

ノードの種類・内容も探索対象とする確率モデル構築型遺伝的ネットワークプログラミング

Probabilistic Model Building Genetic Network Programming for Searching Contents of Node and Connection Genes

元木 達也
Tatsuya Motoki

新潟大学工学部
Faculty of Engineering, Niigata University
motoki@ie.niigata-u.ac.jp

小林 涼
Ryo Kobayashi

日本アルゴリズム株式会社
Nippon Algorithm Company LTD.

keywords: genetic network programming (GNP), estimation of distribution algorithm (EDA), probabilistic model building genetic network programming (PMBGNP).

Summary

In this paper, we propose an estimation of distribution algorithm (EDA) for finding a good individual in genetic network programming (GNP). Our EDA is an extension of Li et al.(2009)'s probabilistic model building genetic network programming (PMBGNP). Each individual in GNP has a directed graph structure composed of a start node, judgment nodes, processing nodes and arcs between nodes. While Li et al.'s PMBGNP builds probabilistic distributions of terminal points of arcs, our PMBGNP also builds probabilistic distributions of function assignments to nodes as well as distributions of terminal points of arcs. Our PMBGNP searches over the space of possible combinations of function assignments to nodes and terminal points of arcs, and so dispenses with any breakdown of the number of nodes. Two maze problems and the 11-multiplexer problem are used to evaluate the performance of the proposed search method. The experimental results show that our PMBGNP finds the optimum solutions of the tested problems in some moderate probability.

1. はじめに

片桐, 平澤らは, ロボットを始めとするエージェントの知的な動作列を進化的に探索するための手法を提案し, それを遺伝的ネットワークプログラミング (genetic network programming, GNP) と呼んだ [Katagiri 00, Hirasawa 01]. GNP においては, 解候補である個体は開始ノード (start node), 判定ノード (judgment node), 処理ノード (processing node) の 3 種類のノードが弧 (arc) で結ばれたグラフ構造をもち, エージェントの動作プログラムとして扱われる. 3 種類のノードの内, 開始ノードは動作の起点を示す役割だけをもち, そこから出ている弧に沿って別のノードに遷移する. 処理ノードは環境に対して作用するエージェントの何らかの基本動作を表し, 動作後にはこの処理ノードから 1 本だけ出ている弧に沿って別のノードに遷移する. 判定ノードはエージェントの置かれた環境の情報を基に何らかの判断を行う基本動作を表し, その判断結果に基づいてこの判定ノードから出ている複数の弧のどれに沿って次のノードに至るのかを決める. この様に GNP 個体はエージェントの動作列の生成を進め

るものである. GNP では個体はグラフ構造をもつため, ノードを再利用する機能を本来的に備えている. そのため, ノード再利用を適切に行うことが出来れば, 比較的にコンパクトな構造でエージェントの行動規則を表すことが出来る. その後, GNP は色々な問題に適用され [(社) 電気学会 10, 12.7 節, 15.5 節], 強化学習の考えを用いるなどの拡張も提案されている [Mabu 02, Mabu 07].

一方, 最近は確率モデル構築型 EA (probabilistic model building EA) の研究も盛んである [(社) 電気学会 10, 11 章]. この種の EA では, 従来の EA で行われている親選択・交叉・突然変異という処理の代わりに, ①良好個体群の選出, ②良好個体群の元になる確率分布の推定, ③推定結果である確率分布 (確率モデル) に従った新しい解候補の生成, といった処理を行うことで解探索を進める. この考えで GNP 個体の探索を行う研究としては Li らによるものがある [Li 09, Li 10a, Li 10b, Li 11a, Li 11b, Li 14]. 彼らは確率モデルを構築する方法として, 良好個体群内の個々の遺伝子の頻度分布に基づくもの [Li 09, Li 10a, Li 10b] や, 強化学習に基づくもの [Li 11a, Li 11b, Li 14] について調べた. ただ, 彼らの確率モデル構築

型 GNP(probabilistic model building GNP,PMBGNP) においては, 良好個体群の元になる確率分布としてノード間の弧の分岐先に関するものだけを考慮する. 従って, 色々な種類・内容 (基本動作) のノードを必要な個数予め用意して固定し, 接続先の部分だけを進化的に探索することになる. これでは, PMBGNP 手法を適用する際に, 各々の種類・内容 (基本動作) のノードを何個用意するかに関して試行錯誤することに成り兼ねない. そこで, この論文ではノードの種類・内容も探索対象とする PMBGNP を提案する. そして, 2 種類の迷路問題と 11 マルチプレクサ問題に適用し, その探索性能を実験的に調べる.

以下, 2 章では GNP の概略を, 3 章では Li らの提案した PMBGNP を紹介し, 4 章ではノードの種類・内容 (基本動作) も探索する PMBGNP を提案する. 続く 5 章では実験に用いるテスト問題について述べ, 主要なパラメータを調整した実験結果, ノードの種類・内容を固定して弧の接続先だけを探索する場合との比較を示す. 6 章では多様性維持の様子, 個体群サイズ設定の影響, 良好個体群選出の手法, 確率モデルを明示的に構成せずに PMBGNP と同等の探索を行う方法についての考察を行う. 最後の 7 章ではまとめを行う.

2. 遺伝的ネットワークプログラミング (GNP)

片桐, 平澤らは, ロボットを始めとするエージェントの知的な動作列を進化的に探索するための手法を提案し, それを遺伝的ネットワークプログラミング (genetic network programming,GNP) と呼んだ [Katagiri 00, Hirasawa 01]. GNP においては, 解候補である個体は図 1 に例示される様に, 複数のノードが弧 (arc) で結ばれたグラフ構造をもつ. ノードとしては, 開始ノード (start node), 判定ノード (judgment node), 処理ノード (processing node) の 3 種類を考える. この内, 開始ノードは動作の起点を示す役割だけをもち, そこから出ている弧の行き先のノードで指示された動作がエージェントの最初の動作となる. 処理ノードは環境に対して作用するエージェントの何らかの基本動作を表し, 動作後にはこの処理ノードから 1 本だけ出ている弧に沿って行く事によりエージェントの次の動作を指示するノードに遷移する. 判定ノードはエージェントの置かれた環境の情報を基に何らかの判断を行う基本動作を表し, その判断結果に基づいてこの判定ノードから出ている複数の弧のどれに沿って次のノードに至るのかを決める. この様に, GNP 個体はエージェントの動作列の生成を進めるものであり, エージェントは (通常の流れ図と同じ様に) 開始ノードから始めてノードを辿り, ノードで指示された基本動作を順次行うことにより行動を進めることができる. GNP 個体は一見すると有限状態機械 (FSM) の様でもあるが, 環境情報を参考にするのは常時 (FSM の場合) ではなく判定ノード上に制御が来ている時だけであり, また基本動作を行うのは状態

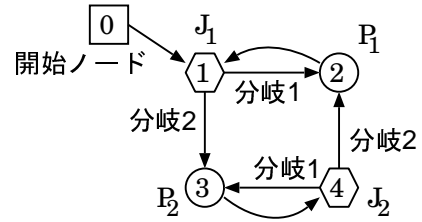


図 1 GNP 個体の例. (ここでは, □は開始ノード, 六角形の箱は判定ノード, ○は処理ノード, ノード内の数値はノード番号, J_i は i 番目の基本判定動作, P_i は i 番目の基本処理動作を表す. 従って, この図は 1 番目,...,4 番目のノードにそれぞれ J_1, P_1, P_2, J_2 という基本動作が割り当てられていることを表す.)

ノード 0	0 0 0	1 0	—
ノード 1	1 1 1	2 0	3 0
ノード 2	2 1 5	1 0	—
ノード 3	2 2 5	4 0	—
ノード 4	1 2 1	3 0	2 0

図 2 図 1 の GNP 個体の遺伝子構造. (ここでは, 各基本判定動作に要する時間=1, 各基本処理動作に要する時間=5, 各ノード分岐に要する時間=0 と仮定する.)

遷移時 (FSM の場合) ではなくノード上に制御が来ている時である.

GNP 個体の遺伝子構造: 図 1 の GNP 個体の遺伝子構造を図 2 に例示する. 一般に, GNP 個体の i 番目のノードとそこから出ている弧は次の形の遺伝子列で表され, GNP 個体全体はこの様な遺伝子列を並べた配列で表される.

K_i	ID_i	d_i	C_{i1}	d_{i1}	C_{i2}	d_{i2}
ノード遺伝子			接続遺伝子 (i.e. 弧に関連する遺伝子)				

ここで, K_i はノードの種類を表す遺伝子で, 開始ノードの場合は $K_i = 0$, 判定ノードの場合は $K_i = 1$, 処理ノードの場合は $K_i = 2$ である. ID_i は可能な基本判定動作/基本処理動作として予め用意されている中の何番目のものを用いるかを表し, d_i はその動作に伴う遅れ (消費) 時間を表す. また, 弧に関連した接続遺伝子の部分において, C_{ij} は i 番目のノードから出ている j 番目の弧の分岐先のノード番号, d_{ij} はその分岐に伴う遅れ (消費) 時間を表す.

GNP は色々な問題に適用され [(社) 電気学会 10, 12.7 節, 15.5 節], 各ノードの種類 (K_i 部) を固定した上で強化学習の考えを用いて動作内容 (ID_i 部) を動的に決められるようにするなどの拡張も提案されている [Mabu 02, Mabu 07]. しかし, 文献 [Katagiri 00] の GNP や文献 [Mabu 07] で比較対象として説明されている GNP においては, 各々の基本動作を行う判定ノード, 処理ノードを十分な

個数もった初期個体を構成した上で、接続遺伝子の部分だけを進化的に探索している。

3. 確率モデル構築型 GNP(PMBGNP)

最近は、確率モデル構築型 EA(probabilistic model building EA, EA based on probabilistic model, estimation of distribution algorithm) の研究も盛んである [Pelikan 02, (社)電気学会 10, 11 章]. この種の EA は、従来の EA の枠組み中の親選択・交叉・突然変異の部分、①良好個体群の選出、②良好個体群の元になる確率分布の推定、③推定結果である確率分布(確率モデル)に従った新しい解候補の生成、といった手順で置き換えて得られた探索手法であり、その探索能力は確率分布を推定する際に個体中の遺伝子要素間の依存関係をどの程度正確に反映できるかによって決まる。一般には、個体中の遺伝子要素間の依存関係を正確に反映した確率分布を推定することは簡単ではなく、かと言って遺伝子要素間の依存関係を全然考慮に入れない確率モデル(i.e. 確率分布の推定結果)を使って新しい解候補を生成したのでは、容易な問題に対してしかうまく最適解を探し出せない。そこで、良好個体群の中から2つの遺伝子要素間の依存関係を多数見出し(i.e. 遺伝子要素間にある依存関係の構造を推定し)、それに沿って確率分布を構築したり、更には、3つ以上の遺伝子要素間の依存関係も考慮して確率分布を構築したり、といった研究が行われている。

GNP に関しては、確率モデル構築型のものが Li らによって提案されている [Li 09, Li 10a, Li 10b, Li 11a, Li 11b, Li 14]. まず文献 [Li 09, Li 10b] では、最も基本的な確率モデル構築型 GNP(probabilistic model building genetic network programming, PMBGNP) として、遺伝子要素間の依存関係を全く考慮に入れない確率モデルを構築するものが考えられた。すなわち、接続遺伝子中の各々の C_{ij} (i 番目のノードから出ている j 番目の分岐先のノード番号) を確率変数と見る。そして、良好個体群 P_{good} 内で i 番目のノードから出ている j 番目の分岐先のノードの番号が c_{ij} となっている割合 $\text{freq}(C_{ij} = c_{ij} | P_{\text{good}})$ を用いて、良好個体群の分布をモデル化した確率分布を

$$\begin{aligned} & \Pr(C_{01} = c_{01}, C_{11} = c_{11}, \dots, C_{ij} = c_{ij}, \dots) \\ & \approx \prod_{i: \text{ノード番号}} \prod_{j: \text{分岐番号}} \Pr(C_{ij} = c_{ij}) \\ & \approx \prod_{i: \text{ノード番号}} \prod_{j: \text{分岐番号}} \text{freq}(C_{ij} = c_{ij} | P_{\text{good}}) \quad (1) \end{aligned}$$

と近似し、これを用いて新しい解候補の生成を行う探索手法が考えられた。(GNP 個体としては、ノード数とノード遺伝子部、遅れ時間 (d_{ij}) の部分が固定されているので、個体中で不確定な残りの遺伝子群 $\{C_{ij}\}$ を式 (1) の確率分布に従って決めれば個体の全遺伝子が確定することになる。) また、文献 [Li 10b, Li 10a] では良好個体群中の接

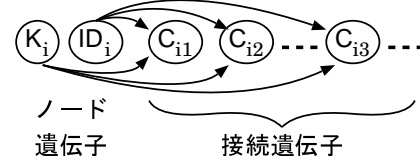


図3 ここで仮定する K_i, ID_i と C_{i1}, C_{i2}, \dots の間の依存関係。(各々の C_{ij} の確率分布が K_i と ID_i に依存していることを表す。)

続遺伝子の情報だけでなく個体評価中に実際に使われた接続遺伝子の情報も考慮して周辺分布 $\Pr(C_{ij} = c_{ij})$ の近似を求める手法が考えられ、文献 [Li 11a, Li 11b, Li 14] では強化学習の考えを利用して周辺分布 $\Pr(C_{ij} = c_{ij})$ の近似を求める手法が考えられた。

しかし、Li らの提案した PMBGNP においては、いずれも、色々な種類・内容(基本動作)のノードを必要な個数予め用意して固定し、接続先の部分だけを進化的に探索する。これでは、PMBGNP 手法を適用する際に、各々の種類・内容(基本動作)のノードを何個用意するかに関して試行錯誤することに成り兼ねない。そこで、この論文ではノードの種類・内容も探索対象とする PMBGNP を考える。

