

重要度指標付き Genetic Network Programming における 機能切替えについて

江藤 慎治[†] 畠山 裕之[†] 間 普 真 吾[†]
平澤 宏太郎[†] 古 月 敬 之[†]

これまで、多くの進化型計算手法がエージェントの行動系列の作成のために開発されてきたが、Genetic Network Programming (GNP) もその 1 つである。本論文では、脳の機能局在を GNP に導入し、人工脳を構成論的に構築することを目指す。脳の機能局在とは、脳の異なる部分が特定の機能に対応していることをいう。GNP は有効グラフを基本構造としており、機能局在を実現するのに適している。本論文では、特に環境に応じて機能の切替えを行う GNP の獲得を重要度指標付き GNP を用いて行った。

Functional Localization Using Genetic Network Programming with Importance Index

SHINJI ETO,[†] HIROYUKI HATAKEYAMA,[†] SHINGO MABU,[†]
KOTARO HIRASAWA[†] and TAKAYUKI HURUZUKI[†]

Many methods of generating behavior sequences of agents by evolution have been reported. A new evolutionary computation method named Genetic Network Programming (GNP) has been also developed recently along with these trends. The aim of this paper is to build an artificial model to realize functional localization of GNP considering the fact that the functional localization of the brain is realized in such a way that a different part of the brain corresponds to a different function. In this paper, it is especially stated that the switching function for functional localization can be realized using GNP with Importance Index (GNP IMX).

1. はじめに

従来より、人工知能やロボティクスの分野においてプログラミングに関する様々な研究がなされている¹⁾。これらの研究は、ある環境下に存在するエージェントが与えられた目標を達成するために必要な行動の系列を生成することに関する。これは、エージェントを生物と見なすと、その生物の脳を設計することであるともいえる。このような人工脳の実現、すなわちエージェントの行動系列の作成は、従来、設計者によって行われてきた。しかし、エージェントの目的や環境がより複雑になるにつれ、そのような行動系列の設計が困難をきわめることは容易に想像できる。さらに、特定の設計者の考えに基づいて設計された行動では、エージェントの能力が設計者のスキルに大きく依存するといった問題や、環境変化が起これると与えられた行動系

列では対処できないケースが多いといった問題があげられる。このことから、自律性と創造性に富んだエージェントの行動系列は、設計者によって与えられるよりはむしろ進化的手法により、設計者への依存度を低くし自動的に獲得されることが望ましい。

こうした背景から、エージェントの行動系列の自動設計に関する多くの研究がなされてきた^{2)~6)}。そのようなエージェントの行動系列を実現する手法に、Genetic Algorithm (GA), Evolution Strategy (ES), Evolutionary Programming (EP), Genetic Programming (GP), Parallel Algorithm Discovery and Orchestration (PADO) などの進化型計算が多数提案されている^{2)~6)}。しかし、これまでの手法では、エージェントが現在置かれている環境の情報をすべて用いて今何をするべきかという点を重視した行動系列の生成を試みているものが多い。これに対し、本論文では、現在必要と思われる環境に関する情報のみを必要に応じて用いるといった、現実の脳が行っている機能に近い手法で行動系列の生成を行う。脳は目や耳などの感覚器

[†] 早稲田大学大学院情報生産システム研究科
Graduate School of Information, Production and Systems, Waseda University

官から得られる情報のすべてを利用して行動を決定するのではなく、必要に応じて情報の取捨選択を行い、それを利用して行動を決定する。そのような必要な情報のみを必要に応じて用いるという脳に近い機能を持ち、部分観測マルコフプロセス環境下での行動系列の生成を行うことのできる Genetic Network Programming (GNP)^{7)~10)} を用いてエージェントの行動系列の自動設計を行う。GNP は単純な機能を持った複数のノードをネットワーク状に接続し、ノード間の遷移規則により行動系列を表現するモデルである。

ところで、一般に人間の脳は、異なる機能を異なる部分で処理するという機能局在構成をしている。また、現実の問題の多くが複雑な構造をしている。このような複雑な問題を機能局在構成の人間の脳とは異なる均一構造の従来の進化型的方式で解決することは難しい。一方、脳の機能局在をニューラルネットワーク (NNs) によって実現しようとする研究^{11),12)} も行われている。本論文では、NNs で検討されてきた機能局在を構成論的進化型計算手法である GNP 上で展開することを試みる。

本論文での機能局在モデルとは、複雑な問題を複数個のタスクに分割し、各タスクに対応した Sub-GNP もしくは固定のプログラムが状況に応じて適宜動作するモデルである。なお、タスク分割の自動化については本論文では検討していない。本論文では機能局在型 GNP の基本的な検討を行っており、機能の切替えを行う GNP に重要度指標付き GNP (GNP with IMX) を用い、切替え GNP を進化により獲得する自己充足的ゴミ拾い問題¹³⁾ を取り扱っている。GNP with IMX では重要度指標 (Importance Index, IMX) を計算することにより、従来の GNP と異なり、判定ノードの分岐の重要度の結果をも考慮して処理を行うことが可能となる。

