Vol.2010-MPS-77 No.3
2010/3/4

ホップフィールドネットワーク間の協調によるロボットの行動計画 — 倉庫番を例にして —

宮崎海理† 松田聖††

人間の知的活動はすべて何らかの最適化過程であるとの仮説に基づき，ロボットの行動計画をホップフィールドネットワーク間の最適化機能の協調により求めることを試みた．行動計画は論理性を必要とする典型的な人間の知的活動であるが，ニューラルネットワークでそれを実現することは難しいと考えられてきた．今回，この行動計画を倉庫番パズルという，押すことしかできないロボットが障害物を避けながらブロックを目的地へと運ぶというパズルを例にして行った．

Robot Planning by Cooperation of Hopfield Networks – A Case of Warehouse Keeper Puzzle(Soukoban) -

KAIRI MIYAZAKI† SATOSHI MATSUDA††

Based on the hypothesis that all intellectual activities of human being are performed as optimization processes, we design Hopfield networks that can coordinately achieve a robot planning. Planning is a typical intellectual activity of logical task, which is considered difficult for neural networks to do. As an example of robot planning, we take the Warehouse Keeper Puzzle, called Soukoban, the goal of which is for the robot to carry a block to a goal only by pushing it and by avoiding many obstacles.

1. はじめに

人間には自身の計画を適切に修正できる制御機構が存在すると考えられる．人間の行動決定は，ある環境下において，目標を設定し，その目標を達成するための計画を立てた上で行われる．その計画に沿った行動を行っていく過程で，自身の行動によって，または他の要因によって環境に変化が起こり，現行の計画では不都合が生じる場合がある．その際に，計画自体を修正することによって，状況に則した柔軟な行動制御が行われていると考えることができる．この人間的な「計画を適切に修正できる制御」が技術的に可能となれば，より柔軟な対応をできるシステムが構築できるであろう．一方，著者らは人間の知的活動はすべて何らかの最適化過程であるとの仮説に基づき，意思決定，画像処理，パズルの求解等に対して，最適化ニューラルネットワークによるモデル化を提案してきた[1][2][3]．そこで本研究では，倉庫番パズルを例にとり，適切な修正を行うロボットの行動計画を複数のホップフィールドネットワーク間の最適化機能の協調により求めることを試みた．今回取り扱う行動計画のような複雑な処理を単一ニューラルネットワークで実現するのは現実的ではないことから，複数のニューラルネットワーク間の協調により実現する方法が必要となる．

ニューラルネットワークは，連想記憶やパターン認識などのモデル化に用いられてきた．神経細胞のモデル化であるという点から考えても，人間の知的活動を表現することに対して有効であり，近年でも様々な分野で用いられている．しかし，人間の行動計画に対して利用された例は少なく，その中でも最適化によって解へと収束するホップフィールドネットワーク[4][5]が用いられることはあまりなかった．

また，ロボットの行動計画についても，かつての論理指向の試みの後は，センサー等を用いた疑似実環境下での関心が中心となっており，やはり脳のモデルであるニューラルネットワークとの関連で考えられることは少なかった[6][7][8]．

倉庫番パズルに関しては，PSPACE 完全問題であることが知られており，如何に効率良く解の探索を行うかに焦点があてられているため[9]，人間の知的活動モデルという観点から検討されることはあまりなかった．しかし，その論理性や自らの行動から環境が変化する点を考えると，パズルとしてだけでなく，ロボットの行動計画という知的活動であるととらえることができる．

本論文では，倉庫番パズルを例にとり，複数のホップフィールドネットワークの最適化機能の協調行動によるロボットの行動計画の可能性を示し，簡単な問題に対して

†日本大学大学院生産工学研究科数理情報工学専攻
Graduate Course of Mathematical Information Engineering, Graduate School of Industrial Technology, Nihon University
†† 日本大学生産工学部数理情報工学科
Department of Mathematical Information Engineering, College of Industrial Technology, Nihon University

シミュレーションを行った.

2. 倉庫番パズル

本研究でのロボット行動計画の例題とする倉庫番パズルの例を図１に示す.