4. ノードの種類・内容も探索する PMBGNP

確率モデルの構築: ノードの種類・内容も探索対象とする場合は、当然、分岐先のノード番号 (C_{ij}) だけでなくノードの種類 (K_i) や基本動作の番号 (ID_i) も確率変数として扱わなければならない。しかし、基本動作に要する時間 (d_i) や分岐に要する時間 (d_{ij}) はノードの種類と内容の組 (K_i, ID_i) の表す基本動作に依存して一意に決まるものと考えられるので、これらは確率モデルを記述する際に考慮に入れない。以上の考えの下、ここでは確率変数 K_i, ID_i と C_{i1}, C_{i2}, \dots の間に図3の様な依存関係を仮定し、良好個体群 P_{good} 内で $(K_i, ID_i) = (k_i, id_i)$ となっている割合 $\text{freq}(K_i = k_i, ID_i = id_i | P_{\text{good}})$ と良好個体群 P_{good} に属し $(K_i, ID_i) = (k_i, id_i)$ という条件を満たす個体群中で $C_{ij} = c_{ij}$ となっている割合 $\text{freq}(C_{ij} = c_{ij} | K_i = k_i, ID_i = id_i, P_{\text{good}})$ を用いて、選出された良好個体群 P_{good} の分布をモデル化した確率分布を

$$\begin{aligned} & \Pr(C_{01} = c_{01}, \dots, (K_i, ID_i) = (k_i, id_i), C_{i1} = c_{i1}, \dots) \\ & \approx \prod_{i: \text{ノード番号}} \Pr((K_i, ID_i) = (k_i, id_i), C_{i1} = c_{i1}, \dots) \\ & \approx \prod_{i: \text{ノード番号}} \Pr((K_i, ID_i) = (k_i, id_i)) \prod_{j: \text{分岐番号}} \Pr(C_{ij} = c_{ij} | (K_i, ID_i) = (k_i, id_i)) \\ & \approx \prod_{i: \text{ノード番号}} \text{freq}(K_i = k_i, ID_i = id_i | P_{\text{good}}) \prod_{j: \text{分岐番号}} \text{freq}(C_{ij} = c_{ij} | K_i = k_i, ID_i = id_i, P_{\text{good}}) \quad (2) \end{aligned}$$

と近似し、これを用いて新しい解候補の生成を行うことを考える。

解探索の手順:PMBGNP も確率モデル構築型 EA の一種なので、その解探索手順の大枠は、従来の EA 手順中の親選択・交叉・突然変異の部分を「良好個体群の選出・確率モデルの構築・確率モデルに沿った解候補のサンプリング」という作業で置き換えて得られ、次の形で表すことができる。

```

個体のランダムな生成を繰り返す
    ことによって初期個体群  $P(0)$  を構成;
 $P(0)$  内の個々の個体の適合度を評価;
for ( $t \leftarrow 0$ ; 終了条件を満たさない;  $t \leftarrow t + 1$ ) {
     $P_{\text{good}} \leftarrow P(t)$  内の良好個体の集合;
     $P_{\text{good}}$  の分布を近似する確率モデル  $M$  を構築;
    モデル  $M$  に沿ったサンプリングを繰り返す
        ことによって次世代個体群  $P(t+1)$  を構成;
     $P(t+1)$  内の個々の個体の適合度を評価;
}

```

これを基本的な枠組みとして、ここでは更に次の方針 1~4 を採用する。

(方針 1) 初期個体の生成は、① K_i を (1 と 2 が等確率で出る) 一様分布に従ってランダムに決定、② ID_i として可能な値を一様分布に従ってランダムに決定、③ 関連する C_{i1}, C_{i2}, \dots について可能な値 (0 と i 以外のノード番号) をランダムに決定、という作業を各々のノード番号 $i=1, 2, \dots$ に対して行い、 $K_0 = ID_0 = 0, C_{01} = 1$ と設定することにより行う。

(方針 2) 個体群 $P(t)$ 中の一定の割合 ($\text{rate}_{\text{good}}$) の上位個体群を良好個体群 P_{good} と考える。

(方針 3) 構築される確率モデル M が世代と伴に不安定に変化するのを避けるために、Li ら [Li 09, Li 10a, Li 10b] は前世代の確率分布も慣性項として考慮する “exponential smoothing method” と呼ばれる方法を採用していたが、ここでは個体群 $P(t)$ 中の一定の割合 ($\text{rate}_{\text{elite}}$) の上位個体を次世代個体としてそのまま残すことにする。

(方針 4) 個体の多様性維持を図るために、確率モデル M に沿ったサンプリングを忠実に行うのではなく、 c_{node} , c_{connect} をパラメータとした

$$p_{\text{node}} = \frac{1}{\text{個体中の開始ノードを除くノード数}} \times c_{\text{node}} \quad (3)$$

$$p_{\text{connect}} = \frac{1}{\left(\frac{M \text{ に沿ったサンプリングで得られる個体中の弧の平均本数} \right)} \times c_{\text{connect}} \quad (4)$$

という量を考える。そして、次世代個体中のノード遺伝子 (K_i, ID_i) を決める際には確率 p_{node} で一様分布に沿ってランダムに決め、残りの確率 ($1 - p_{\text{node}}$) で確率モデル M に沿ってランダムに決める。また、次世代個体中の接続遺伝子 (C_{ij}) を決める際には確率 p_{connect} で一様分布に沿ってランダムに決め、残りの確率 ($1 - p_{\text{connect}}$) で確率モデル M に沿ってランダムに決める。従って、サンプリングによって構成される個体中で、平均 c_{node}

個のノード遺伝子と平均 c_{connect} 個の接続遺伝子が一様分布に沿ってランダムに決定されることになる。

これらの方針 1~4 を取り込むことによって、ここで考える PMBGNP の探索手順は次の形に表すことができる。

ノードの種類・内容も探索する PMBGNP の探索手順 :
 パラメータ Num_nodes(個体中のノード数), Popsiz(個体群サイズ) の値を決め、個体を保持するための配列 $\text{ind}[0] \sim \text{ind}[\text{Popsiz}-1]$ を用意する。また、個数を数える作業用に大きさが基本動作数の配列 $\text{countA}[]$ と、大きさが基本動作数 \times 最大分岐数 \times ノード数の配列 $\text{countB}[][]$ を用意し、番号 i のノードにおけるノード遺伝子の頻度分布 $\{\text{freq}(K_i = k_i, ID_i = id_i | P_{\text{good}})\}$ (を表すオブジェクト) を保持するための領域 $\text{dist}_{\text{node}}[i]$ と、番号 i のノードから出ている j 番目の弧の分岐先に関する条件付き頻度分布 $\{\text{freq}(C_{ij} = c_{ij} | K_i = k_i, ID_i = id_i, P_{\text{good}})\}$ (を表すオブジェクト) を保持するための領域 $\text{dist}_{\text{connect}}[i][k_i, id_i][j]$ を確保する。そして、探索を制御する 4 つのパラメータ $\text{rate}_{\text{elite}} \in [0, 1)$, $\text{rate}_{\text{good}} \in (0, 1)$, $c_{\text{node}} \geq 0$, $c_{\text{connect}} \geq 0$ の値を決め、次の手順に従って解探索を行う。

個々の個体 $\text{ind}[i]$ を (方針 1) の手順に従ってランダムに生成することによって初期個体群 $P(0) = \{\text{ind}[0], \dots, \text{ind}[\text{Popsiz}-1]\}$ を構成 ;

$P(0)$ 内の個々の個体の適合度を評価;

for ($t \leftarrow 0$; 終了条件を満たさない; $t \leftarrow t + 1$) {

// エリート個体, 良好個体の選別

$n_{\text{elite}} \leftarrow \lfloor \text{Popsiz} \times \text{rate}_{\text{elite}} \rfloor$;

$n_{\text{good}} \leftarrow \lfloor \text{Popsiz} \times \text{rate}_{\text{good}} \rfloor$;

次の 2 つの条件を満たす様に個体 $\text{ind}[0] \sim \text{ind}[\text{Popsiz}-1]$ を並べ替える。

(条件 1) ($\forall i < n_{\text{elite}})(\forall j \geq n_{\text{elite}}) \text{ind}[i]$ の適合度は $\text{ind}[j]$ の適合度以上

(条件 2) ($\forall i < n_{\text{good}})(\forall j \geq n_{\text{good}}) \text{ind}[i]$ の適合度は $\text{ind}[j]$ の適合度以上 ;

// $P_{\text{good}} = \{\text{ind}[0], \dots, \text{ind}[n_{\text{good}}-1]\}$ を基に

// 確率モデルを構築

for (each ノード番号 i) {

配列 $\text{countA}, \text{countB}$ の各要素を 0 に設定;

for (each 良好個体番号 k) {

$k_i \leftarrow \text{ind}[k]$ の i 番目のノードの種類として設定された値;

$id_i \leftarrow \text{ind}[k]$ の i 番目のノードの基本動作番号として設定された値;

$\text{countA}[(k_i, id_i)] \leftarrow \text{countA}[(k_i, id_i)] + 1$;

for (each 分岐番号 j) {

$\text{dest_node} \leftarrow \text{ind}[k]$ の i 番目のノードから出ている j 番目の分岐先として設定された値;

$\text{countB}[(k_i, id_i)][j][\text{dest_node}] \leftarrow \text{countB}[(k_i, id_i)][j][\text{dest_node}] + 1$;

}

}

配列 $countA[]$ の情報を基に頻度分布 $freq(K_i = k_i, ID_i = id_i | P_{good})$ を構成して $dist_{node}[i]$ に格納

for (each ノード遺伝子の組 (k_i, id_i))

for (each 分岐番号 j)

$countA[(k_i, id_i)]$ と配列 $countB[(k_i, id_i)][j]$ の情報を基に頻度分布 $freq(C_{ij} = c_{ij} | K_i = k_i, ID_i = id_i, P_{good})$ を構成して $dist_{connect}[i][(k_i, id_i)][j]$ に格納

}

//確率モデルに沿って新規個体を生成

for ($k \leftarrow n_{elite}; k < Popsiz; k \leftarrow k + 1$)

//式 (2) を基に新規個体を生成し $ind[k]$ に格納

for (each ノード番号 $i > 0$)

$ind[k]$ の i 番目のノードにおけるノード遺伝子について、式 (3) によって定まる確率 p_{node} で一様分布に基づいて確率的に設定し、残りの確率 $(1 - p_{node})$ で $dist_{node}[i]$ に基づいて確率的に設定

$k_i \leftarrow ind[i]$ の i 番目のノードの種類として設定された値;

$id_i \leftarrow ind[i]$ の i 番目のノードの基本動作番号として設定された値;

for (each 分岐番号 j)

$ind[k]$ の i 番目のノードにおける j 番目の接続遺伝子については、式 (4) によって定まる確率 $p_{connect}$ で (0 と i 以外が等確率で出る) 一様分布に基づいて確率的に設定し、残りの確率 $(1 - p_{connect})$ で $dist_{connect}[i][(k_i, id_i)][j]$ に基づいて確率的に設定

}

//個体評価

$ind[n_{elite}] \sim ind[Popsiz - 1]$ の適合度を評価;

}

この探索手順は個体群サイズや最大世代数といったパラメータの他に、次世代に残すエリートの割合 ($rate_{elite}$)、良好個体と考える割合 ($rate_{good}$)、ノード遺伝子を一様分布に沿って決める確率を制御する係数 (c_{node})、接続遺伝子を一様分布に沿って決める確率を制御する係数 ($c_{connect}$)、個体中のノード数 (Num_nodes) という 5 つのパラメータをもつことになる。

5. 実験

本章では、4 章で提案した手法の評価実験を行う。まず 5.1 節で実験に用いた 3 つのテスト問題を説明し、5.2 節で実装上のメモ、探索手法のパラメータ設定について述べ

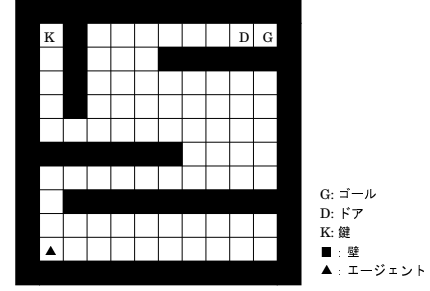


図 4 迷路問題 1 (鍵・ドア問題, 初期状態としてエージェント▲は北方向を向いていると仮定)

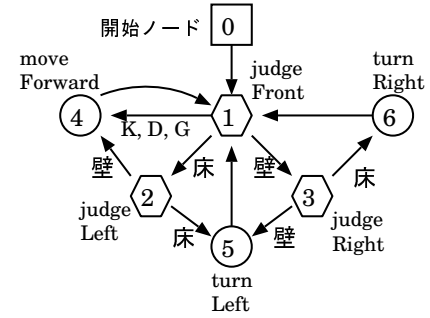


図 5 迷路問題 1 における最適解の例

る。5.3 節～5.4 節では予備実験として 3 つのテスト問題で良好な探索結果をもたらす $rate_{elite}$, $rate_{good}$, c_{node} , $c_{connect}$ に関する共通のパラメータ設定を探る。そして、5.5 節で探索対象である GNP 個体のノード数を変化させた実験を行い、最後の 5.6 節でノードの種類・内容を固定し弧の接続先だけを探索する場合との比較を行う。

5.1 対象問題

性質の異なる問題として迷路問題 (2 題), 11 マルチプレクサ問題の 2 種類を選び、これらを評価実験に用いた。

迷路問題 1 (鍵・ドア問題): 2 次元格子状の空間上に壁 (障害物), ゴール (G), ドア (D), 鍵 (K) が配置された図 4 の環境の中を 1 体のエージェントに動いてもらう。ここでは、エージェントが図 4 の左下のマスに北向きに配置された状態から始め、内部に保持した GNP 個体の生成する動作系列に沿ってマスの移動等を行うということを想定し、出来るだけ少ないステップ数でエージェントがゴール地点 G (右上) にたどり着ける様な GNP 個体を探し出すことを考える。この迷路においてはドア地点 D (右上) を通り抜けるためには地点 K (左上) に置かれた鍵を予め取っておかなければならないという制約が課せられているため、エージェントには地点 K, D, G を順に通ってもらうことになる。エージェント内部の GNP 個体で可能な基本処理動作としては moveForward (前進), turnRight (右に 90° 回転), turnLeft (左に 90° 回転) の 3 つを考える。また、可能な基本判定動作としては, judgeFront (すぐ前のマスの状態 (床, 壁, 鍵, ドア, or ゴール) に応じて 5 つの分岐先の内の 1 つに分岐), judgeRear (すぐ後ろのマスの状態に応じて 5 つの分岐先の内の 1 つに分岐), judgeLeft (す

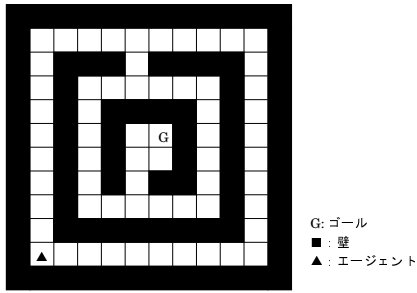


図 6 迷路問題 2(部分観測迷路, 初期状態としてエージェント▲は北方向を向いていると仮定)

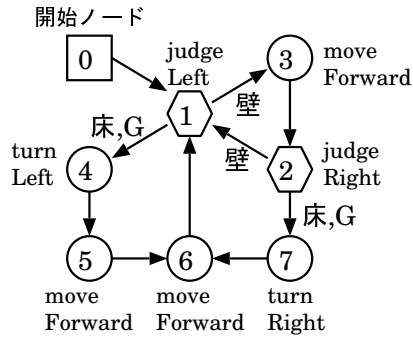


図 7 迷路問題 2 における最適解の例

ぐ左のマスの状態に応じて 5 つの分岐先の内の 1 つに分岐), judgeRight (すぐ右のマス状態に応じて 5 つの分岐先の内の 1 つに分岐), の 4 つを考える. そして, GNP 個体の生成する基本動作列を前から順に見て, 5 回の連続した基本判定動作列, もしくは 4 回以下の連続した基本判定動作列とそれに続く 1 回の基本処理動作, の各々をエージェントの 1 ステップの動作として数えることにする. GNP 個体の適合度は, GNP 個体の生成する基本動作列を用いてエージェントを最大 300 ステップ動作させた時に, ゴール地点もしくは最大ステップ数に至るまでにエージェントの費やしたステップ数を用いて次により定める.

$$\text{適合度} = \begin{cases} 300 - \text{費やしたステップ数} & \text{if ゴールに到達} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

この問題設定の下では, 長さが 47 の基本処理動作列でエージェントは出発点から地点 K, 地点 D を経てゴール地点に至ることが可能であるので, 最大適合度は $300 - 47 = 253$ となる. この最大適合度をもたらす GNP 個体の例を図 5 に示す.