2. Genetic Network Programming (GNP)

2.1 GNP の構成要素

GNP の基本構成を図 1 に示す。GNP は個体を有向グラフで表現し、各ノードには判定ノード J_m と処理ノード P_n が設定されている。 J_m は判定ノードのライブラリに記載されている m 番目の判定を表し、 P_n は処理ノードのライブラリに記載されている n 番目の処理を表す。ライブラリには設計者が用意したエージェントの最小の行動単位が記載されており、内容に重複はないものとする。また、S は開始ノードを示し、GNP はこのノードから開始される。

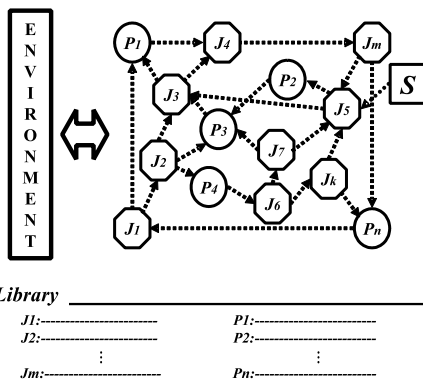


図 1 GNP の基本構成
Fig. 1 The basic structure of GNP individual.

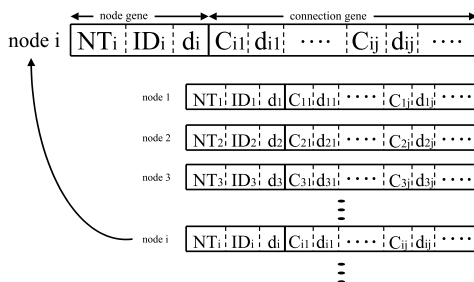


図 2 GNP の遺伝子表現
Fig. 2 The genotype expression of the nodes in GNP.

2.2 GNP の遺伝子表現

GNP の遺伝子表現を図 2 に示す。ノード i の遺伝子はノードの内容と接続に関する以下の 2 種類の情報よりなる。

- ノードの内容に関する情報
 - NT_i : ノード i の種類 (判定ノード/処理ノード)
 - ID_i : 判定/処理内容
 - d_i : 判定/処理時間
- ノード間の接続に関する情報
 - C_{ij} : ノード i の j 番目の分岐に接続するノード番号
 - d_{ij} : ノード i からノード C_{ij} への遷移の遅れ時間

d_i や d_{ij} のような時間を設定するのは、判定・処理に要する時間や遷移の遅れ時間を与えることによって、より現実的なモデルをつくるためである。また、エージェントが一連の行動を行う時間を 1 ステップと定義し、1 ステップの最大時間を適切な値に設定することにより、エージェントが設定された時間内に何らかの環境への動作 (何もしないを含む) を行うことも

できる．1ステップ後は他のエージェントが同様に一連の判定および処理を実行する．具体的には，各エージェントは，判定/処理時間と遅れ時間の総和が最大時間を超えるまで GNP 内の 1 ステップのノード遷移を行う．本論文のシミュレーションでは，判定時間を 1 単位時間，処理時間を 5 単位時間，ノード間の遅れ時間をすべて 0 単位時間，最大時間を 5 単位時間に設定した．

2.3 GNP の進化

各 GNP 個体のノード数は同一であり，GNP 内の各ノードはユニークな番号を持っている，同一番号のノードの種類，機能は同一とする．また，GNP 内の種類，機能ごとのノード数は同一とし，ノード間の初期接続はランダムとする．

本論文での，GNP の進化は，簡単のためにノード遺伝子の接続に関する情報のみを進化させることによって実現する．すなわち，ノード間の接続のみを進化的に変更することによって適切な構造を得ようというものである．

個体の進化のためのオペレータとして，交差は「2親個体内の対応するノード間接続遺伝子を交差率 P_c で確率的に交換する」オペレータを，また，突然変異は「ノード間接続遺伝子を突然変異率 P_m で確率的に変更する」オペレータを採用した．ここで，個体とは GNP プログラムを構成するために必要なノード遺伝子の集合のことを意味する．すなわち，1 個体は複数のノード遺伝子からなる 1 つの有向グラフである．

