また，エッジの数が減った場合には余分なエッジを削除する．

突然変異木によるランダム木との入れ替えでは選択されたノードを削除し，そこに新たにランダムに生成した部分木を接続する．図 2.6 に突然変異木による操作の例を示す．例では，選択された非終端ノード B が削除され，ランダムに生成された非終端ノード A を根ノードとする部分木を接続する．

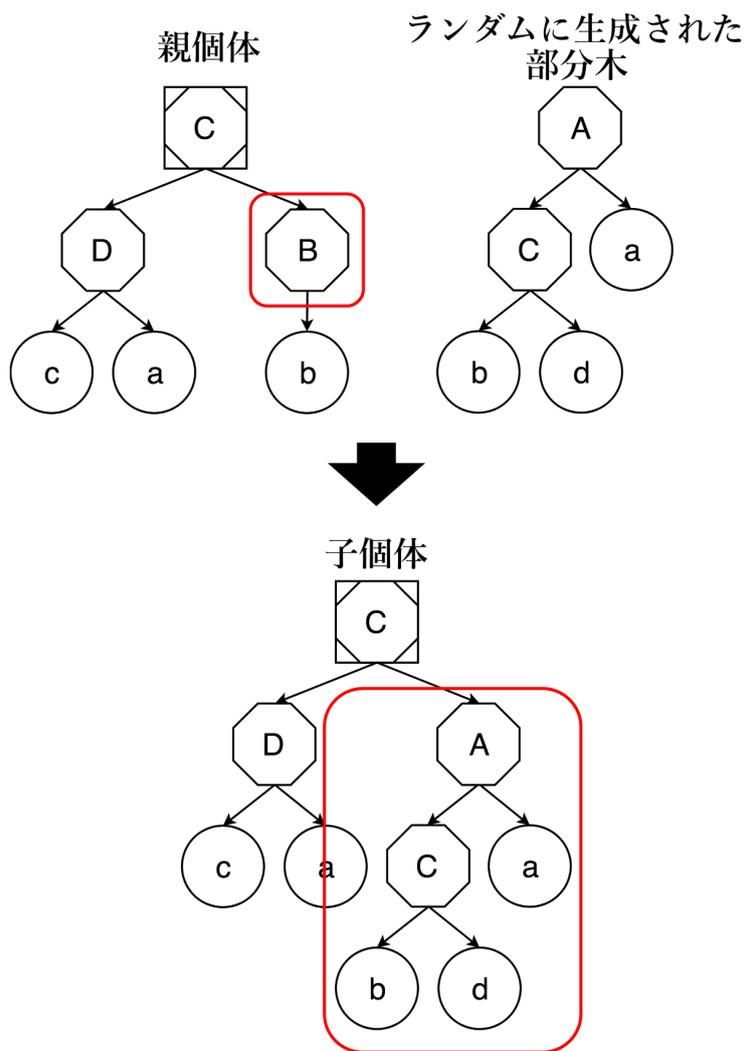


図 2.6 GP におけるランダムな部分木との入れ替えの例

● 逆位

逆位とは、選択された1つの親個体からランダムに非終端ノードを選び、それを親とする2つの子ノードを交換する操作である。逆位の例を図2.7に示す。まず、親個体の全ノードの中からランダムに非終端ノードを1つ選択する。例では実線で囲われた非終端ノードAが選択されている。次に、そのノードを親とする子ノードの中からランダムにノードを2つ選択する。最後に、選ばれたそれぞれのノードを根ノードとする部分木を交換する。ただし、選択された個体の根ノードが終端ノードの場合は、逆位を行わない。

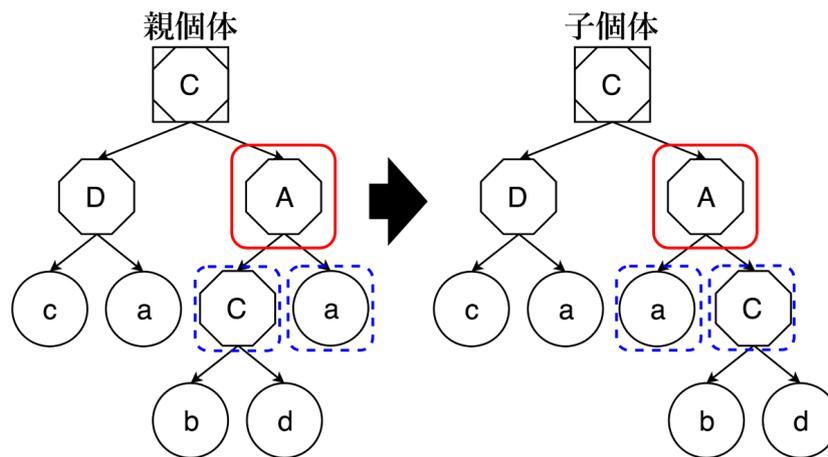


図 2.7 GP における逆位の例

第3章 調整ノード付き GP (GP_{CN})

3.1 GP_{CN} のアルゴリズム

我々の研究では、遺伝的プログラミング (GP) において1個体が複数の木を持つように拡張した調整ノード付き GP (GP_{CN}) を提案している [19], [20]. GP_{CN} の個体例を図 3.1 に示す. GP_{CN} の個体は、それぞれが規則に対応する複数の木から構成されている. 各木は、参照順を表す識別番号と行動回数を表すプロセス数 P を持っている. 1個体の木の数 M (つまり調整ノードの数) は、あらかじめ定義されているとする. また、プロセス数 P は木ごとに独立しており、初期値として $(TotalSteps)/M$ の値が設定されている. ここで、 $TotalSteps$ は問題ごとに設定されているエージェントの最大行動回数である. 非終端ノードには知覚情報に関する分岐が、終端ノードにはエージェントが実行できる行動が記されている. 木の参照は、識別番号の最も小さい木から行い、その木におけるエージェントの行動回数がその木に設定されているプロセス数 P を超えたとき、次の識別番号を持つ

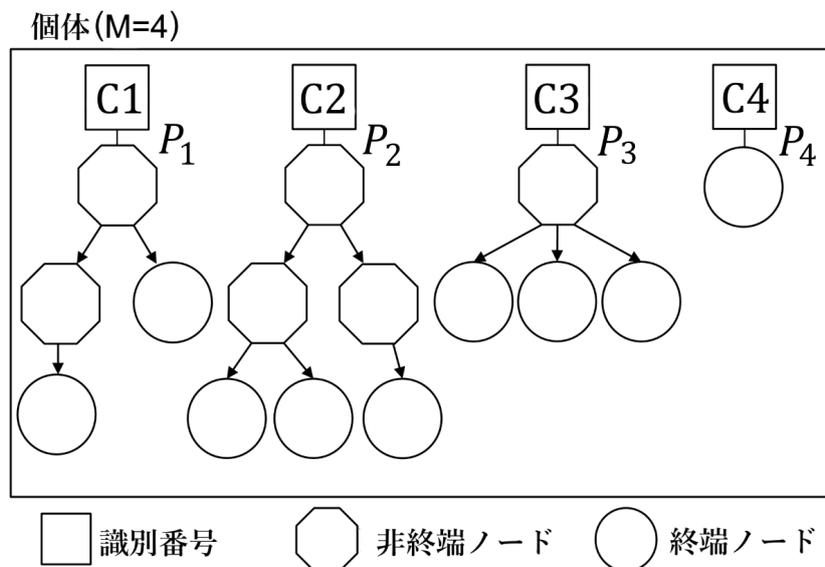


図 3.1 GP_{CN} の個体例

木に遷移する。また、識別番号の最も大きい木においてその木におけるエージェントの行動回数がプロセス数 P を超えたとき、次に遷移するのは、識別番号の最も小さい木である。その際、各木におけるエージェントの行動回数は初期化される。エージェントは、遷移してきた木の行動回数の合計が $TotalSteps$ になるまで、環境から知覚情報を受け取り行動するまでの動作を繰り返す。また、参照している木におけるエージェントの行動回数がその木に設定されているプロセス数 P を超えない場合が、エージェントが遷移してきた木の行動回数の合計が $TotalSteps$ に達した場合に起こる。その場合、シミュレーションはそこで終了し、以降の木は参照されない。

GP_{CN} のアルゴリズムは GP と同様である。はじめに、個体集団の初期生成を行う。次に、生成した各個体の適合度を求める。 GP_{CN} の個体は複数の木を持っているが、適合度の計算は個体に対して行い、個々の木については行わない。ここで、終了条件が満たされていなければ、個体集団の再生及び遺伝操作を行い、次世代の個体集団を生成する。なお、終了条件は「世代数が実行時に与える最大世代数になる」とする。また、 GP_{CN} における遺伝操作は、GP と同様に交叉、突然変異、逆位の3つである。

3.2 GP_{CN} の遺伝操作

- 交叉

GP_{CN} の交叉の例を図 3.2 に示す。 GP_{CN} の交叉では個体が複数の木構造を持つために、はじめに2つ親個体からそれぞれ交叉する木をランダムに選択する。図 3.2 ではそれぞれの親個体で選択された木を実線の枠で示している。次に、それぞれ選択された木の全ノードからランダムに交叉するノードを選択する。図 3.2 の例では破線で囲まれた非終端ノード D と終端ノード a が選択されている。最後に、選択されたノードを根ノードとする部分木を交換する。しかしながら、選択された木がどちらも終端ノードを根ノードとする場合は、交叉は行わない。

また、子個体の各木に設定されるプロセス数 P の値については、2つの親個体のうち、似た木構造を多く持つ親個体のプロセス数 P の値を引き継ぐ。

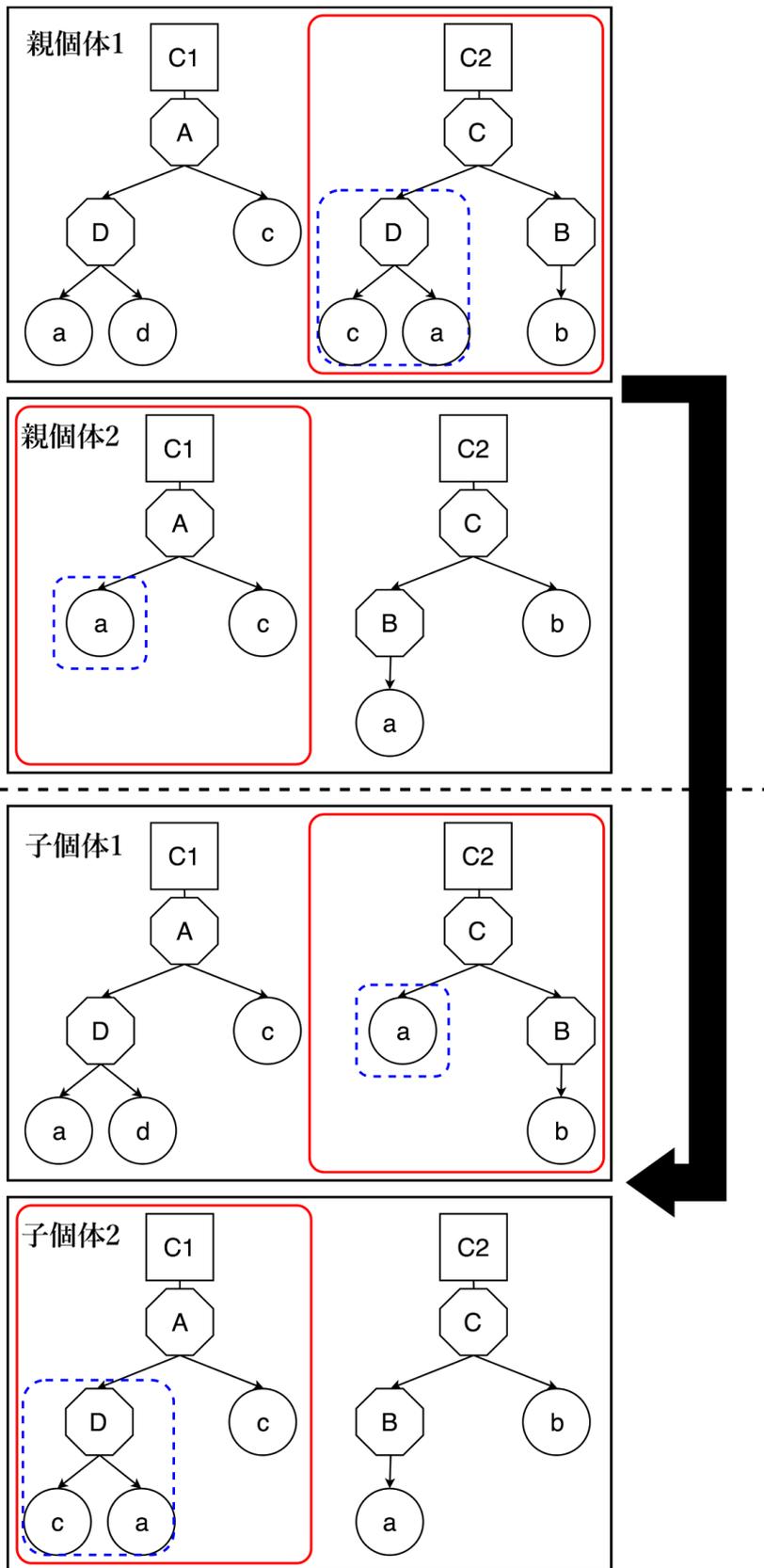


図 3.2 GP_{CN} における交叉の例

図 3.2 では，子個体 1 は親個体 1 のプロセス数 P を引き継ぎ，子個体 2 は親個体 2 のプロセス数 P を引き継ぐ．

● 突然変異

GP_{CN} の突然変異の例を図 3.3, 図 3.4, 図 3.5 に示す． GP_{CN} の突然変異では，交叉と同様に親個体の複数の木構造の中から 1 つの木を選択する．それぞれの図で，選ばれた木を実線の枠で示す．次に突然変異木率によってノードの種類の変更，または選択された部分木の入れ替えを行う．図 3.3 と図 3.4 はノードの種類の変更を行う場合の例である． GP_{CN} における突然変異のノードの種類の変更では，GP と同様の操作が行われる．選択された木の全ノードの中から突然変異を行うノードをランダムに決定し，図 3.3 に破線の枠で示すように選択されたノードが終端ノードであればランダムに異なる終端ノードに変更する．また，図 3.4 のように非終端ノードであればラン

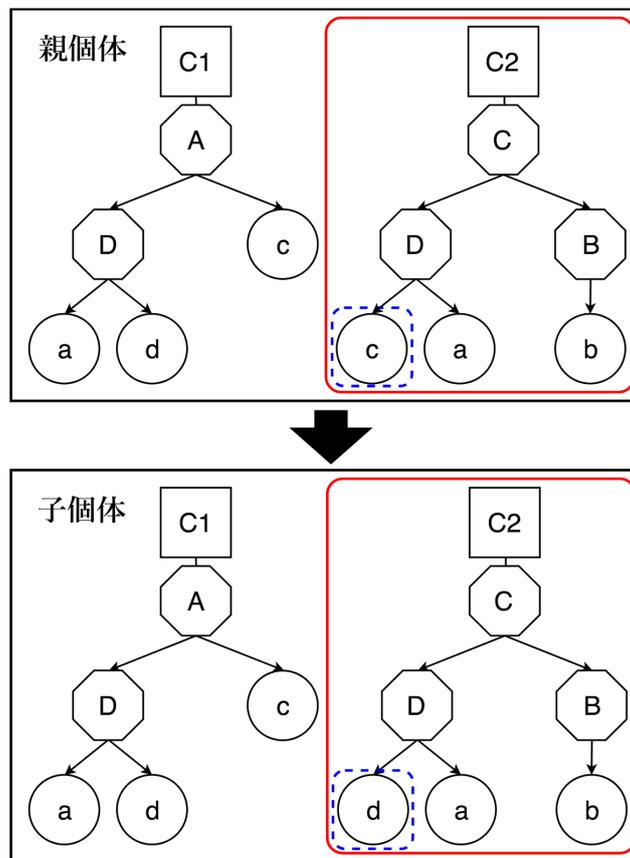


図 3.3 GP_{CN} におけるノードの種類の変更で終端ノードが選択された場合の例

ダムに異なる非終端ノードに変更し，エッジの数が増えれば新たにランダムに生成した部分木を接続し，図 3.4 のようにエッジが減った場合には，余剰なエッジを削除する．

部分木の入れ替えの例を図 3.5 に示す．部分木の入れ替えも GP と同様に，全ノードからランダムにノードを選択し，選択されたノードを根ノードとする部分木とランダムに生成した部分木の入れ替えを行う．図 3.5 では，選択された木を実線の枠，選択されたノードを破線の枠で示している．

また，これらの操作によって生成された子個体の各木には，親個体の各木に設定されているプロセス数 P の値を引き継ぐ．

- 逆位

GP_{CN} の逆位の例を図 3.6 に示す．GP_{CN} の逆位では，図 3.6 の実線の枠で示すようにランダムに逆位を行う木を選択し，その木の全ノードから逆

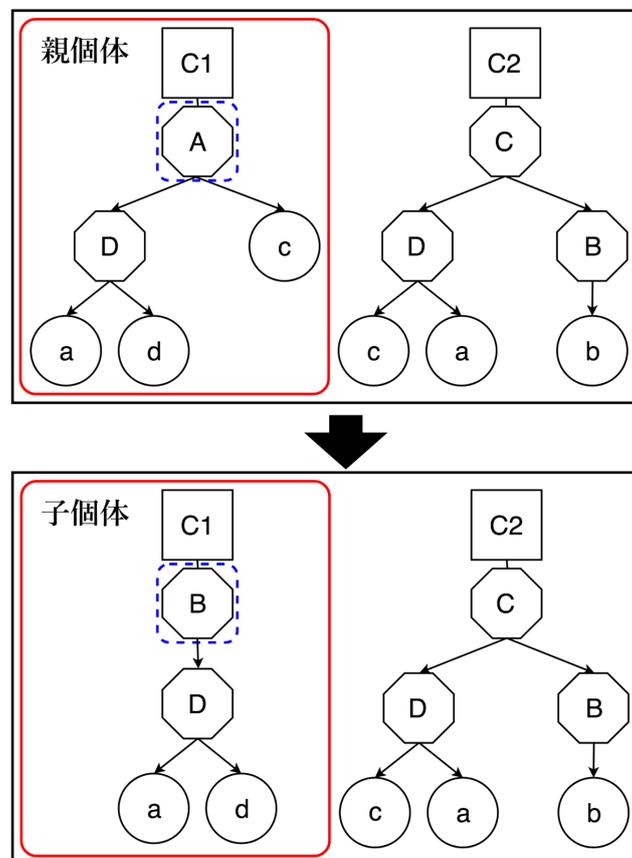


図 3.4 GP_{CN} におけるノードの種類の変更で非終端ノードが選択された場合の例

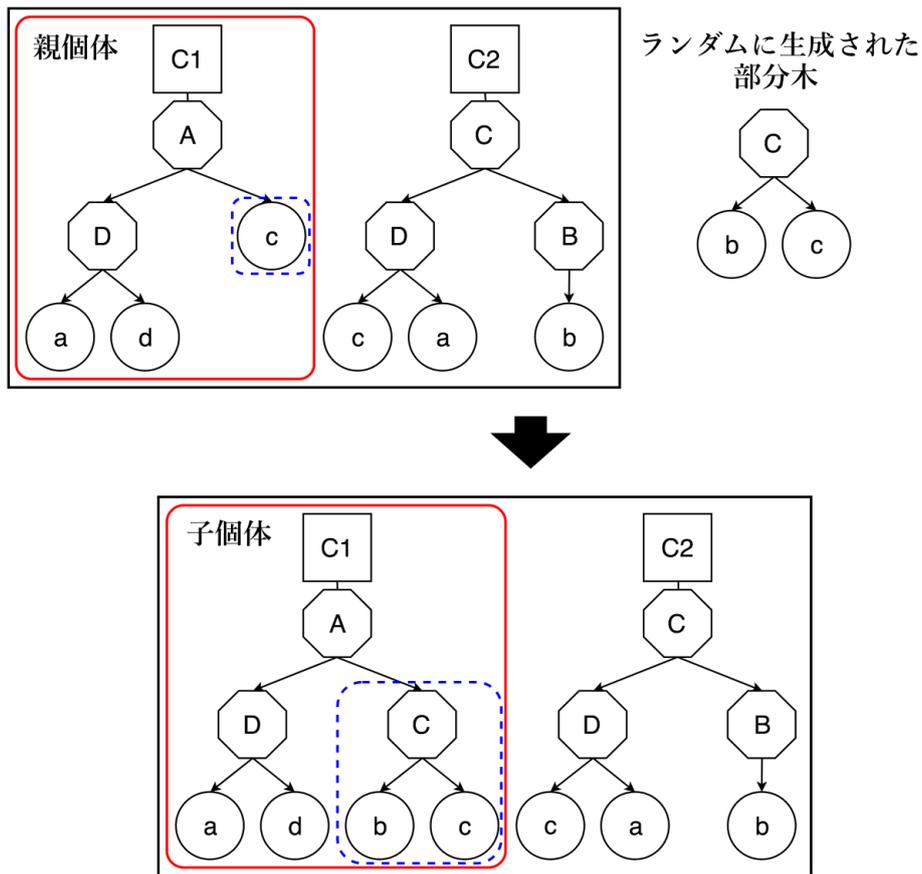
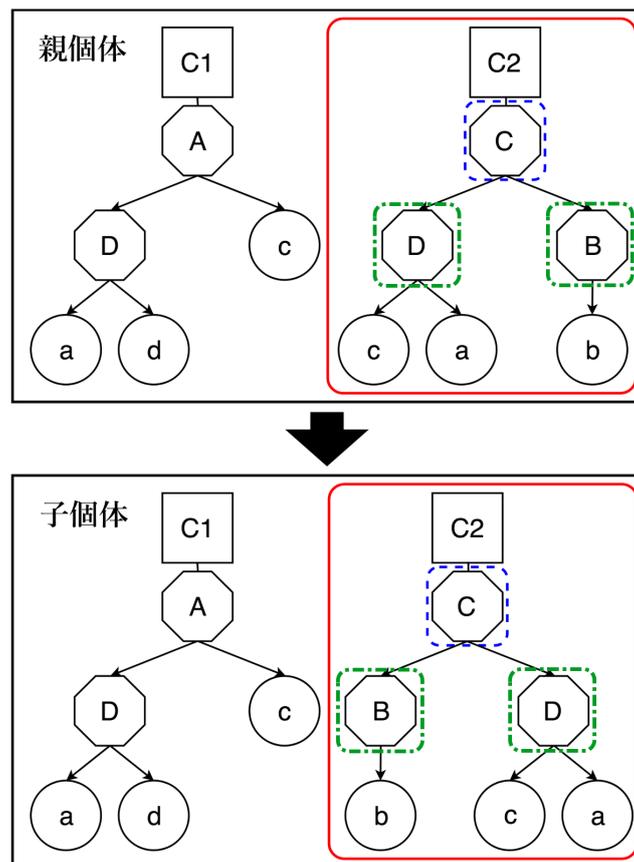