迷路問題 2 (部分観測迷路): 迷路問題 1 と同様の迷路問題を考える. ここでの初期配置は図 6 の通りで, 鍵やドアは無い. エージェントはこの状態から始めてゴール地点 (G) を目指すことになる. この迷路においても, 迷路問題 1 の場合と同様に, 内部に保持した GNP 個体の生成する動作列に沿ってエージェントがマスの移動等を行うことを想定し, 出来るだけ少ないステップ数でエージェントがゴール地点 G(中央) にたどり着ける様な GNP

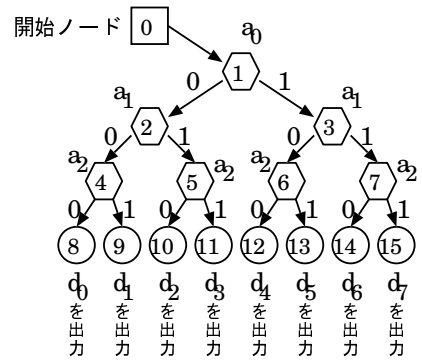


図 8 11 マルチプレクサ問題における最適解の例

個体を探し出すことを考える. GNP 個体を構成する基本動作に関しても, 迷路問題 1 で考えたものをそのまま用いる. すなわち, 基本処理動作としては moveForward , turnRight , turnLeft の 3 つを考え, 基本判定動作としては judgeFront , judgeRear , judgeLeft , judgeRight の 4 つを考える. ただ, ここでの迷路においては各マスにドアや鍵が配置されることはないので, マスの状態としては床, 壁, ゴールの 3 種類だけが可能となる. それゆえ, 各々の基本判定動作の分岐先としては 5 箇所ではなく 3 箇所だけが用意される. ステップ数や適合度に関しても迷路問題 1 と同じ考え方を採用し, GNP 個体の適合度は式 (5) により定める. この問題の場合は, 長さが 35 の基本処理動作列でエージェントは出発点からゴール地点に至ることが可能であるので, 最大適合度は $300 - 35 = 265$ となる. この最大適合度をもたらす GNP 個体の例を図 7 に示す.

11 マルチプレクサ問題 [Kozs 92]: 11 マルチプレクサはアドレス部 3 ビット $a_0 \sim a_2$ とデータ部 8 ビット $d_0 \sim d_7$ の合計 11 ビットの入力から成るブール関数で, アドレス部の指定したアドレス $\alpha = \sum_i a_i \times 2^i$ を基にそのアドレスのデータ値 d_α を出力する関数 $\text{multiplexer11}(a_0, a_1, a_2, d_0, \dots, d_7) = d_{4a_2+2a_1+a_0}$ である. 文献 [Kozs 92] に載っている元々の 11 マルチプレクサ問題を参考に, ここでは, その全ての入力の種類 (2048 組) に対して 11 マルチプレクサ関数と同じ出力をもたらす GNP 個体を探し出すことを考える. 可能な基本処理動作としては d_0 の値を出力する動作, d_1 の値を出力する動作, ..., d_7 の値を出力する動作, の 8 つを考える. また, 可能な基本判定動作としては a_0 の値を調べその値 (0 または 1) に応じて用意された 2 つの分岐先の内の 1 つに分岐する動作, a_1 の値を調べその値に応じて分岐する動作, a_2 の値を調べその値に応じて分岐する動作, の 3 つを考える. GNP 個体の動作は, 与えられた入力値の組に対してノード遷移を繰り返した後に基本処理ノードに至り, そこで出力を出した時点で終了すると考える. それゆえ, ここでの探索目標となるのは, 図 8 に例示されている様な, 基本処理ノードが配置された葉ノードと基本判定動作が配置された葉でないノードから成る木構造を含む GNP 個体であ

る．特に，ここでは，目的の出力特性を満たす GNP 個体の中で，各入力に対して出力までに為される基本判定動作の回数が最小 (3 回) になっているものを探し出すことを願って，GNP 個体 ind を入力事例 c に対して動作させた時に開始ノードから最初の処理ノードに至るまでのノード遷移の回数 $\text{num_steps}(ind, c)$ を用いて，入力事例 c に対する GNP 個体 ind の動作の評価値 $f(ind, c)$ を

$$f(ind, c) = \begin{cases} 0 & \text{if } ind \text{ は } c \text{ に対して不正解を出す} \\ 0 & \text{if } \text{num_steps}(ind, c) > 10 \\ (10 - \text{num_steps}(ind, c))/6 & \text{otherwise} \end{cases} \quad (6)$$

と定める．そして，GNP 個体 ind の適合度を次により定める．

$$ind \text{ の適合度} = \sum_{\text{入力事例: } c} f(ind, c) \quad (7)$$

常に正しく振舞う個体の場合 $f(ind, c)$ の最大値は 1 となり，GNP 個体の最大適合度は 2048 となる．この最大適合度をもたらす GNP 個体の例を図 8 に示す．(木構造は本来の GNP の探索対象ではないが，目的の動作を果たす GNP 個体を構成するために必要な基本動作の個数が 1~4 個と様々であるため，必要な基本動作が割り当てられたノードが必要な個数 GNP 個体の中に組込まれるかどうかを調べるために，この問題を評価実験に用いる．)

5.2 実装上のメモ，実験パラメータの設定

11 マルチプレクサ問題における適合度計算の実装：標準的な GP[Koza 92] においては $d_0 \sim d_7$ も $a_0 \sim a_2$ も終端子として扱われ，個体はブール式を表す木として処理されるので，個々の入力事例を別個に扱って適合度計算を行う必要がある．しかし，ここでの問題設定においては $a_0 \sim a_2$ は判定動作， $d_0 \sim d_7$ は処理動作として扱われ，個体は決定木として処理される．入力事例 $c = (a_0, a_1, a_2, d_0, d_1, \dots, d_7)$ に対する GNP 個体 ind の評価値 $f(ind, c)$ を求めるためには，0 個以上の判定ノードを経て処理ノードに至る経路を調べる．この経路が

開始ノード $\rightarrow a_{n_1}$ の値に関する判断ノード
 $\rightarrow a_{n_2}$ の値に関する判断ノード
 $\rightarrow \dots \rightarrow a_{n_k}$ の値に関する判断ノード
 $\rightarrow d_m$ 出力の処理ノード

となっていた時，式 (6) に従うと，

$$f(ind, c) = \begin{cases} 0 & \text{if } d_{4a_2+2a_1+a_0} \neq d_m \\ 0 & \text{if } k+1 > 10 \\ (9-k)/6 & \text{otherwise} \end{cases}$$

となる．開始ノードから処理ノードに至る経路は $d_0 \sim d_7$ に依存せず $a_0 \sim a_2$ の値だけによって決まる．そして， $\{n_1, n_2, \dots, n_k\} = \{0, 1, 2\}$ かつ $4a_2 + 2a_1 + a_0 = m$ の時にこの経路は $a_0 \sim a_2$ の部分が c と同じ $2^8 = 256$ 個全ての入力事例に対して $\max\{(9-k)/6, 0\}$ という評価値

をもたらす．また， $\{n_1, n_2, \dots, n_k\} \neq \{0, 1, 2\}$ や $4a_2 + 2a_1 + a_0 \neq m$ の時でも，この経路は $a_0 \sim a_2$ の部分が c と同じ $2^8 = 256$ 個の入力事例の内半数に対して (偶然正解を出し) $\max\{(9-k)/6, 0\}$ という評価値をもたらす，残りの半数の入力事例に対して 0 という評価値をもたらす．従って，上記経路中の n_1, n_2, \dots, n_k や k, m を $a_0 \sim a_2$ の関数と見て， $a_0 \sim a_2$ に対する開始ノードから処理ノードに至る経路を

開始ノード $\rightarrow a_{n_1(a_0, a_1, a_2)}$ の値に関する判断ノード
 $\rightarrow a_{n_2(a_0, a_1, a_2)}$ の値に関する判断ノード
 $\rightarrow \dots \rightarrow a_{n_k(a_0, a_1, a_2)}(a_0, a_1, a_2)$ の値に関する判断ノード
 $\rightarrow d_{m(a_0, a_1, a_2)}$ 出力の処理ノード

と表すことにすると，

$$\sum_{d_0} \sum_{d_1} \dots \sum_{d_7} f(ind, c) = \begin{cases} \frac{(9-k(a_0, a_1, a_2))}{6} \times 256 & \text{if } \{n_1(a_0, a_1, a_2), \dots, n_k(a_0, a_1, a_2)(a_0, a_1, a_2)\} \\ & = \{a_0, a_1, a_2\} \\ & \& 4a_2 + 2a_1 + a_0 = m(a_0, a_1, a_2) \\ \frac{(9-k(a_0, a_1, a_2))}{6} \times 128 & \text{otherwise} \end{cases}$$

となる．実装においては，計算量を抑えるために式 (7) に忠実に従って適合度計算するのを避け，この式の $a_0 \sim a_2$ に関する総和をとることにより式 (7) と同等の値を求めた．

探索手法実装に関するメモ：次世代に残すエリートと確率モデル構築に使う良好個体を選び出すために，クイック整列法の `partition` 処理を変形して用いた (6.3 節)．また，擬似乱数の生成には Mersenne Twister[松本 98] を用いた．

実験パラメータの設定：第 4 章で提案した，ノードの種類・内容も探索する PMBGNP は，個体群サイズ (Popsiz) や最大世代数 (Max_gen) といった通常の EA パラメータの他に，個体中のノードの総数 (Num_nodes)，次世代に残すエリートの割合 (rate_{elite})，良好個体として確率モデル構築に用いる個体の割合 (rate_{good})，ノード遺伝子を一様分布に沿って決める確率を制御する係数 (c_{node})，接続遺伝子を一様分布に沿って決める確率を制御する係数 (c_{connect}) という 5 つのパラメータをもつ．以上の実験パラメータのうち，Popsiz については 1000 をデフォルト値とした．Max_gen は 10000 と設定して，最適解が得られた時点で探索を終了する様にした．Num_nodes は，迷路問題 1 では 15，迷路問題 2 では 8，11 マルチプレクサ問題では 16 をデフォルト値として扱った．残りの 4 つのパラメータ rate_{elite}, rate_{good}, c_{node}, c_{connect} については，予備実験として次の 5.3 節 ~ 5.4 節で 3 つのテスト問題全てで良好な探索結果をもたらす組合せを探る．

5.3 予備実験 1 : rate_{elite}, rate_{good} 設定の影響

パラメータ rate_{elite} と rate_{good} は適合度設定とともに選択圧を決める重要な要素で，提案 PMBGNP (4 章) の

表 1 $rate_{elite}, rate_{good}$ による成功率の変動. ($c_{node}=c_{connect}=1$ と固定して各 300 回探索実行した結果; 迷路問題 1 では 4000 世代目, 残りの 2 つの問題では 10000 世代目の成功率を表示; 迷路問題 1 では 1.000 を, 迷路問題 2 では 0.8 以上の数値を, 11 マルチプレクサ問題では 0.4 以上の数値を太字表示; 更に 3 つの問題で太字の場合 下線 も付けて表示)

問題	$rate_{elite}$	$rate_{good}$								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
迷路問題 1 (4000 世代目)	0.00	0.337	0.360	0.373	0.363	0.377	0.347	0.347	0.367	0.360
	0.05	1.000	1.000	1.000	1.000	1.000	0.997	0.963	0.913	0.840
	0.10	1.000	1.000	<u>1.000</u>	<u>1.000</u>	1.000	0.997	0.987	0.950	0.880
	0.15	1.000	1.000	<u>1.000</u>	0.963	0.963	0.913	0.830	0.803	0.767
	0.20	1.000	0.900	0.840	0.720	0.633	0.580	0.613	0.520	0.523
	0.25	1.000	0.760	0.440	0.453	0.477	0.413	0.380	0.367	0.380
迷路問題 2 (10000 世代目)	0.00	0.000	0.007	0.007	0.007	0.007	0.003	0.000	0.000	0.007
	0.05	0.707	0.800	0.863	0.697	0.493	0.197	0.067	0.017	0.010
	0.10	0.677	0.757	<u>0.853</u>	<u>0.870</u>	0.843	0.803	0.700	0.520	0.310
	0.15	0.710	0.607	<u>0.823</u>	0.857	0.870	0.827	0.850	0.780	0.733
	0.20	0.670	0.793	0.757	0.877	0.880	0.883	0.857	0.820	0.823
	0.25	0.647	0.750	0.787	0.810	0.833	0.890	0.877	0.840	0.847
11 マルチプレ クサ問題 (10000 世代目)	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.05	0.440	0.607	0.543	0.000	0.000	0.000	0.000	0.000	0.000
	0.10	0.363	0.543	<u>0.620</u>	<u>0.493</u>	0.003	0.000	0.000	0.000	0.000
	0.15	0.363	0.503	<u>0.480</u>	0.520	0.297	0.010	0.000	0.000	0.000
	0.20	0.373	0.397	0.443	0.453	0.310	0.100	0.000	0.003	0.010
	0.25	0.363	0.417	0.487	0.450	0.273	0.117	0.003	0.033	0.007