2.4 重要度指標付き GNP

従来の GNP は，判定ノードによって環境の情報を判定し，判定結果によって次のノードに遷移し，処理ノードへ接続されるとその処理内容を実行する．

重要度指標付き GNP では判定ノードの判定変数 $J_l (l \in L)$ に処理 $P_n (n \in N)$ の重要度指標 $IMX(J_l, P_n)$ を設定し，処理ノードでは，処理ノードに至る判定ノードの重要度指標を使用して処理 $P_n (n \in N)$ の強度 $S(P_n) (n \in N)$ を計算する．

$$S(P_n) = \sum_{l \in L'} IMX(J_l, P_n) \tag{1}$$

ここで

L' : 処理ノードに至る判定ノードの判定変数の添字の集合

L : 判定ノードでの判定変数の添字の集合

N : 処理の種類添字の集合

なお，処理ノードでは，強度 $S(P_n) (n \in N)$ が最大となる処理を選択する．この結果，従来の GNP と異なり，重要度指標付き GNP では，

- 重要度指標に対象の先見情報を導入することにより，GNP のノード数を小さく設定でき，より効率的な GNP の進化が可能となる，
 - 先見情報による重要度指標を活用するのではなく，重要度指標そのものを進化により獲得することで，より性能の高い GNP を構成できる，
- などの特徴を有する．また，判定変数 J_l を $|K(l)|$ 個の領域 $(D_1, D_2, \dots, D_{k(l)}, \dots, D_{|K(l)|})$ に分割し， $J_l \in D_{k(l)}$ の場合は判定変数 J_l の判定ノードの第 $k(l)$ 番目の分岐を行い，このときの処理 P_n の重要度指標を $IMX(k(l), n)$ とすると処理 P_n の強度 $S(P_n)$ は次のように計算できる．

$$S(P_n) = \sum_{l \in L'} IMX(k(l), n) \tag{2}$$

3. 機能局在型 GNP

本章では GNP の拡張である，機能局在型 GNP について説明する．

3.1 GNP の機能局在モデル

脳はすべての神経細胞が他の細胞と接続されているのではなく，いわゆる機能局的なモジュール構造をしている．この脳の機能局在をモデル化するため，本論文では，異なる機能を持つ 1 個のモジュールを Sub-GNP として，これを組み合わせることで機能局在型の人工脳を構築することを考える．

機能局在型 GNP の基本構造を図 3 に示す．機能局在型 GNP は異なる動作を実現する複数の Sub-GNP と Sub-GNP に切替え動作指令を出力する Switch-GNP より構成される．すなわち，機能局在型 GNP は複数の Sub-GNP を GNP 内部に有し，問題の状況（環境）に応じて Switch-GNP によりそれらを使い分けることによって，従来の GNP と比べ高い処理能力を目指す GNP である．

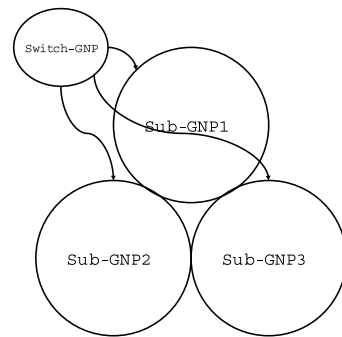


図 3 機能局在型 GNP の基本構造

Fig. 3 The basic structure of functional localization of GNP.

複数の機能を持つ機能局在型 GNP では、それぞれの Sub-GNP の機能分担とその構成も重要ではあるが、その機能の切替えも重要である。タスクを適切に機能分担し、これに Sub-GNP を割り当て、Sub-GNP を適切に構築しても、状況に応じて Sub-GNP を切り替えることができれば、全体として問題を解くことが困難となるからである。

本論文では機能局在型 GNP の初期検討として、Sub-GNP と Switch-GNP を同時に進化で獲得するのではなく、Sub-GNP はあらかじめ得られているものとして、Switch-GNP のみを進化により求める検討を行った。Sub-GNP が与えられた場合の Switch-GNP の獲得に適したアプリケーションとしてエレベータ群管理制御¹⁴⁾がある。Sub-GNP は、その性質上、問題の先見知識から設定できるケースも多く、Switch-GNP のみの獲得であっても十分な価値があると考えられる。

3.2 Switch-GNP with IMX

機能局在型 GNP の Switch-GNP では、前述の重要度指標付き GNP を用いる。具体的には、Switch-GNP が Sub-GNP の切替え指示を行うために必要な重要度指標と処理の強度を計算する機構を判定ノードと処理ノードに設ける。Sub-GNP の数が、処理の数に対応する。提案する Switch-GNP では、判定ノードと処理ノードで得られる重要度指標の値をノード遷移にともなって加算し、得られた処理の強度の中で最も高い値を持つ Sub-GNP への切替え指示を行う。