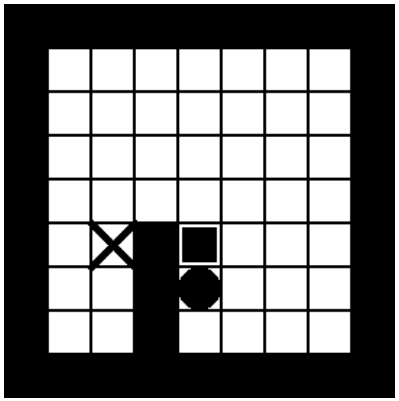

図 1 倉庫番パズル
Fig.1 Example of Soukoban Puzzle.

倉庫番とは，特定の広さを持ったマス目状のフィールドに，1 ステップ時間で縦か横に 1 マスずつ移動ができるロボット（図中の●），押すことができる物（■），ロボットや物がその上に存在できず，動かせない壁（黒い塗りつぶし）と，物を最終的に移動させる目的地（×）が存在するパズルである．ロボットは物を進行方向へ押すことしかできず，引っ張ったり横にどけたりはできない．この中で，ロボットが物を如何にして目的地まで運ぶかを考えるパズルである．このパズルでは，ただ単純に目的地の方向へ物を押すだけではなく，壁を迂回する，また方向転換のために一度物や目的地から離れるなどの先を見越した計画と，状況が遷移する度に計画を修正する柔軟性が必要となる．

3. ホップフィールドネットワーク

ホップフィールドネットワークとは，ニューラルネットワークの一種であり，ニューロンが相互に結合することによって構成されている. また, エネルギーと呼ばれる，ネットワーク全体の目的に対する適合度を示す値が定義され，各ニューロンが互いに影響し合い，値が変化するのに伴いエネルギーは減少し，最終的にネットワークはエネルギー最小，つまり最適化を目指すものである．

二次のホップフィールドネットワークのダイナミクスは，次のようになる．

$$\frac{du_i}{dt} = \sum_j \sum_k w_{ijk} v_j v_k + \sum_j w_{ij} v_j + h_i \qquad (1)$$

$$v_i = \frac{1}{2}\left(1 + \tanh\left(\frac{u_i}{T}\right)\right) \qquad (2)$$

ここで，$v_i \in [0, 1]$，$u_i \in R$，$h_i \in R$ はそれぞれニューロン i の出力値，内部値，バイアス値を表す．$w_{ij} \in R$，$w_{ijk} \in R$ はニューロン i，j および，ニューロン i，j，k の結合荷重を表し，結合荷重は対称であるとする．T は適当な定数である．このときのエネルギー関数は次のように定義される．

$$E = -\frac{1}{3}\sum_i \sum_j \sum_k w_{ijk} v_i v_j v_k - \frac{1}{2}\sum_i \sum_j w_{ij} v_i v_j - \sum_i h_i v_i \qquad (3)$$

このとき，

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial v_i} \qquad (4)$$

$$T\frac{dv_i}{dt} = -2v_i(1 - v_i)\frac{\partial E}{\partial v_i} \qquad (5)$$

が成り立つことから，各ニューロンの値はエネルギーが減少するように変化し，平衡点である超立方体の頂点 $V \in [0, 1]$, あるいは内部のエネルギーの極小点に収束する.

今回，このホップフィールドネットワークを，倉庫番を解くときに考えられる論理的ステップ１つ１つに対して構築する．それらのネットワークが論理的順序で最適化していくことによって，次に行うべき行動を決定する．次の図 2 が，今回倉庫番を解くために使った論理とそのアルゴリズムの概要である.