探索能力に大きな影響を及ぼすと考えられる. そこで, その影響を調べるために, $Popsiz=1000$, $c_{node}=c_{connect}=1$, $Num_nodes=15$ (迷路問題 1), 8(迷路問題 2), 16(11 マルチプレクサ問題) と固定した上で, 3 つの問題に対して $rate_{elite} \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$, $rate_{good} \in \{0.1, 0.2, \dots, 0.9\}$ の各々の組合せの下で提案 PMBGNP の探索を 300 回ずつ行なった. その結果を表 1 に示す. この表は $rate_{elite}$, $rate_{good}$ の値の組合せによって成功率 (i.e. 行なった探索実行の内 で最適解が得られた割合) がどう変わるかを問題毎に示したものである. 迷路問題 1 では 4000 世代目, 迷路問題 2 と 11 マルチプレクサ問題では 10000 世代目の成功率が表示されている. この表 1 を見ると, 迷路問題 1 の 4000 世代目で成功率 1.000(太字), 迷路問題 2 の 10000 世代目で成功率 0.8 以上(太字), 11 マルチプレクサ問題の 10000 世代目で成功率 0.4 以上(太字) という 3 つの条件を満たしている組合せは $(rate_{elite}, rate_{good})=(0.1, 0.3), (0.1, 0.4), (0.15, 0.3)$ の 3 つ(下線表示の場所)だけであることが分かる. そこで, 以後 $Popsiz=1000$ の場合は, $(rate_{elite}, rate_{good})=(0.1, 0.3)$ を安定した探索結果をもたらす組合せと考え, この値の組を用いることにする.

5.4 予備実験 2: $c_{node}, c_{connect}$ 設定の影響

パラメータ c_{node} と $c_{connect}$ は新規個体の多様性を維持するための重要な要素で, 提案 PMBGNP の探索能力に大きな影響を及ぼすと考えられる. そこで, その影響を調べるために, $Popsiz=1000$, $rate_{elite}=0.1$, $rate_{good}=0.3$, $Num_nodes=15$ (迷路問題 1), 8(迷路問題 2), 16(11 マルチ

プレクサ問題) と固定した上で, 3 つの問題に対して $c_{node} \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$, $c_{connect} \in \{0.0, 0.5, 1.0, \dots, 3.0\}$ の各々の組合せの下で提案 PMBGNP の探索を 300 回ずつ行なった. その結果を表 2 に示す. この表は c_{node} , $c_{connect}$ の値の組合せによって成功率がどう変わるかを問題毎に示したものである. 迷路問題 1 では 4000 世代目, 迷路問題 2 と 11 マルチプレクサ問題では 10000 世代目の成功率が表示されている. この表 2 を見ると, 迷路問題 1 の 4000 世代目で成功率 1.000(太字), 迷路問題 2 の 10000 世代目で成功率 0.8 以上(太字), 11 マルチプレクサ問題の 10000 世代目で成功率 0.4 以上(太字) という 3 つの条件を満たしている組合せは $(c_{node}, c_{connect})=(0.5, 1.5), (1.0, 1.0), (1.0, 1.5), (1.5, 1.0)$ の 4 つ(下線表示の場所)だけであることが分かる. そこで, 以後 $Popsiz=1000$ の場合は, $(c_{node}, c_{connect})=(1.0, 1.0)$ を安定した探索結果をもたらす組合せと考え, この値の組を用いることにする.

5.5 実験: ノード数設定の影響

提案 PMBGNP(4 章) においては, 個体中の各ノードの種類 (i.e. K_i の値)・内容 (i.e. ID_i の値) が固定されていないので, 十分大きなノード数 (Num_nodes) を指定して提案 PMBGNP の探索実行を行えば必要な種類・内容のノードが必要なだけ出現して望ましい振舞いをする個体が得られることが期待できる. しかし, 実際にノード数をどの程度に設定すれば最良の探索に繋がるのかは不明である. そこで, 参考のため, ここでは, 良好なノード数設定の仕方を探るため, 3 つの問題に対して

表 2 $c_{\text{node}}, c_{\text{connect}}$ による成功率の変動. ($\text{rate}_{\text{elite}}=0.1, \text{rate}_{\text{good}}=0.3$ と固定して各 300 回探索実行した結果; 迷路問題 1 では 4000 世代目, 残りの 2 つの問題では 10000 世代目の成功率を表示; 迷路問題 1 では 1.000 を, 迷路問題 2 では 0.8 以上の数値を, 11 マルチプレクサ問題では 0.4 以上の数値を太字表示; 更に 3 つの問題で太字の場合下線も付けて表示)

問題	c_{node}	c_{connect}						
		0.0	0.5	1.0	1.5	2.0	2.5	3.0
迷路問題 1 (4000 世代目)	0.0	0.933	0.977	0.987	0.990	0.983	1.000	1.000
	0.5	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	1.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	1.5	1.000	1.000	1.000	1.000	1.000	0.993	0.977
	2.0	1.000	1.000	0.997	0.997	0.990	0.987	0.937
迷路問題 2 (10000 世代目)	0.0	0.007	0.160	0.190	0.213	0.223	0.237	0.277
	0.5	0.023	0.607	0.790	0.900	0.940	0.970	0.967
	1.0	0.030	0.647	0.853	0.917	0.913	0.953	0.863
	1.5	0.027	0.663	0.800	0.850	0.847	0.750	0.597
	2.0	0.017	0.643	0.710	0.730	0.687	0.513	0.207
11 マルチプレクサ問題 (10000 世代目)	0.0	0.000	0.023	0.070	0.090	0.100	0.117	0.067
	0.5	0.433	0.437	0.480	0.433	0.397	0.320	0.103
	1.0	0.653	0.573	0.620	0.463	0.320	0.100	0.023
	1.5	0.660	0.577	0.530	0.333	0.100	0.003	0.000
	2.0	0.573	0.497	0.260	0.083	0.000	0.000	0.000

色々な Num_nodes の設定を試し, それらの結果を比較することにする. 具体的には, Popsiz=1000, $\text{rate}_{\text{elite}}=0.1$, $\text{rate}_{\text{good}}=0.3$, $c_{\text{node}}=c_{\text{connect}}=1.0$ と固定した上で, 迷路問題 1 に対しては Num_nodes=7,9,11,20,50,100 の 6 通り, 迷路問題 2 に対しては Num_nodes=8,10,12,20,50,100 の 6 通り, 11 マルチプレクサ問題に対しては Num_nodes=16,18,20,30,50,100 の 6 通りを考える. そして, 各々の設定の下で提案 PMBGNP の探索を 300 回ずつ行い, 世代と共に成功率と最良適合度 (300 回探索の平均) がどう推移するかを観察した. その結果を図 9~図 14 に示す. これらの図を見る限りにおいては, 迷路問題 1 では Num_nodes=9~20 が, 迷路問題 2 では Num_nodes=12~20 が, 11 マルチプレクサ問題では Num_nodes=50~100 が良好なノード数設定と認められる. ノード数を可能な最小値に設定するよりも多少の余裕を持たせて設定した方が, 早い世代での探索成功に繋がる傾向が確認される. 特に 11 マルチプレクサ問題ではノード数が多ければ多いほど早い世代で解探索に成功に至るという結果になっているが, これは, ノード数が多ければ多い程 GNP 個体を問題に適用した時に生成されるノード遷移列に閉路が出来にくくなり, 11 マルチプレクサ問題で要求されている木構造が出来易くなるためと考えられる.

5.6 実験: ノードの種類・内容を固定し弧の接続先だけを探索する場合との比較

4 章で示した PMBGNP は弧の接続先だけでなくノードの種類・内容も探索するので, どの種類・内容のノードを何個用意すべきかについての事前知識を要しない. この点を見ただけでも, ノードの種類・内容を固定して探

索する場合よりも有用である. しかし, ノードの種類・内容も探索する様になると, (予めノードの種類・内容の設定を与えて) 固定する場合と比べて探索空間が広くなり探索自体の難易度が上がることが予想される. そこで, この節では 5.1 節で考えた 3 つの問題に対してノードの種類・内容を固定して探索する実験を行い, ノードの種類・内容も探索した 5.5 節の結果と比較する. ここでは, ノードの種類・内容を固定した探索手順としては 4 章で示した PMBGNP に次の変更を加えたものを考える.

- 4 章で示した PMBGNP においては, ノードの種類・内容も探索するので開始ノードから出ている弧の接続先を 1 番目のノードに固定していた. しかし, 各ノードの種類・内容を固定する場合に開始ノードからの接続先を 1 番目のノードに固定したのでは最初に行う判断/処理を確定してしまうことになる. そこで, 開始ノードから出ている弧の接続先についても他の処理ノードと同様に確率モデルを構築し, 新規個体を生成する際は確率モデルに基づいたサンプリングを行うことにする.
- 初期個体生成の際は, 各ノードの種類・内容は予め指定された通りに設定し, 確率的な設定は行わない. また, 開始ノードから出ている弧の接続先 (C_{01} 遺伝子) については, (0 以外が等確率で出る) 一様分布に従ってランダムに決定する.
- ノード遺伝子に関する頻度分布は構成しない.
- 新規個体を生成する際, ノード遺伝子に関しては頻度分布に基づくサンプリングは行わず, 予め指定された設定を用いる.
- パラメータ c_{node} については $c_{\text{node}} = 0.0$ と固定する. 最も基本的な PMBGNP においては $c_{\text{connect}} = 0.0$ である

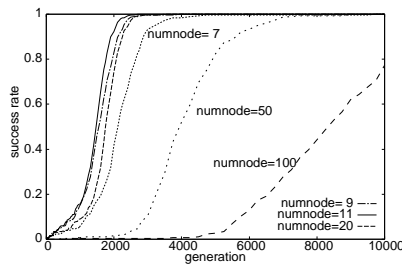


図 9 Num_nodes を変化させた時、成功率の推移がどう変わるか (迷路問題 1)

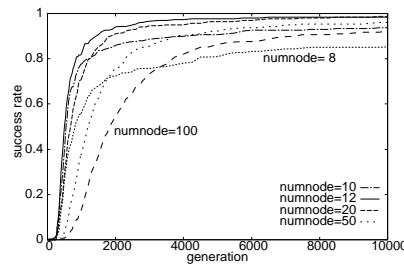


図 10 Num_nodes を変化させた時、成功率の推移がどう変わるか (迷路問題 2)

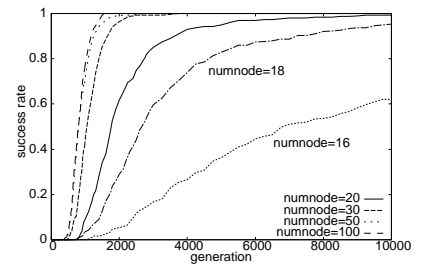


図 11 Num_nodes を変化させた時、成功率の推移がどう変わるか (11 マルチプレクサ問題)

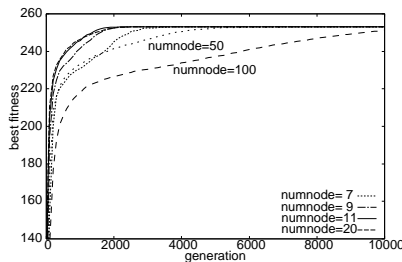


図 12 Num_nodes を変化させた時、最良適合度の平均的な推移がどう変わるか (迷路問題 1)

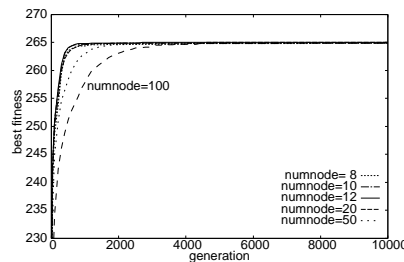


図 13 Num_nodes を変化させた時、最良適合度の平均的な推移がどう変わるか (迷路問題 2)

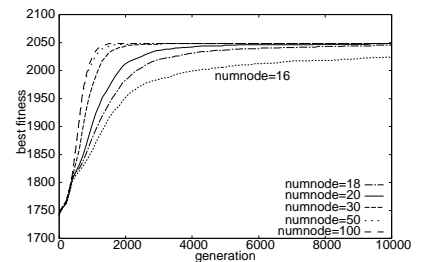


図 14 Num_nodes を変化させた時、最良適合度の平均的な推移がどう変わるか (11 マルチプレクサ問題)

