障害物回避ファジィロボットのＧＰ並列化によるメンバーシップ関数自動生成

村井　保之[*]　松村　幸輝[**]　巽　久行[***]　徳増　眞司[*]
神奈川工科大学[*]　鳥取大学[**]　筑波技術短期大学[***]

１．はじめに

環境から得られた情報に基づき自律的に障害物回避が行えるロボットの構築が望まれている[1]. これを実現する方法として，ファジィ制御が有効であると考えられる．この場合，良好な制御を行うためには適切なメンバーシップ関数の設定とチューニングがきわめて重要となる．

この観点から，筆者らは，進化システムのひとつである遺伝的プログラミング[2]（以下，GP）を用いてメンバーシップ関数を自動的に生成し，これをロボットの軌道制御に用いる方法について検討している[3]．本論文では，GP の探索効率向上のため，地域交配集団(deme) [4]を用いた並列処理について，その有効性を論ずる．

２．遺伝的プログラミング

ここでは，GP の基本的な表現方法をそのまま適用し，数式で構成した木構造を用いてメンバーシップ関数を表現し，交叉，突然変異などの遺伝操作を行いプログラム個体の進化を試みる．

３．障害物回避モデル

ロボットは移動障害物と目標物が存在する作業空間上で，センサーからの情報に基づき，移動障害物を回避しながら，作業空間左端のスタート位置から右端の目標物に到達するように移動する．その際，ロボットは，ｘ軸の正の方向へは常に一定の移動量で移動するものとし，障害物回避および目標物接近のためにｙ軸方向にのみ移動量を増減できるものとする．

４．ファジイ制御による軌道修正

ロボットの軌道修正は，障害物および目標物との位置関係に基づき，ファジィ制御でｙ軸方向の移動量を増減させることによって行う．

GP により生成されたメンバーシップ関数の例を図１に示す．図１(a)は，入力関数でロボットと対象物（障害物あるいは目標物）との距離をラベル「近い，遠い」の２つで表したものである．図１(b)は，ロボットと対象物に対する相対位置（ロボットの中心と対象物の中心の差）をラベル「左側，やや左，中央，やや右，右側」の５つで表した．図１(c)は，出力関数で，ロボットのｙ軸方向の移動量（移動方向と移動量）を「左，中央，右」の３つで表すものとする．

Automatic generation of membership functions for obstacle avoidance fuzzy robots based on GP parallelism.

Yasuyuki Murai*, Koki Matsumura**, Hisayuki Tatsumi***
Shinji Tokumasu*,
Kanagawa Institute of Technology*
Tottori University **
Tsukuba College of Technology ***

５．遺伝的プログラミングによるメンバーシップ関数の生成

メンバーシップ関数は以下の遺伝操作によって生成する．まず，乱数を用いて初期個体集団を生成する，この集団の各個体の適応度を計算し，ルーレット法で交叉を行う．交叉により発生した個体は適応度の低い個体と入れ替える．また，突然変異は，集団中の個体を乱数で選出し，個体の長さに基づいて変異させるノードの位置を乱数で決定しその内容を変更する．以上の遺伝操作を，適応度が所定値を超えるか，一定の世代になるまで繰り返す．なお，個体の適応度は,次に述べる手続きに従って設定する．そして，この処理によって最終的に得られる最良個体をメンバーシップ関数としてファジィ制御に用いる．なお, 実際の遺伝操作では 10 個のメンバーシップ関数を連結して１つの個体として扱うことにした．各個体の適応度は,遺伝操作により生成した個体を，メンバーシップ関数として用いて回避シミュレーションを実行し，ロボットが障害物に接触せずに，かつ走行距離が短いものほど高くなるように設定する．この処理を，１つの個体について，障害物の位置を変え所定の回数だけ行い, この合計を適応度とする．