図 3.5 GP_{CN} における部分木との入れ替えの例

位を行う非終端ノードをランダムに1つ選択する．図 3.6 の例では，選択されたノードを破線で示しており，非終端ノード C が選択されている．次に，そのノードを親とする子ノードの中からランダムにノードを2つ選択する．最後に，選ばれたそれぞれのノードを根ノードとする部分木を交換する．ただし，選択された個体の根ノードが終端ノードの場合は，逆位を行わない．

子個体の各木のプロセス数 P については，親個体の各木に設定されているプロセス数 P の値を引き継ぐ．

図 3.6 GP_{CN} における逆位の例

第4章 エージェントシミュレーション (ベンチマーク問題)

4.1 概要

提案手法の性能を評価するため、2つのベンチマーク問題を用いて実験を行った。ごみ拾い問題は、フィールドが狭く、またエージェントの知覚できる情報が多いため比較的難易度の低い問題となっている。一方、Santa Fe Trail問題はフィールドが広く、また知覚できる情報も少ない。さらに、2種類の無条件遷移を含むため難易度の高い問題となっている。

4.2 ごみ拾い問題

ごみ拾い問題とは、決められた行動回数の中でできるだけ多くのごみをごみ収集所に廃棄することを目的とした問題である。ごみ拾い問題のフィールドの例を図4.1に示す。フィールドは縦11×横11（一番外側のマスは壁とする）とし、ごみを2つまで保持できるエージェントが1体、ごみが10個、ごみ収集所が1箇所設置されている。エージェントとごみは環境ごとにランダムな位置に配置され、設置後のエージェントの向きもランダムに決定される。エージェントは1つのマスを占領し、1ステップで前方に1マス動く、左または右に90度回転する、何もしないのいずれかの行動を実行することができる。またエージェントは、ごみのあるマスに到達するとごみを拾うことができ、ごみ収集所のマスに到達するとごみを廃棄することができる。

この問題に用いられるノードの種類を表4.1に示す。種別の非終端は非終端ノード、終端は終端ノードを示す。また、表中の()内は、それぞれのノードの省略記号とエッジの数を示している。非終端ノードは、それぞれの判定結果に従ういくつかのエッジを持っている。“Check the distance from the agent to the collection place”では、ごみ収集所までの距離を3段階(0-4, 5-7, 8-)で判定し、その

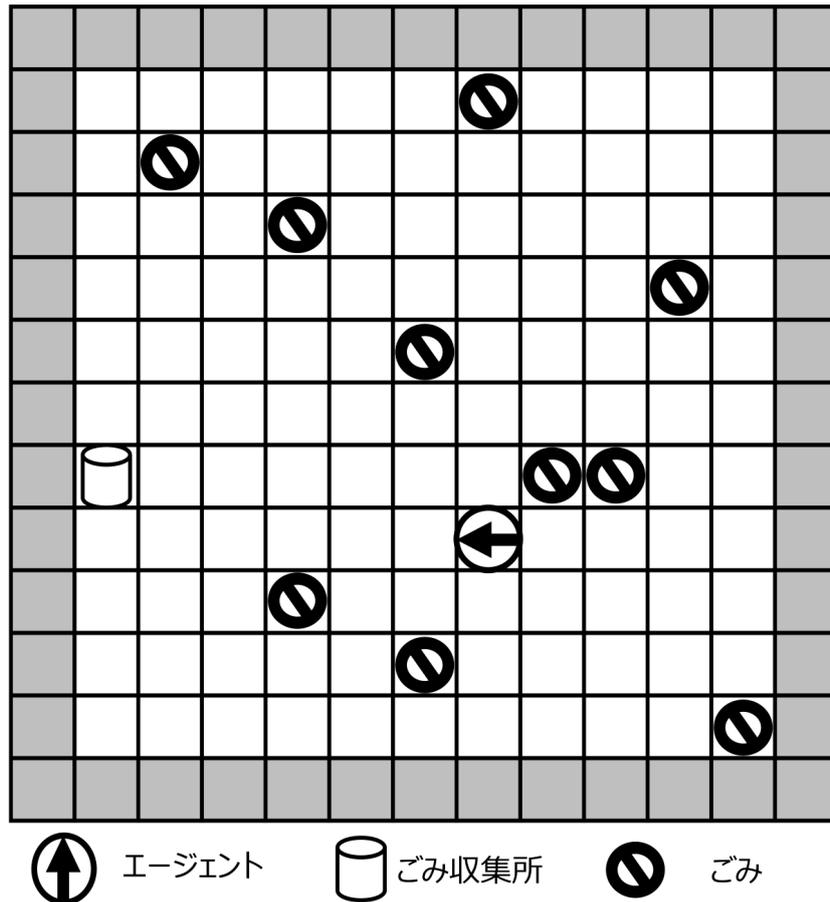


図 4.1 ごみ拾い問題のフィールド例

うち1つがエージェントからごみ収集所までの距離として選択される。したがって、3つのエッジを持つ。“How many trash the agent has”は、エージェントが保持しているごみの数 (0, 1, 2) によって3段階の判定を行うため、3つのエッジを持つ。“Check the direction from the agent to the collection place”では、エージェントから見たごみ収集所の方向を8方向(前, 右前, 左前, 右, 左, 右後, 左後, 後)に分けて判定するため、8つのエッジを持つ。“Check the direction from the agent to the nearest trash”と“Check the direction from the agent to the second nearest trash”では、上記に示したエージェントから見たごみの方向(8方向)と環境内にごみが存在しない(NONE)の9つの判定があるため、9つのエッジを持つ。終端ノードには、エージェントが実行する行動が格納されている。エージェントの行動には、前進(Move forward), 右回転(Turn right), 左回転(Turn left), 行動しない(Stay)の4種類がある。

表 4.1 ごみ拾い問題に用いるノードの種類

種別	記号 (省略記号, エッジ数)	機能
非終端	Check the distance from the agent to the collection place (DisCP, 3)	エージェントからごみ収集所までの距離の判定.
非終端	How many trash the agent has (HMT, 3)	エージェントの保持するごみ数の判定.
非終端	Check the direction from the agent to the collection place (DirCP, 8)	エージェントから見たごみ収集所の方向の判定.
非終端	Check the direction from the agent to the nearest trash (DirNT, 9)	エージェントから見た最も近いごみの方向の判定.
非終端	Check the direction from the agent to the second nearest trash (DirST, 9)	エージェントから見た 2 番目に近いごみの方向の判定.
終端	Move forward (MF, 1)	前方へ 1 マス進む.
終端	Turn right (TR, 1)	右へ 90 度回転.
終端	Turn left (TL, 1)	左へ 90 度回転.
終端	Stay (ST, 1)	何もしない.

本論文で用いたごみ拾い問題は, $TotalSteps$ を 250 としている. エージェントとごみの配置を変えた 10 種類の環境を用意し, それぞれの環境の中で $TotalSteps$ 以内に廃棄できたごみの総数を E_{GC} とする. 環境の種類を i , その環境で捨てられたごみの数を N_i とすると E_{GC} は式 (4.1) により求めることができる.

$$E_{GC} = \sum_{i=1}^{10} N_i \quad (4.1)$$

各環境に配置されているごみの数はそれぞれ 10 個であるため, E_{GC} の最大値は 100 である.

4.3 Santa Fe Trail 問題

Santa Fe Trail 問題とは，フィールドに並べられた餌の道を辿り，できるだけ多くの餌を集めることを目的とした問題である．Santa Fe Trail 問題のフィールドを図 4.2 に示す．フィールドは縦 32 × 横 32 とし，エージェントが 1 体，餌が 89 個決められたマスに配置されている．エージェントは 1 つのマスを占領し，1 ステップで前方に 1 マス動くか，左または右に 90 度回転することができる．またエージェントは，餌のあるマスに到達するとそのマスの餌を回収することができる．

この問題に用いられるノードの種類を表 4.2 に示す．ごみ拾い問題で示したノードの種類を表と同様に，種別の非終端は非終端ノード，終端は終端ノードを示し，表中の () 内はそれぞれのノードの省略記号とエッジの数を示している．

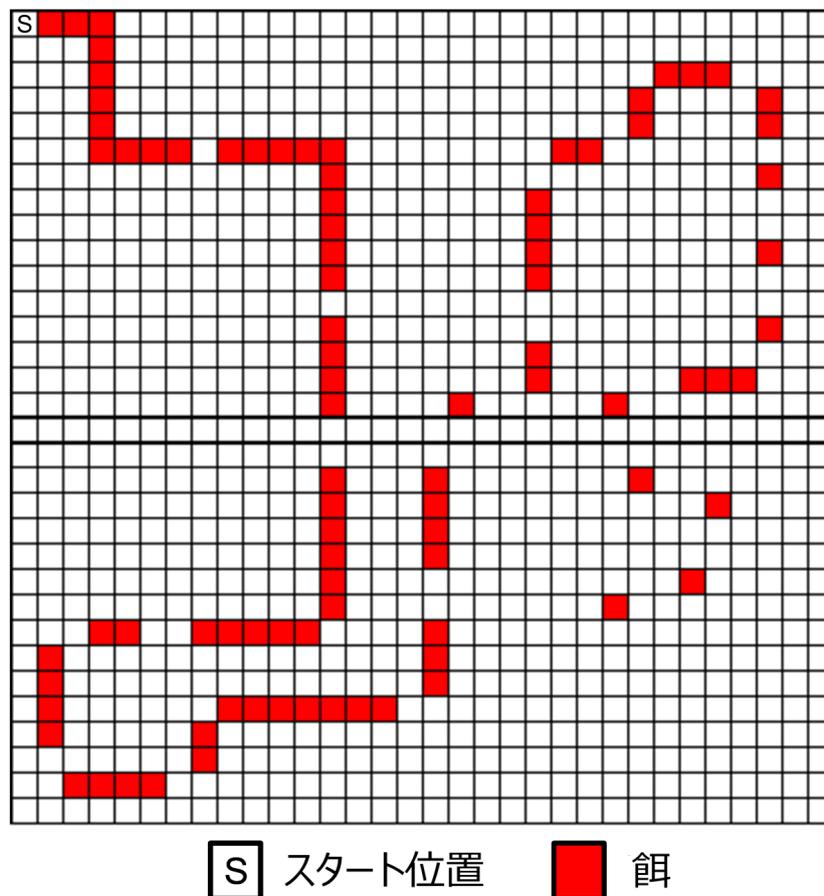


図 4.2 Santa Fe Trail 問題のフィールド

表 4.2 Santa Fe Trail 問題に用いるノードの種類

種別	記号 (省略記号, エッジ数)	機能
非終端	If Food Ahead (FA, 2)	前方に餌が有るかどうかの判定.
非終端	ProgX(x,y) (PX, 2)	x を実行して y を実行する.
非終端	ProgY(x,y,z) (PY, 3)	x を実行して y を実行し, z を実行する.
終端	Move forward (MF, 1)	前方へ 1 マス進む.
終端	Turn right (TR, 1)	右へ 90 度回転.
終端	Turn left (TL, 1)	左へ 90 度回転.

Santa Fe Trail 問題では, 非終端ノードはそれぞれの判定結果に従ういくつかのエッジを持つ条件分岐と, 複数あるエッジの先に接続された行動規則を順番に実行する無条件分岐に分けられる. 条件分岐にあたる “If Food Ahead” は, エージェントの前方のマスに餌があるか, ないかの判定を行う非終端ノードであり, 2つのエッジを持つ. 無条件分岐にあたる非終端ノードは “Prog” という特殊なノードであり, “ProgX” は 2つ, “ProgY” は 3つのエッジを持ち, それぞれのエッジの先に接続されている行動規則を順に実行する. したがって, ごみ拾い問題と異なり, エージェントはルートノードからの 1度の参照で複数回の行動を行う可能性がある. ただし, 行動するたびに行動回数がカウントされるため, 行動回数が *TotalSteps* に達した時点でエージェントシミュレーションは終了し, 以降の行動は実行されない. また, GP_{CN} では各木に繰り返し実行回数を表すプロセス数 P が設定されているが, これはその木が実行した行動の回数としているため, その木の行動回数がプロセス数 P に達した時点で次の木に参照を移す. 終端ノードは, エージェントが実行する行動が定義されている. エージェントの行動には, 前進 (Move forward), 右回転 (Turn right), 左回転 (Turn left) の 3種類がある.

この問題の適合度は集めた餌の総数であり, 最大が 89 である. *TotalSteps* は 400 である.

第5章 GP_{CN} の性能を向上させるための研究

5.1 概要

本章では、複数の木を持つために進化効率が悪くなった GP_{CN} の性能を向上させるための研究を説明する。はじめに、先行研究で提案されている条件付確率を用いた個体生成手法、異文化型島モデル、これらを組み合わせた条件付確率による個体生成手法を導入した島モデルについて述べる。次に、 GP_{CN} の複数の木が識別番号順に参照されるという特徴に基づいて提案した木の順序入れ替え手法について述べる。実験では、これらの手法を GP_{CN} に導入し、ごみ拾い問題と Santa Fe Trail 問題に適用することで性能の比較を行っている。

5.2 先行研究

5.2.1 条件付き確率を用いた個体生成手法

GP の改良手法である GP_{CN} は、1 個体が複数の木構造を持つことができ、多くの行動規則を内包することができる。また、各木は独立しているため役割ごとに行動規則を分散させることができるため、木が肥大化してしまうブロー現象の抑制が期待できる。しかしながら、 GP_{CN} には 1 個体が複数の木を持つために適合度の上がり方が遅いという問題があった。そこで、条件付き確率を用いた個体生成手法が提案されており、 GP_{CN} に導入することで進化の効率が向上することが確認されている [20, 21]。条件付き確率を用いた GP_{CN} (GP_{CN_CP}) の進化の流れを次に示す。

- 1: 初期個体集団を生成。
- 2: 個体の適合度を評価。
- 3: 世代数が実行時に与える最大世代数になっていれば終了、超えてなければ

- 4を実行.
- 4: 優良個体を用いて頻度表を作成.
- 5: 頻度表から条件付き確率表を作成.
- 6: 総個体数の半数を条件付き確率を用いた個体生成により生成.
- 7: 総個体数の残り半数を遺伝操作により生成.
- 8: 2に戻る.

条件付き確率を用いた個体生成では、木の識別番号ごとに頻度表及び条件付き確率表を作成する。これにより、それぞれの識別番号を持つ木の役割を考慮した行動規則の獲得を行うことができる。はじめに、識別番号ごとに各非終端ノードのそれぞれの知覚情報に対する子ノードの出現頻度を数えることで頻度表を作成する。なお、頻度表の作成には現世代の優良個体を用いる。例として図5.1を用いて、識別番号C1の木の非終端ノードAに対する頻度表の作成方法を説明する。図5.1の個体の非終端ノードAからはエッジが2本伸びており、それぞれに記載されている0と1が知覚情報を表している。個体1においては、非終端ノードAが1つ出現しており、知覚情報0に非終端ノードBが、1に終端ノードcが接続されている。また、個体2においては、非終端ノードAが1つ出現しており、知覚情報0に終端ノードaが、1に終端ノードcが接続されている。これらを数え頻度表にしたものが表5.1である。次に、この頻度表から条件付き確率表を作成する。表5.1

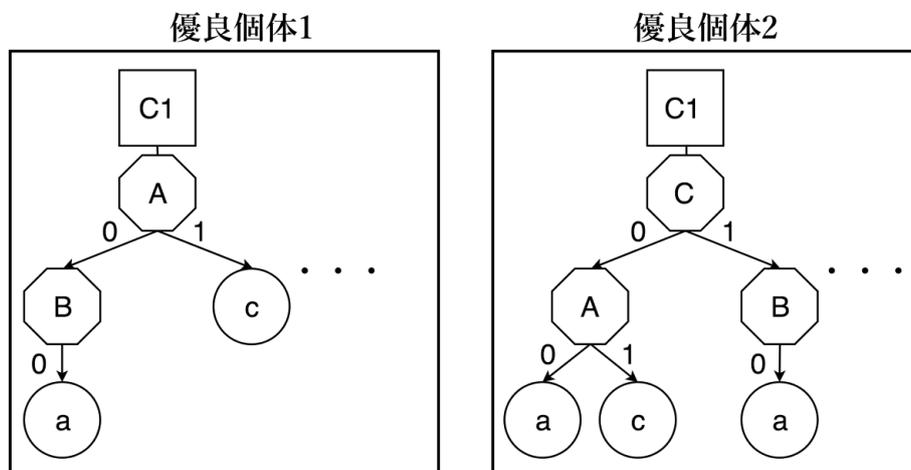


図 5.1 2つの優良個体の例

から作成した条件付き確率表を表 5.2 に示す。

条件付き確率を用いた個体生成では、はじめに木の根におけるノードの出現確率に基づき、乱数を用いてルーレット選択と同じ操作によって根ノードを決定する。次に、親となるノードの条件付き確率に基づき、根ノードと同様にルーレット選択と同じ操作によって子となるノードを決定する。条件付き確率に基づいた子ノードの追加は、現在のノードが終端ノードになるか、木が最大の深さになるまで繰り返される。例として、表 5.2 を用いて説明する。ある個体の識別番号 C1 の木の根ノードが非終端ノード A に決まったとする。次に、識別番号 C1 の非終端ノード A に対する条件付確率表である表 5.2 を用いて各知覚情報の子ノードを決定する。表 5.2 より、知覚情報 0 の子ノードには非終端ノード B か終端ノード

表 5.1 識別番号 C1 の非終端ノード A に対する頻度表

子ノード	知覚情報に対応した 各ノードの頻度	
	0	1
B	1	0
a	1	0
c	0	2
合計	2	2

表 5.2 識別番号 C1 の非終端ノード A に対する条件付き確率表

子ノード	知覚情報に対応した 各ノードの条件確率	
	0	1
B	0.5	0
a	0.5	0
c	0	1.0
合計	1.0	1.0

a が選択される可能性がある。ここで、非終端ノード B が選択された場合、次に識別番号 C1 の非終端ノード B に対する条件付確率表を用いて子ノードの追加を行っていく。また、終端ノード a が選択された場合には知覚情報 0 に対するノードの追加を終了する。次に、知覚情報 1 についても同様に子ノードの追加を行っていく。

また、条件付確率を用いた個体生成によって、新たに生成された個体の各木のプロセス数 P は、個体の初期生成と同様の方法で設定される。

5.2.2 異文化型島モデル

進化的計算手法では、進化の過程で局所解に陥ってしまうことがある。それを改善するために、GP の拡張手法として提案されている異文化型島モデル [22] を GP_{CN} に応用した [21, 23]。異文化型島モデルでは、多様性を維持しながら適合度を上げることを目的しており、母集団を適合度重視の島と多様性重視の島の 2 つのサブ集団に分割し探索を行う。異文化型島モデルを用いた GP_{CN} (GP_{CN,IL}) の進化の流れを図 5.2 に示す。まず、各島において初期集団を生成する。次に個体集団を評価し、島モデルの特徴の 1 つである移住を行う。移住とは、各島の個体の一部を別の島に移動させる操作である。移住は毎世代行われ、適合度重視の島の適合度が低い個体と多様性重視の島の適合度が高い個体を設定した移住数だけ交換する。その後、終了条件を満たしていなければ、それぞれの島の目的に合わせて設定されている個体生成方法を用いて次世代個体を生成する。なお、終了条件は、GP_{CN} と同じく「世代数が実行時に与える最大世代数（ここでは N_g ）になる」とする。

適合度重視の島では、個体集団の適合度の向上に重点を置き進化を行っていく。個体生成には、エリート保存と複数ある遺伝操作うちの交叉のみが用いられる。遺伝操作には他にも突然変異や逆位があるが、これらの操作は優良な行動規則を破壊してしまう可能性があるため適合度重視の島では用いない。また、交叉には GP や GP_{CN} で用いられていた交叉ではなく、深さ依存交叉 [36] を用いる。深さ依存交叉は破壊的な操作が起こりにくく、適合度の高い個体が生成されやすいとされている。図 5.3 に深さ依存交叉の例を示す。図 5.3 では、選択された木を実線の枠、選択された深さを破線、深さ依存交叉が行われたノードを一点鎖線で示

している。深さ依存交叉の操作の流れを次に示す。ここで、木の根の深さを0、最大深さを $MaximumDepth$ 、交換するノードの深さを d とする。

- 1: トーナメント選択により選ばれた2個体から、それぞれ木を1つランダムに選択。
- 2: $[0, MaximumDepth]$ からランダムに d の値を選択。
- 3: それぞれの木の深さ d にあたるノードの中から、ランダムにノードを1ずつ

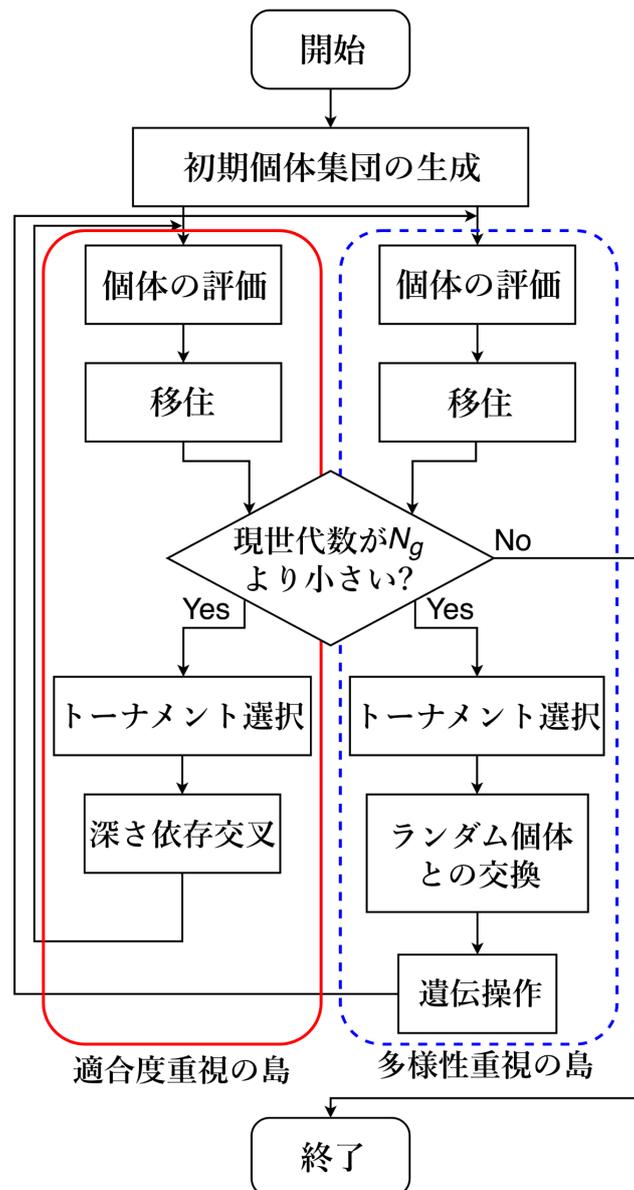


図 5.2 GPCN_IL の進化の流れ

選ぶ。

4: 選ばれたノードを根ノードとする部分木を交換。

深さ依存交叉によって生成された子個体の各木のプロセス数 P は、交叉と同様な方法で設定される。すなわち、より多く同じ行動規則を持つ親個体のプロセス数 P を引き継ぐため、図 5.3 では子個体 1 は親個体 1 のプロセス数 P を引き継ぎ、子個体 2 は親個体 2 のプロセス数 P を引き継ぐ。