提案方式の特徴の 1 つに、処理ノードに遷移しなくても Sub-GNP への切替え指示が可能にある。たとえば、GNP の遷移において、判定ノードのみのループを検出した場合である。GNP の実行中にループを検出した場合には、そこで GNP の実行を停止し、その時点で最も大きな処理の強度の値を持つ Sub-GNP への切替え指示を行うことになる。

なお、本論文では以下の 5 つの切替え GNP についての比較を行った。

- (1) 重要度指標可変型 Switch-GNP (GNPFL(e) : 重要度指標を進化によって獲得する)。
- (2) 重要度指標固定 Switch-GNP (GNPFL(f) : 前もって重要度指標を設定する)。
- (3) 重要度指標ランダム固定 Switch-GNP (GNPFL(r) : 前もって重要度指標をランダムに設定する)。
- (4) 重要度指標なし Switch-GNP (GNPFL(n) : 従来型 GNP, ノード数により 2 種類)。

GNPFL(e) については、重要度指標の変更を行う以下の遺伝的オペレータの追加を行う。

- 重要度指標の突然変異：判定ノードおよび処理ノードの重要度指標を突然変異確率 P_{im} で確率的にランダムに変更する。

4. シミュレーション

4.1 自己充足的ゴミ拾い問題

自己充足的ゴミ拾い問題とは、エネルギーレベルを維持しつつ、環境内のゴミを拾い集めるという問題である。エージェントは、ゴミを拾い集めるという外部環境に関するタスク 1 と、エネルギーレベルの維持という内部の指標に関するタスク 2 の、2 つのタスクを同時に解くことになる。このゴミの収集とエネルギーレベルの維持の 2 つのタスクをうまく切り替えることができてはじめて、全体の問題を解くことができる。

自己充足的ゴミ拾い問題はチェスボードのような 2 次元格子上に、エージェントと複数のゴミをランダムに配置し、固定の場所にゴミ収集所とエネルギー充電ステーションを配置した環境である (図 4 参照)。エージェントは 1 つのセル上にちょうど収まる大きさで、一度に前方に 1 セル動くか左か右に 90 度回転することができ、行動することでエネルギーを消費する。エージェントはゴミのあるセルに到達することでゴミを保持することができ、ゴミを保持した状態で収集所に到達することでゴミの収集を行う。エージェントの保持できるゴミの数は有限である。また、エネルギーレベルが 0 以下になるとエージェントは行動を停止する。エージェントは充電ステーションに到達するだけではエネルギーの充電を完全に行うことはできず、一定ステップとどまることで十分な量のエネルギーを得ることができる。

このエージェントのタスク切替え機能を GNP により進化で獲得することを試みた。

4.2 シミュレーション条件

環境の広さは 11 × 11 マスで端が壁で囲まれている

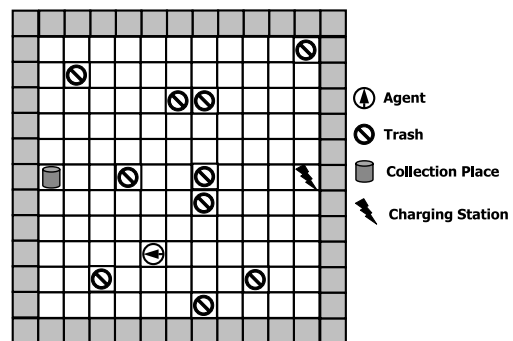


図 4 自己充足的ゴミ拾い問題

Fig. 4 Self-sufficient garbage collector problem.

表 1 パラメータ条件

Table 1 Parameter conditions for evolving Switch-GNP.

Generation	1,000
Number of GNP	100
Number of elite GNP	1
Number of crossover individuals	40
Number of mutation individuals	59 or 30
Number of mutation individuals using importance index	0 or 29
Crossover probability P_c	0.1
Mutation probability P_m	0.01
Mutation probability for importance index P_{im}	0.01
Number of nodes per each kind Q	1 or 3

環境とする。エージェントは環境内をすべて見渡すことができる。エージェントは1体、ゴミは10個、ゴミ収集所とエネルギー充電ステーションはそれぞれ1カ所とした。エージェントは250ステップ内ですべてのゴミの収集を行うことを目指す。エージェントが一度に保持できるゴミの数は2個までとした。この条件で、エージェントの初期位置および向きとゴミの初期位置をランダムに設定した10環境を用意し、それぞれの環境の得点の合計に応じてGNPを進化させる。

エージェントは行動を行うたびにエネルギーを消費するため、定期的に充電を行わなければならない。今回のシミュレーションでのエージェントのエネルギーの初期値は40とし、エージェントが前に進むと3消費し、左右に回転すると1消費する。充電量は式(3)で表現される。

$$\Delta E(t) = \frac{(100 - E(t))^2}{200}. \quad (3)$$

ここで ΔE はエネルギー充電量、 $E(t)$ は現在のエネルギーレベルである。 $E(t)$ が低ければ低いほど ΔE も多くなり、逆に $E(t)$ が高いと ΔE は少なくなる。

