

自律移動ロボット群による搬送問題の解法

函館高専 ○佐野 雅泰 浜 克己

要 旨

複数のエージェントが共通の目標を達成する過程は協調問題解決と呼ばれ、実現のためには各エージェントが個々の局所的な行動戦略と大域目標達成のための戦略を整合させる必要がある。本稿では搬送問題を例に取り、各エージェントの行動戦略の獲得方法について試みる。

1. 緒 言

複数のエージェントが独立な動作をしながらも、全体として共通の目標を達成する過程は協調問題解決と呼ばれ、多くの方式が提案されている¹⁾。本稿では、何種類かの荷物と障害物が存在する環境を設定し、複数のロボットが荷物をそれぞれ指定された位置に運ぶための分散解法を提案する。

各ロボットは自律移動エージェントと見なされ、内蔵された挙動関数にしたがって、障害物を回避しながら最小ステップ数で荷物およびその指定された位置へ到達できる行動戦略の獲得を目指す。方式として、この挙動関数をニューラルネットワークで構成し、これを問題向きに進化させる方法を試みる²⁾。計算機実験を通じて、本方式の有効性を確認する。

2. 問題設定

環境モデルを図1に示す。また、問題空間は次の5項組で表現される。

$$p = (S, \Delta, \Phi, \Theta, \Psi) \tag{1}$$

S : 2次元セル空間内の状態集合

$$S = \{(x, y, b) \mid x \in X, y \in Y, b \in B\}$$

$$X = \{1, 2, \dots, N_x\}, Y = \{1, 2, \dots, N_y\}$$

$$B = \{0, 1\}, 0 : \text{通行可}, 1 : \text{障害物}$$

Δ : エージェントの挙動関数

$$\Delta = (U, C, W)$$

U : ユニット集合, C : 結合集合, W : 荷重集合

Φ : エージェントの状態集合

$$\Phi = \{(a_i, q_i, s_i, k_i) \mid i = 1, \dots, N_a\}$$

a_i : 番号, $q_i = (x_i, y_i)$: 位置, s_i : 状態

k_i : 把持荷物番号

Θ : 荷物の状態集合

$$\Theta = \{(l_i, r_i, c_i) \mid i = 1, \dots, N_l\}$$

l_i : 番号, r_i : 位置, c_i : 状態

Ψ : 荷物の目標位置集合

$$\Psi = \{d_i \mid i = 1, \dots, N_l\}$$

エージェントの移動は以下のように表現される。

$$q_i^{t+1} = \delta_i(q_i^t, m) \quad , \quad m : \text{行動 (4方向の移動)}$$

$$\text{if } s_i = 1 \text{ then } r_{k_i} = q_i$$

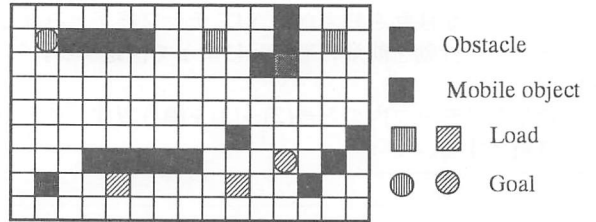


Fig.1 Environmental model

以上より、この問題は次のように定式化され、これを満たすような各エージェントの行動列が問題の解となる。

$$\text{minimize } \sum_{i=1}^{N_a} \text{step}_i \quad \text{subject to } r_j = d_j, \quad j = 1, \dots, N_l \tag{2}$$

3. エージェントの挙動関数

各エージェントは、挙動関数としてのニューラルネットワーク構造とセンサからの入力情報によって行動を決定する。ここでは、エージェントの行動系列を取り扱う必要性から、過去の状態に依存した動作を遂行できるようにするため、その基本構造を中間層に自己ループを持つリカレント型 (RNN) とする。ここでは、異なる層のユニット間の結合、並びに中間層内の自己ループ結合についてはあらかじめ固定とするが、中間層内の異なるユニットの間の結合は可変とし、学習の際に段階的に増やしていくものとする。このうち、中間層ユニットからの出力は、過去の時間ステップの出力を反映して次式で計算される。

$$y_i(t) = \sigma \left(\sum_{j=1}^{NI} w_{ij} x_j(t) + \sum_{k=1}^{NH} w_{ik} y_k(t-1) \right) \tag{3}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

NI, NH : 入力層と中間層のユニット数

x_j : ユニットjへの入力, w_{ij}, w_{ik} : 結合荷重

$\sigma(x)$: シグモイド関数

挙動関数の入力には、以下の各内容に対するセンサからの上下左右4方向の情報がバイナリ値として使用される。

(1) 周囲の障害物の状況

(2) 移動(目標)先の方向

ここで、移動先の方向は同時に複数入力を持つことが可能であり、一例として移動方向が右上方の場合には上と右に対応する入力の設定される。出力は移動可能な4方向のいずれかでありエージェント毎に最も大きい出力値に対応する行動が選択される。また、各時点で行動できるエージェ