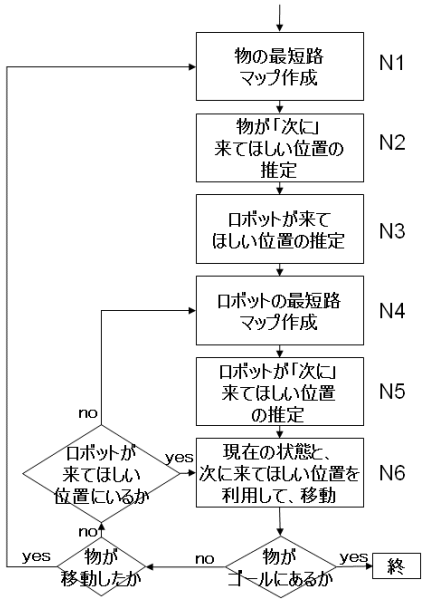

図 2 倉庫番を解くアルゴリズム
Fig.2 Algorism of Solving Soukoban.

まずニューラルネットワーク N1 が物とその目的地までの最短路を決定する．最短路が決定したならば，その最短路を通るときに，物が次の段階で来て欲しい位置，すなわち次にどこに押せばいいのかを N2 が決定することができる．物を次に押す位置が決定したならば，物をそこに押すためには，ロボットがどこに来ればよいのか，すなわちロボットが来るべき場所を N3 が決定する．ロボットの来るべき場所は，ロボットにとっての目的地となるので，次に N4 はロボットに関する最短路を考えることができる．物の場合と同じように，最短路が決まれば，N5 はロボットが次に来るべき場所を決定することができる．そこまで，順番に決定すると，ロボットがどう移動すればよいのかがわかるので，N6 は状態を 1 ステップ進める，すなわち移動を行うことができる．移動を終えて，物が目的地に到着していれば目的が達成されるので終了することができるが，到着していなければ次のロボットの移動を考えなければならない．その際，ロボットや物が動いているならば，それぞれ最短路を考え直さなければならない場合がある．その最短路を計画し直すことによって，その下の論理的段階である物が次に来てほしい位置などがドミノ式に変更されていくので，状態に応じた柔軟な計画の修正が行われるのである．

なお，ロボットの最短路を決定するとき，物の位置は計画通りに動かす場合以外はなるべく動かさないほうが良いことから，まず物を壁と同じく動かせない障害物として考える．その上で，物を障害物として考えると最短路が決定できないときは，物を障害にならないものとして最短路を決定することとした．また，計画に従っていった結果，どこにも物を押せない状態になってしまった場合，その状態での物の位置は物を移動させるべきではない場所，すなわち仮想的に壁として扱い，問題を解きなおすようにした．これにより，移動させるべきではない場所を記録していくことになるので，その部分を除いた新たな計画を立てることができるようになる．

このアルゴリズムのそれぞれの論理的段階に対して，ホップフィールドネットワークを構築し，エネルギー関数を与える．ホップフィールドネットワークの構成は，それぞれの論理的段階において 1 つの格子が 1 つのニューロンに対応しており，発火をしているニューロンがその論理的段階の解を表している．物の最短路マップ作成を例にとれば，各格子上にニューロンが対応しており，その中で発火しているニューロンが物の最短路を表現している．

（6）式は物の最短路マップ作成に関するエネルギー関数である．

$$
\begin{aligned}
E_1 = {} & A_1 \sum_i^M \sum_j^N M_b(i,j) \\
& + B_1 \sum_i^M \sum_j^N M_b(i,j) \times \left[\begin{matrix} M_b(i-1,j) + M_b(i,j-1) + \\ M_b(i+1,j) + M_b(i,j+1) - 2 \end{matrix} \right]^2 \\
& + C_1 \left[\begin{matrix} M_b(s_i^b - 1, s_j^b) + M_b(s_i^b, s_j^b - 1) + \\ M_b(s_i^b + 1, s_j^b) + M_b(s_i^b, s_j^b + 1) - 1 \end{matrix} \right]^2 \\
& + D_1 \left[\begin{matrix} M_b(g_i^b - 1, g_j^b) + M_b(g_i^b, g_j^b - 1) + \\ M_b(g_i^b + 1, g_j^b) + M_b(g_i^b, g_j^b + 1) - 1 \end{matrix} \right]^2 \qquad (6)
\end{aligned}
$$