ゴミの収集、エネルギーレベルの維持の両タスクを、GNPの進化で獲得することも可能であるが、今回は切替え機能についての検討を行うために、上記の両タスクについては固定のプログラムを使用する。これらは、ゴミを拾うと収集所にゴミを運ぶタスク1と、充電ステーションに移動しそこにとどまり続けるタスク2の単純なプログラムで構成されている。

今回のシミュレーションで用いたパラメータを表1に示す。表中の右側の値は重要度可変型Switch-GNP (GNP_{FLE}(e))の場合の値である。これらの条件下で、Switch-GNPが適切にタスクを切り替えることができるかどうかの検討を行った。具体的には、ゴミを拾い集める機能を持つタスク1と充電を行うタスク2を適切に切り替え、より多くのゴミを集めることができるかどうかである。なお、表1中のQはノード内容ご

表 2 開始/判定/処理ノード関数

Table 2 Functions of a start node, processing nodes and judgment nodes.

	function
0	start node (SN)
1	check the state of the current energy (CSE)
1	check the distance to the collection place (DCP)
1	check the distance to the charging station (DCS)
1	check the task of one previous step (TASK)
2	choose Task1 (T1)
2	choose Task2 (T2)

とのノード数である。

4.3 GNPのノードの設定

以下は、今回のシミュレーションで用いる、Switch-GNPのノードについての説明である。ノードの内容およびノード内の重要度指標値の計算方法について説明する。

4.3.1 開始/判定/処理ノード

Switch-GNP内の開始/判定/処理ノードとして、表2で示されるノード群を用意した。0が開始ノード、1が判定ノード、2が処理ノードを示す。表中の()内の文字列はそれぞれのノードの省略表記である。開始ノードは単純にGNPの開始を示している。

判定ノードでは、それぞれの判定を行いその結果に従って次に実行するノードの選択を行う。なお、シミュレーションの判定ノードでは基本的に式(2)で処理の強度を計算する。「エネルギーの状態(CSE, J_1)」では、1~100のエネルギーレベルを10段階($D_1 = \{1-10\}$, $D_2 = \{11-20\}$, ..., $D_9 = \{81-90\}$, $D_{K(1)} = D_{10} = \{91-100\}$)に分け、エージェントのエネルギーレベルに応じて10分岐を行う。「収集所までの距離(DCP, J_2)」ではエージェントから収集所までの距離を3段階($D_1 = \{0-4\}$, $D_2 = \{5-7\}$, $D_{K(2)} = D_3 = \{8-\}$)に分け、距離に応じて3分岐を行う。「充電ステーションまでの距離(DCS, J_3)」では、エージェントから充電ステーションまでの距離

表 3 各ノード分岐での重要度指標の設定
Table 3 Setting of importance index in each node branch.

Function	Importance index																					
	T1	5		0																		
T2	0		5																			
CSE	1-10		11-20		21-30		31-40		41-50		51-60		61-70		71-80		81-90		91-100			
	-2	5	-1	5	0	5	1	5	2	4	3	3	4	2	5	1	5	0	5	-5		
DCP	0-4		5-7		8-																	
	3	0	2	0	1	0																
DCS	0-4		5-7		8-10		11-13		14-													
	0	2	-1	1	-2	0	-3	0	-4	0												
TASK	T1		T2		NONE																	
	4	0	1	3	0	0																

を5段階 ($D_1 = \{0-4\}$, $D_2 = \{5-7\}$, $D_3 = \{8-10\}$, $D_4 = \{11-13\}$, $D_{K(3)} = D_5 = \{14-\}$) に分け、距離に応じて5分岐を行う。「1ステップ前のタスク (TASK, J_4)」では1ステップ前に選択したタスクがタスク1か、タスク2か、なし (NONE) か、によって3分岐する。基本的にタスクの切替えに重要な判定ほど多くの分岐を持つように設計されている。これについては次節で述べる。

処理ノードは1つの接続先を持ち、タスク1あるいはタスク2のいずれかを選択した後、接続先へ遷移する。

4.3.2 重要度指標の設定

重要度指標は、本論文で提案する Switch-GNP において最も重要な指標である。従来の GNP では、処理ノードに遷移することによってのみ GNP の処理を行ってきた。しかし、Switch-GNP の処理ノードではタスク1の処理ノードおよびタスク2の処理ノードでかならずしもタスク1あるいはタスク2が選択指示されるわけではない。