なお，最大適応度は 2,500,000 とした．

６．地域交配集団による並列処理

GP では，遺伝操作が進むにつれ，個体の長さが増大する傾向にある．個体の長さが増すと，遺伝操作にかかる時間も増大する．そのため，より良い解を探索するために，遺伝操作の回数を増やすことは，処理時間の増大を招く．また，集団内の個体数を増やすことで，個体の多様性を維持し，探索効率を上げることも考えられるが，個体数が増えると遺伝操作の時間も増えてしまう．そこで，これらの問題を解決するため，地域交配集団による並列処理を行っ

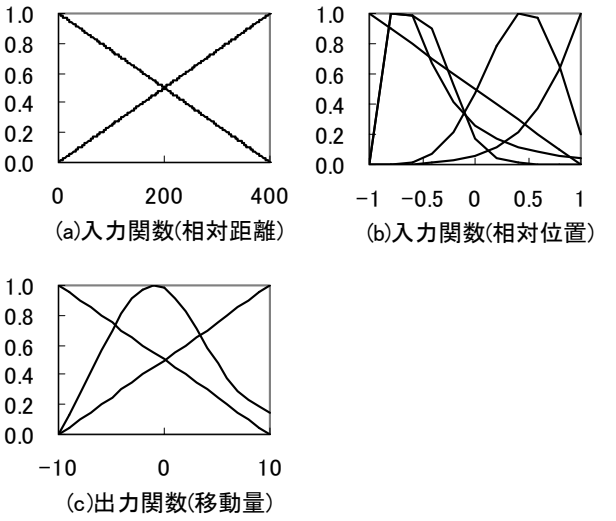

図1　自動生成されたメンバシップ関数の例

た. これは, 個体集団をいくつかのサブ集団に分け, 別々のコンピュータ上で同時並行して遺伝操作を行う事で, 個々のコンピュータの個体数を少なくし処理時間の増大を抑え, かつ, 集団間での個体交換を行うことで, 個体の多様性を維持しようというものである. 本論文では, 各個体集団を TCP/IP により接続された LAN 上のコンピュータに割り当て, 同時並列に遺伝操作を行った. 集団間での個体交換は TCP/IP によるパケット通信で行う. 図 2 にシステム構成を示す. システムは, 個体の交換と遺伝操作クライアントの同期をとるためのサーバ, 遺伝操作を行うクライアントから構成される. 本論文の実験では, サーバ 1 台とクライアント 4 台の 5 台構成とした. また, 全てのコンピュータは同じ仕様 (Pentium Ⅲ, 1GHz, Windows2000) とした.

サーバと遺伝操作クライアント間の個体交換は次の手順で行う.

(1)クライアントがサーバに接続する.
(2)全てのクライアントがサーバに接続後, サーバからクライアントへ遺伝操作開始コマンドを送信する.
(3)クライアントは遺伝操作を行う.
(4)クライアントは最良個体をサーバへ送信し, サーバからの個体受信待ちに入る.
(5)サーバは, 接続している全てのクライアントから最良個体を受信後, 個体を交換しクライアントへ送信する. その際の個体交換は次式に従う.

Cs= (Cr+1) Mod n

Cs : 送信クライアントの番号,
Cr : 受信クライアントの番号, n はクライアント数

(6)クライアントはサーバからの個体を受信する.

受信された個体は, 各クライアントの個体集団に追加され, 次回の遺伝操作対象となる.

(7)(3)～(6)の操作を, 指定された回数繰り返す.

7. 実験による評価

図 3 は, 提案手法と並列処理しない場合の, 最大適応度を比較したものである. 並列処理した場合の適応度は, 4 台の遺伝操作クライアントのうち, 最大のものを使用した. 実験は, それぞれ, 乱数の初期値を変え, 交叉率 5%, 突然変異率 5%, 初期個体数 100, 最大個体数 100, 遺伝操作回数 3000 回の条件で, 5 回行った.