M，N は盤面の大きさ，M_b は解を求めたい物の最短路マップ，(s^b_i, s^b_j) は物の現在の位置（スタートの位置），(g^b_i, g^b_j) は物の目的地（ゴールの位置）を表す．A_1，B_1，C_1，D_1 は定数である．式（6）は最短路の条件となる，「ニューロンの発火数が最小であること」，「スタート，ゴールからの隣はひとつだけ発火していること」，「それ以外の発火しているニューロンの 4 方隣のうち 2 つは発火していること」を満たしていない場合，エネルギーが高くなるように作られている．このエネルギー関数を求めたい解である M_b で微分することによって，ニューロンの更新式が決定される．

次に，物が「次に」来てほしい位置の推定に関するエネルギー関数を示す．

Vol.2010-MPS-77 No.3
2010/3/4

$$
\begin{aligned}
E_2 = {} & A_2 M_b(s_i^b, s_j^b) \times M_b(s_i^b - 1, s_j^b) \times \left[S_b^{t+1}(s_i^b - 1, s_j^b) - 1 \right]^2 \\
& + B_2 M_b(s_i^b, s_j^b) \times M_b(s_i^b, s_j^b - 1) \times \left[S_b^{t+1}(s_i^b, s_j^b - 1) - 1 \right]^2 \\
& + C_2 M_b(s_i^b, s_j^b) \times M_b(s_i^b + 1, s_j^b) \times \left[S_b^{t+1}(s_i^b + 1, s_j^b) - 1 \right]^2 \\
& + D_2 M_b(s_i^b, s_j^b) \times M_b(s_i^b, s_j^b + 1) \times \left[S_b^{t+1}(s_i^b, s_j^b + 1) - 1 \right]^2 \qquad (7)
\end{aligned}
$$

S_b^{t+1} は物の次に来てほしい位置を表すニューロン群である．A_2，B_2，C_2，D_2 は定数である．(7) 式は，「物の最短路マップが物の現在の位置で発火し，そのすぐ隣にも物の最短路マップが発火しているのなら，そのすぐ隣にある物の最短路マップと同じ位置が物の次に来てほしい位置として発火する」という条件が，物の現在の位置の 4 方隣について記述されている．このエネルギー関数を求めたい解である S_b^{t+1} で微分することによって，ニューロンの更新式が決定される．なお，ここで用いられている M_b は，アルゴリズム的に (6) 式を用いて先に最適化されているため，S_b^{t+1} を更新する際には固定値として扱われる．変化が起こるニューロンは S_b^{t+1} のみである．

$$
\begin{aligned}
E_3 = {} & A_3 \sum_i^M \sum_j^N S_b^t(i-1, j) \times S_b^{t+1}(i-2, j) \times \left[G_a(i, j) - 1 \right]^2 \\
& + B_3 \sum_i^M \sum_j^N S_b^t(i, j-1) \times S_b^{t+1}(i, j-2) \times \left[G_a(i, j) - 1 \right]^2 \\
& + C_3 \sum_i^M \sum_j^N S_b^t(i+1, j) \times S_b^{t+1}(i+2, j) \times \left[G_a(i, j) - 1 \right]^2 \\
& + D_3 \sum_i^M \sum_j^N S_b^t(i, j+1) \times S_b^{t+1}(i, j+2) \times \left[G_a(i, j) - 1 \right]^2 \qquad (8)
\end{aligned}
$$

(8) 式は，ロボットが来てほしい位置を推定するためのエネルギー式である．G_a がロボットの目的地を表現するニューロン群である．S_b^t は現在の物の位置を表すニューロン群であり，現在物のある位置が 1，それ以外が 0 となっている．A_3，B_3，C_3，D_3 は定数である．この式には「物の現在の位置の隣に，物の次に来てほしい位置があるとき，物の次に来てほしい位置とは現在の位置に関して反対側がロボットの目的地となる」という条件が 4 方向に関して記述されている．このエネルギー関数を G_a に関して微分することで，ロボットの来てほしい位置を推定する更新式となる．なお，ここで用いられている S_b^{t+1} は，アルゴリズム的に (7) 式を用いて先に最適化されているため，G_a を更新する際には固定値として扱われる．変化が起こるニューロンは G_a のみである．