が、 c_{connect} は弧の接続先の探索に関わるものであるもので、ここでは調整パラメータとして扱う。

実験 (ノードの種類・内容を適切に設定・固定して探索した場合): 図 5 に迷路問題 1 に対する最適解の例が示されている。これを参考に、逆に、ノード遺伝子部を図 5 のものと同じに (すなわち、番号 1,2,3,4,5,6 のノードにそれぞれ `judgeFront`, `judgeLeft`, `judgeRight`, `moveForward`, `turnLeft`, `turnRight` という動作が来る様に) 固定して弧の接続先だけを探索する実験を行った。その際、基本的なパラメータ設定は 5.5 節の実験と同じく $\text{Popsize}=1000$, $\text{rate}_{\text{elite}}=0.1$, $\text{rate}_{\text{good}}=0.3$, 探索回数=300 とし、 $c_{\text{connect}}=0.0, 1.0, 2.0, 3.0, 4.0, 5.0$ の 6 通りの設定の下で世代と共に成功率がどう推移するかを観察した。その結果を図 15 に示す。ノードの種類・内容も探索する場合の実験結果 (図 9, $\text{Num_nodes}=7$ のグラフ) と比較すると、 $c_{\text{connect}}=3.0, 4.0, 5.0$ の場合にはより早い世代で 100% の成功率に至っている。しかし、 $c_{\text{connect}}=0.0, 1.0, 2.0$ の場合にはより速く成功率は上がるものの途中で探索が停滞し 10000 世代目の時点で 100% の成功率に至っていない。同様に、迷路問題 2, 11 マルチプレクサ問題に対する最適解の例が図 7, 図 8 に示されているので、ノード遺伝子部をそれぞれ図 7, 図 8 のものと同じに固定して、 $\text{Popsize}=1000$, $\text{rate}_{\text{elite}}=0.1$, $\text{rate}_{\text{good}}=0.3$, 探索回数=300, $c_{\text{connect}} \in \{0.0, 1.0, 2.0, 3.0, 4.0, 5.0\}$ というパラメータ設定の下で弧の接続先だけを探索する実験を行い、世代と共に成功率がどう推移するかを観察した。その結果をそれぞれ図 16, 図 17 に示す。その内迷路問題 2 の結果 (図 16) を見ると、 $c_{\text{connect}} \in \{1.0, 2.0, 3.0, 4.0, 5.0\}$ の場合にはノードの種類・内容も探索する場合の実験結果 (図 10, $\text{Num_nodes}=8$ のグラフ) より早い世代で 100% の成功率に至っているもの

の、 $c_{\text{connect}}=0.0$ の場合には全然成功に至っていない。11 マルチプレクサ問題の結果 (図 17) に関しては、いずれの $c_{\text{connect}} \in \{0.0, 1.0, 2.0, 3.0, 4.0, 5.0\}$ の設定下でも最終の 10000 世代目で 20% 未満の成功率にしか至らず、ノードの種類・内容も探索する場合 (図 11, $\text{Num_nodes}=16$ のグラフ) よりも 40% 以上低い成功率しか達成されていない。この原因としては、ノードの種類・内容を固定することによって解の形が限定され、色々な形の最適解が探索空間から消え、解探索を行いにくくなった、という可能性が考えられる。最後に、 c_{connect} の設定に関しては、 $c_{\text{connect}}=2.0, 3.0, 4.0$ が 3 つの問題で (他より) 良好な結果をもたらしていると認められるので、以下では $c_{\text{connect}}=3.0$ という設定だけを用いることにする。

実験 (各基本動作を同数ずつ用意してノードに固定し探索した場合): 一般に、どの種類・内容のノードを何個用意すべきかを最初から分かっているとは限らないので、ノードの種類・内容を固定した探索を行う場合は、(ノード数が過剰になるのを承知で) 各々の基本動作を十分な個数用意してノードに固定することになる。この場合の探索効率を調べるために、まず、各々の基本動作を同数ずつ用意してノードに固定して弧の接続先を探索する実験を行った。具体的には、迷路問題 1, 2 については基本動作を 1 個ずつ, 2 個ずつ, 3 個ずつ, 4 個ずつ, 8 個ずつ, 14 個ずつ用意することを考え、11 マルチプレクサ問題については基本動作を 1 個ずつ, 2 個ずつ, 3 個ずつ, 4 個ずつ, 6 個ずつ, 9 個ずつ用意することを考えた。そして、これまでと同様に $\text{Popsize}=1000$, $\text{rate}_{\text{elite}}=0.1$, $\text{rate}_{\text{good}}=0.3$, 探索回数=300, $c_{\text{connect}}=3.0$ というパラメータ設定の下で弧の接続先だけを探索する実験を行い、世代と共に成功率がどう推移するかを観察した。その結果を図 18~ 図 20 に示

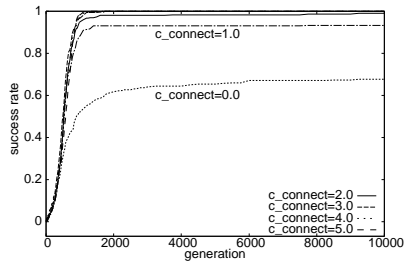


図 15 図 5 で使われているラベルを固定して様々な c_{connect} の下で弧の接続先を探索した場合の成功率の推移 (迷路問題 1)

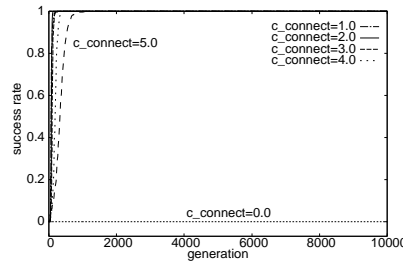


図 16 図 7 で使われているラベルを固定して様々な c_{connect} の下で弧の接続先を探索した場合の成功率の推移 (迷路問題 2)

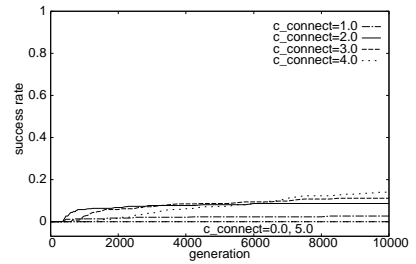


図 17 図 8 で使われているラベルを固定して様々な c_{connect} の下で弧の接続先を探索した場合の成功率の推移 (11 マルチプレクサ問題)

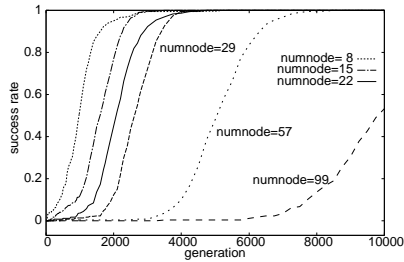


図 18 各ラベルを同じ個数用意しノードに固定して弧の接続先を探索した場合の成功率の推移 (迷路問題 1)

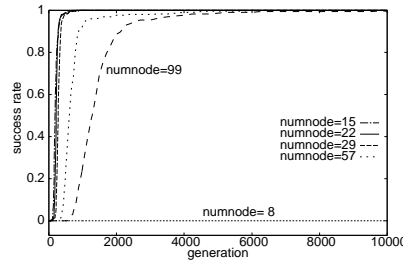


図 19 各ラベルを同じ個数用意しノードに固定して弧の接続先を探索した場合の成功率の推移 (迷路問題 2)

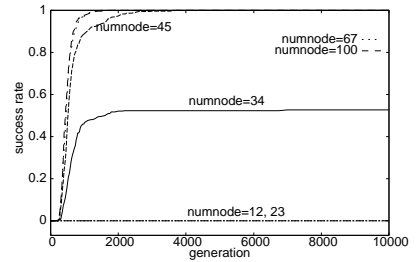


図 20 各ラベルを同じ個数用意しノードに固定して弧の接続先を探索した場合の成功率の推移 (11 マルチプレクサ問題)

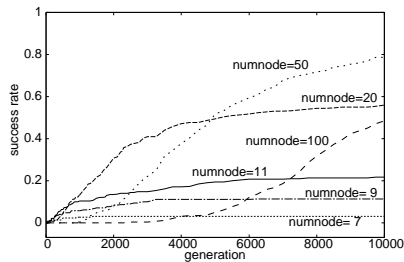


図 21 各ノードにラベルをランダムに割り当て固定し弧の接続先を探索した場合の成功率の推移 (迷路問題 1)

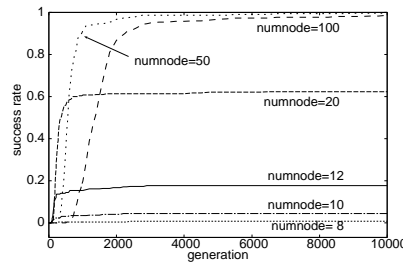


図 22 各ノードにラベルをランダムに割り当て固定し弧の接続先を探索した場合の成功率の推移 (迷路問題 2)

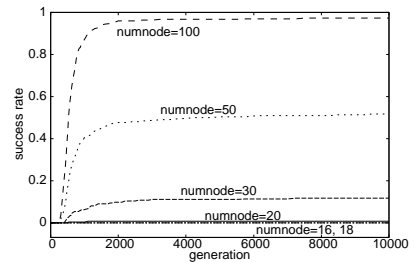


図 23 各ノードにラベルをランダムに割り当て固定し弧の接続先を探索した場合の成功率の推移 (11 マルチプレクサ問題)

す。その内、迷路問題 1 の結果 (図 18) に関しては、ノードの種類・内容も探索する場合の実験結果 (図 9) とほぼ同程度の速さで探索成功に至っていると判断できる。迷路問題 2 の結果 (図 19) に関しては、2 個ずつ以上の基本動作が個体中に用意されている場合にはノードの種類・内容も探索した場合の結果 (図 10) よりも速く探索成功に至る一方で、基本動作が 1 個ずつしか個体中に用意されていない場合には全然探索成功に至っていない。11 マルチプレクサの結果 (図 20) に関しては、4 個ずつ以上の基本動作が個体中に用意されている場合にはノードの種類・内容も探索した場合の結果 (図 11) と同程度の速さで探索成功に至るが、3 個ずつ以下しか基本動作が個体中に用意されていない場合には同程度のノード数でノードの種類・内容も探索した結果 (100% の成功率) よりもはつきりと低い成功率 (0~53%) しか達成できていない。迷路問題 2, 11 マルチプレクサ問題の実験結果より、ノードの種類・内容を固定して探索する方法においては、各々の基本動作を何個用意するかの設定を適切に行うことが重要で、これが適切に行われていないと探索がうまく進

まないことが確認された。

実験 (各ノードにランダムに基本動作を割り当てて固定し探索した場合): どの基本動作を何個用意すればよいのかについても調べたい場合、そのための方法として、各々のノードにランダムに基本動作を割り当て固定して探索する実験を繰り返す、という方法も考えられる。そこで、次に $c_{\text{connect}}=3.0$ とする以外は 5.5 節と全く同じ設定の下でこの方法による実験を行った。具体的には、迷路問題 1 に対しては Num_nodes=7,9,11,20,30,50,100 の 6 通り、迷路問題 2 に対しては Num_nodes=8,10,12,20,30,50,100 の 6 通り、11 マルチプレクサ問題に対しては Num_nodes=16,18,20,30,30,50,100 の 6 通りを考えた。そして、Popsiz=1000, rate_{elite}=0.1, rate_{good}=0.3, $c_{\text{connect}}=3.0$ というパラメータ設定の下で各々のノードにランダムに基本動作を割り当て固定し弧の接続先を探索する実験を 300 回繰り返し、世代と共に成功率がどう推移するかを観察した。その結果を図 21~ 図 23 に示す。迷路問題 2 で Num_nodes=50,100 とした場合はノードの種類・内容も探索した結果 (図 10, Num_nodes=50,100 のグラフ) よりも高い成功率に至っている

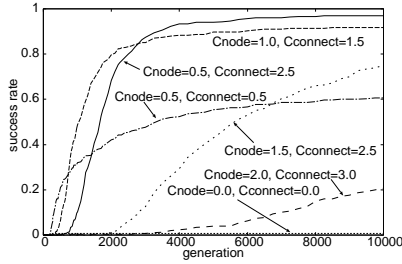


図 24 様々な $C_{node}, C_{connect}$ 設定に対する、成功率の推移 (迷路問題 2)

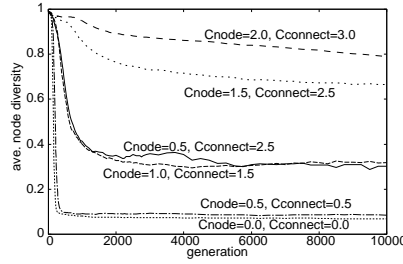


図 25 様々な $C_{node}, C_{connect}$ 設定に対する、ノード遺伝子部の多様性の推移 (迷路問題 2)

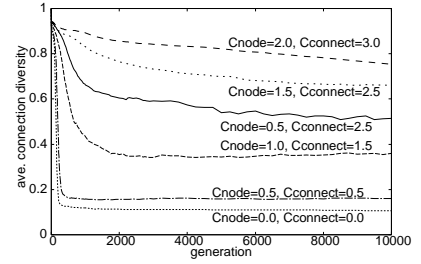


図 26 様々な $C_{node}, C_{connect}$ 設定に対する、接続遺伝子部の多様性の推移 (迷路問題 2)

が、それ以外の場合はどれもノードの種類・内容も探索した結果 (図 9～図 11) よりも劣った結果になっている。Num_nodes ≤ 30 の場合のこの原因としては、ランダムに基本動作をノードに割り当てたために、個体内に十分な基本動作が用意されない可能性が考えられる。

以上の実験により、ノードの種類・内容も探索することの有用性が確認された。

6. 考 察

6.1 多様性維持の様子

個体内の遺伝子の個数が一定の場合に、Shapiro は個体群内の個体の多様性の尺度として次の様なもの考えた [Shapiro 06].

$$\sum_{i: \text{遺伝子座}} \frac{\sum_{a: i \text{ で可能な遺伝子}} \nu_i^a (1 - \nu_i^a)}{i \text{ で可能な遺伝子の個数}}$$

但し、 $\nu_i^a \stackrel{\text{def}}{=} \text{個体群内で遺伝子座 } i \text{ に遺伝子 } a \text{ が入っている割合}$

この多様性測度の定義に沿って考えると、提案 PMBGNP で構成される個体群に関して、

- 所属 GNP 個体に含まれるノード遺伝子の多様性の度合いは、個体群構成の基になった確率モデル M を用いて次の式で見積もることができる。

ノード遺伝子の多様性の度合い

$$\begin{aligned} &= \text{「開始ノード以外のノードにおけるノード遺伝子の多様性の度合い」の平均} \\ &= \frac{\sum_{i: \text{開始ノード以外のノード}} \text{diversity}_{\text{node}}(i)}{\text{開始ノード以外のノード数}} \quad (8) \end{aligned}$$

ここで、 $\text{diversity}_{\text{node}}(i)$ はノード i におけるノード遺伝子部の多様性の度合いを表し、最大値が 1 になる様に標準化したものである。ノードの種類 (i.e. K_i の値) と内容 (i.e. ID_i の値) の組を簡単のためラベルと呼ぶことにすると、これは Shapiro の多様性測度の定義に沿って次の式で表すことができる。

$$\begin{aligned} \text{diversity}_{\text{node}}(i) &= \frac{\sum_{a: i \text{ で可能なラベル}} \nu_i^a (1 - \nu_i^a)}{\text{ノード } i \text{ で可能なラベルの種類数}} \\ &\quad / \left(\frac{1 - 1/(\text{ノード } i \text{ で可能なラベルの種類数})}{\text{ノード } i \text{ で可能なラベルの種類数}} \right) \end{aligned}$$

$$= \frac{\sum_{a: i \text{ で可能なラベル}} \nu_i^a (1 - \nu_i^a)}{1 - 1/(\text{ノード } i \text{ で可能なラベルの種類数})}$$