多様性重視の島では、多様性の維持に重点を置き、ランダム個体との交換と GP_{CN} で用いられた遺伝操作によって次世代個体を生成する。ランダム個体との交換の例を図 5.4 に示す。ランダム個体との交換とは、一部の個体をランダムに生成した個体と交換する操作である。GP_{CN} では 1 個体が複数の木を持っているため、その中からランダムに 1 つの木を選択し、ランダムに生成された木と選ばれた木を交換する。図 5.4 では、実線で囲まれた識別番号 C1 の木が選択されランダムに生成された木構造と交換されている。ランダム個体との交換によって生成された子個体の各木のプロセス数 P は、交換が行われていない木については、親個体の木に設定されているプロセス数 P の値をそのまま引き継ぐ。交換によって新しくなった木については、木構造を生成する際に、初期個体生成と同様の方法でプロセス数 P が新たに設定されるため、その値を使用する。

5.2.3 条件付き確率を用いた個体生成手法を導入した島モデル

本節では、先行研究で示した異文化型島モデルを拡張した、条件付き確率を用いた個体生成を導入した島モデル型 GP_{CN} (GP_{CN_ILCP}) を説明する [25, 26]。異文化型島モデルでは、適合度重視の島においてエリート保存と深さ依存交叉のみを行っていた。これは、突然変異や逆位を用いると優良な個体の破壊が起こる可能性があり、それを防ぐためである。しかしながら、この 2 つの操作だけでは優良個体の保存と優良個体同士の交叉しか行われないため、優良な行動規則を持つ全く新しい個体が生成されないという問題がある。そこで、適合度重視の島の 2 つの操作に加え、新たに条件付き確率を用いた個体生成手法を導入する。条件付き確率を用いた個体生成では、優良個体を用いて条件付き確率表を作り、その条件付き確率により次世代個体を生成するため優良な行動規則を持つ個体が生成され

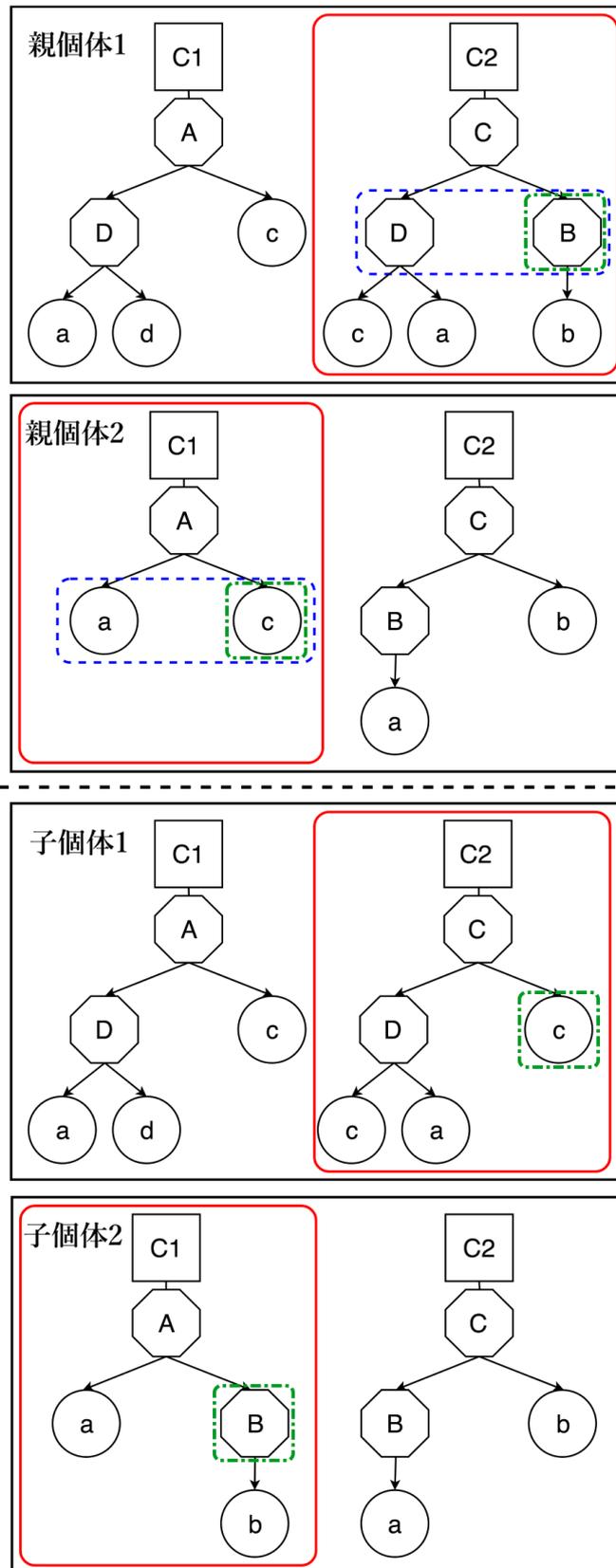


図 5.3 深さ依存交叉の例

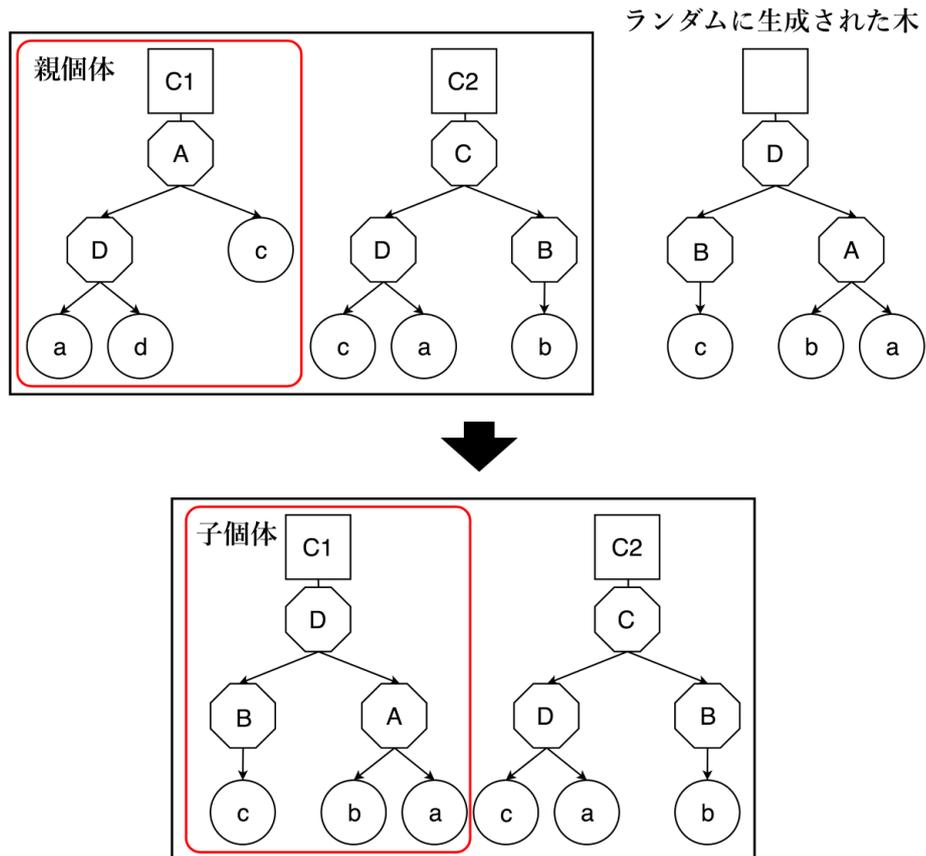


図 5.4 ランダム個体との交換の例

る可能性が高く、適合度重視の島の目的と一致している。条件付き確率表の作成には、適合度重視の島の優良個体を用いる。

さらに、多様性重視の島にも条件付き確率を用いた個体生成手法を導入する。しかしながら、適合度重視の島に導入した条件付き確率を用いた個体生成手法では世代が進むにつれ確率が収束していき、似たような行動規則を持つ個体が生成されやすくなるため、多様性を重視する島の目的を妨害してしまう可能性がある。そこで、適合度重視の島で作成した条件付き確率表を多様性が補えるような形に変換し、その多様性を考慮した条件付き確率を用いて個体生成を行う。これから表 5.3 を用いて、確率表の変換について説明していく。

まず、式 (5.1) を用いて適合度重視の島で作成された条件付き確率表における高い確率を小さく、低い確率を大きくするために 1.0 から各ノードの確率を引く。

$$C'_t(i) = 1.0 - C_t(i) \quad (5.1)$$

ここで、 i は木の識別番号、 $C'_t(i)$ は変換過程にある条件付き確率表、 $C_t(i)$ は適合度重視の島で作成された条件付き確率表を示す。また、確率が0のノードについては優良個体には必要のないノード、確率が1.0のノードについては子ノードとして最適なノードであると考えられるため、式(5.1)を用いた計算は行わない。式(5.1)を用いた結果を表5.4に示す。

次に表5.4を正規化し、多様性を考慮した条件付き確率表に変換する。変換された表を表5.5に示す。多様性重視の島では、この多様性を考慮した条件付き確率率を用いて個体生成を行う。多様性を考慮した条件付き確率表は、適合度重視の島で作成された条件付き確率表では選ばれにくいが優良な行動規則に含まれるノードが選出されやすいように変換されている。また、式(5.1)を用いて変換する

表 5.3 適合度重視の島で作成された条件付き確率表の例

子ノード	知覚情報に対応した 各ノードの条件確率	
	0	1
B	0.6	0
a	0.3	0
c	0.1	1.0
合計	1.0	1.0

表 5.4 変換過程にある条件付き確率表の例

子ノード	知覚情報に対応した 各ノードの条件確率	
	0	1
B	0.4	0
a	0.7	0
c	0.9	1.0
合計	2.0	1.0

表 5.5 多様性を考慮した条件付き確率表の例

子ノード	知覚情報に対応した 各ノードの条件確率	
	0	1
B	0.2	0
a	0.35	0
c	0.45	1.0
合計	1.0	1.0

ことで確率が平滑化され偏った選出が起こりにくくなり、多様性を考慮した個体生成が可能となっている。

5.3 木の順序入れ替え手法

本論文で提案する木の順序入れ替え手法について説明する。GP_{CN}では、1つの個体が複数の木構造を持ち、それぞれの木には参照順を表す識別番号と実行回数を表すプロセス数 P が設定されている。これにより、それぞれの木は行動規則を独立させることができるため、行動の順序やフィールドの状態を考慮した行動規則を獲得することができる。例えば、ごみ拾い問題であれば、ごみを拾うことに特化した木やごみを捨てることに特化した木、Santa Fe Trail問題であれば餌が密集したエリアに特化した木や餌が分散したエリアに特化した木というように、GP_{CN}ではそれぞれの木が役割を持つように進化していく。しかしながら、各木の参照順は識別番号により決まっており、それぞれの木が潜在的に持つ行動規則と実際に割り振られている識別番号に期待される役割が一致しているとは限らない。また、進化の過程で、木の持つ行動規則を大きく変更するには多くの時間を必要としてしまう。そこで、個体内の複数の木の順番を入れ替え、それぞれの木に適する順序に並び変わることを期待し、木の順序を入れ替える手法を提案する[26, 26]。木の順序入れ替え手法の例を図5.5に示しアルゴリズムを以下に示す。

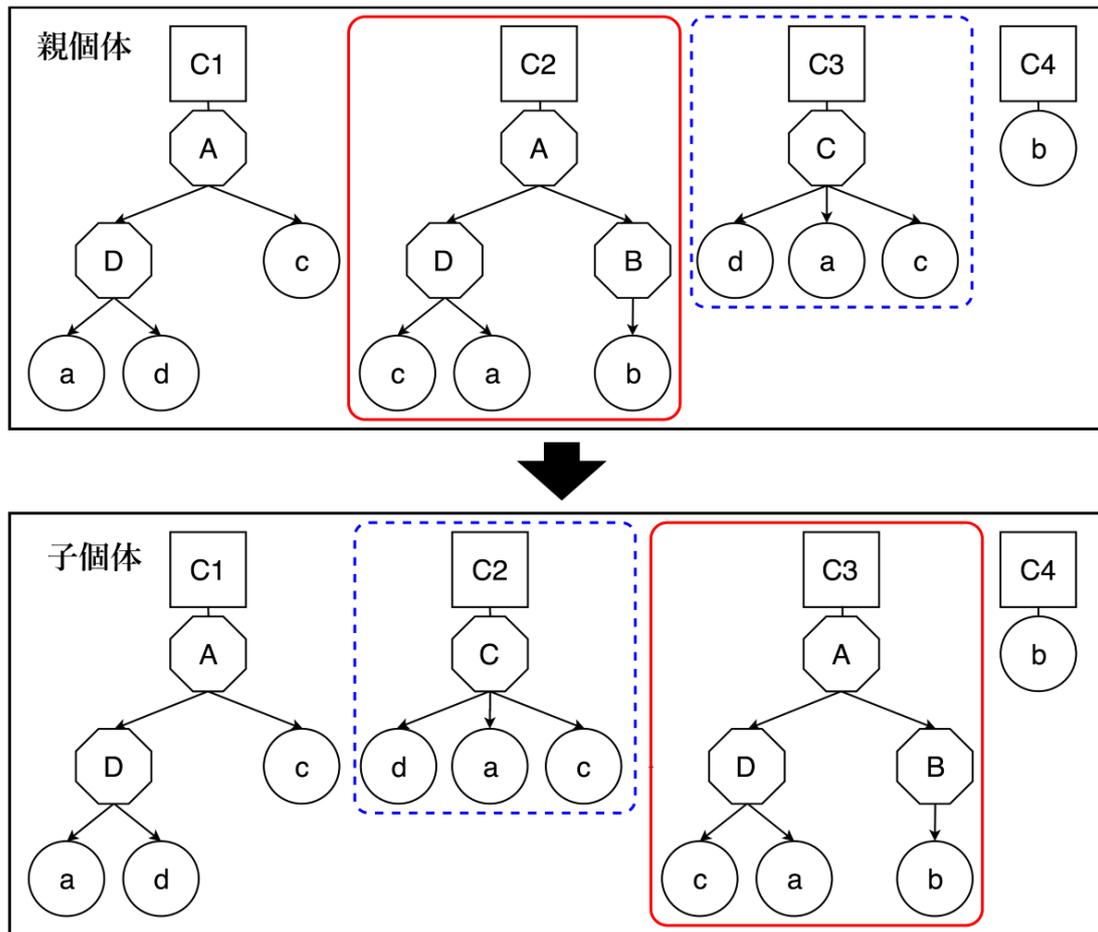


図 5.5 木の順序入れ替え手法の例

- 1: 1つの親個体の複数木から1つの木をランダムに選択する。
- 2: 同じ親個体の複数木の中から1つの木をランダムに選択する。
- 3: 1と2で選んだ木が同じ識別番号を持つ木であれば2へ戻る。
- 4: 2つの木に設定されている識別番号を入れ替える。

図 5.5 では、識別番号 C2 の木と識別番号 C3 の木が入れ替えられている。例では、それぞれの非終端ノードの働きについては詳しく定義していないが、親個体の識別番号 C1 の木と識別番号 C2 の木を見ると、同様のノードが並んでおり似たような行動規則を持つことが予測される。木の順序の入れ替えを行うことで、子個体では同様の行動規則を持つ木が並ぶような無駄になりやすい状況を解消している。

木の順序入れ替えを用いた GP_{CN} (GP_{CN_ET}) では、先に述べた遺伝操作と木の

順序入れ替え手法を用いて次世代個体を生成する。

5.4 評価実験

5.4.1 実験設定

ごみ拾い問題では、個体数を 300、世代数を 1000、個体の木の数を 10、各木のプロセス数 P の初期値を 25 とする。初期個体集団の生成方法は決められた範囲からランダムな木の深さを持つ個体を生成する *grow* 方式を採用し、木の最大深さは 6 としている。その他のパラメータは表 5.6 に載せている値を用いている。Santa Fe Trail 問題では、個体数を 2000、世代数を 1000、個体の木の数を 2、各木のプロ

表 5.6 ごみ拾い問題に用いたパラメータ

	GP, GP _{CN}	GP _{CN_CP}	GP _{CN_ET}	GP _{CN_IL} , GP _{CN_ILCP}	
				適合度重視	多様性重視
突然変異率		0.05		-	0.2
突然変異木率		0.1		-	0.1
交叉率		0.8		1.0	0.8
逆位率		0.2		-	0.1
ランダム個体との 交換確率		-		-	0.2
トーナメントサイズ		2		2	3
エリート保存数		1		1	-
木の順序入れ替え 確率	-	-	0.7	-	-
条件付確率を用いた 個体生成数	-	75	-	50	25
移住数	-	-	-	100	

セス数 P を 200 とする．初期個体集団の生成方法は *ramped half-and-half* 方式を採用し，木の最大深さは 6 としている．*ramped half-and-half* 方式は，個体集団の 50% を *grow* 方式で生成し，残りの 50% として木の最大深さを持つ個体を生成する方法である．その他のパラメータは表 5.7 に載せている値を用いている．また，本論文で用いたパラメータは， GP_{CN} や各手法が導入された GP_{CN} が最高性能となるようにパラメータ調整を行っている．これらは予備実験により得られた値である．なお，パラメータの値が多少変化しても急激な性能の低下は発生しない．

また，本章で用いた GP と GP_{CN} 及び各手法を導入した GP_{CN} の個体の評価関数には，ごみ拾い問題では式 (4.1) の値を，Santa Fe Trail 問題では *TotalSteps* 内に集めた餌の数を用いている．

表 5.7 Santa Fe Trail 問題に用いたパラメータ

	GP, GP_{CN}	GP_{CN_CP}	GP_{CN_ET}	GP_{CN_IL} , GP_{CN_ILCP}	
				適合度重視	多様性重視
突然変異率		0.05		-	0.2
突然変異木率		0.1		-	0.11
交叉率		0.77		1.0	0.98
逆位率		0.2		-	0.2
ランダム個体との 交換確率		-		-	0.1
トーナメントサイズ		2		2	3
エリート保存数		1		300	-
木の順序入れ替え 確率	-	-	0.15	-	-
条件付確率を用いた 個体生成数	-	200	-	175	25
移住数	-	-	-	850	

5.4.2 性能評価

はじめに、提案手法の単体での性能を確認するために、木の順序入れ替え手法を単体で導入した GP_{CN} (GP_{CN_ET}) と、先行研究で提案されている条件付き確率を用いた個体生成手法を導入した GP_{CN} (GP_{CN_CP})、異文化島モデルを導入した GP_{CN} (GP_{CN_IL}) の比較を行う。ごみ拾い問題の 1000 世代の E_{GC} の変化を図 5.6 に示す。図 5.6 より、ごみ拾い問題において複数木構造を持つ GP_{CN} は GP よりも高い性能を示している。また、GP_{CN} と GP は 0 世代の時点で差が開いており、GP は 500 世代程度で進化が緩やかになっているのに対して、GP_{CN} は進化が続いていることがわかる。性能を向上させるための各手法を導入した GP_{CN_CP}, GP_{CN_IL}, GP_{CN_ET} は、なにも導入されていない GP_{CN} よりも高い性能を示しており、各手法を導入することで性能が向上することがわかった。特に、提案手法

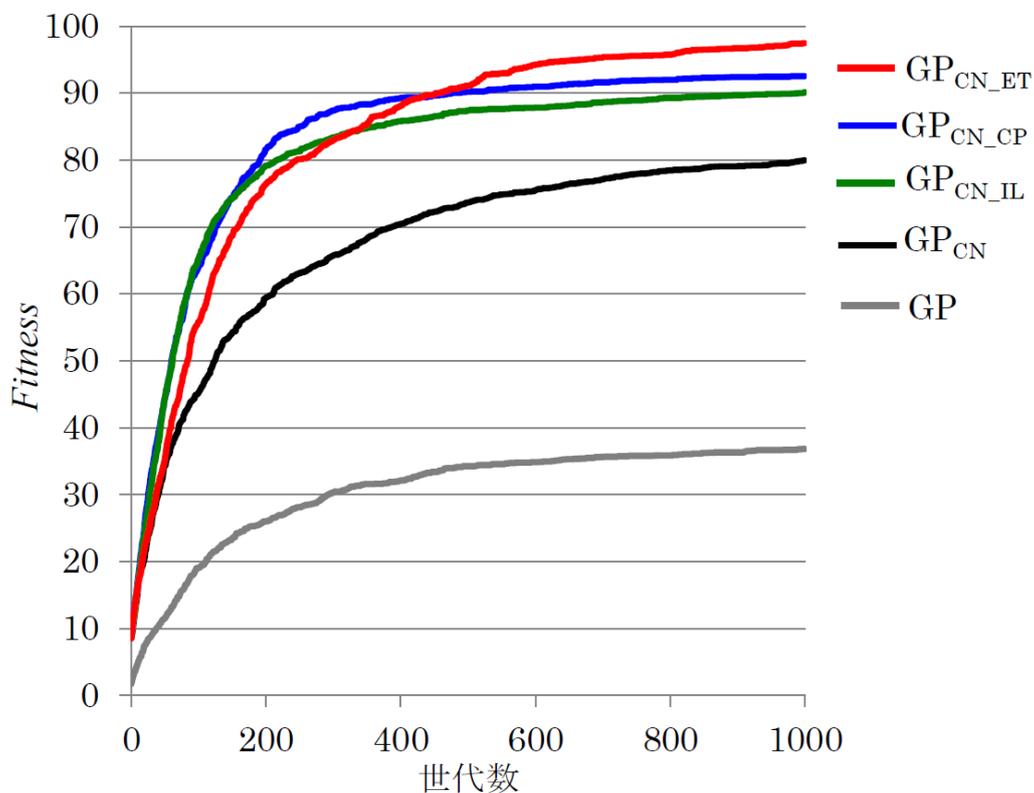


図 5.6 ごみ拾い問題における GP, GP_{CN}, GP_{CN_CP}, GP_{CN_IL}, GP_{CN_ET} の 1000 世代の適合度の変化

である木の順序入れ替え手法は最も高い適合度を示しており、複数の木構造を持つ GP_{CN} において識別番号により管理される木の順序を意図的に入れ替えることが有効であることが示されている。また、 GP_{CN_CP} や GP_{CN_IL} は 500 世代程度で進化が緩やかになっているのに対し、 GP_{CN_ET} は 1000 世代になってもなお進化を続けている。したがって、木の順序を入れ替えることによって、1つの役割にとどまることがなくなるため、多様性の維持にも繋がっていると考えられる。

次に、先行研究において提案されている条件付確率を用いた個体生成を導入した島モデルを用いた GP_{CN} (GP_{CN_ILCP}) との比較を行う。先ほどと同様に、ごみ拾い問題の 1000 世代の適合度の変化を図 5.7 に示す。 GP_{CN_ILCP} は GP_{CN_CP} や GP_{CN_IL} よりも高い適合度を示しており、従来手法の中で最も高い性能であることが確認された。次に、提案手法である GP_{CN_ET} と GP_{CN_ILCP} を比較すると、単体にもかかわらず GP_{CN_ILCP} よりも高い適合度を示していることがわかる。進化速度で見ると GP_{CN_ILCP} の方が速く、200 世代程度で適合度 90 を超えている。し