ロボットの最短路マップ作成とロボットの次に来てほしい位置に関するエネルギー式は，(6)，(7) 式と同じ記述で，使っている変数に違いがあるだけである．

$$
\begin{aligned}
E_6 = {} & A_6 S_a^t(s_i^b - 1, s_j^b) \times S_b^t(s_i^b, s_j^b) \times S_b^{t+1}(s_i^b + 1, s_j^b) \\
& \quad \times \left\{ (newS_a^t(s_i^b, s_j^b) - 1)^2 + (newS_b^t(s_i^b + 1, s_j^b) - 1)^2 \right\} \\
& + B_6 S_a^t(s_i^b, s_j^b - 1) \times S_b^t(s_i^b, s_j^b) \times S_b^{t+1}(s_i^b, s_j^b + 1) \\
& \quad \times \left\{ (newS_a^t(s_i^b, s_j^b) - 1)^2 + (newS_b^t(s_i^b, s_j^b + 1) - 1)^2 \right\} \\
& + C_6 S_a^t(s_i^b + 1, s_j^b) \times S_b^t(s_i^b, s_j^b) \times S_b^{t+1}(s_i^b - 1, s_j^b) \\
& \quad \times \left\{ (newS_a^t(s_i^b, s_j^b) - 1)^2 + (newS_b^t(s_i^b - 1, s_j^b) - 1)^2 \right\} \\
& + D_6 S_a^t(s_i^b, s_j^b + 1) \times S_b^t(s_i^b, s_j^b) \times S_b^{t+1}(s_i^b, s_j^b - 1) \\
& \quad \times \left\{ (newS_a^t(s_i^b, s_j^b) - 1)^2 + (newS_b^t(s_i^b, s_j^b - 1) - 1)^2 \right\} \\
& + F_6 S_a^t(s_i^a, s_j^a) \times S_b^{t+1}(s_i^a - 1, s_j^a) \\
& \quad \times \left\{ (newS_a^t(s_i^a - 1, s_j^a) - 1)^2 + (newS_b^t(s_i^b, s_j^b) - 1)^2 \right\} \\
& + G_6 S_a^t(s_i^a, s_j^a) \times S_b^{t+1}(s_i^a, s_j^a - 1) \\
& \quad \times \left\{ (newS_a^t(s_i^a, s_j^a - 1) - 1)^2 + (newS_b^t(s_i^b, s_j^b) - 1)^2 \right\} \\
& + H_6 S_a^t(s_i^a, s_j^a) \times S_b^{t+1}(s_i^a + 1, s_j^a) \\
& \quad \times \left\{ (newS_a^t(s_i^a + 1, s_j^a) - 1)^2 + (newS_b^t(s_i^b, s_j^b) - 1)^2 \right\} \\
& + I_6 S_a^t(s_i^a, s_j^a) \times S_b^{t+1}(s_i^a, s_j^a + 1) \\
& \quad \times \left\{ (newS_a^t(s_i^a, s_j^a + 1) - 1)^2 + (newS_b^t(s_i^b, s_j^b) - 1)^2 \right\} \\
& + J_6 S_a^t(s_i^a, s_j^a) \times S_a^{t+1}(s_i^a - 1, s_j^a) \times S_b^t(s_i^a - 1, s_j^a) \\
& \quad \times \left\{ (newS_a^t(s_i^a - 1, s_j^a) - 1)^2 + (newS_b^t(s_i^a - 2, s_j^a) - 1)^2 \right\} \\
& + K_6 S_a^t(s_i^a, s_j^a) \times S_a^{t+1}(s_i^a, s_j^a - 1) \times S_b^t(s_i^a, s_j^a - 1) \\
& \quad \times \left\{ (newS_a^t(s_i^a, s_j^a - 1) - 1)^2 + (newS_b^t(s_i^a, s_j^a - 2) - 1)^2 \right\} \\
& + L_6 S_a^t(s_i^a, s_j^a) \times S_a^{t+1}(s_i^a + 1, s_j^a) \times S_b^t(s_i^a + 1, s_j^a) \\
& \quad \times \left\{ (newS_a^t(s_i^a + 1, s_j^a) - 1)^2 + (newS_b^t(s_i^a + 2, s_j^a) - 1)^2 \right\} \\
& + M_6 S_a^t(s_i^a, s_j^a) \times S_a^{t+1}(s_i^a, s_j^a + 1) \times S_b^t(s_i^a, s_j^a + 1) \\
& \quad \times \left\{ (newS_a^t(s_i^a, s_j^a + 1) - 1)^2 + (newS_b^t(s_i^a, s_j^a + 2) - 1)^2 \right\} \qquad (9)
\end{aligned}
$$