但し、 $\nu_i^a \stackrel{\text{def}}{=} \text{モデル } M \text{ を使った時、ノード } i \text{ 上のラベルとして } a \text{ が生成される確率}$

- 所属 GNP 個体に含まれる接続遺伝子の多様性の度合いは、ノードの種類 (i.e. K_i の値) と内容 (i.e. ID_i の値) の組をやはりラベルと呼ぶことにすると、集団構成の基になった確率モデル M を用いて次の式で見積もることができる。

接続遺伝子の多様性の度合い

$$\begin{aligned} &= \text{「開始ノード以外のノードにおける接続遺伝子の多様性の度合い」の平均} \\ &= \frac{1}{\left(\text{開始ノード以外のノード数} \right)} \sum_{i: \text{開始ノード以外のノード}} \left(\text{モデル } M \text{ を使った時、} i \text{ で } a \text{ が生成される確率} \right) \\ &\quad \frac{\sum_{a: i \text{ で可能なラベル}} \sum_{k: a \text{ の分岐番号}} \text{diversity}_{\text{connect}}(i, a, k)}{a \text{ の分岐数}} \quad (9) \end{aligned}$$

ここで、 $\text{diversity}_{\text{connect}}(i, a, k)$ はノード i にラベル a が配置された時の、 k 番目の分岐先の多様性の度合いを表し、最大値が 1 になる様に標準化したものである。これは Shapiro の多様性測度の定義に沿って次の式で表すことができる。

$$\begin{aligned} \text{diversity}_{\text{connect}}(i, a, k) &= \frac{\sum_{b: \text{可能な分岐先のノード番号}} \nu_{i,a,k}^b (1 - \nu_{i,a,k}^b)}{\text{可能な分岐先 } (i \text{ と開始ノード以外) の個数}} \\ &\quad / \left(\frac{1 - 1/(\text{可能な分岐先の個数})}{\text{可能な分岐先の個数}} \right) \\ &= \frac{\sum_{b: \text{可能な分岐先のノード番号}} \nu_{i,a,k}^b (1 - \nu_{i,a,k}^b)}{1 - 1/(\text{ノード数} - 2)} \end{aligned}$$

但し、 $\nu_{i,a,k}^b \stackrel{\text{def}}{=} \text{モデル } M \text{ を使った時、ラベル } a \text{ の配置されたノード } i \text{ からの } k \text{ 番目の分岐先としてノード番号 } b \text{ が生成される確率}$

例 (多様性維持の様子): 5.4 節では 2 つのパラメータ C_{node} と $C_{connect}$ の値を調整する実験を繰り返したが、このうち迷路問題 2 で $(C_{node}, C_{connect}) = (0.0, 0.0), (0.5, 0.5), (0.5, 2.5), (1.0, 1.5), (1.5, 2.5), (2.0, 3.0)$ とした場合について

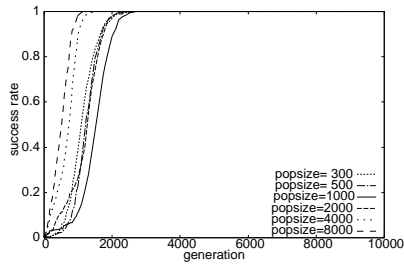


図 27 Popsiz を変化させた時、成功率の推移がどう変わるか (迷路問題 1)

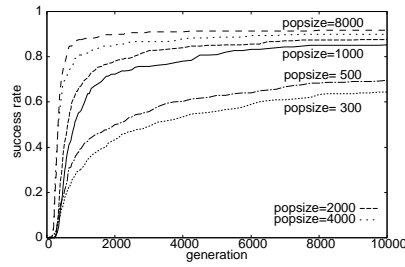


図 28 Popsiz を変化させた時、成功率の推移がどう変わるか (迷路問題 2)

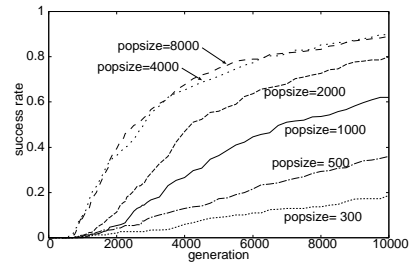


図 29 Popsiz を変化させた時、成功率の推移がどう変わるか (11 マルチプレクサ問題)

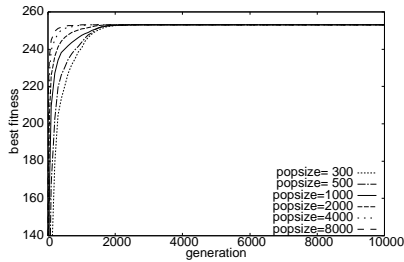


図 30 Popsiz を変化させた時、最良適合度の平均的な推移がどう変わるか (迷路問題 1)

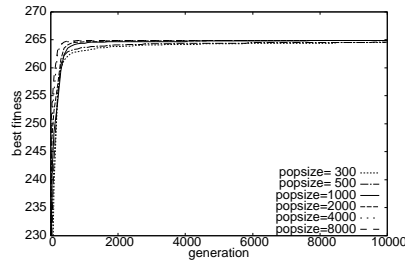


図 31 Popsiz を変化させた時、最良適合度の平均的な推移がどう変わるか (迷路問題 2)

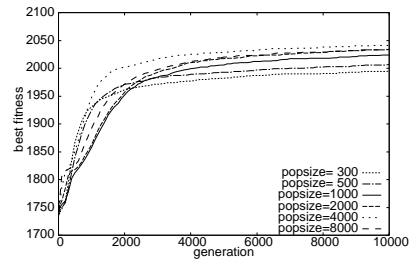


図 32 Popsiz を変化させた時、最良適合度の平均的な推移がどう変わるか (11 マルチプレクサ問題)

て、成功率と上記のノード遺伝子の多様性の度合い (300 回探索実行の平均)、接続遺伝子の多様性の度合い (300 回探索実行の平均) が世代と共にどう推移するのかを示したグラフを図 24～図 26 に示す。($c_{\text{node}}, c_{\text{connect}}$)=(0.5, 2.5), (1.0, 1.5) の時の様に探索が良好に進み高い成功率を達成できている設定においては、探索の進行と共にノード遺伝子部、接続遺伝子部の多様性の度合いは下がるが、極端な下がり方にならずに最後まである程度 (0.3) 以上の値が維持されている。これに対して、($c_{\text{node}}, c_{\text{connect}}$)=(0.0, 0.0), (0.5, 0.5) の場合は、急速に多様性の度合いが落ちている様子が図 25～図 26 から読み取れ、結果として **exploitation** の傾向が強くなり過ぎて探索が停滞し成功率も上がらないと解釈できる。また、($c_{\text{node}}, c_{\text{connect}}$)=(1.5, 2.5), (2.0, 3.0) の場合は、世代交代を進めても多様性の度合いは高いまま (0.6 以上) 維持されている様子が図 25～図 26 から読み取れ、結果として **exploration** の傾向が強くなり過ぎて成功になかなか至らないと解釈できる。

以上で例示した様に、 c_{node} と c_{connect} はノード遺伝子部、接続遺伝子部の多様性の度合いを制御するための重要なパラメータとなっている。

6.2 個体群サイズ設定の影響

個体群サイズパラメータ **Popsiz** も新規個体の多様性を維持するための重要な要素で、提案 **PMBGNP** の探索能力に大きな影響を及ぼすと考えられるので、その影響を具体的に観察したい。ただ、5.3 節～5.4 節 (予備実験 1～2) で得られた 4 つのパラメータ $\text{rate}_{\text{elite}}$, $\text{rate}_{\text{good}}$, c_{node} , c_{connect} の良好な設定は **Popsiz**=1000 の場合に対するものであり、パラメータ **Popsiz** の値によって良好な $\text{rate}_{\text{elite}}$, $\text{rate}_{\text{good}}$, c_{node} , c_{connect} の値も変わってくるものと思われる。

そこで、まず **Popsiz**=4000 の場合に対して 5.3 節～5.4 節 (予備実験 1～2) と同様の実験を行なった。その結果、

- ($c_{\text{node}}, c_{\text{connect}}$)=(1.0, 1.0) の時、迷路問題 1 の 3000 世代目で成功率 1.0, 迷路問題 2 の 10000 世代目で成功率 0.85 以上, 11 マルチプレクサ問題の 10000 世代目で成功率 0.7 以上 という 3 つの条件を満たしている $\text{rate}_{\text{elite}}$, $\text{rate}_{\text{good}}$ の値の組合せは ($\text{rate}_{\text{elite}}, \text{rate}_{\text{good}}$) = (0.0125, 0.05), (0.0125, 0.1), (0.025, 0.05), (0.025, 0.1), (0.05, 0.1), (0.05, 0.2), (0.10, 0.1), (0.15, 0.1), (0.20, 0.1) の 9 通りであった。

- ($\text{rate}_{\text{elite}}, \text{rate}_{\text{good}}$)=(0.05, 0.1) の時、迷路問題 1 の 3000 世代目で成功率 1.0, 迷路問題 2 の 10000 世代目で成功率 0.85 以上, 11 マルチプレクサ問題の 10000 世代目で成功率 0.7 以上 という 3 つの条件を満たしている c_{node} , c_{connect} の値の組合せは ($c_{\text{node}}, c_{\text{connect}}$) = (0.5, 1.0), (0.5, 1.5), (0.5, 2.0), (1.0, 1.0), (1.0, 1.5), (1.0, 2.0), (1.0, 2.5), (1.5, 1.0), (1.5, 1.5), (1.5, 2.0), (2.0, 1.5) の 11 通りであった。

これらの結果を参考に、**Popsiz**=4000 の場合は $\text{rate}_{\text{elite}}$ =0.05, $\text{rate}_{\text{good}}$ =0.1, c_{node} =1.0, c_{connect} =1.5 を安定した探索結果をもたらす組合せと考える。

次に、**Popsiz**=300 の場合に対しても 5.3 節～5.4 節 (予備実験 1～2) と同様の実験を行なった。その結果、

- ($c_{\text{node}}, c_{\text{connect}}$)=(1.0, 1.0) の時、迷路問題 1 の 4000 世代目で成功率 1.0, 迷路問題 2 の 10000 世代目で成功率 0.5 以上, 11 マルチプレクサ問題の 10000 世代目で成功率 0.15 以上 という 3 つの条件を満たしている $\text{rate}_{\text{elite}}$, $\text{rate}_{\text{good}}$ の値の組合せは ($\text{rate}_{\text{elite}}, \text{rate}_{\text{good}}$) = (0.05, 0.2), (0.05, 0.3), (0.10, 0.3), (0.10, 0.4), (0.15, 0.2), (0.15, 0.3),

(0.15,0.4), (0.15,0.5) の 8 通りであった。(問題毎に成功率の基準が全然違うが、これは各々の問題において 30 通りの組合せの中で他より良好なものを選別するためである。例えば 11 マルチプレクサ問題においては、成功率が 0.15 以上かどうかによって上位 13 通りの組合せと下位 17 通りの組合せに分かれる。)

●(rate_{elite}, rate_{good})=(0.10,0.3) の時、迷路問題 1 の 4000 世代目で成功率 1.0, 迷路問題 2 の 10000 世代目で成功率 0.5 以上, 11 マルチプレクサ問題の 10000 世代目で成功率 0.15 以上 という 3 つの条件を満たしている c_{node} , c_{connect} の値の組合せは $(c_{\text{node}}, c_{\text{connect}}) = (1.0, 1.0), (1.0, 1.5), (1.0, 2.0), (1.5, 1.0)$ の 4 通りであった。

これらの結果を参考に、Popsiz=300 の場合は rate_{elite}=0.1, rate_{good}=0.3, c_{node} =1.0, c_{connect} =1.0 を安定した探索結果をもたらす組合せと考える。

以上の予備実験結果を基に、Popsiz=300, 500, 1000, 2000, 4000, 8000 の場合に対して、それぞれ

	Popsiz					
	300	500	1000	2000	4000	8000
rate _{elite}	0.1	0.1	0.1	0.08	0.05	0.04
rate _{good}	0.3	0.3	0.3	0.2	0.1	0.08
c_{node}	1.0	1.0	1.0	1.0	1.0	1.0
c_{connect}	1.0	1.0	1.0	1.2	1.5	1.7

という値を実験に用いることにする。(Popsiz=500, 2000, 8000 の場合は推定。)そして、Popsiz の影響を調べるために、Num_{nodes}=15(迷路問題 1), 8(迷路問題 2), 16(11 マルチプレクサ問題) と固定した上で、3 つの問題に対して Popsiz=300, 500, 1000, 2000, 4000, 8000 の提案 PMBGNP の探索を 300 回ずつ行ない、世代と共に成功率と最良適合度(300 回探索の平均)がどう推移するかを観察した。その結果を図 27~図 32 に示す。Popsiz を大きくすると概して探索成功の速度も速くなっている。しかし、一部の例外(11 マルチプレクサ問題, Popsiz≤1000)を除いて、それは Popsiz の増加に伴う計算量の増加に見合うものとはなっていない。Popsiz を大きくすると推定する個体分布が安定的に細かく表せる様になる訳であるが、「近くの最適解を推定分布から生成する可能性を増やす」といった直接的な探索性能の向上にはあまり反映されないことが確認される。

6.3 エリートと良好個体を選び出す方法の影響

提案 PMBGNP を実装するに当たって、次世代に残すエリートと確率モデル構築に使う良好個体を選び出すために、ここではクイック整列法の partition 処理を変形して用いた。具体的には、次の手順をエリート個体を選び出す際にも、確率モデル構築に使う良好個体を選び出す際にも用いた。

上位個体選別の手順(クイック整列法の partition 処理を変形した版): 個体 a が個体 b より良い適合度をもつ時 $a > b$, a が b 以上に良い適合度をもつ時 $a \geq b$ と書

