

O10 遺伝的プログラミングによる RARS ロボットの探索

黒田 信幸

元木 達也

新潟大学大学院 自然科学研究科

あらまし 本研究ではRARS [1]という、自動車レースシミュレータ上で動作するロボットの動作制御プログラムを、遺伝的プログラミングを用いて探索する。このシミュレータはロボットに対して数多くの環境情報を与えるが、それらを遺伝的プログラミングで効率良く扱うにはどうするかを検討する。

1 RARS について

RARS(Robot Auto Racing Simulator)は図1のようなある一定の幅のコースをまわるロボットの動きをシミュレートする。コースは、ストレートとカーブのセグメントを組合せてできていて、ロボットがコースから外れてしまった場合は減速するペナルティーが課せられる。本研究では図1のようにある程度の複雑さをもったコースを、1台のロボットができるだけ速く1周する制御プログラムを探索する。

シミュレータ上では各時点のロボットの動きは、ロボットの置かれた環境の情報(表1)を引数として受け取り、戻り値として、ステアリングホイールの角度(alpha)と望むスピード(vc)の組を返す関数で表される。この関数が呼ばれるごとに少しずつ進むので、同じ距離を動く場合、呼ばれる回数が少いほど速く動いたということになる。

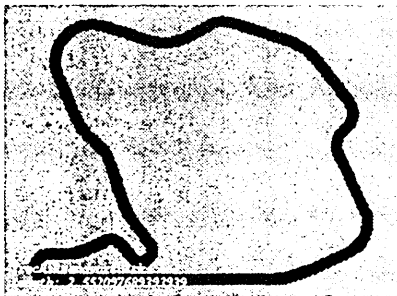


図1: 実験で使用するコース

シミュレータがロボットに与える情報としては表1に挙げた17種類のものがある。コースの状況については、今いるセグメントと3つ先のセグメントまでの情報を得られることになっている。

表1: シミュレータから与えられる情報

*1: ストレートのとき長さ, カーブのとき radian
*2: ストレートのとき 0

cur_len	今いるセグメントの長さ*1
cur_rad	今いるセグメントの半径*2
to_lft	ロボットから左の壁までの距離
to_rgt	ロボットから右の壁までの距離
to_end	セグメントの終わりまでの距離*1
v	現在のロボットのスピード
vn	vの左の壁方向への成分
nex_len	次のセグメントの長さ*1
nex_rad	次のセグメントの半径*2
after_len	2つ先のセグメントの長さ*1
after_rad	2つ先のセグメントの半径*2
aftaft_len	3つ先のセグメントの長さ*1
aftaft_rad	3つ先のセグメントの半径*2
cen_a	求心方向加速度
tan_a	接方向加速度
alpha	前に設定した alpha
vc	前に設定した vc

2 実験 1

最初はシミュレータから与えられる情報を終端子に、四則演算と if 文を非終端子に設定し、単純に遺伝的プログラミングを適用した。

個体の表し方 1つの個体は alpha と vc を決定するために2つの木を持つ。それぞれの木の終端子は表1のものと [0,2] 内の乱数を発生させるものを合わせて全部で18種類、非終端子は四則演算と if 文の5種類とする。

適合度の与え方 ロボットが3000ステップ費しても1周できなかった場合は、進んだ距離に応じて0.0から0.2までの適合度を与える。1周できた場合は、ロボットの動作ステップ数に応じて0.2から0.8となるように次の式で適合度を与える。

$$\left(\frac{3000 - \text{動作ステップ数}}{3000} \right) \times 0.8 + 0.2$$

シミュレータに付いている人間が作ったサンプルのロボットは約2000ステップで1周するので適合度は約0.467となる。

パラメータ

1世代の個体数	500
木の高さの上限	12
交叉率	0.8
突然変異率	0.2
選択方法	トーナメント(サイズ5)
実行世代数	500

実験結果 実験を10回行ったときの最良適合度の平均の推移を図2に示す。500世代目での10回中の最良の適合度は0.401333(2245ステップで1周)であった。

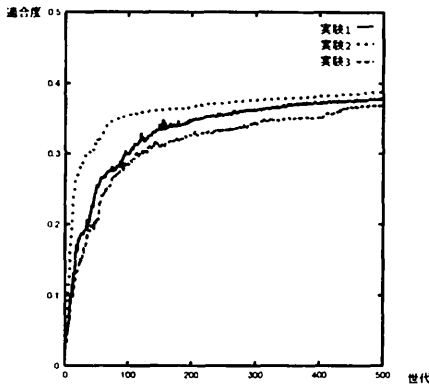


図2: 実験結果

3 実験2

実験1において得られた個体を調べてみると、例えばradianと速度の足し算など、現実的に意味がないような計算が多数見られた。これが原因で本来必要であるはずの計算の探索が遅れていると考えられた。そこで終端子、非終端子に型を付け、探索範囲を狭くしてやることにした。

個体の表し方 終端子(表1)と非終端子を6つの型(単位なし、長さ、速度、加速度、radian、時間)に分類した。また、例えばcur_lenのようにストレートのとき長さ、カーブのときradianとなっていたものは、cur_len(長さ),cur_radian(radian)のように分けた。その際、カーブのときには正確にセグメントの長さを表すようにcur_len(長さ)=cur_len(radian)×cur_rad(半径)とし、ストレートのときにはcur_radianを大きい値に設定した。

実験結果 適合度とパラメータを実験1と同じに設定して実験を10回行ったときの最良適合度の平均の推移を図2に示す。500世代目での10回中の最良の適合度は0.412(2205ステップで1周)であった。

4 実験3

実験1、実験2の個体を実際にRARSシミュレータの下で走らせて動きを観察してみると、コースアウトしないようになるべくコースの中央を走ろうとしているものばかりであり、スピードが制限されているように見えた。これはコース情報による分岐がうまくいっていないことが原因と思われる。そこでロボットのプロプログラムを、知ることのできる4つ(aftaft,after,nex,cur)のセグメントの状態(ストレートかカーブか)によってあらかじめ $2^4 = 16$ 種類に分けることにした。

個体の表し方 alphaとvcを決定するための2つの木をそれぞれ16種類に分けるので、1つの個体は合計32本の木を持つことになる。実験2の終端子の6つの型のうち、長さとradianについては、記号の意味は実験1の表1に戻し、セグメントがストレートのときは半径の値(常に0)は意味がなくなるので省略した。このようにして長さとradianの終端子は16種類作られた。それ以外の型については実験2と同じである。条件分岐をあらかじめやっていることと、探索空間が広がってしまうことを考えて木の高さの上限は10にした。

実験結果 実験を10回行ったときの最良適合度の平均の推移を図2に示す。実験2よりも平均が悪くなっているのは探索空間を広げてしまったことが原因であろうと思われる。しかし500世代目での10回中の最良の適合度は0.435733(2116ステップで1周)であり、少数ではあるが実験2の最良の適合度よりも高い個体が作られるようになった。

5 まとめ

終端子となりうる情報が多いこの問題では、終端子や非終端子に型を付けることは有効であった。また問題を分割しそれぞれに木を割り当てることによって、よりよい個体を作ることが可能であることがわかった。しかしここまでの最良のロボットでもRARSではまだ遅い方である。今後は情報の重要度を調べ、終端子や非終端子の出現頻度を変化させて実験してみる予定である。

参考文献

- [1] RARS main page
URL <http://www.ebc.ee/~mrenmm/rars/rars.htm>
- [2] 伊庭育志,“遺伝的プログラミング”,東京電機大学出版局,1996