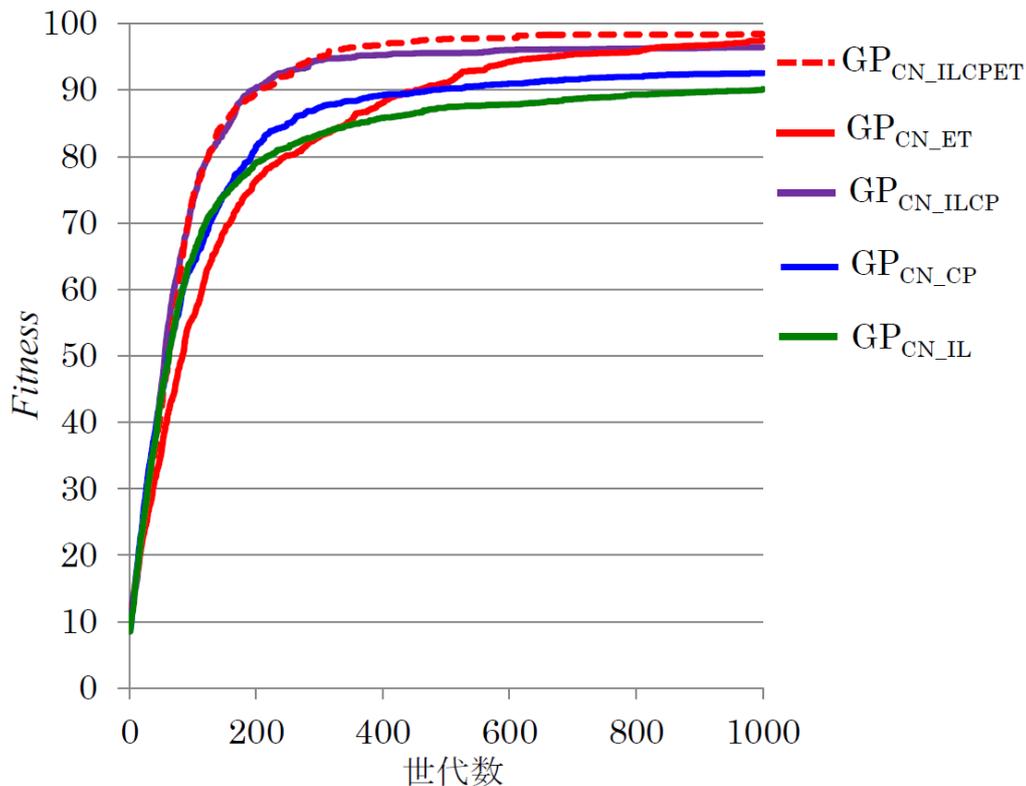


図 5.7 ごみ拾い問題における GP_{CN_CP} , GP_{CN_IL} , GP_{CN_ET} , GP_{CN_ILCP} , GP_{CN_ILCPET} の 1000 世代の適合度の変化

かしながら 400 世代付近で進化が停滞してしまっている。一方、提案手法である GP_{CN_ET} は、着実に進化を続け 800 世代付近で GP_{CN_ILCP} の適合度を超えている。また、 GP_{CN_ILCP} に木の順序入れ替え手法を組み合わせた GP_{CN_ILCPET} は最も高い適合度を示し、進化の速い GP_{CN_ILCP} と局所解に落ちにくい GP_{CN_ET} の良い面が組み合わされたことにより性能が向上したと考えられる。

Santa Fe Trail 問題を用いて提案手法の有効性を確認する。Santa Fe Trail 問題においても提案手法の単体での性能を確認するために、はじめに木の順序入れ替え手法を単体で導入した GP_{CN} (GP_{CN_ET}) と、先行研究で提案されている条件付き確率による個体生成手法を導入した GP_{CN} (GP_{CN_CP}) と異文化島モデルを導入した GP_{CN} (GP_{CN_IL}) の比較を行う。Santa Fe Trail 問題の 1000 世代の適合度の変化を図 5.8 に示す。ごみ拾い問題と同様に、本研究で使用している GP_{CN} は通常の GP よりも高い性能を示している。また、 GP_{CN} よりも各手法を導入した GP_{CN_CP} , GP_{CN_IL} , GP_{CN_ET} のほうが高い性能を示している。しかしながら、提案手法である木の順序入れ替え手法は、 GP_{CN_IL} よりも低い適合度になってしまっている。また、 GP_{CN_CP} も GP_{CN_IL} よりも低い適合度を示し、 GP_{CN_CP} と GP_{CN_ET} がほぼ同じ適合度を示している。 GP_{CN_CP} と GP_{CN_ET} の共通する点は、どちらも GP_{CN} の 1 個体が複数の木を持つ特徴に着目して提案されている手法である点である。Santa Fe Trail 問題では、人手によるパラメータ調整により木の数 2 が最適とされており、各個体の木の数が少ないため識別番号ごとに条件付き確率を作成する GP_{CN_CP} や木の順序を入れ替える GP_{CN_ET} の効果が十分に発揮できなかったのではないかと予想する。

次に、先行研究において提案されている条件付き確率を用いた個体生成を導入した島モデルを用いた GP_{CN} (GP_{CN_ILCP}) との比較を行う。先ほどと同様に、Santa Fe Trail 問題の 1000 世代の適合度の変化を図 5.9 に示す。ごみ拾い問題と同様に、 GP_{CN_CP} や GP_{CN_IL} よりも GP_{CN_ILCP} が高い適合度を示しており、従来手法の中で最も性能が良いことが確認できた。しかしながら、 GP_{CN_ILCP} に提案手法である木の順序入れ替え手法を組み合わせた GP_{CN_ILCPET} は、 GP_{CN_ILCP} よりも性能が低下してしまった。図 5.9 より、 GP_{CN_ILCP} は局所解に陥ることなく 400 世代付近で最大適合度に達している。したがって、木の順序入れ替え手法によって木の順序を入れ替えることで、役割に最適化された木の位置が頻繁に入れ替えられ

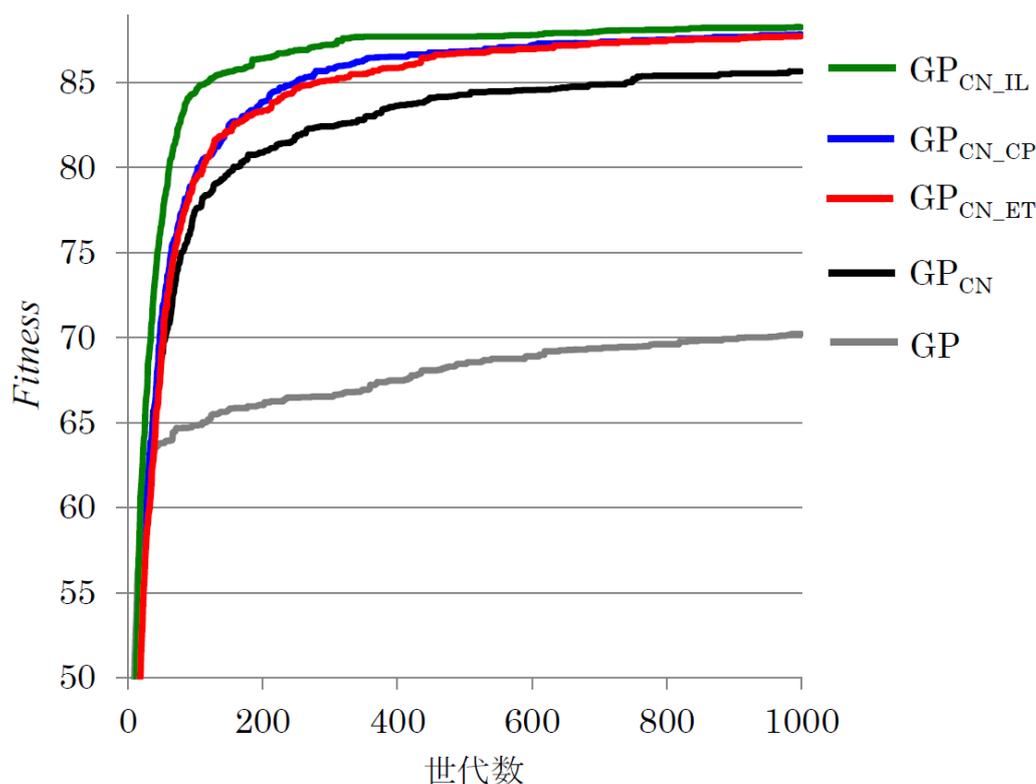


図 5.8 Santa Fe Trail 問題における GP, GP_{CN}, GP_{CN_CP}, GP_{CN_IL}, GP_{CN_ET} の 1000 世代の適合度の変化

性能が低下したと考えられる。

5.5 まとめ

本章では提案している木の順序入れ替え手法の有効性を確認するため、先行研究で提案されている条件付確率を用いた個体生成手法、異文化型島モデル、条件付確率を用いた個体生成を導入した島モデルとの比較を行った。ごみ拾い問題においては、木の順序入れ替え手法は高い適合度を示し、先行研究において最も性能の高かった条件付確率を用いた個体生成を導入した島モデルと組み合わせることで、さらに性能が向上した。一方、Santa Fe Trail 問題では条件付確率を用いた個体生成手法や異文化型島モデルよりも適合度が低くなってしまった。また、条件付確率を用いた個体生成を導入した島モデルと組み合わせても性能が向上しなかった。2つの実験結果の違いとして個体の木の数があげられる。このパラメー

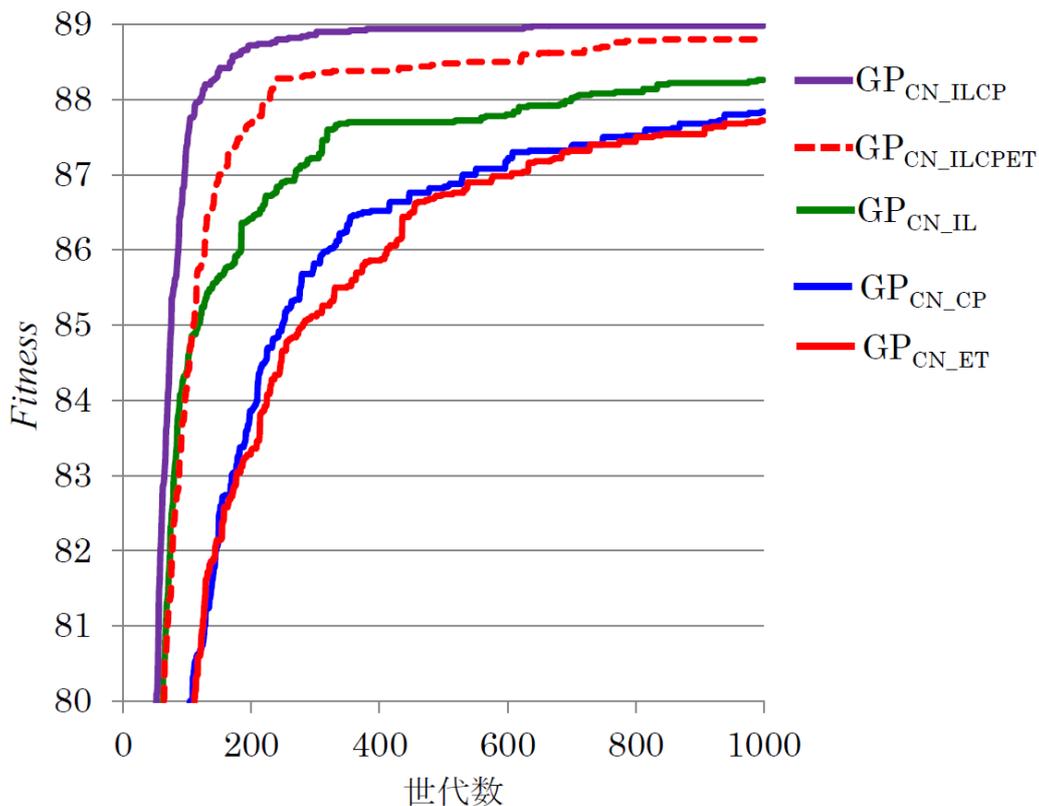


図 5.9 Santa Fe Trail 問題における GP_{CN_CP} , GP_{CN_IL} , GP_{CN_ET} , GP_{CN_ILCP} , GP_{CN_ILCPET} の 1000 世代の適合度の変化

々は、 GP_{CN} の木の数を 2 から 15 まで変化させ実験を行い、各問題で最も適合度が高くなった値を採用している。ごみ拾い問題では 10, Santa Fe Trail 問題では 2 を採用していた。木の順序入れ替え手法は、識別番号によって管理されている木の順番を入れ替える手法であり、Santa Fe Trail 問題では木の数が 2 と少なく設定されているため、効果が発揮されなかったのではないかと考えられる。また、この手法は個体内の木を 2 つ選んで行われるため、木の数を 2 とする場合、頻繁に順序が入れ替えられてしまい、木がうまく進化できなかった可能性がある。よって、複数木構造を持つ GP_{CN} において木の順序を入れ替える手法は有効であるが、木の数が少なく設定されている場合には、進化を妨げてしまう可能性があることが分かった。

第6章 GP_{CN} の性能を向上させるための研究

6.1 概要

本章では、 GP_{CN} の特徴的なパラメータであるプロセス数 P と木の数 M の自動決定について行った研究を説明する。本章は、プロセス数 P の自動決定に関する研究と木の数 M の自動決定に関する研究の2段構成になっている。はじめに、プロセス数 P の自動決定に関する先行研究と提案手法について説明し、ごみ拾い問題と Santa Fe Trail 問題に適用することで性能の比較を行っている。次に、木の数 M を自動決定するための提案手法を説明し、ごみ拾い問題と Santa Fe Trail 問題に適用して、人手で決定したパラメータと自動決定によって得られたパラメータの比較考察を述べている。

6.2 プロセス数 P を自動決定するため手法

6.2.1 プロセス数 P の更新手法

1 個体が複数の木を持つ GP_{CN} では、解決すべき問題によって1個体が持つ木の数 M と各木におけるエージェントの行動回数を表すプロセス数 P の適する値が異なると考えられる。そこで、進化の過程でプロセス数 P を更新していく手法を提案し GP_{CN} に導入する [23, 24]。プロセス数 P の更新手法を導入した GP_{CN} は $GP_{CN}(e)$ と示す。プロセス数 P の更新手法では、選択された個体の中からランダムに1つの木を選び、その木のプロセス数 P の値を更新する。 i を木の識別番号、 $P_c(i)$ を現世代の木 i のプロセス数としたとき、更新された木 i のプロセス数 $P(i)$ は式 (6.1) により求められる。

$$P(i) = P_c(i) + \beta \tag{6.1}$$

式 (6.1) において、 β はプロセス数 P の更新値であり、 $[-\gamma, +\gamma]$ からランダムに

選択される． γ の初期値は16とし，各問題に対して(最大世代数)/10世代ごとに減少させ，進化が進むにつれ1に収束する．本研究で用いる問題はどちらも1000世代であるため，100世代ごとに γ が減少していく．プロセス数 P の更新は，各木で独立して実行される．

6.2.2 提案手法

A. 終端ノードのみの木を淘汰するための評価関数

通常のGPでは，1個体が1つの木で構成されるため，終端ノードのみで構成された木を持つ個体は1つの行動規則しか持たない．そのため，そのような個体は進化の中で排除されていく．例えばごみ拾い問題では，右を向く，前進するといった行動規則1つだけでは，ごみを集めることができない．

しかしながら，複数の木で構成されるGP_{CN}では，個体内の木が全て終端ノードのみの木で構成される場合であっても，複数の行動規則を持つことができる．例えば，先ほどと同じ問題を考える場合，右を向く，前進するといった行動規則を組み合わせることで，進化の序盤であってもフィールドを動き回り，ごみを集めることができる．一方，進化の序盤において，複雑な木構造を持つ個体は知覚情報に対する行動が適切でないため，ほとんどごみを集めることができない．そのため，GP_{CN}において終端ノードのみの木を持つ個体が生き残りやすいという問題を発見した．また，終端ノードのみの木には遺伝操作が適用されないため，進化の妨げになってしまう．そこで本研究では，終端ノードのみの木を淘汰するために新たな適合度を提案する[28, 29]．ごみ拾い問題における集めたごみの数のようなエージェントの評価値を E とすると，個体 i の適合度は式(6.2)のように表される．

$$Fitness(i) = E(i) - \left(\alpha \times \frac{M_{SingleNode}(i)}{M(i)} \right) \quad (6.2)$$

ここで， α は終端ノードのみの木に対する重み， $M_{SingleNode}(i)$ は個体 i に含まれる終端ノードのみの木の数， $M(i)$ は個体 i の木の数である．また，計算の結果，適合度が0より小さくなる場合は0に変更する．本研究では，ごみ拾い問題とSanta Fe Trail問題においてGP_{CN}の $E(i)$ が最大になるように α を調整し，ごみ拾い問題では10.0，Santa Fe Trail問題では13.0と設定した．

B. プロセス数 P の継承手法

先行研究 [23, 24] において，進化の中で木に適したプロセス数 P の値を獲得するためにプロセス数 P の更新手法が提案されている．この手法は，個体内の木のプロセス数 P の値をランダムに決められた値で加算もしくは減算することによって徐々に変更していく．しかしながら，プロセス数 P の取りうる範囲が広くなると効率が悪くなるという問題があった．そこで，我々は新たなプロセス数 P の決定手法としてプロセス数 P の継承手法を提案する [30, 31, 32]．プロセス数 P の継承手法を適用した GP_{CN} は GP_{CN_I} と表す．

プロセス数 P の継承手法のアルゴリズムを次に示す．

- 1: 親個体 1 の中からランダムに木を選択．
- 2: 親個体 2 の中からランダムに木を選択．
- 3: 親個体 1 と親個体 2 から選択された木 A と木 B に対して交叉を実行し，子個体 1 と子個体 2 を生成．
- 4: 親個体の適合度を比較し，大きい方の親個体から選ばれた木に設定されているプロセス数 P の値を交叉によって生成された木に引き継ぐ．

プロセス数 P の継承手法の例を図 6.1 に示す．図 6.1 では，実線で囲われた木が交叉のために選択された木，破線で囲われたノードが交叉のために選択されたノードである．また，識別番号の横に表示している P と P' がその木に設定されているプロセス数 P を示す．図では親個体 1 と親個体 2 が交叉され，優良個体である親個体 1 の木 1 のプロセス数 P が子個体に引き継がれている．

この手法を用いることで，個体集団中の子個体は優良な個体の木に設定されているプロセス数 P の値を引き継ぐことができるため，個体間に優良個体のプロセス数 P の値を広めることができる．

C. プロセス数 P の突然変異

プロセス数 P の継承手法では，交叉によって生成された木に，優良個体のプロセス数 P の値を引き継ぐことによって問題に適した値にプロセス数 P が収束すると考えられる．しかしながら，この手法では 1 度の操作で 2 つの個体に同じプ

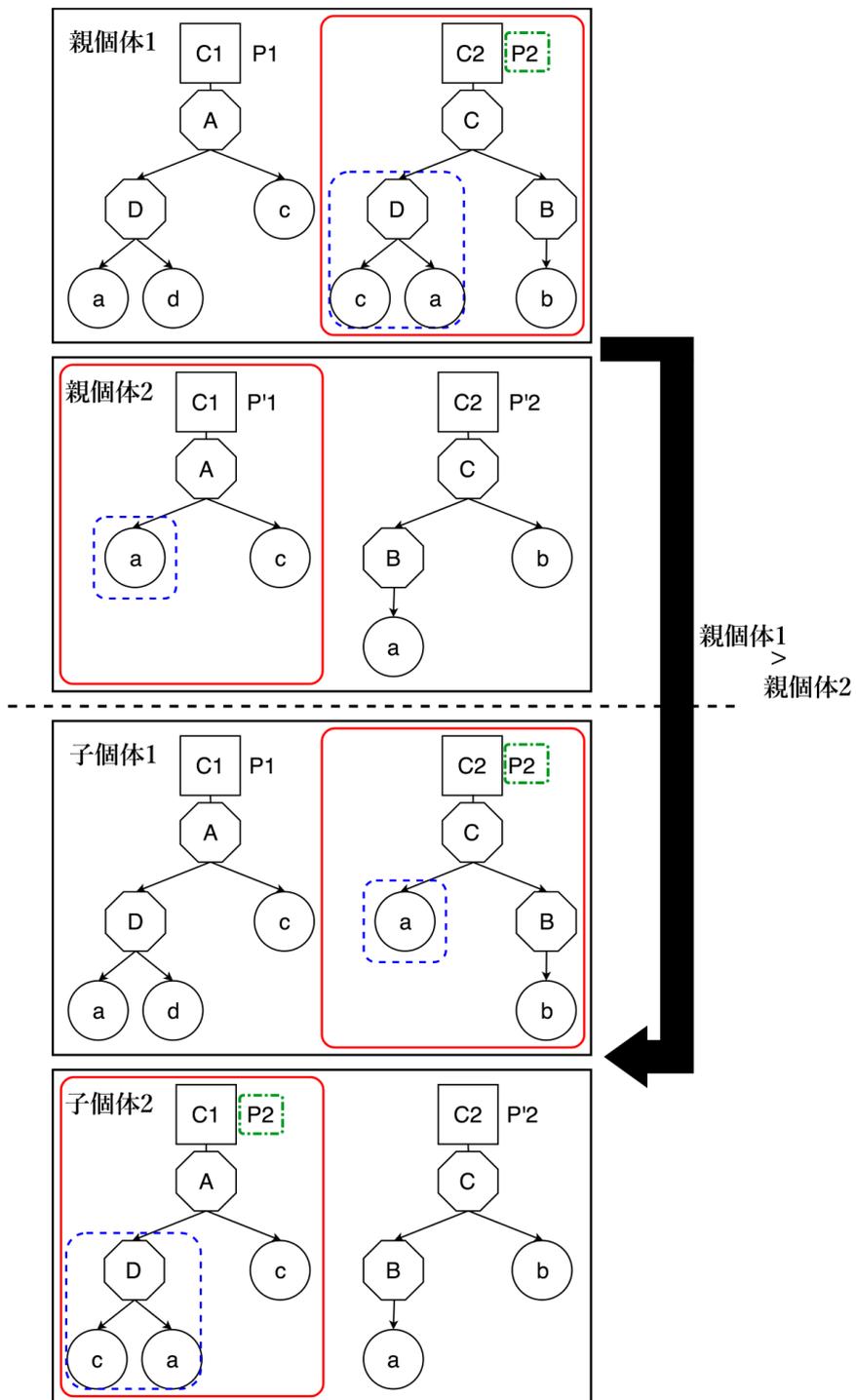


図 6.1 プロセス数 P の継承手法の例

プロセス数 P の値が引き継がれるため、プロセス数 P の多様性が急速に失われる可能性がある。そこで、それを防ぐためにプロセス数 P の突然変異を提案する。プロセス数 P の突然変異を適用した GP_{CN} は GP_{CN_Mp} と表す。