前述したように、処理ノードに至るまでの判定ノードで各 Sub-GNP の重要度指標を加算し、これに処理ノードの各 Sub-GNP の重要度指標を加えて各 Sub-GNP の強度を計算し、最も強度の大きい Sub-GNP (タスク) が選択指示されることになる。

表3に GNPFL(f) における判定ノードの判定・分岐および処理ノードでの重要度指標を示す。表中の左側がタスク1、右側がタスク2についての重要度指標である。今回のシミュレーションにおいて、切替えに関して最も重要な情報はエージェントのエネルギーの状態であることは明らかである。そこで、判定ノード「エネルギーの状態 (CSE)」ではエネルギーレベルを10段階に分け、重要度指標に関して一番きめ細かい設定をしている。具体的には、エネルギーが少なくなると、タスク2が出力されやすくなり、エネルギーが高くなるとタスク1が出力されやすくなるように設計さ

れている。判定ノード「収集所までの距離 (DCP)」では、収集所に近い場所にいるときほどタスク1が出力されやすくなるように設定されている、ただし判定ノード DCP は切替えにはさほど重要であるとは考えられないため、タスク2に関しては判定結果にかかわらずその重要度指標の値は0となっている。エージェントが充電ステーションから離れれば離れるほど、充電ステーションに戻るためのエネルギーが大きくなるため、判定ノード「充電ステーション (DCS)」では、エージェントからエネルギー充電ステーションまでの距離が離れれば離れるほど、タスク1の重要度指標の値を低くする設定になっている。「1ステップ前のタスク (TASK)」では、極力同じタスクが連続して指示されやすくなるように設計されたノードである。ただし、エネルギーレベルを維持するタスク2は、あくまでもゴミの収集のタスク1のために存在しているタスクなので、タスク1とタスク2の重要度指標の差の絶対値は T2 の場合が T1 の場合より小さくなっている。また、この重要度指標の設定が有用であるかを測るために、重要度指標をランダムに設定して固定した GNPFL(r) についてもシミュレーションを行った。GNPFL(r) は、他の場合と同様に10回の独立したシミュレーションを行っているが、それぞれの試行で重要度指標をランダムに設定している。

なお、タスク1の処理ノードでは、タスク1の重要度指標が値を持ち、タスク2の重要度指標は0である。逆に、タスク2の処理ノードでは、タスク2の重要度指標が値を持ち、タスク1の重要度指標は0となっている。また、タスク切替えの指示は Switch-GNP の中で処理ノードに遷移するか、あるいは、判定ノードによるループが検出されたときに行われる。

表3を使用して、Switch-GNP の実行の一例を示す。まず、ノード CSE に遷移し、その判定結果が $J_1 \in D_5 = \{41-50\}$ で接続先が判定ノード DCP であった場合、タスク1の重要度指標に +2、タスク2

の重要度指標に +4 を行いノード DCP に遷移する．ノード DCP での判定結果が $J_2 \in D_2 = \{5-7\}$ でその接続先がタスク 1 の処理ノードであった場合、タスク 1 の重要度指標に +2 を行い、タスク 2 の重要度指標の更新は行われず、タスク 1 の処理ノードに遷移する．タスク 1 の処理ノードではタスク 1 の重要度指標に +5 を行い、GNP の実行が終了する．タスク 1 の重要度指標の加算値、換言するとタスク 1 の強度が 9、タスク 2 の重要度指標の加算値、換言するとタスク 2 の強度が 4 であるため、強度の高いタスク 1 が Switch-GNP により指示される．なお、タスク切替え指示が終了すると各タスクの強度はクリアされる．

GNPFL(e) の場合は、重要度指標の初期値は -5 から 5 の間でランダムである．また、重要度指標の突然変異もランダムに -5 から 5 の間の値となる．

4.4 適合度関数

シミュレーションで用いた適合度関数を式 (4) と式 (5) に示す．GNP の適合度はゴミを集めた個数とエネルギーの回復の総量によって行う．

$$Fitness = \sum_{n=1}^{10} (100T_n + CE_n), \tag{4}$$

$$Fitness = 100T. \tag{5}$$

ここで T_n は n 回目のシミュレーションで収集したゴミの数で、 CE_n は n 回目のシミュレーションで回復したエネルギーの総量である．また、 n は 10 個の環境の番号である．式 (4) は GNP の進化のときに用いた適合度関数で、10 環境の合計値である．また、式 (5) は得られた最良の GNP の汎化能力テストで用いた適合度関数で、 T は収集したゴミの数である．