Vol.2010-MPS-77 No.3
2010/3/4

（9）式は現在の状態と次に来てほしい位置を利用して移動を行うことに関するエネルギー関数である．$newS_b^t$ は物の新しい位置を，$newS_a^t$ はロボットの新しい位置を表現するニューロン群である．A_6，B_6，C_6，D_6，F_6，G_6，H_6，I_6，J_6，K_6，L_6，M_6 は定数である．3 種類の移動のケースが，それぞれ 4 方向について，エネルギー関数 E_6 の中に表現されている．その 3 種類の移動とは，「現在の物の位置の隣に，物の次に来てほしい位置があり，かつその逆側にロボットが存在していたとき，物の新しい位置は物の次に来てほしい位置と同じ位置が発火し，ロボットの新しい位置は現在の物の位置と同じ位置が発火する」，「ロボットの現在の位置の隣に，ロボットの次に来てほしい位置があり，ロボットの次に来てほしい位置に物が現在ない場合，物の新しい位置は現在の物の位置と同じ位置が発火し，ロボットの新しい位置はロボットの次に来てほしい位置と同じ位置が発火する」，「ロボットの現在の位置の隣に，ロボットの次に来てほしい位置があり，そこに現在物があるのなら，物の新しい位置は現在の位置からロボットの進行方向に一つずれた位置が発火し，ロボットの新しい位置はロボットの次に来てほしい位置と同じ位置が発火する」である．これらの条件を式として表したのが（9）式である．このエネルギー関数を $newS_b^t$，$newS_a^t$ に関してそれぞれ微分することで，物の新しい位置，ロボット新しい位置を推定する更新式となる．なお，ここで用いられている $newS_b^t$，$newS_a^t$ 以外の変数は，アルゴリズム的に（9）式を用いて最適化されるまでに決定しているため，$newS_b^t$，$newS_a^t$ を更新する際には固定値として扱われる．変化が起こるニューロンは $newS_b^t$，$newS_a^t$ のみである．

これらの論理的段階の条件を記述したホップフィールドネットワークをアルゴリズムに従い順番に最適化していくことによって，取るべき行動を決定し，また必要に応じて計画を変更していくことが可能な構造を作成した．

4. シミュレーション

以上のアルゴリズムとホップフィールドネットワークを用いて，実際に倉庫番の問題を解くシミュレーションを行った．問題は，ロボットが 1 体，物と目的地が 1 つずつという設定とした．次の図 3，4 がそのシミュレーション結果である．N1 から N6 までの一回の動作を 1step として，初期状態から終了までの例を示している．図中の B は N1 で生成した物の最短路を，R は N4 が生成したロボットの最短路を表す．1step ごとに移動を行い，その次の移動のための最短路を改めて求め，表示している．

1 つ目のパズルの例では（図 3），step 0 で初期状態からの物の最短路が推定され，すでにロボットはロボットが来てほしい位置にあるので，物の最短路に従うように物を押すことになる．すると，step 1 では，物を最短路に従って移動させるために，ロボットは物の次の位置の逆側へ移動しなければならなくなる．そこがロボットの目的地となり，目的地に到達するための最短路を推定する．その最短路に従いロボットは動く．このように，物の移動経路は step 0 に示されたとおりに最後まで実現するという直感的なものであるが，ロボットの動作は物の位置が変化するたびに再計画を行うことによって，状況に応じた行動を取ることができる．