3章の“3.2 GP_{CN} の遺伝操作”の中で示すように、突然変異にはノードの種類の変更とランダムに生成された部分木との交換の2種類がある。その中でもランダムに生成された部分木との交換は、木に大きな変化をもたらす。プロセス数 P の突然変異も木に与える影響が大きいと考えられるため、ランダムに生成された部分木との交換が行われた木に適用する。図 6.2 にプロセス数 P の突然変異の例を示す。図 6.2 では、突然変異のために選択された木を実線の枠で、選択されたノードを破線の枠で示している。ランダムな部分木を生成する際に、 $[1, TotalSteps/M]$ からランダムに値を選択することによってランダムなプロセス数 P の値を生成し、その値をランダムな部分木が接続された木のプロセス数 P に設定する。

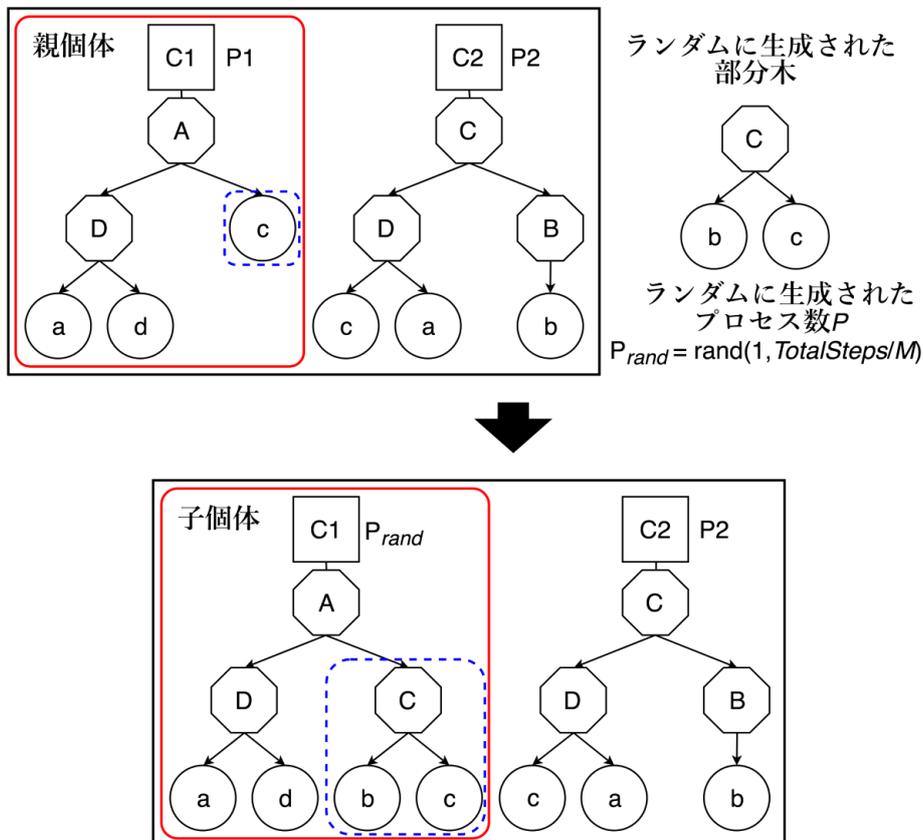


図 6.2 プロセス数 P の突然変異の例

6.2.3 終端ノードのみの木を淘汰するための評価関数の変更

終端ノードのみの木を淘汰するための評価関数による影響を確認するために、適合度のみの変更を行った GP_{CN} により実験を行う。実験では、終端ノードのみの木を淘汰するための適合度を導入した GP_{CN} (GP_{CN}(ETS)) と、各問題で設定されている E の値を適合度に適用した GP_{CN} (GP_{CN}(E_value)) を用いて、両者の性能の比較を行う。性能の比較には、各 GP_{CN} に設定されている適合度ではなく、各問題で設定されている E の値を用いる。

実験設定は、ごみ拾い問題では、個体数を 300, 世代数を 1000, 個体の木の数を 10, 初期集団の生成方法を *grow* 方式とし, Santa Fe Trail 問題では, 個体数を 2000, 世代数を 1000, 個体の木の数を 2, 初期集団の生成方法を *ramped half-and-half* 方式とする。また, 各問題における木の深さは 6 とし, 各木のプロセス数 P の初期値は固定ではなく, $[1, TotalSteps/M]$ の範囲からランダムに選択するように変更した。その他のパラメータについては, プロセス数 P の設定方法を変更したことに伴い, 新たにパラメータ調整を行った。パラメータ調整により新たに決定された GP_{CN} のパラメータを表 6.1 に示す。また, 終端ノードのみの木を淘汰するための適合度に関する係数 α は, ごみ拾い問題では 10.0, Santa Fe Trail 問題では 13.0 とした。これらのパラメータは, 各問題において GP_{CN}(E_value) と GP_{CN}(ETS) の

表 6.1 終端ノードのみの木を淘汰するための評価関数の変更において 2 つの問題に共通するパラメータ

	GP _{CN} (E_value) GP _{CN} (ETS)
突然変異率	0.05
突然変異木率	0.1
交叉率	0.8
逆位率	0.2
トーナメントサイズ	2
エリート保存数	1

性能が最大となるように調整している。

ごみ拾い問題における $GP_{CN}(E_value)$ と $GP_{CN}(ETS)$ の 1000 世代の E_{GC} の変化を図 6.3 に示す。図 6.3 から、 $GP_{CN}(E_value)$ よりも $GP_{CN}(ETS)$ の方が高い E_{GC} を示していることがわかる。次に、実際に適合度を変更したことにより、個体集団中の終端ノードのみの木がどのように変化していったかを比較する。図 6.4 にごみ拾い問題における $GP_{CN}(E_value)$ と $GP_{CN}(ETS)$ の終端ノードのみの木の変化を示す。ごみ拾い問題では、個体数が 300、各個体の木の数が 10 であるため、個体集団中の木の数は 3000 個となる。図 6.4 から、初期個体集団の生成において、約 1500 個の終端ノードのみの木が発生していることがわかる。また、 $GP_{CN}(E_value)$ では、序盤に終端ノードのみの木の数が上昇してしまっている。その後、減少傾向を示しているが傾きが小さく、1000 世代においても約 700 個の終端ノードのみの木が生き残ってしまっている。一方、終端ノードのみの木を淘汰するための評価関数に変更した $GP_{CN}(ETS)$ では、序盤から終端ノードのみの木が減少傾向にあり、200 世代付近で約 200 個まで終端ノードのみの木の数を減らしている。最終世代である 1000 世代では、 $GP_{CN}(ETS)$ は終端ノードのみの木の数を約 70 まで減らしており、終端ノードのみの木を淘汰するための評価関数によって、終端ノード

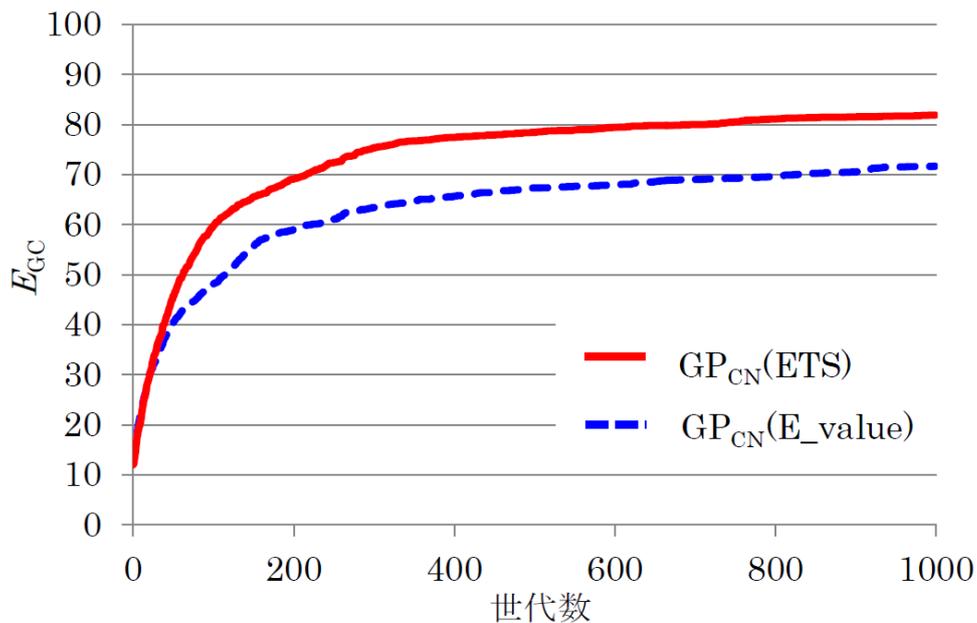


図 6.3 ごみ拾い問題における $GP_{CN}(E_value)$, $GP_{CN}(ETS)$ の 1000 世代の E_{GC} の変化

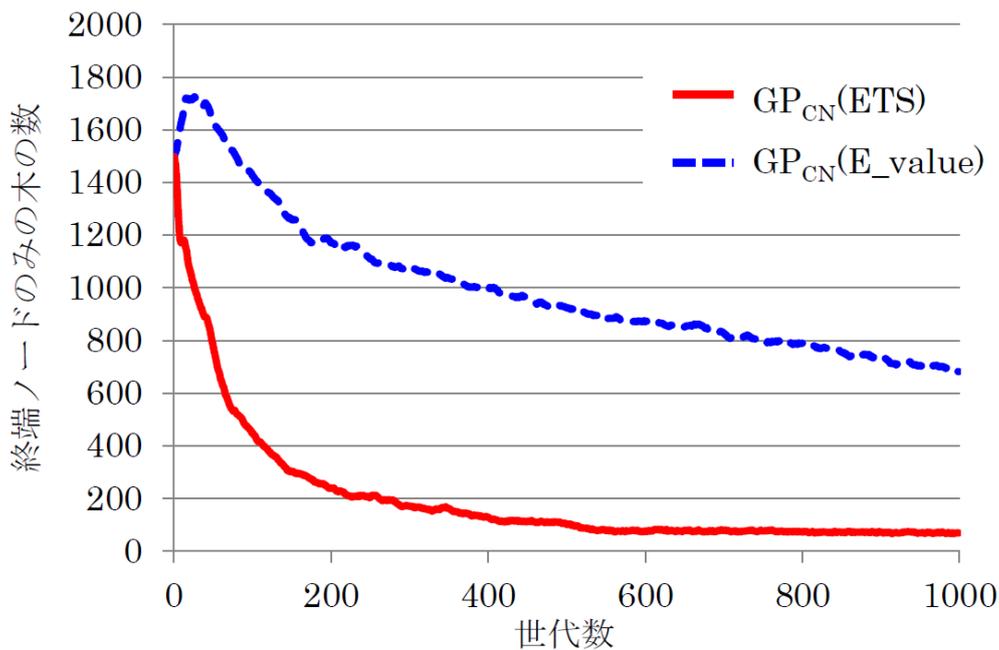


図 6.4 ごみ拾い問題における GP_{CN}(E_value), GP_{CN}(ETS) の 1000 世代の終端ノードのみの木の数の変化

のみの木が大幅に減少することが確認できた。

Santa Fe Trail 問題における GP_{CN}(E_value) と GP_{CN}(ETS) の 1000 世代の E_{SFT} の変化を図 6.5 に示す。図 6.5 から、GP_{CN}(E_value) よりも GP_{CN}(ETS) の方が高い E_{SFT} を示していることがわかる。ごみ拾い問題と同様に、実際に適合度を変更したことにより、個体集団中の終端ノードのみの木がどのように変化していったかを比較する。図 6.6 に Santa Fe Trail 問題における GP_{CN}(E_value) と GP_{CN}(ETS) の終端ノードのみの木の変化を示す。Santa Fe Trail 問題では、個体数が 2000、各個体の木の数が 2 であるため、個体集団中の木の数は 4000 個となる。図 6.6 から、初期個体集団の生成において、約 1000 個の終端ノードのみの木が発生していることがわかる。また、ごみ拾い問題よりも個体数が多いにもかかわらず終端ノードのみの木の数が少ないのは、初期生成に用いた方式の違いによるものである。Santa Fe Trail 問題では、GP_{CN}(E_value) と GP_{CN}(ETS) のどちらにおいても終端ノードのみの木の数は急峻な減少傾向を示している。これは、木の数が少ないことによって、ごみ拾い問題の時よりも終端ノードのみの木が淘汰されやすいことを示している。しかしながら、それでも GP_{CN}(E_value) では 1000 世代で、終端ノード

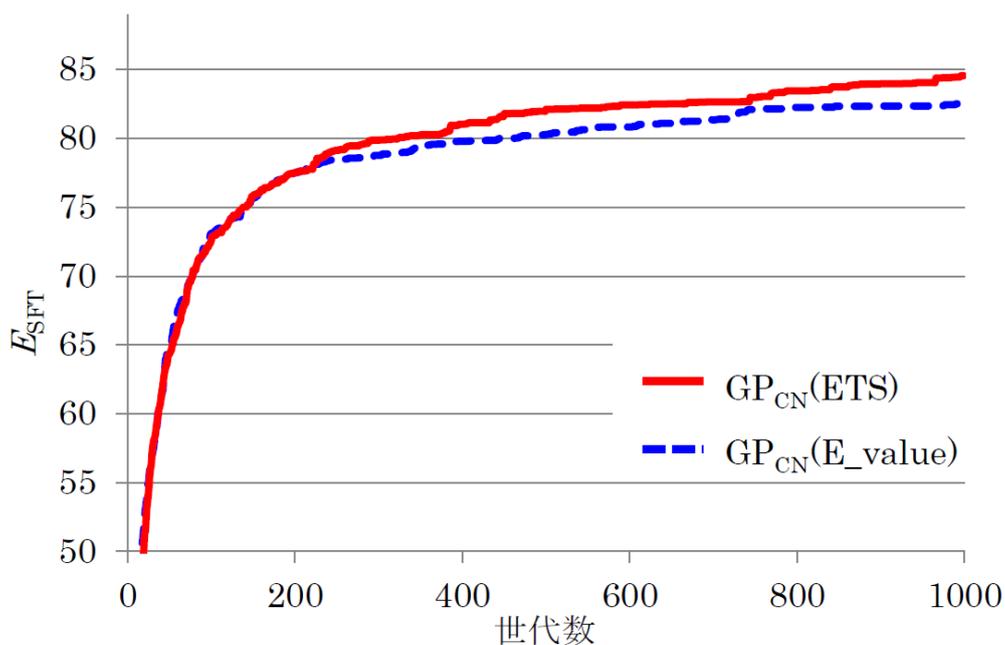


図 6.5 Santa Fe Trail 問題における $GP_{CN}(E_value)$, $GP_{CN}(ETS)$ の 1000 世代の E_{SFT} の変化

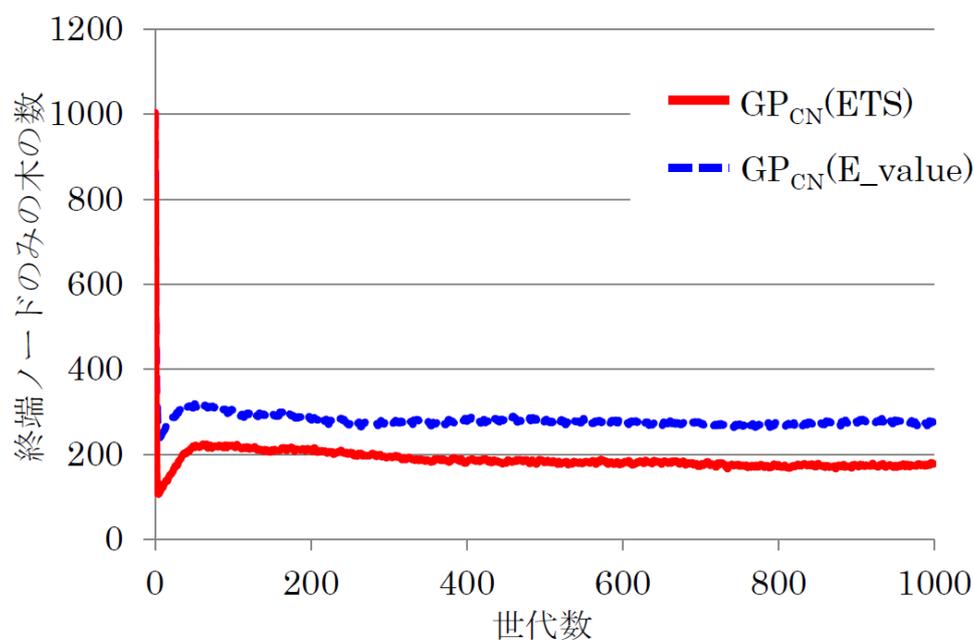


図 6.6 Santa Fe Trail 問題における $GP_{CN}(E_value)$, $GP_{CN}(ETS)$ の 1000 世代の終端ノードのみの木の数の変化

のみの木の数を約300個までしか減らすことができていない。一方、GP_{CN}(ETS)では約200個まで終端ノードのみの木の数を減らすことができており、終端ノードのみの木を淘汰するための評価関数の変更の有効性を確認できた。

したがって、次の節の“プロセス数 P を自動決定する手法の実験”では、終端のノードのみの木を淘汰するための評価関数を個体の適合度に用いる。

6.2.4 プロセス数 P を自動決定する手法の実験

提案手法であるプロセス数 P の継承手法及びプロセス数 P の突然変異の有効性を確認するために、これらの手法を導入したGP_{CN_I}とGP_{CN_{Mp}}をごみ拾い問題とSanta Fe Trail問題に適用し実験を行う。比較手法として、GPとGP_{CN}、先行研究で提案されているプロセス数 P の更新手法を導入したGP_{CN}(GP_{CN}(e))を用いた。また、手法の組み合わせによる性能の変化を確認するために、2つの提案手法を組み合わせたGP_{CN_{IMp}}とそれぞれの提案手法にプロセス数 P の更新手法を組み合わせたGP_{CN_I}(e)とGP_{CN_{Mp}}(e)を実装し、ベンチマーク問題に適用した。

実験設定は、ごみ拾い問題では、個体数を300、世代数を1000、個体の木の数を10、初期集団の生成方法を*grow*方式とし、Santa Fe Trail問題では、個体数を2000、世代数を1000、個体の木の数を2、初期集団の生成方法を*ramped half-and-half*方式とする。また、各問題における木の深さは6とし、各木のプロセス数 P の初期値はの範囲からランダムに選択するように変更している。各問題の適合度には、終端ノードのみの木を淘汰するための評価関数を用いた。その他パラメータについては、各手法を導入したGP_{CN}ごとに性能が最も良くなるようにパラメータ調整を行った。パラメータ調整により新たに決定されたGP_{CN}のパラメータを表6.2、表6.3に示す。また、終端ノードのみの木を淘汰するための適合度に関する係数 α は、ごみ拾い問題では10.0、Santa Fe Trail問題では13.0とした。

ごみ拾い問題における1000世代の E_{GC} の変化を図6.7に示す。図6.7では、各GP_{CN}を E_{GC} の高い順に凡例に示している。図6.7より、プロセス数 P を自動決定するための手法を導入していないGP_{CN}よりも、プロセス数 P の更新手法を導入したGP_{CN}(e)、プロセス数 P の継承手法を導入したGP_{CN_I}やプロセス数 P の突然変異を導入したGP_{CN_{Mp}}が高い E_{GC} を示している。したがって、プロセス数 P を問題に合わせて適切に調整することは、GP_{CN}の性能の向上につながる

表 6.2 ごみ拾い問題に用いたパラメータ

	GP	GP_{CN} $GP_{CN}(e)$	GP_{CN_I} , $GP_{CN_I}(e)$	$GP_{CN_{Mp}}$, $GP_{CN_{Mp}}(e)$	$GP_{CN_{IMp}}$, $GP_{CN_{IMp}}(e)$
突然変異率	0.05				
突然変異木率	0.1			0.05	
交叉率	0.8		0.95	0.8	0.95
逆位率	0.2				
トーナメントサイズ	2				
エリート保存数	1				
プロセス数 P の 更新確率	-	0.05			-

表 6.3 Santa Fe Trail 問題に用いたパラメータ

	GP	GP_{CN} $GP_{CN}(e)$	GP_{CN_I} , $GP_{CN_I}(e)$	$GP_{CN_{Mp}}$, $GP_{CN_{Mp}}(e)$	$GP_{CN_{IMp}}$, $GP_{CN_{IMp}}(e)$
突然変異率	0.05				
突然変異木率	0.1			0.25	
交叉率	0.8		0.85	0.8	0.85
逆位率	0.2				
トーナメントサイズ	2				
エリート保存数	1				
プロセス数 P の 更新確率	-	0.05			-

ことがわかる。また、先行研究である $GP_{CN}(e)$ よりも提案手法である GP_{CN_I} と $GP_{CN_{Mp}}$ の方が高い E_{GC} を示しており、提案手法はプロセス数 P を自動決定するための手法として有効であることが確認できた。多様性を維持するためにプロセス数 P の更新手法を組み合わせた $GP_{CN_I}(e)$ やプロセス数 P の突然変異を組み合わせた $GP_{CN_{IMp}}$ は、 GP_{CN_I} よりも性能が低くなってしまった。これは、プロセス数 P が GP_{CN} において重要なパラメータであり、この値によって木の役割が変化してしまうため、早期に収束してしまう方が良いこと示している。

次に、Santa Fe Trail 問題における 1000 世代の E_{SFT} の変化を図 6.8 に示す。図 6.8 では、ごみ拾い問題と同様に、各 GP_{CN} を E_{SFT} の高い順に凡例に示している。図 6.8 より、Santa Fe Trail 問題においても、 GP_{CN} よりも $GP_{CN}(e)$ 、 GP_{CN_I} や $GP_{CN_{Mp}}$ の方が高い E_{SFT} を示しており、プロセス数 P の自動決定手法が有効であることを示している。また、プロセス数 P の更新手法とプロセス数 P の継承手

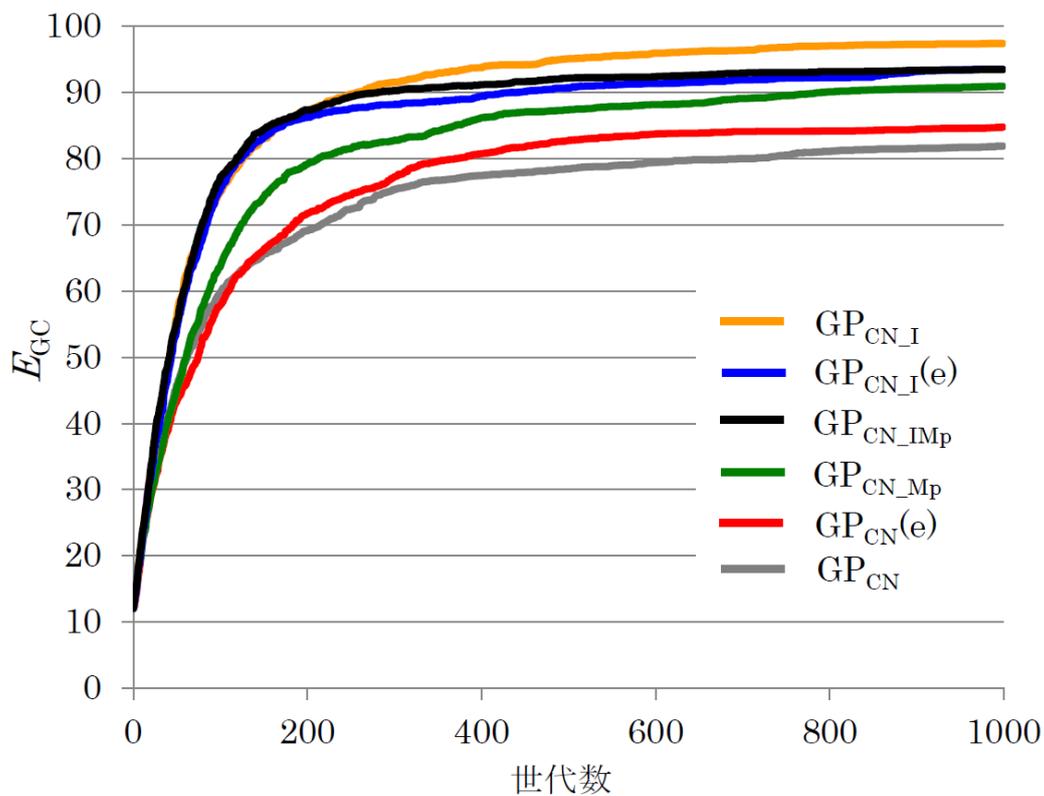


図 6.7 ごみ拾い問題における GP_{CN} 、 $GP_{CN}(e)$ 、 GP_{CN_I} 、 $GP_{CN_{Mp}}$ 、 $GP_{CN_I}(e)$ 、 $GP_{CN_{IMp}}$ の 1000 世代の E_{GC} の変化