くことにする。個体列 $a[from], a[from+1], \dots, a[to]$ と整数 $num(from \leq num < to)$ が与えられた時、個体列 $a[from], a[from+1], \dots, a[to]$ を並べ直して

$(\forall i < num+1) a[i] \geq a[num+1]$

$\wedge (\forall j > num+1) a[num+1] \geq a[j]$, もしくは

$(\forall i < num) a[i] \geq a[num] \wedge (\forall j > num) a[num] \geq a[j]$

とする(これで上位 $num - from + 1$ 個の選別が出来たことになる)ための手順として、次の擬似コードを考えることが出来る。

```

i ← from;
j ← to;
while (i ≤ num ∧ num < j) {
    //partition 操作
    pivot ← a[i];
    while (i < j) {
        while (i < j ∧ a[j] ≤ pivot) j ← j - 1;
        if (i = j) break;
        a[i] ← a[j];
        i ← i + 1;
        while (i < j ∧ a[i] > pivot) i ← i + 1;
        if (i = j) break;
        a[j] ← a[i];
        j ← j - 1;
    }
    a[j] ← pivot;
    //注目区間を縮小して繰り返し
    if (num < j - 1) {
        to ← j - 1;
        i ← from;
        j ← to;
    } else if (i < num) {
        from ← i + 1;
        i ← from;
        j ← to;
    }
}

```

上位個体選別の手順としては他の方法も考えられる。この部分の実装によって探索結果が大きく変わることもあることを例示するために、上位個体選別の手順として、

与えられた個体 $a[from], a[from+1], \dots, a[to]$ をヒープ整列法で並べ直して

$a[from] \geq a[from+1] \geq \dots \geq a[to]$ となる様にする

というものも考え、これを組み込んだ提案 PMBGNP の探索を迷路問題 1 に対して Popsiz=1000, rate_{elite}=0.1, rate_{good}=0.3, $c_{\text{node}}=c_{\text{connect}}=0$, Num_{nodes} ∈ {7, 15, 40, 100} というパラメータ設定の下でそれぞれ 300 回ずつ行なった。そして、比較のため、これまで通りのクイック整列法の partition 処理を変形した版の上位個体選別手順を組み込んだ提案 PMBGNP の探索も同じ条件の下でやはり 300 回ずつ行なった。いずれの探索実行においても、世

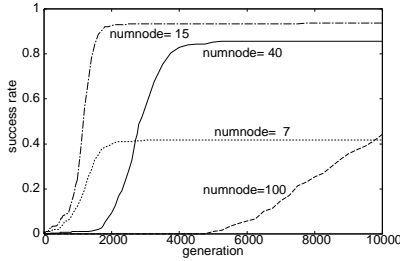


図 33 Num_nodes を変化させた時, 成功率の推移がどう変わるか (迷路問題 1, $c_{node}=c_{connect}=0$)

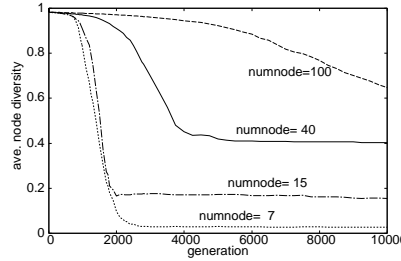


図 34 Num_nodes を変化させた時, ノード遺伝子部の多様性の度合いの推移がどう変わるか (迷路問題 1, $c_{node}=c_{connect}=0$)

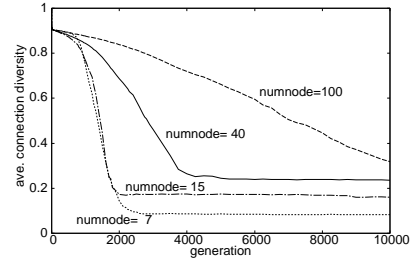


図 35 Num_nodes を変化させた時, 接続遺伝子部の多様性の度合いの推移がどう変わるか (迷路問題 1, $c_{node}=c_{connect}=0$)

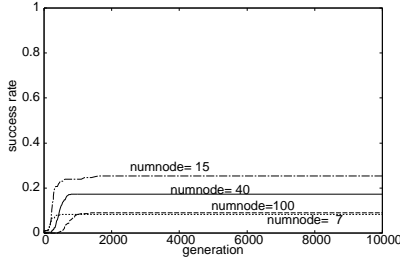


図 36 Num_nodes を変化させた時, 成功率の推移がどう変わるか (迷路問題 1, $c_{node}=c_{connect}=0$, 上位個体選別にヒープ整列法を利用)

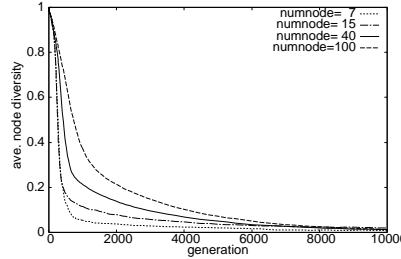


図 37 Num_nodes を変化させた時, ノード遺伝子部の多様性の度合いの推移がどう変わるか (迷路問題 1, $c_{node}=c_{connect}=0$, 上位個体選別にヒープ整列法を利用)

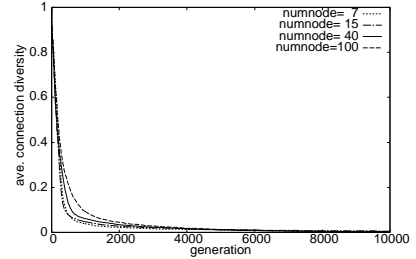


図 38 Num_nodes を変化させた時, 接続遺伝子部の多様性の度合いの推移がどう変わるか (迷路問題 1, $c_{node}=c_{connect}=0$, 上位個体選別にヒープ整列法を利用)

代と共に成功率とノード遺伝子部の多様性の度合い, 接続遺伝子部の多様性の度合いがどう推移するかを観察した. その結果を図 33~図 38 に示す.

ヒープ整列法版の上位個体選別手順を用いた場合 (図 36~図 38) は, 急速に多様性が失われ, 探索も失敗に終わることが多くなっている. ヒープ整列法版を用いた場合の多様性の急速な減少は次の 2 つのことに起因すると考えられる.

●(原因 1) ヒープ整列法版を用いた場合, 適合度値の同じ個体が多くありそれらの適合度値の改善が殆ど起きない状況でも, 個々の個体が格納されている配列上の位置が世代と共に大幅に変わり, その結果, エリート個体として次世代に残る個体に大幅な変動が起こる.

●(原因 2) 確率モデルに従ったランダムサンプリングを繰り返して次世代個体群を構成した場合, 次世代個体群の (確率) 分布における個体の多様性の度合いは, 元のランダムサンプリングに使った確率分布での個体の多様性の度合いより下がる傾向にある. (すなわち, 多様性の度合いが下がる確率の方が上がる確率より僅かに大きい.)

その根拠: 例えば, 個体の遺伝子長が 1 で, 良好個体群中に type-a の個体が k 個, type-b の個体が $n-k$ 個, 合わせて n 個の個体がある場合は, 次の様な確率分布が確率モデルとして構成されることになる.

$$\Pr(X = \text{type-a}) = k/n \stackrel{\text{def}}{=} p_k,$$

$$\Pr(X = \text{type-b}) = (n-k)/n = 1 - p_k$$

この確率分布に従ってランダムに n 個の次世代個体を生成した場合, n 個のうち type-a が i 個生成される確

率は $\binom{n}{i} p_k^i (1-p_k)^{n-i}$ である. それゆえ,

分布の偏り傾向が強くなる確率

$$= \begin{cases} \text{type-a の個体が } 0 \sim k-1 \text{ 個生成される確率} & \text{if } k \leq n/2 \\ \text{type-a の個体が } k+1 \sim n \text{ 個生成される確率} & \text{otherwise} \end{cases}$$

$$= \begin{cases} \sum_{i=0}^{k-1} \binom{n}{i} p_k^i (1-p_k)^{n-i} & \text{if } k \leq n/2 \\ \sum_{i=k+1}^n \binom{n}{i} p_k^i (1-p_k)^{n-i} & \text{otherwise} \end{cases}$$

分布の偏り傾向が弱くなる確率

$$= \begin{cases} \sum_{i=k+1}^n \binom{n}{i} p_k^i (1-p_k)^{n-i} & \text{if } k \leq n/2 \\ \sum_{i=0}^{k-1} \binom{n}{i} p_k^i (1-p_k)^{n-i} & \text{otherwise} \end{cases}$$

である. 具体的に $n=1000$ の場合を考えて k を変化した時に, 偏り傾向が強くなる確率, 弱くなる確率がどう推移するかのグラフを図 39 に示す. この図を見る限りは, 確かに偏り傾向が強くなる (従って多様性の度合いが下がる) 確率の方が弱くなる確率より大きくなっている.

具体的に, ヒープ整列法版を組み込んだ提案 PMBGNP を迷路問題 1 に適用した上記の場合は, 初期の世代でほとんど全ての個体の適合度が 0 であり, ランダムな個体がエリート個体, 良好個体として扱われるだけとなる. その結果, ($c_{node}=c_{connect}=0$ という設定もあり) サンプルングの際に偶然生じた偏りが次世代の個体分布に反映され, その確率分布を用いたサンプリングで更に別の偏りが生じることが繰り返され, 徐々に集団内の遺伝子の多様性が失われてしまったものと推測される.

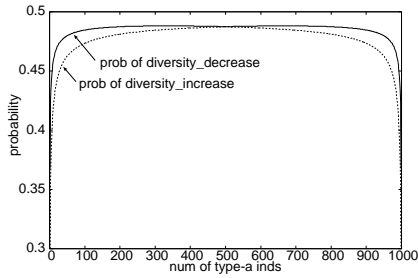


図 39 偏り傾向が強くなる確率 vs. 弱くなる確率

一方、クイック整列法の **partition** 処理を変形した版を組み込んだ提案 **PMBGNP** を迷路問題 1 に適用した場合 (図 33～図 35) は、初期の状況でほとんど全ての個体の適合度が 0 であるという状況は同じであるが、その様な状況の時に個体が格納されている配列上で個体の配置換えがあまり起きない。その結果、その様な状況が続く間は、エリート個体として扱われる個体が変わることはほとんど無く、初期に構成された (それ故、おそらく偏りの余り無い) 遺伝子部品が確率分布の構成に使われることになる。従って、ヒープ整列法版を組み込んだ場合と比べて、構成した確率分布の中に偶然生じる偏りが蓄積しにくいと考えられる。

通常の **GA** の場合は同じ個体が何世代にも渡ってエリート個体として扱われると個体群中の個体の多様性の低下を招くが、以上により、確率モデル構築型の場合は多様性維持に貢献することもある、という事例が示されたことになる。

6.4 確率モデルを明示的に構成せずに同等のサンプリングを行う方法

第 4 章で提案した **PMBGNP** においては、毎世代、各々のノード毎に、ノードの基本動作を変域とする 1 個の確率分布 $\{\Pr(K_i, ID_i)\}$ と、ノード番号を変域とする (ノード上で可能な基本動作数 \times ノードからの平均分岐数) 個の条件付き確率分布 $\{\Pr(C_{ij}|K_i, ID_i)\}$ を構築することになる。従って、構築される確率モデルは、全部で (ノード数 $- 1) \times (\text{可能な基本動作数} + (\text{ノード数} - 1) \times \text{可能な基本動作数} \times \text{ノードからの平均分岐数})$ 個の確率データ (実数値) の組で表されることになる。1 世代当たりの計算量については、確率モデル構築の際に $\mathcal{O}(\text{良好個体数} \times \text{ノード数} \times \text{ノードからの最大分岐数}) + \mathcal{O}(\text{ノード数}^2 \times \text{可能な基本動作数} \times \text{ノードからの平均分岐数})$ の手間がかかり、新規個体生成の際に $\mathcal{O}(\text{生成個体数} \times \text{ノード数} \times \text{ノードからの最大分岐数} \times \log_2 \text{ノード数})$ の手間がかかることになる。従って、確率モデルの構築にはノード数²に比例した時間を要することになり、ノード数が大きくなるにつれて、この処理が全体の計算量のかなりの部分を占める様になる。

一方、[丹治 11] の様に、確率モデルの構築を明示的に

は行わずに、確率モデルに従った新規個体の生成を良好個体のデータを組み合わせて行うという道もある。我々の場合は、良好個体群 P_{good} が明らかになった時点で、各々のノード番号 i と、番号 i のノードで可能なノード遺伝子 $l = (k_i, id_i)$ 毎に、良好個体群の部分集合

$$S_{il} \stackrel{\text{def}}{=} \{ind \in P_{\text{good}} | ind \text{ においては}$$

番号 i のノードのノード遺伝子が $l\}$

について調べておけば、この S_{il} と P_{good} の情報を基にラベルが l の時の弧の接続先に関する頻度分布を作り上げることができ、また、頻度分布が無くてもその頻度分布に沿ったサンプリングを行うことが出来ることに着目する。具体的に P_{good} の分布を近似する確率モデルに沿って新規個体の生成を行うには、次の①,②を順に行えば良い。

- ① 各々のノード番号 i に対して、 P_{good} の中からランダムに個体を選んで、その i 番目のノードのノード遺伝子を新規個体の i 番目のノードのノード遺伝子としてコピーする。
- ② 各々のノード番号 i 、接続遺伝子番号 j に対して、前ステップ①で確定している i 番目のノードのノード遺伝子 l を調べ出す。そして、 S_{il} の中からランダムに個体を選んで、その i 番目のノード中の j 番目の接続遺伝子を新規個体の i 番目のノード中の j 番目の接続遺伝子としてコピーする。

従って、第 4 章で考えた **PMBGNP** の探索手順と同等のものとして、次の形の探索手順を考えることができる。

ノードの種類・内容も探索する **PMBGNP** の探索手順：

パラメータ **Num_nodes**(個体中のノード数)、**Popsizel**(個体群サイズ) の値を決め、個体を保持するための配列 $ind[0] \sim ind[\text{Popsizel}-1]$ を用意する。また、各々のノード番号 i と、番号 i のノードで可能なノード遺伝子 $l = (k_i, id_i)$ 毎に、良好個体の部分集合 S_{il} の情報を保持するための領域を確保する。そして、探索を制御する 4 つのパラメータ $\text{rate}_{\text{elite}} \in [0, 1)$, $\text{rate}_{\text{good}} \in (0, 1)$, $\text{c}_{\text{node}} \geq 0$, $\text{c}_{\text{connect}} \geq 0$ の値を決め、次の手順に従って解探索を行う。