2 つ目のパズルの例（図 4）でも，再計画がうまく働いている．step 3 において，物が障害物となり，単純にはロボットの目的地に達することができない状況におかれる．このときのネットワーク上の行動決定の推移を見てみると，まず物は目的地へ動く（下に動く）ように求めており，それに従ってロボットの目的地は現在の物の位置の上へと設定される．すると，その目的地に達するためのロボットの最短路が，一度は物を障害物として推定しようとし失敗するが，もう一度物を障害物とせずに推定しようとすると，物がある場所もロボットの最短路として判断される．その最短路に従った行動を取ると，step 4 のように物が意図せず移動されてしまうが，その状態に変化に対応し，もう一度物の最短路を推定しなおすことによって，物の次の位置，ロボットの目的地，最短路，次の位置がすべて計画しなおされ，柔軟な行動を取ることができる．

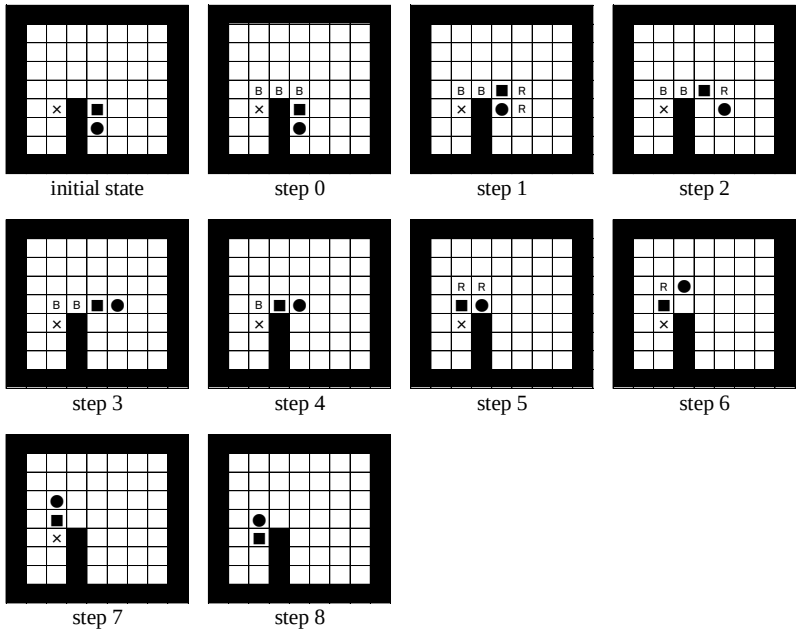

図 3 パズル 1
Fig.3 Puzzle 1.