法を組み合わせさせた $GP_{CN_I}(e)$ が最も高い E_{SFT} を示しており、プロセス数 P の継承手法の有効性が確認できた。

6.2.5 まとめ

本節では、適合度の変更の重要性と、提案しているプロセス数 P の継承手法とプロセス数 P の突然変異の有効性を確認するために、ベンチマーク問題を用いた実験を行った。終端ノードのみの木を淘汰するための評価関数を確認するための実験では、従来の問題ごとに設定されている E の値を評価関数とした GP_{CN} と終端ノードのみの木を淘汰するための評価関数とした GP_{CN} を比較した。実験では、終端ノードのみの木を淘汰するための評価関数とした GP_{CN} は、ごみ拾い問題と Santa Fe Trail 問題のどちらの問題においても、高い性能を示し、ほとんどの終端ノードのみの木が淘汰されていることが確認できた。

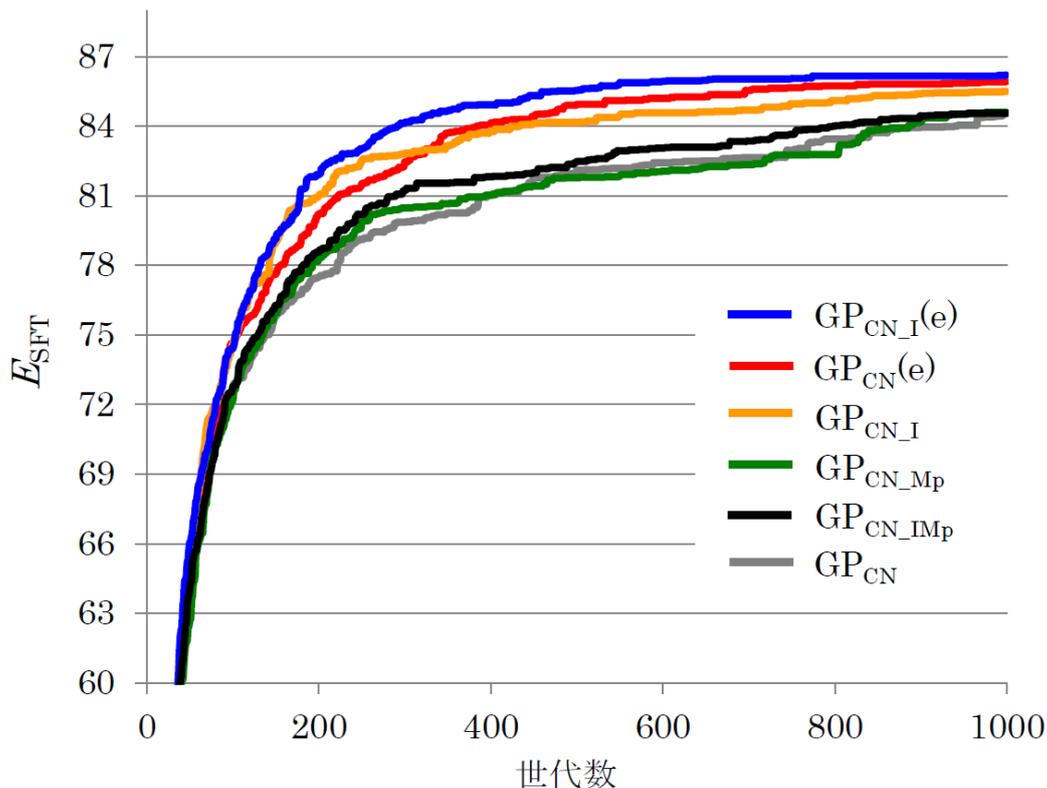


図 6.8 Santa Fe Trail 問題における GP_{CN} , $GP_{CN}(e)$, GP_{CN_I} , $GP_{CN_{MP}}$, $GP_{CN_I}(e)$, $GP_{CN_{IMp}}$ の 1000 世代の E_{SFT} の変化

また、プロセス数 P の自動決定を行う実験では、両問題において、GP_{CN} よりもプロセス数 P を自動決定するための手法を導入した各 GP_{CN} が高い性能を示しており、プロセス数 P を自動決定するための手法の有効性が示された。特に、提案手法であるプロセス数 P の継承手法は高い性能を示しており、プロセス数 P の自動決定手法として最も有効であることが確認できた。

6.3 木の数 M を自動決定するための手法

6.3.1 提案手法

A. 木の数 M を考慮した評価関数

木の数 M は GP_{CN} の個体を特徴づけるパラメータの1つであり、木が多過ぎると進化が遅くなる可能性がある。また、木の数が多いほど問題に柔軟に対応することができる反面、ほとんど機能していない木が複数含まれ進化の妨げになると考えられる。そこで、できる限り少ない木の数 M に決定しシンプルな個体を得るために、先に説明した式 (6.2) に木の数を考慮した評価値を追加する [30, 31, 32]。木の数 M を考慮した適合度は式 (6.3) のように表される。

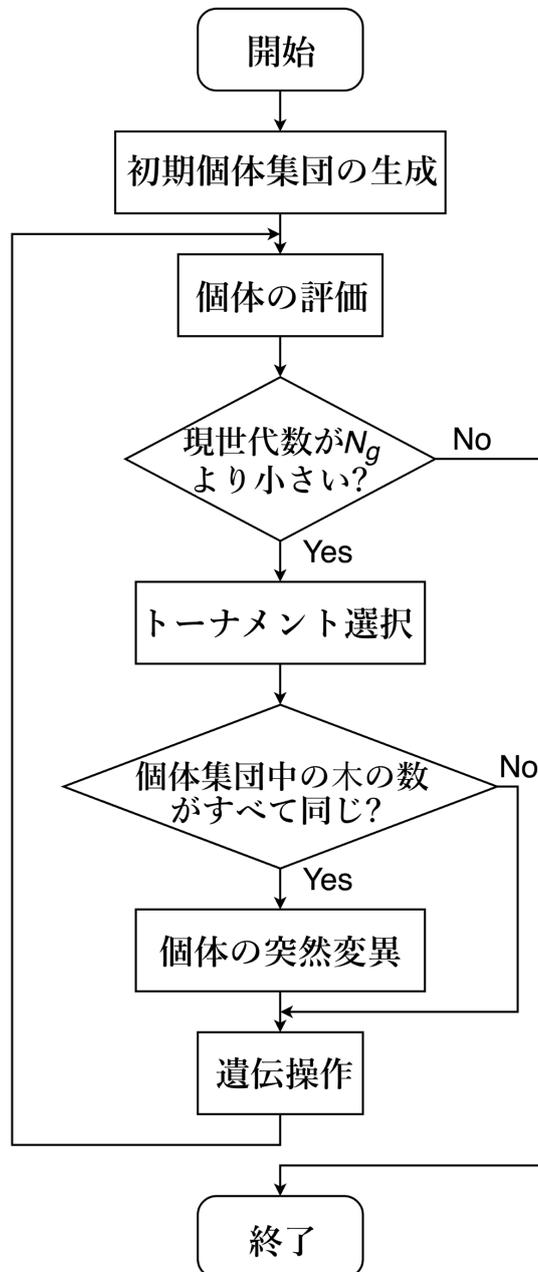
$$Fitness(i) = E(i) + (TotalSteps - M(i)) - \left(\alpha \times \frac{M_{SingleNode}(i)}{M(i)} \right) \quad (6.3)$$

ここで、 $TotalSteps$ は問題ごとに設定されているエージェントの最大行動回数であり、適合度が0より小さくなることを防ぐために用いている。また、各木のプロセス数 P の値を1とした場合、木の数 M が最大で $TotalSteps$ となることからこの値を用いている。 α は終端ノードのみの木を淘汰するための適合度と同様の値を用いた。

B. 個体の突然変異

進化の過程で木の数を獲得するには様々な木の数 M を十分に探索する必要がある。しかしながら、適合度の変更だけでは木の数 M が急速に収束してしまい探索することができない。そこで、個体の突然変異を提案する。個体の突然変異を導入した GP_{CN} (GP_{CN_Mi}) の進化の流れを図 6.9 に示す。個体の突然変異は、遺伝操作の前に実行される。また、個体の突然変異の前に個体集団中の全ての個体

の木の数 M が同じであるかの判定を行う。判定の結果、全ての個体が同じ木の数 M であれば確率によって個体の突然変異が行われる可能性があり、そうでなければ行われない。これは、個体の突然変異が個体を大きく変化させる操作であるため、毎世代行わないようにするためである。個体の突然変異が行われる可能性がある場合、個体の突然変異率 P_{Mi} で個体ごとに個体の突然変異を実行する。個体

図 6.9 GP_{CN_Mi} の進化の流れ

の突然変異では、はじめに、木の増加操作確率 P_{Inc} で木を増やす操作が選択され、 $1 - P_{\text{Inc}}$ で木を減らす操作が選択される。ただし、もし個体の木の数が2であった場合は、必ず木を増やす操作が選ばれる。

次に、木の数 M を増やす操作と木を減らす操作においてそれぞれ2つある操作のうち、どちらを実行するかを選択する。木の数 M を増やす操作の場合は木の複製確率 P_{Copy} で木の複製が、 $1 - P_{\text{Copy}}$ でランダム木の追加が選択される。木を減らす操作の場合はランダムな木の削除確率 P_{Rd} でランダムな木の削除が、 $1 - P_{\text{Rd}}$ でセマンティクスを用いた木の削除が選択される。最後に、選択された操作を実行する。木を増やす操作である木の複製とランダム木の追加、木を減らす操作であるランダムな木の削除とセマンティクスを用いた木の削除について以下に示す。

- 木の複製

木の複製は、個体内の木を複製し木の数 M を増やす操作である。図 6.10 に木の複製の例を示す。はじめに、個体の中からランダムに木を1つ選択し複製する。次に、複製した木の挿入位置をランダムに決定し挿入する。もし選ばれた位置が一番後ろでなければ、同じ識別番号を持つ木とそれより後ろの木を後ろにずらす。最後に木の数 M を1増やす。木の複製は個体内に同じ行動規則を持つ木を増やす操作であり、複数木を複製しても効果が得られないため、1つしか複製を行わない。

- ランダム木の追加

ランダム木の追加は、ランダムに生成した木を追加し木の数 M を増やす操作である。図 6.11 にランダム木の追加の例を示す。はじめに、ランダムに木を生成する。次に、生成した木の挿入位置をランダムに決定し挿入する。もし選ばれた位置が一番後ろでなければ、同じ識別番号を持つ木とそれより後ろの木を後ろにずらす。最後に木の数 M を1増やす。また、探索領域をより大きな木の数 M に広げるために上の操作を2回行い、2つの木を追加する。

- ランダムな木の削除

ランダムな木の削除は、個体内からランダムに選ばれた木を削除する操作である。図 6.12 にランダムな木の削除の例を示す。この操作は木を減ら

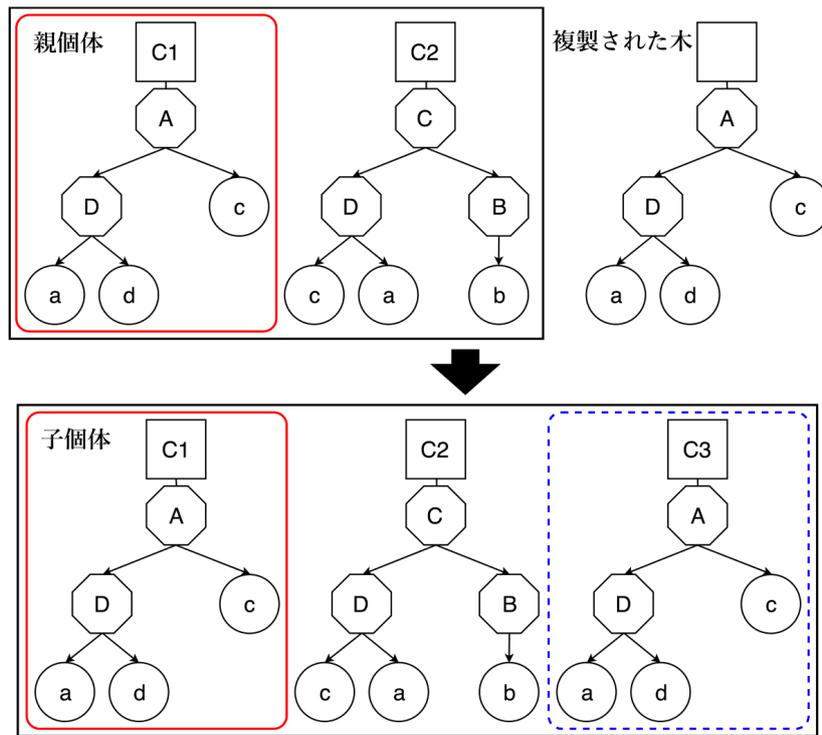


図 6.10 木の複製の例

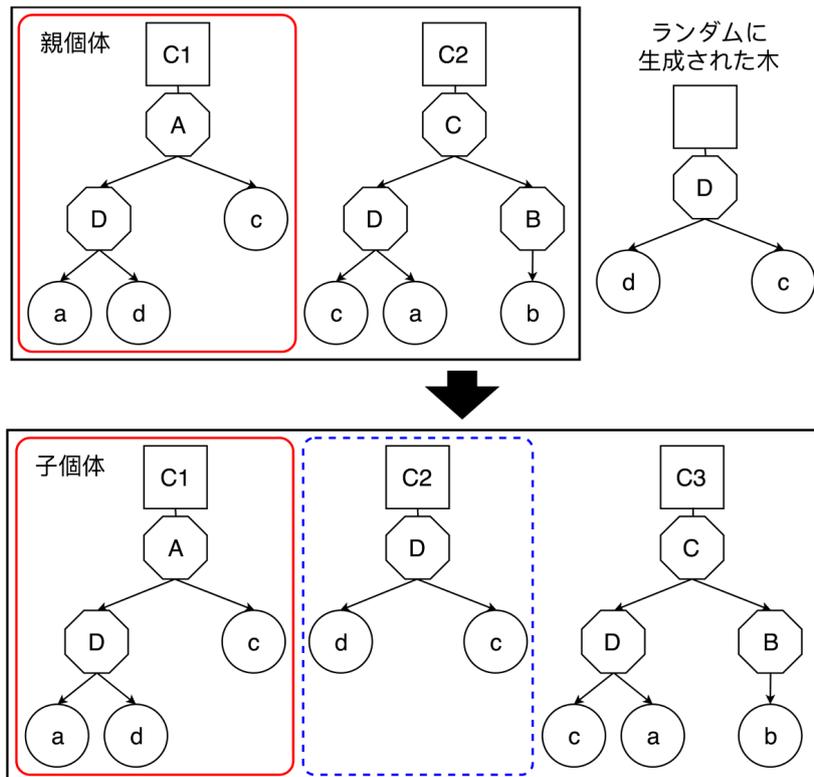


図 6.11 ランダム木の追加の例

す操作であるため、個体集団中の個体の木の数 M が2の場合は選択されない。はじめに、個体内から削除する木をランダムに選択しその木を削除する。もし削除された木の識別番号が一番後ろの番号でなければ、それより後ろの木を前にずらす。最後に、木の数 M を1減らす。

- セマンティクスを用いた木の削除

セマンティクスを用いた木の削除は、隣り合う類似した木を削除することで全体的な行動規則に影響を与えず木の数 M を減らすことができる操作である。GP_{CN}の複数木は識別番号により順番に参照されるため、同じ行動規則が並んでいる場合にはどちらかを削除しても個体の行動規則に影響がない。図6.13にセマンティクスを用いた木の削除の例を示す。この操作は、木を減らす操作であるため、個体集団中の個体の木の数 M が2の場合は選択されない。また、本操作で用いたセマンティクスについては次節で説明する。木の数を M とし、以下にアルゴリズムを示す。

- 1: i を1に初期化。
- 2: セマンティクスを用いて、識別番号 i の木と識別番号 M の木の類似度を計算。
- 3: 類似度がしきい値以上なら識別番号 M の木を削除し、 M を1減らす。
- 4: $M = 2$ であれば10に飛ぶ。
- 5: セマンティクスを用いて、識別番号 i の木と識別番号 $i+1$ の木の類似度を計算。
- 6: 類似度がしきい値未満なら9に飛ぶ。
- 7: 識別番号 $i+1$ の木を削除し、 M を1減らす。
- 8: $i+1 < M$ ならば、削除された木より後ろの木を前にずらす。
- 9: i を1増やし、 $i+1 \leq M$ ならば4に戻る。
- 10: 終了。

GP_{CN}において、エージェントは識別番号の最も大きい木の次に、識別番号の最も小さい木を参照する。したがって、識別番号の最も小さい木と識別番号の最も大きい木のセマンティクスの類似度も計算する必要があるため、はじめにそれ

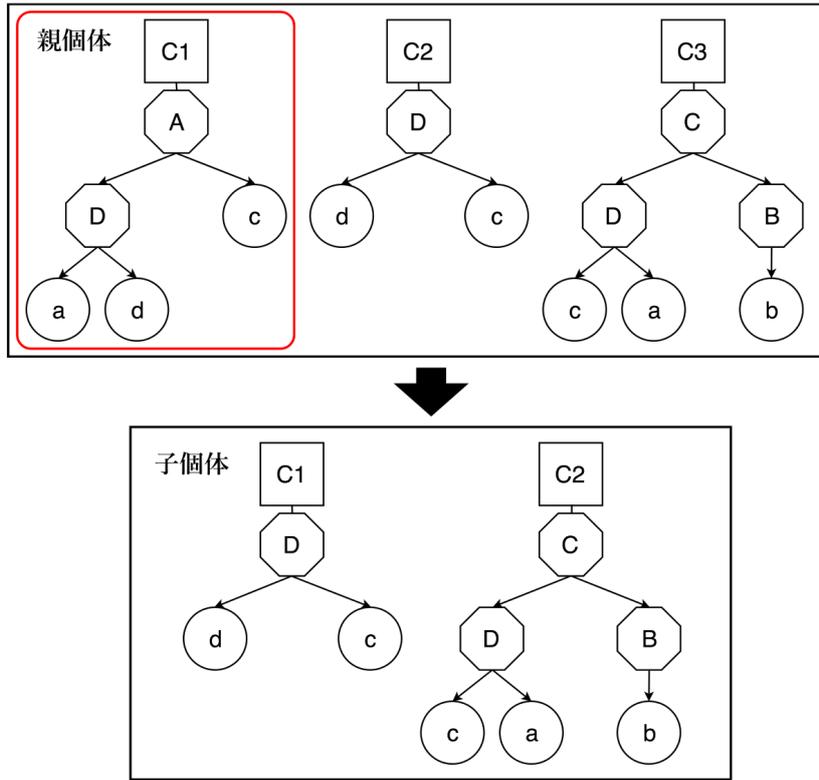


図 6.12 ランダムな木の削除の例

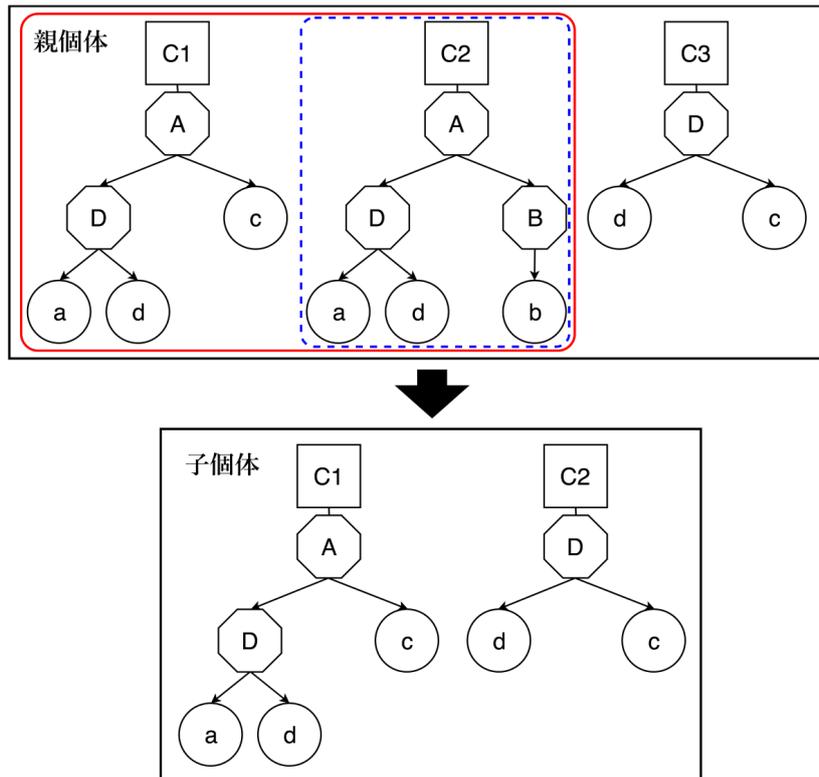


図 6.13 セマンティクスを用いた木の削除の例

を計算する．また，セマンティクスを用いた木の削除では，1度の操作によって複数の木が削除される可能性がある．本研究ではしきい値を0.6とする．

C. エージェント問題のためのセマンティクス

GPにおいて，木の類似度を測るためにセマンティクスがある．セマンティクスを用いた木の類似度では，構造のマッチングではなく，木の意味（働き）を比較する．関数同定問題では，終端ノードから根ノードに向けて計算を行い，その過程で値がどのように変化するかをセマンティクスとする方法がある [14].