ントは唯一とするため、基本的には全エージェントの最大出力値を比較し、その中でさらに最大値を持つエージェントに行動実行権を与えることとした。しかし、これだけでは同一のエージェントのみが連続して選ばれる場合が考えられるので、どのエージェントにも選択可能性を持たせるために、以下のような確率的な方式を採用した。

$$P_i = \frac{o_i}{\sum_{j=1}^{N_a} o_j} \quad (4)$$

ここで、 P_i および o_i はそれぞれエージェント i の選択確率と最大出力値である。

4. 進化学習

目標状態を達成するための行動獲得が目的であるが、それに加えて無駄な移動を軽減する配慮が必要である。そこで、各RNNの適合性値 F_i を以下のように与える。

$$F_i = c_1 \frac{N_{goal}}{N_{load}} + c_2 \frac{N_{get}}{N_{load}} + c_3 \frac{N_{step} - N_{col}}{N_{step}} \quad (5)$$

N_{load} : 空間上の総荷物数

N_{goal} : 目標地点に到達した荷物数

N_{get} : エージェント群が把持(獲得)した総荷物数

N_{step} : 総行動数, N_{col} : 障害物に衝突した回数

c_1, c_2, c_3 : 選択係数 ($c_1 + c_2 + c_3 = 1$)

世代交代毎に各RNNが評価され、ある確率で適合性値の低い一定の割合を淘汰し、生き残った親から子の生成を行う。また、RNNの学習は進化プログラミング(EP)に基づいて行われ、リンク荷重とネットワーク構造に対して突然変異のみを用いる。荷重変化に対するオペレーションは以下のように行われる。

$$w = w + c \cdot N(\mu, \sigma^2) \quad (6)$$

$$c = \frac{F_{max} - F_i}{F_{max}}$$

$N(\mu, \sigma^2)$: ガウス型ランダム変数

構造に関しては、非常に低い確率で中間層内にユニット間結合を追加するものとする。ただし、この場合の初期荷重値を0とする

5. 計算機実験

以下では、提案した方法論に基づき行った実験結果について示す。実験条件は以下の通りである。

- ・ユニット数 入力層: 8, 出力層: 4
- ・個体総数: 100 ・最大世代数: 300
- ・世代毎の最大ステップ数: 600
- ・荷物 種類: 2, 総数: 6 ・エージェント総数: 3
- ・環境モデルの大きさ: 20×20
- ・突然変異率: 0.01 ・淘汰率: 0.4
- ・係数 c_1 : 0.7, c_2 : 0.2, c_3 : 0.1

ここでは、中間層ユニットの数の違いによる目標達成状況を示す。図2に示すような実験用の環境モデルを用意し、ユニット数を6, 8, 10と変えながら上の条件で実験を行った。それぞれについて数回行った結果について、世代

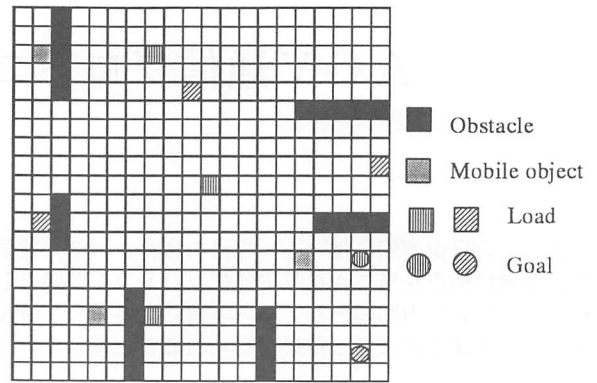


Fig.2 Environmental model for experiment (X=20, Y=20)

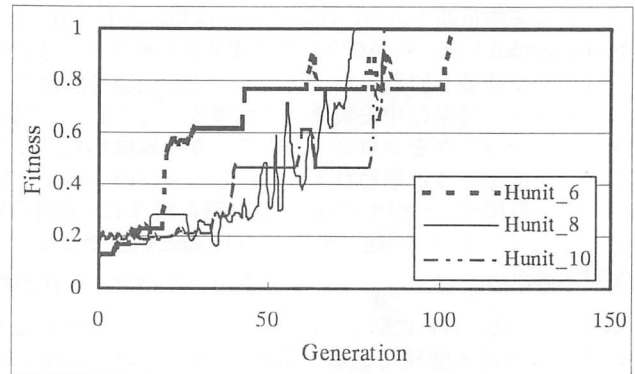


Fig.3 Fitness for the different number of hidden units

毎の適合性値の変化(平均値)を図3に示す。これより、ユニット数が6と10ではある程度一定の値を続けた後に上昇というパターンを繰り返しているのに対し、ユニット数8では変動しながら上昇している。これは、6と10の場合は初期収束は早いだが、その後ある状態に陥り、大きな荷重の変化等が与えられなければそこから抜け出すことが難しい状況にあったと思われる。一方、8の場合は常に変化しているため、そのような状況に会うことがなく、目標達成までの収束時間も最も早かった。この結果だけから断定することはできないが、結合に関して、6の場合には不足し、また10の場合には逆に冗長になっていることが考えられる。

6. 結言

エージェントの挙動関数をリカレント型のニューラルネットで構成し、これを問題向きに進化させる方式を用いることにより、簡単な計算機実験ではあるがその一部について有効性を確認した。

今後は、エージェント同士の協調的行動の検討や挙動関数の適合性を高めることによって、類似の環境やさらに複雑な環境への対応を図る予定である。

参考文献

- 1) 加藤, 武野, 複数の移動ロボットの協調, 日本ロボット学会誌, Vol.12, No.6, pp.785-790, 1994
- 2) Angelino, P.J., Saunders, G.M. and Pollack, J.B., An Evolutionary Algorithm that Construct Recurrent Neural Networks, IEEE Trans. on NN, 5-1, pp.54-65, 1994