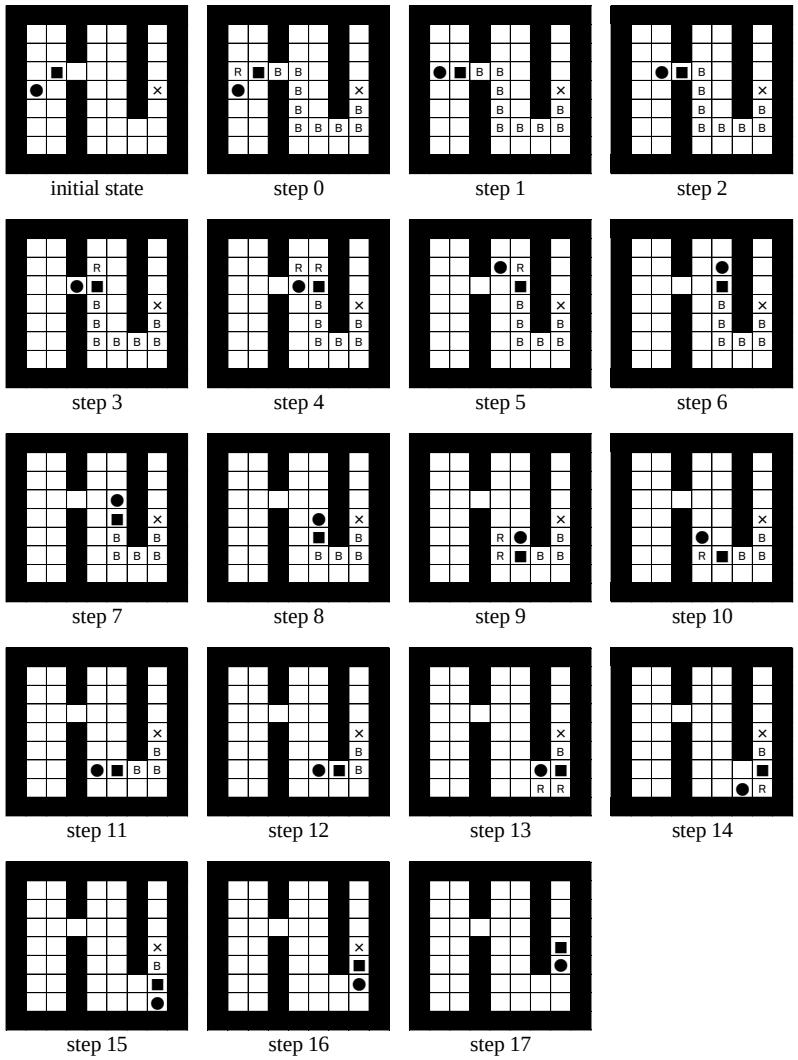

図 4 パズル 2
Fig.4 Puzzle 2.

このように基本的な問題に対しては，ほとんどの場合で行動計画を行うことに成功した．しかし，物の現在の位置とその目的地が隣接しているような問題は，仮想的な壁が機能しない，またエネルギーが収束できない解である問題となるため，それら問題点を改善できる機構を現時点では持っておらず解くことができなかった．

5. まとめ

本研究では，論理的段階をそれぞれホップフィールドネットワークとして構築し，順序通りに最適化を行った．その結果，各ステップのネットワークがドミノ式に最適化することによってトップダウン的に協調し，ロボットの行動計画を産出することができた．すなわち，ニューラルネットワークの最適化を用いた行動計画を表現する一例を示すことができたと考えられる．しかし解けない問題が存在している点，またアルゴリズムという機械的な概念を利用していることから，人間の知的活動はすべて何らかの最適化過程であるとの仮説からすると，まだ問題点は残っている．今後は，うまく動かすことができなくなる最短路を，移動可能な場所からボトムアップ的にも最短路を修正できるような機構，すなわちネットワークの相互のイントラクションによってのみ行動計画を行う機構を構成し，より協調的に移動を決定する方法を考えていく必要があるであろう．

参考文献

1) S. Matsuda, “A neural network model for the decision-making process based on AHP”, ijcnn2005
2) S. Matsuda, “An extended neural network model for the decision-making process based on Analytic Network Process ANP”, ijcnn2006
3) 大谷哲広，松田聖，ニューラルネットワークによるパズルの求解―ホップフィールドネットワークで数独は解ける―，電気情報通信学会「人工知能と知識処理」研究会（SIG-AI），AI2008-88, 2009
4) J. J. Hopfield, “Neurons with graded response have collective computational properties like those of two-state neurons”, in Proc. Nat. Academy Sci. USA, May 1984, vol. 81, pp. 3088-3092
5) J. J. Hopfield and D. W. Tank, ““Neural” computation of decisions of optimization problems”, Biol. Cybern., vol.52, pp.141-152, 1985
6) R. E. Fikes and N. J. Nilsson , “STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving”, Artificial Intelligence, Vol.2, pp.189-208, 1971
7) Rodney A. Brooks, “Intelligence without Representation in MIT Teche Report”, 1988，訳:柴田正，表彰なしの知能,，現代思想, vol. 18-3, 1990
8) N. J. Nilsson, “Artificial Intelligence”, A New Synthesis, 1998
9) 田中哲朗，倉庫番パズル，情報処理学会誌 Vol.43 NO.11 pp.1253-1258, 2002