しかしながらエージェント問題では，根ノードから複数あるうちの1つの終端ノードに向けて処理が行われるため，個体のセマンティクスを単純に考えることが難しい．E. Galvan-Lopezらによってエージェント問題におけるセマンティクスが提案されている [13]. このセマンティクスでは，エージェントがマスを移動するたびに方向を記録し，その軌跡をセマンティクスとしている．しかしながら，この方法では同じマスでの行動を捉えることができない．そこで本研究では，セマンティクス用の環境を1つではなく複数用意し，それぞれの環境で実際にエージェントがとった行動からセマンティクスを求める．

- ごみ拾い問題におけるセマンティクス

ごみ拾い問題におけるセマンティクスは次のように求めた．はじめに，エージェントの情報（位置，向き，保持しているごみの数）とフィールド上のごみをランダムに変更した10種類のセマンティクス用の環境を作成しておく．この環境は，個体の適合度を求めるための環境とは異なる．そして，適合度を求める個体の評価が行われた後，セマンティクス用の環境に個体を適用する．セマンティクス用の環境 i においてエージェントが最初に行った行動を a_i とし，10次元の列ベクトル $St=(a_1, a_2, \dots, a_{10})$ を個体のセマンティクスとする．また，本研究で用いる GP_{CN} は1個体が複数の木を持つため，1つ1つの木に10種類のセマンティクス用の環境を適用し，セマンティクスを記録する．したがって，それぞれの木が独立したセマンティクスを持つ．類似度は，2つの木のセマンティクスを a_1 から順に比較し，一致した行動の割合とする．

- Santa Fe Trail 問題におけるセマンティクス

Santa Fe Trail 問題では、フィールドが決められており、エージェントの用いる知覚情報が“餌が前方にあるか”しかない。そのため、ごみ拾い問題と同様にはセマンティクスを求めることができない。そこで、図 6.14 に示すような、餌が全てのマスに配置されている環境、餌が全くない環境、1 マスおきに餌が配置されている環境を用意し、それらを用いてセマンティクスを求める。ごみ拾い問題と同様に、適合度を求めるための個体の評価が行われた後、セマンティクス用の環境に個体を適用し行動を記録する。この時、Santa Fe Trail 問題では 1 度の参照であっても複数の行動を実行する Prog 関数 (4.3 Santa Fe Trail 問題内の表 4.2 Santa Fe Trail 問題に用いるノードの種類を参照) があるため、1 度の参照で 5 つまでの行動を記録する。環境 i における j 回目の行動を $a_{((5*(i-1)+j))}$ とし、15 次元のベクトル $St=(a_1, a_2, \dots, a_{15})$

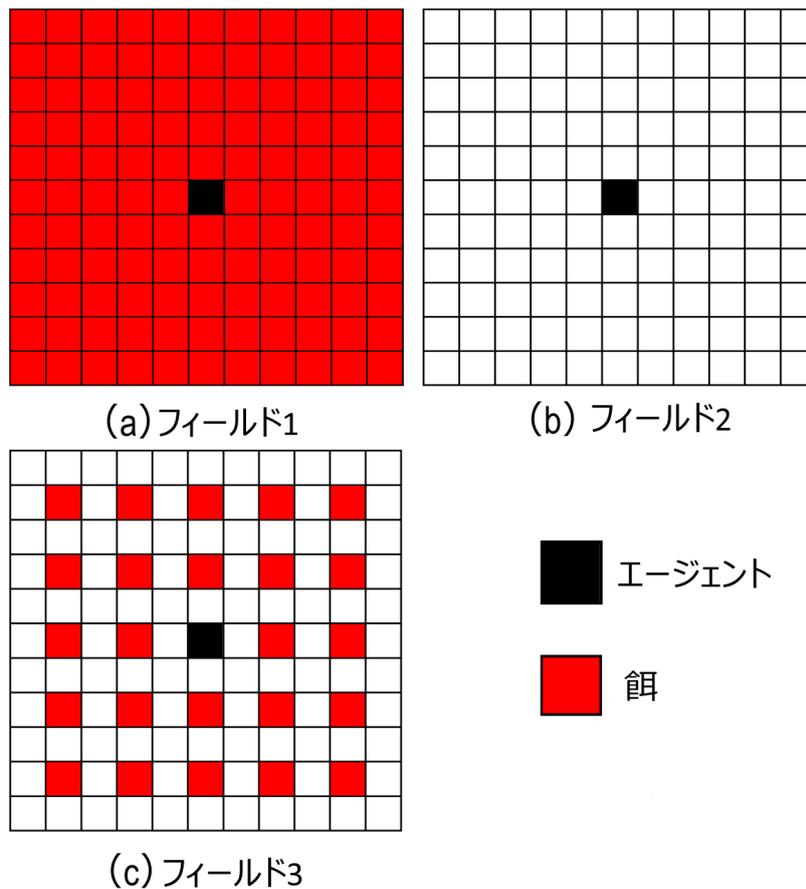


図 6.14 Santa Fe Trail 問題のためのセマンティクス用フィールド

を個体のセマンティクスとする。ただし、Prog 関数の組み合わせ方やプロセス数が小さすぎて行動が5つ存在しない場合、すなわち1度の参照で行動が5つ存在しない場合は、“行動無し”を記録する。また、ごみ拾い問題と同様に1つ1つの木に3種類のセマンティクス用の環境を適用し、セマンティクスを記録する。類似度の計算はごみ拾い問題と同様である。

6.3.2 木の数 M を考慮した適合度の評価実験

木の数を考慮した適合度の影響を確認するために、適合度のみの変更を行った GP_{CN} により実験を行う。実験では、木の数 M を考慮した適合度を用いた GP_{CN} (GP_{CN}(CNT)) と木の数 M を考慮した適合度を用いていない（終端ノードのみの木を淘汰するための適合度のみのみを用いた）GP_{CN} (GP_{CN}(ETS)) を用いて各木の数を獲得した試行の数によって比較を行う。各試行において獲得された木の数とは、最終世代で最良適合度を示した個体が獲得した木の数のことである。

実験設定は、ごみ拾い問題では個体数を300とし、Santa Fe Trail 問題では個体数を2000とする。2つの問題で共通する設定として、世代数を1000、木の深さを6、初期集団の生成方法を *grow* 方式とする。プロセス数 P の初期値は、先ほどの実験と同様に $[1, TotalSteps/M]$ の範囲からランダムに選択するように設定している。また、本実験では個体の木の数 M の初期値は $[2, 20]$ の中からランダムに選択するように変更している。その他パラメータについては、木の数 M の設定方法を変更したことに伴い、新たにパラメータ調整を行った。パラメータ調整により新たに決定された GP_{CN} のパラメータを表6.4に示す。また、終端ノードのみの木を淘汰するための適合度と木の数 M を考慮した評価関数に関する係数 α は、10.0とした。これらのパラメータは、各問題において GP_{CN}(ETS) と GP_{CN}(CNT) の性能が最大となるように調整している。

ごみ拾い問題において各木の数を獲得した試行の数を表6.5に示す。表中の Other は、木の数10以上を示している。木の数 M を考慮した適合度を用いている GP_{CN}(CNT) では、30試行のうち20試行が5以下の木の数を獲得している。木の数 M を考慮していない GP_{CN}(ETS) では、木の数の多い個体が多く生き残ってしまったため20試行が Other となった。また、1000世代における各試行の最良 E_{GC}

表 6.4 2つの問題に共通するパラメータ

	値
突然変異率	0.05
突然変異木率	0.1
交叉率	0.8
逆位率	0.2
トーナメントサイズ	2
エリート保存数	1

の 30 試行の平均で比較すると, $GP_{CN}(CNT)$ は 83.3, $GP_{CN}(ETS)$ は 66.8 となっている. 次に Santa Fe Trail 問題における各木の数を獲得した試行の数を表 6.6 に示す. 木の数 M を考慮した適合度を用いている $GP_{CN}(CNT)$ では, 50 試行のうち 32 試行が 5 以下の木の数を獲得している. 木の数 M を考慮していない $GP_{CN}(ETS)$ では, ごみ拾い問題と同様に木の数の多い個体が多く生き残ってしまったため 46 試行が Other となった. また, 1000 世代における各試行の最良 E_{SFT} の 50 試行の平均で比較すると, $GP_{CN}(CNT)$ は 85.2, $GP_{CN}(ETS)$ は 82.1 となっている. 2つの実験結果から, 木の数を考慮した適合度が無駄に大きな木の数への収束を防ぎ, それにより性能が向上することを示したため, 木の数の自動決定に重要であることが確認された.

表 6.5 ごみ拾い問題において各木の数を獲得した試行の数

	木の数								
	2	3	4	5	6	7	8	9	Other
$GP_{CN}(CNT)$	10	6	2	2	4	1	2	2	1
$GP_{CN}(ETS)$	3	2	0	0	0	1	3	1	20

表 6.6 Santa Fe Trail 問題において各木の数を獲得した試行の数

	木の数								
	2	3	4	5	6	7	8	9	Other
GP _{CN} (CNT)	2	8	8	14	8	6	4	0	1
GP _{CN} (ETS)	0	0	0	0	0	0	1	3	46

6.3.3 木の数 M を自動決定する手法の実験

次に、提案手法である個体の突然変異を用いて木の数とプロセス数 P の自動決定を行う。この実験では全ての手法に木の数 M を考慮した適合度を用いている。各問題における個体の突然変異のためのパラメータを表 6.7 に示す。これらのパラメータは各問題において提案手法である個体の突然変異を導入した GP_{CN} (GP_{CN_Mi}) の E が最大になるように調整している。また、その他設定やパラメータは先ほどの実験と同様である。提案手法の有効性を確認するために、複数木構造を持つ GP (GP_{CN})、プロセス数 P の継承手法を導入した GP_{CN} (GP_{CN_I})、提案手法である GP_{CN_Mi}、プロセス数 P の継承手法と個体の突然変異を導入した GP_{CN} (GP_{CN_IMi}) を各問題に適用し比較する。

ごみ拾い問題において各 GP_{CN} が各木の数を獲得した試行の数を表 6.8 に示す。表中の Other は、木の数 10 以上を示している。単体の GP_{CN} では 9 以上の木の数を獲得した試行があるが、GP_{CN_I} と GP_{CN_Mi}、GP_{CN_IMi} ではそのような試行は

表 6.7 各問題における提案手法のパラメータ

	ごみ拾い問題	Santa Fe Trail 問題
個体の突然変異率 P_{Mi}	0.9	0.85
木の増加操作確率 P_{Inc}	0.45	
木の複製確率 P_{Copy}	0.5	
ランダムな木の削除確率 P_{Rd}	0.3	0.5

なくなっている。したがって、プロセス数 P の獲得手法や個体の突然変異が木の収束に有効であることが示された。また、試行の多くが木の数 2~3 に集中しており、特に木の数 2 が適していることを示唆している。次に、各 GP_{CN} を性能で比較する。ごみ拾い問題における 1000 世代の E_{GC} の変化を図 6.15 に示す。各世代の E_{GC} は、試行 30 回の各試行において最も高い適合度を示した個体の E_{GC} の平均値である。図 6.15 から、 GP_{CN} よりもプロセス数 P の継承手法を用いた GP_{CN_I} の方が高い性能を示している。これはプロセス数 P の継承手法によって問題に適したプロセス数 P が獲得され、さらに適合度の変更によって木の数 M を少ない数に収束できたからと考えられる。プロセス数 P の継承手法と個体の突然変異を組み合わせた $GP_{CN_{IMi}}$ は、 GP_{CN_I} より広い範囲で木の数 M の探索が行えたため、高い性能を得たと考えられる。

Santa Fe Trail 問題における各木の数を獲得した試行の数を表 6.9 に示す。ごみ拾い問題とは異なり、 $GP_{CN_{Mi}}$ の結果が GP_{CN} と近い傾向を示している。これは、プロセス数 P の継承手法がないためにプロセス数 P の探索が行えず、初期生成時のプロセス数 P の値に適した木の数を獲得したからだと考えられる。プロセス数 P と木の数の自動決定を同時に行っている $GP_{CN_{IMi}}$ の結果を見ると少ない木の数に集中しており、期待通りにシンプルな個体が獲得できている。Santa Fe Trail 問題の 1000 世代の E_{SFT} の変化を図 6.16 に示す。各世代の E_{SFT} は、試行 50 回の各試行において最も高い適合度を示した個体の E_{SFT} の平均値である。図 6.16 から、 GP_{CN} よりも、 GP_{CN_I} や $GP_{CN_{Mi}}$ の方が高い性能を示している。また、2 つ

表 6.8 ごみ拾い問題において各木の数を獲得した試行の数

	木の数								
	2	3	4	5	6	7	8	9	Other
GP_{CN}	10	6	2	2	4	1	2	2	1
GP_{CN_I}	12	6	4	3	2	1	2	0	0
$GP_{CN_{Mi}}$	10	8	9	1	1	1	0	0	0
$GP_{CN_{IMi}}$	13	7	5	1	1	1	2	0	0

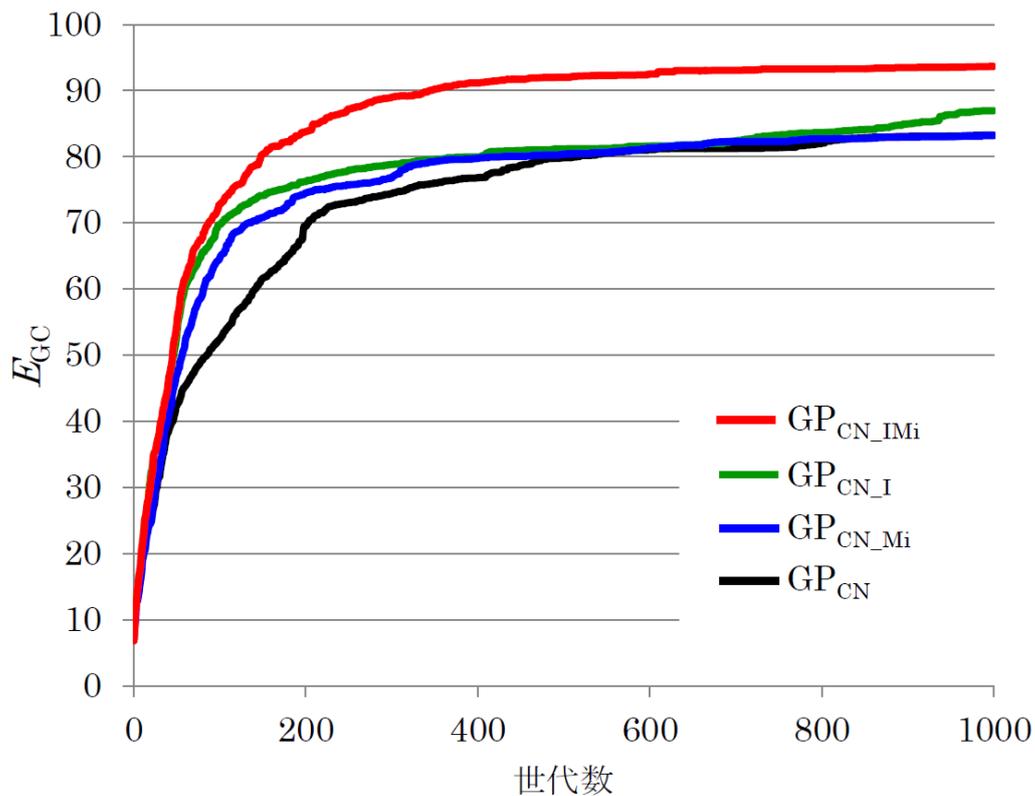


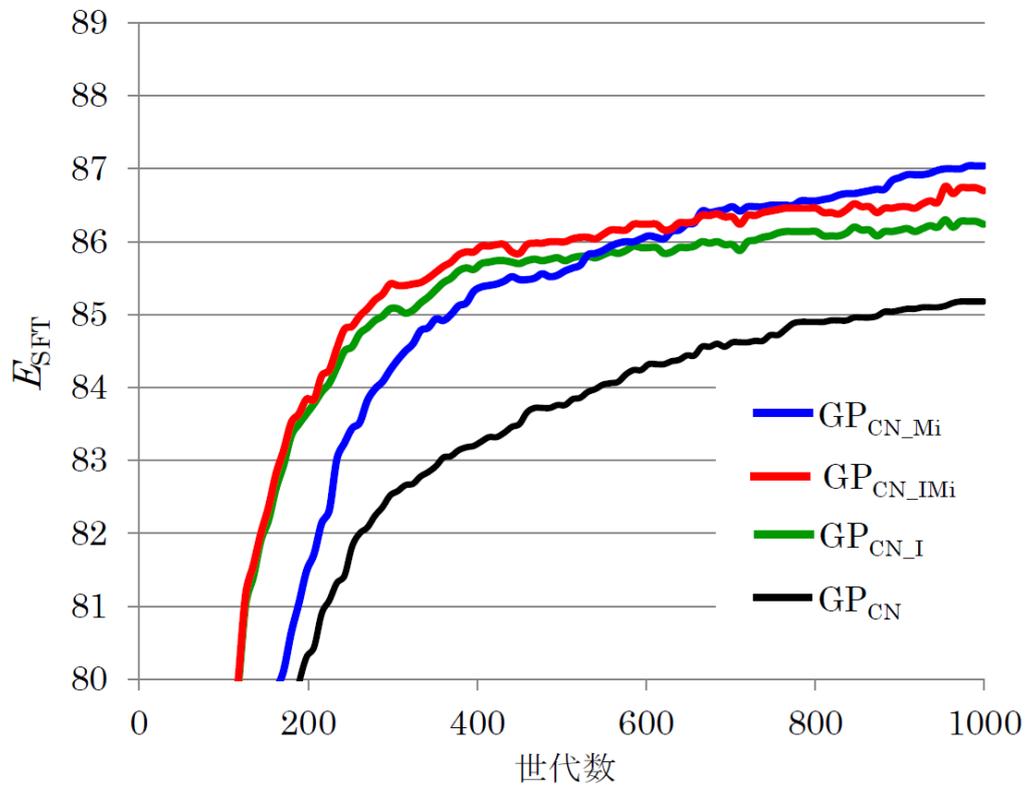
図 6.15 ごみ拾い問題における GP_{CN}, GP_{CN_Mi}, GP_{CN_I}, GP_{CN_IMi} の 1000 世代の E_{GC} の変化

を組み合わせた GP_{CN_IMi} は GP_{CN_Mi} よりも若干性能が劣っているが、問題の最大 E_{SFT} である 89 を獲得した試行数では、GP_{CN_IMi} は 27 試行、GP_{CN_Mi} は 17 試行となり GP_{CN_IMi} のほうが良い結果を得ていることがわかる。

以上の結果から、プロセス数 P の継承手法と個体の突然変異の組み合わせによって、プロセス数 P と木の数 M の探索を同時に行い問題に適したそれぞれのパラメータを獲得できることがわかった。表 6.8 から、ごみ拾い問題においては木の数 2 または 3 が適した値であると考えられる。木の数 2 を獲得した 13 試行の最良個体の平均適合度は 98.1, 木の数 3 を獲得した 7 試行の最良個体の平均適合度は 90.6 となったため、木の数 2 が適していると考えられる。また、Santa Fe Trail 問題では、表 6.9 から、木の数 2 または 3 が適した値であると考えられる。木の数 2 を獲得した 15 試行の最良個体の平均適合度は 88.6, 木の数 3 を獲得した 18 試行の最良個体の平均適合度は 85.6 となったため、木の数 2 が適していると考えられる。

表 6.9 Santa Fe Trail 問題において各木の数を獲得した試行の数

	木の数								
	2	3	4	5	6	7	8	9	Other
GP_{CN}	2	8	8	14	8	6	4	0	1
GP_{CN_I}	13	17	10	4	3	1	0	0	2
$GP_{CN_{Mi}}$	2	3	3	12	6	6	9	7	2
$GP_{CN_{IMi}}$	15	18	8	2	3	3	0	0	1

図 6.16 Santa Fe trail 問題における GP_{CN} , $GP_{CN_{Mi}}$, GP_{CN_I} , $GP_{CN_{IMi}}$ の 1000 世代の E_{SFT} の変化

6.3.4 自動決定により得られたパラメータの評価

自動決定実験により得られた各問題に適した木の数 M とプロセス数 P の有効性を評価するため、人手で決定した木の数 M との比較を行う。人手による木の数の決定方法では、GP_{CN} の木の数 M を 2~10 までのそれぞれの値に固定させて実験を行い、性能が1番高くなるものを問題に適した木の数 M としていた。本実験では、適合度に終端ノードのみの木を淘汰する評価関数を適用し、プロセス数 P の自動決定手法であるプロセス数 P の継承手法を用いた GP_{CN} (GP_{CN,I}) によって人手による木の数の決定を行った。ごみ拾い問題では個体数を 300 とし、Santa Fe Trail 問題では個体数を 2000 とする。2つの問題で共通する設定として、世代数を 1000、木の深さを 6、初期集団の生成方法を *grow* 方式とする。各パラメータは表 6.4 を用い、各木のプロセス数 P の初期値は $[1, TotalSteps/M]$ からランダムに設定する。この方法で得られた各問題に適した木の数は、ごみ拾い問題では 7、Santa Fe Trail 問題では 2 であった。

はじめに、同じ木の数を得た Santa Fe Trail 問題について考察する。Santa Fe Trail 問題では、従来の研究においても木の数 2 が適しているとされてきた。今回の自動決定による結果と人手による結果でも同様の結果が得られたため、Santa Fe Trail 問題において適した木の数は 2 であると考えられ、自動決定を使うことにより求めることができた。木の数の自動決定と木の数の人手による決定の中で得られた問題に適したプロセス数 P の値の一部を表 6.10 に示す。どちらの決定方法においてもプロセス数 P の継承手法によって、各木のプロセス数 P が進化の中で自動的に決定されている。ここでいう問題に適したプロセス数 P の値とは、最終世代において適合度が最大値 (Santa Fe Trail 問題では 89) に達している個体が獲得したプロセス数 P の値である。ただし、木の数の自動決定の場合は木の数 2 を獲得した試行の中で最終世代において適合度が最大値に達している個体が獲得したプロセス数 P の値とする。表 6.10 からどちらの決定方法においても同じようなプロセス数の傾向を持っており、Santa Fe Trail 問題においては問題に適した木の数 M とプロセス数 P の値を同時に獲得できたと言える。

次に、ごみ拾い問題においては自動決定の結果と人手による結果が異なっている。実際に、最終世代で獲得された行動規則で比較を行う。表 6.11 に人手で決定

表 6.10 Santa Fe Trail 問題において各木が獲得したプロセス数 P の値の例

手動		自動	
木 1	木 2	木 1	木 2
183	183	190	190
147	1	179	1
195	13	2	151
95	14	153	153
155	155	55	186
170	6	51	184

された木の数 7 の最良個体のごみを拾った数と捨てた数，獲得したプロセス数 P の例を，表 6.12 に自動決定により得られた木の数 2 の最良個体のごみを拾った数と捨てた数，獲得したプロセス数 P の例を示す．表 6.11 の人手で決定した木の数 7 の個体は，木 1,4 は汎用的な行動規則，木 2,3,5 はごみを拾うことに特化した行動規則，木 6 はごみを捨てることに特化した行動規則を獲得している．木の数 2,3 は隣り合っており，同じ行動規則を持つことから 1 つの木としても問題はない．また，木 7 についてはごみを拾うことも捨てることもできていないため，木を削除しても影響が少ないと考えられる．これらの木のように，木の数が多いい体には無駄な木が多く存在する．一方，表 6.12 の自動決定により得られた木の数 2 の個体は，木 1 はごみを拾うことに特化した行動規則，木 2 はごみを捨てることに特化した行動規則を獲得しており，理想的なシンプルな個体となっている．また，適合度が最大となった個体のうち，試行番号の小さい順から 10 個体の獲得した各木のプロセス数 P の値を表 6.13 に示す．この表からも，木の数 7 の個体には，プロセス数 P が小さくあまり参照されていない無駄な木が多く存在していることがわかった．