その結果, 並列処理しない場合, 3000 回の遺伝操作では, あまり進化が進まず適応度は非常に小さい値となった. そこで, 並列処理しない場合のみ遺伝操作回数を 4000 回とした実験を 1 回行った. この場合, 3500 回目付近から進化が進み, 最終的な適応度は 1,814,232 となり, 並列処理した場合に近い値となった. しかし, 並列処理の場合 5 回中 4 回が 200 万以上と最大適応度に近い値を示しており, 少ない回数の遺伝操作で良好な適応度が得られ, 並列処理の効果が確認できた.

表 1 遺伝操作パラメータと最終適応度

No	交叉率	突然変異率	個体数	適応度
1	20%	10%	100	1,961,776
2	5%	10%	150	1,921,182
3	10%	5%	50	1,947,902
4	5%	5%	100	1,915,136

次に, 遺伝操作のパラメータをクライアント毎に変えた実験を 1 回行った. 表 1 は, その際の各パラメータと最終的な適応度を示したものである. なお, 遺伝操作回数は 3000 回とした. 結果は, 遺伝操作のパラメータを統一した場合よりも適応度は低くなり, 処理時間に関しても, パラメータを統一した場合は, 平均 1 時間 25 分であったが, この場合には 3 時間 35 分と 2 倍以上かかった. これは, 交叉率, 突然変異率に 10%以上のものがあり, これらの影響で遺伝操作に時間がかかったためである. この実験では, サブ集団の環境 (遺伝操作のパラメータ) を変えた場合の探索効率の向上は確認できなかった.

8. むすび

以上, 地域交配集団を用いた並列処理を用いて GP の探索効率を改善する方法について検討した.

これより, この方法がメンバーシップ関数の効率的な生成に有効に働くことがわかった.

参考文献

[1]松村幸輝, 変形ベルヌーイ写像を用いたカオティック移動ロボットの障害物回避シミュレーション, 電学論, vol.119-C, no.5, pp.603-614, 1999.
[2]Koza,J, Genetic Programming, MIT Press,1992.
[3]松村幸輝, 村井保之, 遺伝的プログラミングに基づくファジーロボットの障害物回避モデル, 信学論(A), vol.J83-A, no.12, pp. 1539-1551,2000.
[4]伊庭斉志, 遺伝的プログラミング, 東京電機大学出版局, 1996.

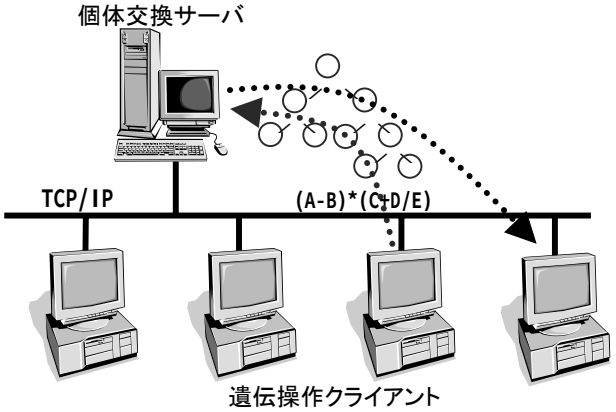

図 2 システム構成

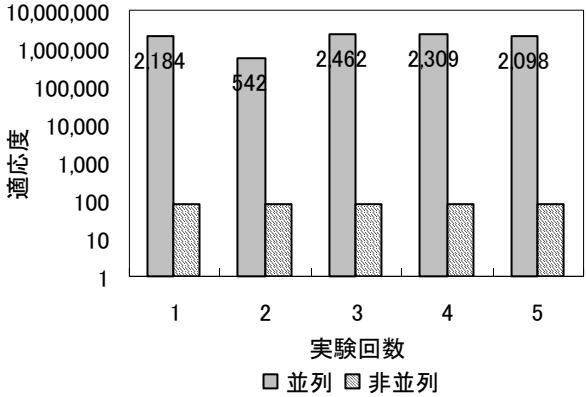

図 3 並列 GP と非並列 GP との比較