5. シミュレーション結果と考察

10 種の異なる乱数系列を用いてシミュレーションを行い、集団の最良個体の適合度 (10 環境の合計値) の平均値で評価を行った．

図 5 より、GNPFL(e) と GNPFL(f) は、早い世代で問題に適合できていることが分かる．一方で、ノード内容ごとのノード数 Q が 1 である GNPFL(n)Q = 1 では進化も遅く、問題に対して完全に適合できていないことも分かる．また、ノード内容ごとのノード数 Q が 3 である GNPFL(n)Q = 3 の場合は、問題に対しては適合できているが、その進化速度は遅いことが分かる．重要度指標を追加したことにより GNPFL(e) と GNPFL(f) ではほぼすべての GNP で確実にタスクが選択できるのに対し、重要度指標を持たない GNPFL(n) はタスクを選択できない GNP が

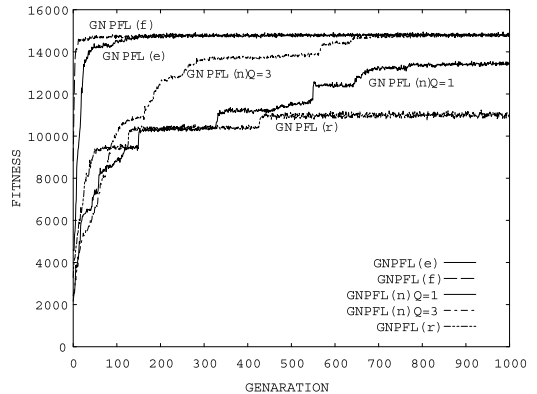


図 5 10 回の試行における最良個体の適合度 (10 環境の合計値) の平均値

Fig. 5 Average of the best fitness values (sum of 10 environments) over ten independent trials.

表 4 汎化能力 (式 (5) の平均適合度)

Table 4 Generalization (Average Fitness of Eq. (5)).

	MAX	AVE	MIN	STDEV
GNPFL(e)	847.9	761.7	733.0	33.8
GNPFL(f)	773.6	759.2	723.4	14.9
GNPFL(n)Q = 1	790.9	693.0	337.1	137.6
GNPFL(n)Q = 3	835.2	752.7	671.8	46.5
GNPFL(r)	800.6	200.2	53.1	215.3

多く発生する．そのため、重要度指標のある場合とない場合は進化の速度および最終的な問題への適合度に差ができています．なお、GNPFL(n)Q = 3 と GNPFL(e) の調整パラメータの数はほぼ同じになっています．また、GNPFL(e) と GNPFL(f) を比較した場合、重要度指標の調整が必要な分だけ GNPFL(e) の進化が遅くなっています．また、GNPFL(r) の問題に対する適合度の低さから、GNPFL(f) における重要度指標の設定が有用であることが分かる．

表 4 に、10 回の独立したシミュレーションで得られた 10 個の最良個体の汎化能力のテスト結果を示す．ここで汎化能力とは訓練した環境以外での問題解決能力である．10 回の試行で得られた 10 個の最良個体について異なる 1,000 環境下で式 (5) の適合度を計算し (合計 10,000 環境の適合度)、その最大値 (MAX)、平均値 (AVE)、最小値 (MIN)、標準偏差 (STDEV) を示している．GNPFL(e) が、他の 4 つに比べて訓練時と同程度の高い汎化能力を獲得していることが分かる．なお図 5 の適合度は 10 環境の合計値であり、回復したエネルギーの総量 CE も含んでいるのに対し、表 4 の適合度は CE がなく 1 環境の適合度である．GNPFL(f) は、重要度指標が固定であるため、訓練時には大きな差がなかったものの汎化能力は

GNPFL(e) に比べ低くなっている。GNPFL(n)Q = 1 では、比較的汎化能力の高い個体を得ることができる反面、平均も低く標準偏差も大きくなっている。また、GNPFL(n)Q = 3 では、その標準偏差は比較的高いものの、汎化能力の高い個体を得ることができている。GNPFL(r) では、重要度指標がランダムであるため、汎化能力の高い個体を得ることは困難である。

6. 結 論

本論文では、Genetic Network Programming (GNP) を拡張した機能局在型 GNP の提案を行い、その基礎的な検討を行った。具体的には、環境に応じてタスクを切り替えることのできる Switch-GNP を提案し、自己充足的ゴミ拾い問題において、Switch-GNP がゴミの収集とエネルギーレベルの維持という2個のタスクを適切に切り替えることができることを示した。

Switch-GNP は、判定ノードおよび処理ノードの内部に重要度指標を設けることで、従来の GNP では比較的困難なタスクの切替えという問題に適合することができることを明らかにした。

以上により重要度指標を持つ Switch-GNP が高い能力を有し、また、重要度指標を進化によって変化させることでさらに問題に適合できるということを示した。このことにより切替えタスクに適したコントローラが存在する問題に提案手法を応用可能である。また、現在、Switch-GNP と Sub-GNP を同時または逐次的に獲得する方法を検討しており、これにより提案手法を様々なアプリケーションに応用することが期待できる。