個々の個体 $ind[i]$ を 4 章で示した (方針 1) の手順に従ってランダムに生成することによって初期個体群 $P(0) = \{ind[0], \dots, ind[\text{Popsizel}-1]\}$ を構成

$P(0)$ 内の個々の個体の適合度を評価;

for ($t \leftarrow 0$; 終了条件を満たさない; $t \leftarrow t+1$) {

// エリート個体, 良好個体の選別

$n_{\text{elite}} \leftarrow \lfloor \text{Popsizel} \times \text{rate}_{\text{elite}} \rfloor$;

$n_{\text{good}} \leftarrow \lfloor \text{Popsizel} \times \text{rate}_{\text{good}} \rfloor$;

次の 2 つの条件を満たす様に個体 $ind[0] \sim ind[\text{Popsizel}-1]$ を並べ替える。

(条件 1) ($\forall i < n_{\text{elite}})(\forall j \geq n_{\text{elite}}) ind[i]$ の適合度は $ind[j]$ の適合度以上

(条件 2) ($\forall i < n_{\text{good}})(\forall j \geq n_{\text{good}}) ind[i]$ の適合度は $ind[j]$ の適合度以上;

// $P_{\text{good}} = \{ind[0], \dots, ind[n_{\text{good}}-1]\}$ を基に

// S_{il} の情報を調べる

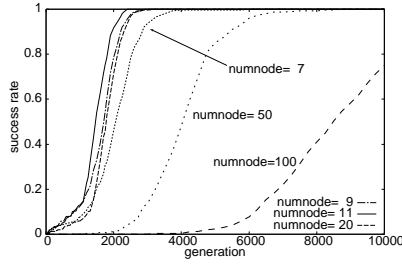


図 40 Num_nodes を変化させた時、成功率の推移がどう変わるか (迷路問題 1)

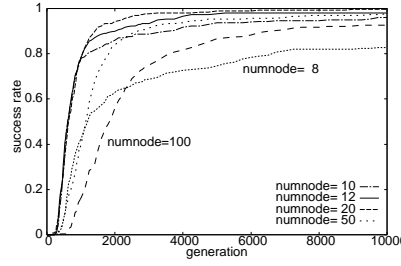


図 41 Num_nodes を変化させた時、成功率の推移がどう変わるか (迷路問題 2)

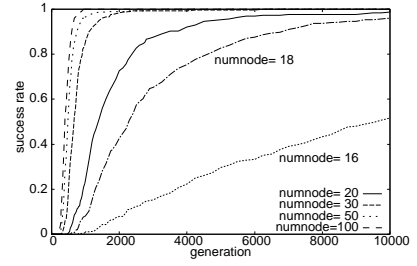


図 42 Num_nodes を変化させた時、成功率の推移がどう変わるか (11 マルチプレクサ問題)

```

for (each ノード番号  $i$ ) {
  for (each ノード遺伝子の組  $l = (k_i, id_i)$ ) {
     $S_{il} \leftarrow \emptyset$ ;
  }
}
for (each 良好個体番号  $k$ ) {
  for (each ノード番号  $i$ ) {
     $k_i \leftarrow ind[k]$  の  $i$  番目のノードの
      種類として設定された値;
     $id_i \leftarrow ind[k]$  の  $i$  番目のノードの
      基本動作番号として設定された値;
     $S_{i,(k_i, id_i)} \leftarrow S_{i,(k_i, id_i)} \cup \{ind[k]\}$ ;
  }
}
// 確率モデルに沿って新規個体を生成
for ( $k \leftarrow n_{elite}$ ;  $k < Popsiz$ ;  $k \leftarrow k + 1$ ) {
  // 式 (2) を基に新規個体を生成し  $ind[k]$  に格納
  for (each ノード番号  $i > 0$ ) {
     $ind[k]$  の  $i$  番目のノードにおけるノード
      遺伝子について、式 (3) によって定まる
      確率  $p_{node}$  で一様分布に基づいて確率的に
      設定し、残りの確率  $(1 - p_{node})$  で ①  $P_{good}$ 
      の中からランダムに個体を選び、② その  $i$ 
      番目のノードのノード遺伝子として新規個体
      の  $i$  番目のノードのノード遺伝子として
      コピーする、ことによって確率的に設定
      ;
     $k_i \leftarrow ind[k]$  の  $i$  番目のノードの
      種類として設定された値;
     $id_i \leftarrow ind[k]$  の  $i$  番目のノードの
      基本動作番号として設定された値;
    for (each 分岐番号  $j$ ) {
       $ind[k]$  の  $i$  番目のノードにおける  $j$  番
        目の接続遺伝子については、式 (4) に
        よって定まる確率  $p_{connect}$  で (0 と  $i$  以外
        が等確率で出る) 一様分布に基づいて確
        率的に設定し、残りの確率  $(1 - p_{connect})$ 
        で ①  $S_{i,(k_i, id_i)}$  の中からランダムに個体
        を選び、② その  $i$  番目のノード中の  $j$ 
        番目の接続遺伝子として新規個体の  $i$  番
        目のノード中の  $j$  番目の接続遺伝子として
        コピーする、ことによって確率的に
        設定
        ;
    }
  }
}

```

表 3 第 4 章の PMBGNP 探索手順を用いた 5.5 節の実験 (図 9～図 11)、確率モデルの明示的な構築を行わない PMBGNP 探索手順を用いた 6.4 節の実験 (図 40～図 42) に要した CPU 時間の比較。単位は分。

ノード数	迷路問題 1		迷路問題 2		11 マルチプレクサ問題	
	確率モデルを構築 (図 9)	確率モデルを構築せず (図 40)	確率モデルを構築 (図 10)	確率モデルを構築せず (図 41)	確率モデルを構築 (図 11)	確率モデルを構築せず (図 42)
7	57.6	52.9	—	—	—	—
8	—	—	61.5	86.9	—	—
9	45.1	45.1	—	—	—	—
10	—	—	37.7	39.2	—	—
11	43.8	42.2	—	—	—	—
12	—	—	23.4	30.1	—	—
16	—	—	—	—	47.7	52.9
18	—	—	—	—	29.2	25.3
20	67.7	63.8	34.8	29.5	21.6	17.4
30	—	—	—	—	19.0	10.9
50	311.2	256.7	105.2	83.8	27.5	13.8
100	1224.5	919.8	308.0	250.8	62.0	24.0

```

}
}
// 個体評価
 $ind[n_{elite}] \sim ind[Popsiz - 1]$  の適合度を評価;
}

```

この探索手順の場合、1 世代当たりの計算量は、 S_{il} の構成の際に $\mathcal{O}(\text{ノード数} \times \text{可能な基本動作数}) + \mathcal{O}(\text{良好個体数} \times \text{ノード数})$ となり、新規個体生成の際に $\mathcal{O}(\text{生成個体数} \times \text{ノード数} \times \text{ノードからの最大分岐数})$ となる。従って、1 世代当たりの計算量は高々ノード数に比例した程度で抑えられる。ノード数が多い場合には、確率モデルを毎世代明示的に構築する PMBGNP を用いるより、こちらの方が速く探索を進められることが期待できる。

実際に、確率モデルの明示的な構築を行わない上記 PM-BGNP 探索手順を用いて第 5.5 節と同じ条件で実験を行い、成功率の推移を観察した。その結果を図 40～図 42 に示す。図 9～図 11 とほぼ同等の結果が得られていることが確認できる。また、比較のために 5.5 節の実験に要した CPU 時間と、この節での実験に要した CPU 時間をまとめたものを表 3 に示す。ここでの実験に用いた計算機は Core i7-4770(3.4GHz) という CPU と 4GB のメモリを有した PC/Linux である。処理時間を抑えるために最適解が得られた時点で探索を打ち切っているため、

表3を見ると、迷路問題1でノード数=7~11, 迷路問題2でノード数=8~12, 11 マルチプレクサ問題でノード数=16~30の場合の様にノード数を大きくすることによって成功率が上がり CPU 時間が少なくなる場合も見受けられる。しかし、確率モデルを毎世代構築する第4章のPMBGNP探索手順を用いた場合にはノード数が大きくなると急激に計算量が増加しているのに対し、確率モデルの明示的な構築を行わないこの節のPMBGNP探索手順を用いた場合にはノード数が大きくなってもそれ程急激な計算量の増加には繋がっていない、ということを確認できる。

7. ま と め

本論文では、ノードの種類・内容(基本動作)も探索対象とする確率モデル構築型GNPを提案し、その探索性能を2種類の迷路問題と11 マルチプレクサ問題でテストした。その結果、提案手法はテストした問題に対しては、個体群サイズが1000程度の設定でも最低限に必要なノード数が与えられていればパラメータ調節により60~100%の割合で探索成功に至ることが確認された(5.5節)。また、ノードの種類・内容を固定し弧の接続先だけを探索する場合と比べて、提案手法の有用性が確認された(5.6節)。個体の多様性維持のためには、(構築した確率モデルに従わずに)ある確率で一様分布に従って新規個体の要素値を決めるということの有用性が明らかになった(5.4節, 6.1節)。適合度が同じ場合は古くから残っているものを優先するという形でエリートを一定の割合で残すことが個体の多様性維持に繋がることもある、ということも明らかになった(6.3節)。最後に、個体を構成するグラフのノード数が大きい場合には、計算量を抑えるために、確率モデルを明示的に構築せずに同等のサンプリングを良好個体群の情報を用いて行う手法が有用であることが実験で確認された(6.4節)。

謝 辞

5.6節は査読者の方々からの照会をきっかけに構成しました。有益な意見を出して下さった査読者の方々に深く感謝します。

◇ 参 考 文 献 ◇

- [Hirasawa 01] Hirasawa, K., Okubo, M., Hu, J., Katagiri, H., and Murata, J.: Comparison between Genetic Network Programming(GNP) and Genetic Programming(GP), in *Proc. of Congress on Evolutionary Computation*, pp. 1276–1282 (2001)
- [Katagiri 00] Katagiri, H., Hirasawa, K., and Hu, J.: Genetic Network Programming — Application to Intelligent Agents —, in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 3829–3834 (2000)
- [Koza 92] Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press (1992)
- [Li 09] Li, X., Mabu, S., Zhou, H., Shimada, K., and Hirasawa, K.: Genetic Network Programming with Estimation of Distribution Algorithms and its Application to Association Rule Mining for Traffic Prediction, in *Proc. of ICROS-SICE International Joint Conference*, pp. 3457–3462 (2009)
- [Li 10a] Li, X., Mabu, S., and Hirasawa, K.: Toward the Maintenance of Population Diversity: A Hybrid Probabilistic Model Building Genetic Network Programming, *進化計算学会論文誌*, Vol. 1, No. 1, pp. 89–101 (2010)
- [Li 10b] Li, X., Mabu, S., Zhou, H., Shimada, K., and Hirasawa, K.: Genetic Network Programming with Estimation of Distribution Algorithms for Class Association Rule Mining in Traffic Prediction, in *Proc. of IEEE Congress on Evolutionary Computation*, pp. 2673–2680 (2010)
- [Li 11a] Li, X., Li, B., Mabu, S., and Hirasawa, K.: A Novel Estimation of Distribution Algorithm Using Graph-based Chromosome Representation and Reinforcement Learning, in *Proc. of IEEE Congress on Evolutionary Computation*, pp. 37–44 (2011)
- [Li 11b] Li, X., Mabu, S., Li, B., and Hirasawa, K.: Probabilistic Model Building Genetic Network Programming Using Reinforcement Learning, *進化計算学会論文誌*, Vol. 2, No. 1, pp. 29–40 (2011)
- [Li 14] Li, X., Mabu, S., and Hirasawa, K.: A Novel Graph-Based Estimation of the Distribution Algorithm and Its Extension Using Reinforcement Learning, *IEEE Trans. on Evolutionary Computation*, Vol. 18, No. 1, pp. 98–113 (2014)
- [Mabu 02] Mabu, S., Hirasawa, K., Hu, J., and Murata, J.: On-line Learning of Genetic Network Programming (GNP), in *Proc. of the World on Congress on Computational Intelligence*, pp. 321–326 (2002)
- [Mabu 07] Mabu, S., Hirasawa, K., and Hu, J.: A Graph-based Evolutionary Algorithms: Genetic Network Programming (GNP) and its Extension Using Reinforcement Learning, *Evolutionary Computation*, Vol. 15, No. 3, pp. 369–398 (2007)
- [松本 98] 松本 眞: コイン投げで一儲けする方法 — 疑似乱数研究の現状 —, *情報処理*, Vol. 39, No. 11, pp. 1166–1170 (1998)
- [Pelikan 02] Pelikan, M., Goldberg, D. E., and Lobo, F. G.: A Survey of Optimization by Building and Using Probabilistic Models, *Computational Optimization and Applications*, Vol. 21, pp. 5–20 (2002)
- [Shapiro 06] Shapiro, J. L.: Diversity Loss in General Estimation of Distribution Algorithms, in *(LNCS 4193) Parallel Problem Solving from Nature — PPSN IX*, pp. 92–101 (2006)
- [(社)電気学会 10] (社)電気学会 進化技術応用調査専門委員会 (編): 進化技術ハンドブック 第I巻 基礎編, 近代科学社 (2010)
- [丹治 11] 丹治 信, 伊庭 斉志: 関係クリック推定を用いた確率モデル遺伝的プログラミング, *進化計算学会論文誌*, Vol. 2, No. 1, pp. 41–52 (2011)

[担当委員: 花田 良子]

2014年3月23日 受理

著 者 紹 介



元木 達也(一般会員)

1977年電気通信大学電気通信学部電子計算機学科卒業。1979年同大学大学院電気通信学研究科電子計算機学専攻修士課程修了。同年茨城大学工学部助手。1987年8月新潟大学教養部講師。1992年2月同部助教授。1994年より同大学工学部に所属。現在に至る。進化的計算の研究に従事。ACM, IEEE, 情報処理学会, 人工知能学会各会員。



小林 涼

2011年新潟大学工学部情報工学科卒業。2013年同大学大学院自然科学研究科電気情報工学専攻情報工学コース修士課程修了。現在、日本アルゴリズム株式会社に勤務。