最後に，ごみ拾い問題において獲得された木の数 M とプロセス数 P の値を用いて比較実験を行う．実験には，通常の GP，木の数を 2 とした $GP_{CN}(GP_{CN}(AUTO))$ と木の数を 7 とした $GP_{CN}(GP_{CN}(MANUAL))$ ，それぞれに条件付確率を用いた個体生成手法を導入した $GP_{CN_CP}(AUTO)$ と $GP_{CN_CP}(MANUAL)$ を用いる．個

表 6.11 木の数 7 の個体の各木がごみを拾った数, 捨てた数, プロセス数 P の値の例

	木 1	木 2	木 3	木 4	木 5	木 6	木 7
拾った数	42	12	7	15	23	1	0
捨てた数	20	2	0	38	2	38	0
プロセス数 P	24	10	10	23	23	24	11

表 6.12 木の数 2 の個体の各木がごみを拾った数, 捨てた数, プロセス数 P の値の例

	木 1	木 2
拾った数	98	2
捨てた数	2	98
プロセス数 P	25	25

表 6.13 ごみ拾い問題において各木が獲得したプロセス数 P の値の例

手動							自動	
木 1	木 2	木 3	木 4	木 5	木 6	木 7	木 1	木 2
24	10	10	23	23	24	11	1	27
16	16	1	5	1	1	1	24	18
7	1	1	1	1	1	8	25	25
5	36	9	13	5	13	13	20	20
14	2	14	2	2	2	2	15	55
11	25	8	25	23	25	11	1	10
3	3	3	3	3	15	20	31	31
3	3	3	3	14	3	14	17	17
11	11	16	14	26	16	5	19	19
9	3	3	9	9	3	3	22	22

体数を 300, 世代数を 1000, 木の深さを 6, 初期集団の生成方法を *grow* 方式とする。また, GP_{CN} のプロセス数 P の値は, それぞれの木の数に対応したプロセス数のセットが表 6.13 から個体ごとにランダムに選択設定される。いずれの手法においても, 適合度には末端ノードのみの木を淘汰するため適合度を適用する。その他パラメータは, 表 6.4 を用いる。1000 世代の E_{GC} の変化を図 6.17 に示す。各世代の E_{GC} は, 試行 30 回の各試行において最も高い適合度を示した個体の E_{GC} の平均値である。図 6.17 から, GP よりも各 GP_{CN} の方が高い E_{GC} を示している。また, 自動決定により得られた木の数 M とプロセス数 P を用いた $GP_{CN}(AUTO)$ の方が $GP_{CN}(MANUAL)$ よりも高い E_{GC} を示している。また, 条件付き確率による個体生成手法を導入した GP_{CN_CP} においても, $GP_{CN_CP}(MANUAL)$ よりも $GP_{CN_CP}(AUTO)$ の方が良い結果を示しており, 自動決定は人手では発見できなかった問題に最適な木の数を発見できたといえる。また, 人手による木の数の決定では, 条件を変更し GP_{CN} を複数回実行する必要があったが, 本論文の自動決

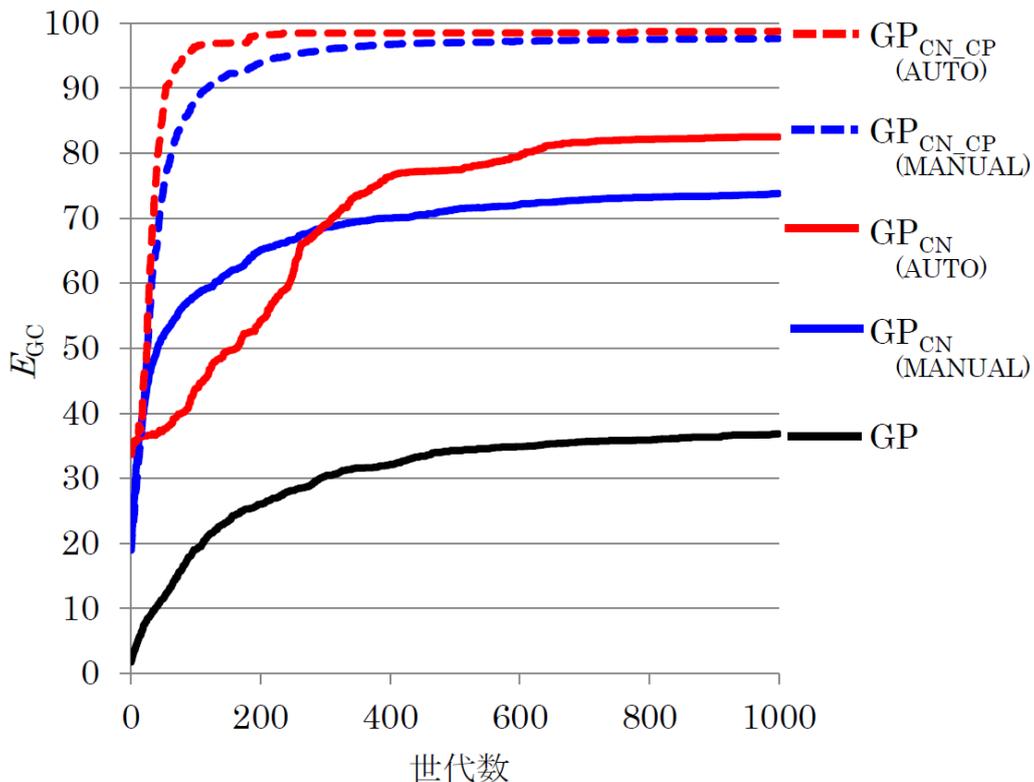


図 6.17 ごみ拾い問題における GP, $GP_{CN}(AUTO)$, $GP_{CN}(MANUAL)$, $GP_{CN_CP}(AUTO)$ と $GP_{CN_CP}(MANUAL)$ の 1000 世代の E_{GC} の変化

定を用いれば1度の実行で木の数を決定することができる。

6.3.5 まとめ

本節では、GP_{CN}における木の数の自動決定を行うために、木の数を考慮した評価関数の導入と個体の突然変異を提案した。木の数 M を考慮した評価関数は、問題に適したできるだけ小さな木の数を決定するための評価関数である。この評価関数の有効性を確認する実験では、前の研究で提案されている終端ノードのみを淘汰するための評価関数と木の数 M を考慮した評価関数の比較を行い、木の数 M を考慮した評価関数によって小さな木の数に決定されることが確認できた。

また、個体の木の数を自動決定する実験では、Santa Fe Trail問題においては人手で設定した木の数と同様の木の数を得ることができた。また、プロセス数 P においても同様の傾向が見られたため、木の数とプロセス数の同時決定が期待通り行えることが確認できた。ごみ拾い問題においては、人手で決定した結果と異なる木の数を自動決定により得た。それぞれの木の行動を確認した結果、人手により決定された個体の木には無駄な行動が多く見られ、自動決定により決定された個体は2つの木を有効に活用していた。また、個体の性能から見ても自動決定により得られた木の数を適用した個体の方が高かったため、人手では見つけることができない木の数を獲得できたと考えられる。したがって、提案手法によって、問題ごとに有効なプロセス数と木の数を自動的に決定することができた。

第7章 むすび

人工知能の分野において、目的を達成するためのエージェントの判断と行動の規則を自動生成する手法の研究が行われてきた。我々は、その中の1つである遺伝的プログラミング (GP) を拡張した調整ノード付き GP (GP_{CN}) を用いた研究を行ってきた。

本論文では、はじめに GP_{CN} の性能の向上を図るため、木の順序入れ替え手法を提案した。木の順序入れ替え手法は、 GP_{CN} の個体が複数の木を持ち識別番号によって管理されていることに着目した手法である。木の順序入れ替え手法は、木の数が多い場合に効力を発揮し性能の大幅な向上につながった。しかしながら、木の数が少ない場合には役割ごとに成長する木の進化を妨害してしまうために、有効でないことがわかった。したがって、木の順序入れ替え手法は役割が多い複雑な問題において有効であると考えられる。

次に、 GP_{CN} において特徴的なパラメータである木の行動回数を示すプロセス数 P の自動決定のために、プロセス数 P の継承手法とプロセス数 P の突然変異の提案を行った。実験では、2つの問題を通してプロセス数 P の継承手法が有効的に働くことが確認できた。継承手法の欠点と思われた急速なプロセス数 P の収束を防ぐため提案したプロセス数 P の突然変異は、プロセス数 P の値が大幅に変更されてしまうため、役割ごとの進化を妨害してしまった。したがって、プロセス数 P の自動決定手法としては、プロセス数 P の継承手法が有効であった。

最後に、木の数を自動決定するための手法として、評価関数の変更と個体の突然変異の提案を行った。実験では、木の数とプロセス数の自動決定を行い Santa Fe Trail 問題では人手と同様の結果を、ごみ拾い問題では自動決定により人手と異なる結果を得た。また、人手と自動決定により得られた木の数を用いて評価実験を行った結果、自動決定により得られた木の数を用いた GP_{CN} の方が高い性能を示した。したがって、提案手法によって、人手では見つけることができない問題に適した木の数を獲得できたといえる。

謝辞

本研究において、ご指導頂いた広島市立大学 情報科学研究科 知能工学専攻 知的メディア工学研究室 高橋健一教授に深く感謝いたします。高橋教授の下で研究した5年半はとても充実した日々で、国際会議での発表など様々な経験をさせていただくことができました。また、同研究室の准教授であった三村和史教授、同研究室の稲葉通将助教授にも日頃から多くの助言を頂き深く感謝いたします。ならびに、広島市立大学 情報科学研究科 高濱徹行教授にはお忙しい時間を割いていただき、本論文の完成に有益なご教示、ご助言を頂きました。ここに深く謝意を表します。

また、母校である大島商船高等専門学校においてご指導をいただき、現在に至るまで私を支えてくださった岡崎秀俊教授にも深く感謝いたします。最後に、市立大学での5年間で出会い共に研究を行ってきた学生の皆さんや、ここまで私を支えてくれた家族にも感謝の気持ちでいっぱいです。ありがとうございました。

参考文献

- [1] Nathan R. Sturtevant, “An Analysis of UCT in Multi-player,” The 6th International Conference on Computers and Games (CG 2008), pp.37–49, 2008.
- [2] 齋藤雄太, 鶴岡慶雅, “線形関数近似によるトリックテイキングゲームの Q 学習,” ゲームプログラミングワークショップ 2016 論文集, vol.2016, pp.196–200, 2016.
- [3] 笠原和真, 二本健太, 伊藤崇, 高橋健一, 稲葉通将, “SVR を適用した FALCON によるトランプゲームに対する学習実験,” 日本知能情報ファジィ学会誌: 知能と情報, vol.30, no.4, pp.643–651, 2018.
- [4] 伊庭斉志, 遺伝的アルゴリズム, 医学出版, 2002.
- [5] Koza, John Reed, Genetic programming: on the programming of computers by means of natural selection, MIT Press, Cambridge, 1992.
- [6] 伊庭斉志, 遺伝的プログラミング入門, 東京大学出版会, 2001.
- [7] 平澤宏太郎, 大久保雅文, 片桐広伸, 胡敬焮, 村田純一, “蟻の行動進化における Genetic Network Programming と Genetic Programming の性能比較,” 電学論 C, vol.121, no.6, pp.1001–1009, 2001.
- [8] 丹治信, 伊庭斉志, “木構造のランダムサンプリングによる GP の個体生成手法の提案と評価,” 電学論 C, vol.130, no.5, pp.775–781, 2010.
- [9] Keiko Ono, Yoshiko Hanada, Masahito Kumano, and Masahiro Kimura, “Depth-dependent crossover in genetic programming with frequent trees,” 2012 IEEE International Conference on Systems (SMC 2012), Man, and Cybernetics, pp.359–363, Seoul, Korea, 2012.
- [10] 小野景子, 花田良子, 熊野雅仁, 木村昌弘, “頻出木と深さの情報を用いた照明制御のための遺伝的プログラミング,” 電学論 C, vol.133, no.11, pp.2044–2052, 2013.
- [11] Keiko Ono, Yoshiko Hanada, Masahito Kumano, and Masahiro Kimura, “Island model genetic programming based on frequent trees,” 2013 IEEE Conference on Evolutionary Computation (CEC 2013), pp.2988–2995, Cancun, Mexico, 2013.

- [12] Nguyen Quang Uy, Nguyen Thi Hien, Nguyen Xuan Hoai, and Michael O’Neill, “Improving the generalisation ability of genetic programming with semantic similarity based crossover,” Genetic Programming 13th European Conference (EuroGP 2010), pp.184–195, Istanbul, Turkey, 2010.
- [13] Edgar Galv’an-L’opez, Brendan Cody-Kenny, Leonardo Trujillo and Ahmed Kattan, “Using semantics in the selection mechanism in Genetic Programming: A simple method for promoting semantic diversity,” 2013 IEEE Congress on Evolutionary Computation (CEC 2013), pp.2972–2979, Cancun, Mexico, 2013.
- [14] Tomasz P. Pawlak, Bartosz Wieloch, Krzysztof Krawiec, “Semantic backpropagation for designing search operators in genetic programming,” Evolutionary Computation, IEEE Transactions, 19(3), pp.326–340, 2015.
- [15] 沢田賢治, 狩野均, “多目的手法を用いた構造化進化戦略による最適化問題の解法,” 計測自動制御学会 第30回知能システムシンポジウム, pp.91–96, 東京, 2003.
- [16] 松村康平, 花田良子, 小野景子, “木構造最適化におけるノードの依存関係を考慮した近傍探索に基づく多段階探索交叉,” 第26回インテリジェント・システム・シンポジウム (FAN 2016), pp.178–181, 大阪, 2016.
- [17] 杉本かい, 加納政芳, 中村剛士, “ヒューマノイドロボットの動作獲得のための進化的二分決定グラフの交叉操作,” 日本知能情報ファジィ学会誌: 知能と情報, vol.27, no.6, pp.845–853, 2015.
- [18] 菅沼雅徳, 土屋大樹, 白川真一, 長尾智晴, “遺伝的プログラミングを用いた階層的な特徴構築による画像分類,” 情報処理学会研究報告 第108回数理モデル化と問題解決 (MPS) 研究会, vol.2016-MPS-108, no.4, pp.1–6, 2016.
- [19] 峰崎敏昌, 上田祐彰, 高橋健一, “調整ノード付き Genetic Network Programming を用いた比較実験,” 電気情報関連学会中国支部 第60回 連合大会講演論文集, p.546, 広島, 2009.
- [20] Toshiyuki Morioka, Hiroaki Ueda, and Kenichi Takahashi, “Efficient Evolutionary Learning of Agent Behavior by Genetic Programming Using the Conditional Probabilities,” Proc. of 12th International Symposium on Advanced Intelligent System 2011 (ISIS 2011), pp.342–345, Suwon, Korea, 2011.

- [21] 伊藤崇, 高橋健一, 稲葉通将, “複数の木構造を持つ遺伝的プログラミングの改良,” 2013 IEEE SMC Hiroshima Chapter 若手研究会, pp.9–12, 広島市立大学, 2013.
- [22] 原裕一, 金川明弘, 山内仁, 高橋浩光, “異文化型島モデルを組み込んだGPにおける効率改善,” 信学技報, vol.106, no.344, pp.11–16, 2006.
- [23] Takashi Ito, Kenichi Takahashi, and Michimasa Inaba, “Experiments Assessing Learning of Agent Behavior Using Genetic Programming with Multiple Trees,” The 6th International Conference on Agents and Artificial Intelligence (ICAART 2014), pp.264–271, Angers, France, 2014.
- [24] 伊藤崇, 高橋健一, 稲葉通将, “自律エージェントに特化した遺伝的プログラミングの拡張,” 合同エージェントワークショップ& シンポジウム 2014 (JAWS 2014), pp.253–256, 宮崎, 2014.
- [25] 伊藤崇, 高橋健一, 稲葉通将, “自律エージェントのための遺伝的プログラミングの拡張,” 電子情報通信学会誌 (D), vol.J98-D, no.6, pp.905–915, 2015.
- [26] Takashi Ito, Kenichi Takahashi, and Michimasa Inaba, “Extension of Genetic Programming with Multiple Trees for Agent Learning,” The 8th International Conference on Computer and Electrical Engineering (ICCEE 2015), 12-pages, 2015.
- [27] 伊藤崇, 高橋健一, 稲葉通将, “複数の木構造を持つ遺伝的プログラミングの拡張,” 第20回日本知能情報ファジィ学会中国・四国支部大会, pp.37–42, 広島市立大学, 2015.
- [28] Takashi Ito, Kenichi Takahashi, and Michimasa Inaba, “Obtaining repetitive actions for genetic programming with multiple trees,” The 20th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2016), pp.120–128, 2016.
- [29] 伊藤崇, 高橋健一, 稲葉通将, “複数木構造を持つGPのための木の行動回数 of 獲得,” 2016 IEEE SMC Hiroshima Chapter 若手研究会, pp.17–22, 広島大学, 2016.
- [30] 伊藤崇, 高橋健一, 稲葉通将, “複数木構造を持つGPのための木の数の自動決定,” 情報処理学会第79回全国大会 (IPSJ 2017), pp.2-55–2-56, 名古屋大学, 2017.

- [31] Takashi Ito, Takahashi Kenichi, and Michimasa Inaba, “Obtaining the Number of Tree in the Evolution for Genetic Programming with Multiple Trees,” The 5th IIAE International Conference on Intelligent Systems and Image Processing 2017 (ICISIP 2017), pp.425–432, Hawaii, USA, 2017.
- [32] 伊藤崇, 高橋健一, 稲葉通将, “複数の木を持つ GP における自動化のための手法,” 日本知能情報ファジィ学会誌 : 知能と情報, vol.30, no.3, pp.571–580, 2018.
- [33] 伊庭齊志, 佐藤泰介, “システム同定アプローチに基づく遺伝的プログラミング,” 人工知能学会誌, vol.10, no.4, pp.590–600, 1995.
- [34] Shinji Eto, Shingo Mabu, Kotaro Hirasawa, and Takayuki Huruzuki, “Genetic Network Programming with Control Nodes,” 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp.1023–1028, Singapore, 2007.
- [35] Bertrand Mesot Eduardo, Eduardo Sanchez, Carlos-andres Pena, and Andres Perez-uribe, “SOS++: finding smart behaviors using learning and evolution,” The 8th International Conference on the Simulation and Synthesis of Living Systems (Artificial Life VIII), pp.264–273, Sydney, Australia, 2002.
- [36] 岩下誠, 伊庭齊志, “移民世代数と深さ依存型交叉を用いた並列分散 GP,” 情報学論 : 数理モデル化と応用, vol.43, no.SIG10, 2002.

研究業績

学術論文誌

- [1] 伊藤崇, 高橋健一, 稲葉通将, “自律エージェントのための遺伝的プログラミングの拡張,” 電子情報通信学会誌 (D) , vol.J98-D, no.6, pp.905–915, 2015.
- [2] Hidetoshi Okazaki and Takashi Ito, “Simulation of Illusion Based on Visual Characteristic of Spatial Frequency,” Journal of Advanced Control, Automation and Robotics (JACAR), 1 (1), pp.12–17, 2015.
- [3] 伊藤崇, 高橋健一, 稲葉通将, “複数の木を持つ GP における自動化のための手法,” 日本知能情報ファジィ学会誌 : 知能と情報, vol.30, no.3, pp.571–580, 2018.
- [4] 笠原和真, 二本健太, 伊藤崇, 高橋健一, 稲葉通将, “SVR を適用した FALCON によるトランプゲームに対する学習実験,” 日本知能情報ファジィ学会誌 : 知能と情報, vol.30, no.4, pp.643–651, 2018.

国際会議

- [5] Takashi Ito, Kenichi Takahashi, and Michimasa Inaba, “Extension of Genetic Programming with Multiple Trees for Agent Learning,” The 8th International Conference on Computer and Electrical Engineering (ICCEE 2015), 12-pages, 2015.
- [6] Takashi Ito, Kenichi Takahashi, and Michimasa Inaba, “Obtaining repetitive actions for genetic programming with multiple trees,” The 20th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2016), pp.120–128, 2016.
- [7] Takashi Ito, Takahashi Kenichi, and Michimasa Inaba, “Obtaining the Number of Tree in the Evolution for Genetic Programming with Multiple Trees,” The 5th IIAE International Conference on Intelligent Systems and Image Processing 2017 (ICISIP 2017), pp.425–432, Hawaii, USA, 2017.
- [8] Takashi Ito, Shuhei Kotake, Takahashi Kenichi, and Michimasa Inaba, “Detecting Concentration of Students Using Kinect in E-learning,” The 5th IIAE Inter-

national Conference on Intelligent Systems and Image Processing 2017 (ICISIP 2017), pp.451–456, Hawaii, USA, 2017.

その他論文

- [9] 伊藤崇, 高橋健一, 稲葉通将, “複数の木構造を持つ遺伝的プログラミングの拡張,” 第20回日本知能情報ファジィ学会中国・四国支部大会, pp.37–42, 広島市立大学, 2015.
- [10] 伊藤崇, 高橋健一, 稲葉通将, “複数木構造を持つ GP のための木の行動回数の獲得,” 2016 IEEE SMC Hiroshima Chapter 若手研究会, pp.17–22, 広島大学, 2016.
- [11] 川口諄, 伊藤崇, 高橋健一, 稲葉通将, “e ラーニングにおける休憩の過ごし方の影響,” 2016年電子情報通信ソサイエティ大会, p.151, 北海道大学, 2016.
- [12] 伊藤崇, 高橋健一, 稲葉通将, “複数木構造を持つ GP のための木の数の自動決定,” 情報処理学会第79回全国大会 (IPSJ 2017), pp.2-55–2-56, 名古屋大学, 2017.
- [13] 伊藤崇, 伊東啓裕, 高橋健一, 稲葉通将, “Kinect を用いた学習者の状態識別実験,” 2017 IEEE SMC Hiroshima Chapter 若手研究会, pp.26–29, 広島工業大学, 2017.
- [14] 笠原和真, 伊藤崇, 高橋健一, 稲葉通将, “トランプゲームに対する SVR を適用した FALCON の改良,” Human-Agent Interaction Symposium 2017 (HAI シンポジウム 2017) , 6-pages, 金沢歌劇座, 2017.
- [15] 伊藤崇, 神谷健太, 高橋健一, 稲葉通将, “Kinect を用いた学習者の状態識別の改良,” 2018 IEEE SMC Hiroshima Chapter 若手研究会, pp.3–8, 広島工業大学, 2018.

受賞

- [15] Takashi Ito, Takahashi Kenichi, and Michimasa Inaba, “Obtaining the Number of Tree in the Evolution for Genetic Programming with Multiple Trees,” The 5th IIAE International Conference on Intelligent Systems and Image Processing 2017 (ICISIP 2017), pp.425–432, Hawaii, USA, 2017. Best Presentation Award.