参 考 文 献

- 1) Brooks, R.A: Robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, Vol.2, No.1, pp.14–23 (1986).
- 2) 馬場口登, 山田誠二: 人工知能の基礎, 昭晃堂 (1999).
- 3) Holland, J.: *Adaptation in Natural and Artificial Systems — An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, University of Michigan Press, Ann Arbor (1975).
- 4) 伊庭斉志: 遺伝的プログラミング入門, 東京大学出版会 (2001).
- 5) Koza, J.R.: *Genetic Programming*, MIT Press, Cambridge, MA (1992).
- 6) Teller, A. and Veloso, M.: PADO: Learning Tree-structured Algorithms for Orchestration

into an Object Recognition System, Carnegie Mellon University, Technical Report Library (1995).

- 7) Eguchi, T., Hirasawa, K., Hu, J. and Ota, N.: A Study of Evolutionary Multiagent Models Based on Symbiosis, *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.36, No.1, pp.179–193 (Feb. 2006).
- 8) Hirasawa, K., Okubo, M., Hu, J. and Murata, J.: Comparison between Genetic Network Programming (GNP) and Genetic Programming (GP), *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp.1276–1282 (2001).
- 9) 平澤宏太郎, 大久保雅文, 片桐広伸, 胡敬炉, 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, *Trans. IEE of Japan*, Vol.121-C, No.6, pp.1001–1009 (2001).
- 10) 片桐広伸, 平澤宏太郎, 胡敬炉, 村田純一: Genetic Network Programming とそのマルチエージェントシステムへの応用, *Trans. IEE of Japan*, Vol.122-C, No.12, pp.2149–2156 (2002).
- 11) Happel, B.L.M. and Murre, J.M.J.: Design and evolution of modular neural network architectures, *Neural Networks*, Vol.7, pp.985–1004 (1994).
- 12) Xiong, Q., Hirasawa, K., Hu, J. and Murata, J.: A functions localized neural network with branch gate, *Neural Networks*, Vol.16, pp.1461–1481 (2003).
- 13) Pfeifer, R. and Scheier, C. (著), 石黒章夫, 小林宏, 細田 耕 (訳): 知の創成—身体性認知科学への招待, 共立出版 (2001).
- 14) 平澤宏太郎, 河内好一, 弓仲武雄, 岩坂達夫: 学習機械のエレベータ群管理制御への応用, *Trans. IEE of Japan*, Vol.90, No.8, pp.1568–1576 (1970).

(平成 17 年 9 月 26 日受付)

(平成 18 年 6 月 1 日採録)



江藤 慎治

昭和 53 年生。平成 15 年九州大学工学部電気情報工学科卒業。平成 17 年早稲田大学大学院情報生産システム研究科情報生産システム工学専攻修士課程修了。現在、早稲田大学大学院情報生産システム研究科情報生産システム工学専攻博士後期課程在学中。計測自動制御学会学生会員。



島山 裕之

昭和 57 年生。現在，早稲田大学大学院情報生産システム研究科情報生産システム工学専攻修士課程在学中。電気学会学生会員。



間普 真吾

昭和 53 年生。平成 13 年九州大学工学部電気情報工学科卒業。平成 15 年九州大学大学院工学府修士課程修了。現在，早稲田大学大学院情報生産システム研究科情報生産システム工学専攻博士後期課程在学中。電気学会，計測自動制御学会，IEEE の各学生会員。



平澤宏太郎（正会員）

昭和 17 年生。昭和 39 年九州大学工学部電気工学科卒業。昭和 41 年九州大学大学院工学研究科修士課程電気工学専攻修了。同年（株）日立製作所入社，日立研究所勤務，平成 1 年同研究所副所長。平成 3 年同大みか工場主管技師長。平成 4 年九州大学工学部教授。平成 8 年同大学大学院システム情報科学研究科教授。平成 12 年同大学院システム情報科学研究院教授。平成 14 年早稲田大学大学院情報生産システム研究科教授，現在に至る。電気学会，計測自動制御学会，IEEE の各会員。工学博士。



古月 敬之

昭和 37 年生。昭和 60 年中国中山大学大学院修士課程修了。同年同大学電子工学科助手，昭和 63 年同講師。平成 5 年来日，平成 9 年九州工業大学情報工学研究科博士後期課程修了。同年九州大学ベンチャービジネスラボラトリ非常勤研究員を経て，同大学大学院システム情報科学研究科助手，平成 12 年同大学院システム情報科学研究院助手。平成 15 年早稲田大学大学院情報生産システム研究科助教授，現在に至る。電気学会，計測自動制御学会の各会員。情報工学博士。